

Typology and literature review for dial-a-ride problems

Peer-reviewed author version

MOLENBRUCH, Yves; BRAEKERS, Kris & CARIS, An (2017) Typology and literature review for dial-a-ride problems. In: ANNALS OF OPERATIONS RESEARCH, 259 (1-2), p. 295-325.

DOI: 10.1007/s10479-017-2525-0

Handle: <http://hdl.handle.net/1942/24042>

Typology and Literature Review for Dial-a-Ride Problems

Yves Molenbruch · Kris Braekers · An Caris

Received: date / Accepted: date

Abstract Dial-a-ride problems consist of designing vehicle routes and time schedules in a system of demand-dependent, collective people transportation. In the standard problem, operational costs are minimized, subject to full demand satisfaction and service level requirements. However, to enhance the practical applicability of solution methods, authors increasingly focus on problem variants that adopt additional real-life characteristics. First, this work introduces an up-to-date classification that distinguishes multiple categories of real-life characteristics. Second, the wide range of solution methods proposed in the literature is reviewed in a structured manner. Although the existing literature is reviewed exhaustively, specific attention is devoted to recent developments. Third, an extensive overview table provides full details on all problem characteristics and solution methods applied in each paper discussed. Fourth, lacunae in research conducted to date and opportunities for future work are identified.

Keywords dial-a-ride · demand-responsive transportation · patient transportation · rich vehicle routing · real-life characteristics · literature review

Yves Molenbruch
Hasselt University, Agoralaan Gebouw D, BE-3590 Diepenbeek
Research Foundation Flanders (FWO), Egmontstraat 5, BE-1000 Brussels
Tel.: +32-11-268783
E-mail: yves.molenbruch@uhasselt.be

Kris Braekers
Hasselt University, Agoralaan Gebouw D, BE-3590 Diepenbeek
Research Foundation Flanders (FWO), Egmontstraat 5, BE-1000 Brussels

An Caris
Hasselt University, Agoralaan Gebouw D, BE-3590 Diepenbeek

1 Introduction

A **dial-a-ride** system is an application of demand-dependent, collective people transportation [21]. Each user requests a trip between an origin and a destination of choice, to which a number of service level requirements are linked. The service provider attempts to develop efficient vehicle routes and time schedules, respecting these requirements and the technical constraints of a pickup and delivery problem [98]. A frequent objective is to minimize operational costs subject to full demand satisfaction and side constraints, but service level criteria may be optimized as well. Balancing the human and economic perspectives involved in solving such a Dial-a-Ride Problem (DARP) is essential for organizing quality-oriented, yet efficient transportation of users with special needs, such as door-to-door transportation for elderly and disabled. In light of the ageing population, dial-a-ride systems are gaining importance to complement regular transportation modes. They also fulfill a social role, preventing isolation of vulnerable groups in society. However, due to the rising demand, service providers no longer manage to compose vehicle routes and time schedules manually. Planning algorithms are required to safeguard cost efficiency and service quality.

This work analyzes the existing literature on planning algorithms for dial-a-ride problems, which delivers a fourfold **contribution** to the research field. First, a thorough classification of existing problem variants is established. Originally, most authors adopted the standard DARP definition [20], which explains why early literature reviews [21, 98] only distinguish between static or dynamic variants, as well as single-vehicle or multi-vehicle systems. Nowadays, more and more authors tend to address additional real-life problem characteristics, enhancing the practical applicability of their solution methods. Although some extensions have been mentioned in recent book chapters [35, 99], this work provides a more comprehensive classification to distinguish multiple categories of real-life problem characteristics. Within each category, similarities and differences between the corresponding papers are analyzed. As a second contribution, this work reviews the wide range of solution approaches that have been applied to various problem variants. Due to space limitations, the discussion emphasizes general evolutions in the literature. However, full details on problem characteristics and solution methods in each single paper are listed in an exhaustive overview table (see Appendix A), which is a third contribution. The overview table also reflects the focus of this work, being the literature on the definition of problem variants and the development of solution approaches. The increasing relevance of this subject is evidenced by the fact that half of all reviewed papers have been published after 2010. As a final contribution, promising directions for future research are identified by revealing trends and lacunae in the current state of the art.

The remainder of this work is organized as follows. Section 2 clarifies the methodology used to review the literature. Section 3 classifies the variety of problem variants defined to date. Section 4 distinguishes multiple solution approaches and discusses widely used benchmark data. Section 5 covers the scheduling subproblem. Finally, Section 6 summarizes the main findings and derives opportunities for future research.

2 Methodology

Two **search strategies** were adopted to collect all relevant literature in a structured way. First, electronic bibliographic databases, such as EBSCOhost and Web of Knowledge, were consulted. A broad search was carried out for papers having the concepts ‘dial-a-ride’, ‘on-demand transportation’, ‘demand-responsive transportation’ or ‘patient transportation’ in their title, content or keywords. Closely examining these search results, papers focusing on problem variants and solution methods have been retained for inclusion in the actual literature review process. Second, additional literature was identified on the basis of references by other authors. These papers were obtained by

searching other sources or by contacting the authors. Once again, the aforementioned criteria were employed to decide which of them have been included. After gathering all literature, a thorough examination of the problem characteristics and solution methods in each paper allowed to establish the structure of this work.

3 Problem characteristics

As a first contribution, this work identifies and classifies all **problem variants** presented in the literature. Section 3.1 discusses the standard DARP [20] and its mathematical formulations. Although this standard problem has been studied extensively, practical applications of dial-a-ride systems often exhibit numerous additional problem characteristics. Planning algorithms ignoring these extensions may deliver infeasible or unrealistic solutions. Therefore, the current trend of studying rich vehicle routing problems with a broader practical applicability [14] is also noticeable in research on the DARP. More specifically, the classification scheme presented in this work is based on the observation that real-life systems may require three types of extensions to the standard problem definition. The first type involves additional characteristics related to the design of the service. On this matter, Section 3.2 presents problem variants that consider heterogeneous users, vehicles or drivers, more complex routing properties and different service level specifications. The second type of extension relates to the objective in the optimization process. Whereas the standard problem definition includes a single operational objective, Section 3.3 discusses a wide range of operational and service-related objectives whose practical relevance has been described in the literature. The third type of extension concerns the nature of the information available to the service provider. The standard problem assumes that data is deterministic and known before vehicle routes and schedules are designed. In contrast, Section 3.4 focuses on problem variants that involve dynamic or stochastic information on travel times, requests or user behavior. Table 1 outlines the detailed classification scheme of additional real-life problem characteristics and indicates the references associated with each category. Papers delivering multiple contributions may appear in several categories. Finally, to conclude the discussion on problem characteristics, Section 3.5 identifies lacunae in the current state of the art and emerging research topics for future work.

3.1 Standard DARP

3.1.1 Definition

A **standard definition** of the DARP has been established by Cordeau and Laporte [20]. The problem consists of designing a number of minimum-cost vehicle routes in a complete graph of nodes and arcs, provided that all user requests are met. Nodes correspond to the pickup and delivery locations of users, supplemented with the vehicle depot. Each directed link between two nodes is an arc, characterized by a travel time and an associated cost which is incurred if the arc is part of the solution. Each route starts and ends at the depot within fixed time intervals and respects a maximum route duration. For the users served in between, the service at each location starts within a time window. The maximum user ride time cannot be exceeded and a vehicle's load cannot violate its capacity. To ensure a correct physical route construction, precedence and pairing of a user's origin and destination should be respected by visiting them in the right order, using the same vehicle. A service duration indicates the time required for loading and unloading users. As indicated in the classification scheme of Parragh et al. [98], this definition distinguishes the DARP from other problems in vehicle routing. Most closely related is the pickup and delivery problem with time windows (PDPTW), which also involves demand-dependent transportation between paired pickup and delivery locations. However, the definition of the PDPTW is based on the transportation of goods, which explains why fewer quality constraints need to be included. Particularly the maximum

3.2 Advanced service design

3.2.1 Heterogeneity

Beaudry et al. [7], Braekers et al. [12], Detti et al. [32], Hanne et al. [55], Liu et al. [76], Parragh [96], Parragh et al. [97], Qu and Bard [106], Wong and Bell [129], Xiang et al. [130], Zhang et al. [134]

3.2.2 Routing properties

Beaudry et al. [7], Braekers et al. [12], Braekers and Kovacs [13], Detti et al. [32], Espinoza et al. [37], Häll et al. [49], Hanne et al. [55], Karabuk [69], Liu et al. [76], Masmoudi et al. [81], Masson et al. [83], Molenbruch et al. [87], Parragh et al. [97], Parragh et al. [102], Reinhardt et al. [109], Zhang et al. [134]

3.2.3 Quality specifications

Diana and Dessouky [33], Jaw et al. [67], Jørgensen et al. [68], Melachrinoudis et al. [85], Paquette et al. [94], Toth and Vigo [127]

3.3 Alternative objectives

3.3.1 Single objective

Attanasio et al. [5], Berbeglia et al. [9], Diana et al. [34], Feuerstein and Stougie [39], Garaix et al. [45], Krumke et al. [71], Parragh et al. [102], Rekiek et al. [110], Ritzinger et al. [111], Yi and Tian [132]

3.3.2 Multiple objectives

Diana and Dessouky [33], Guerriero et al. [48], Jørgensen et al. [68], Lehuédé et al. [72], Mauri and Lorena [84], Melachrinoudis et al. [85], Paquette et al. [95], Parragh et al. [101], Schilde et al. [119], Schilde et al. [120], Wong and Bell [129], Xiang et al. [130]

3.4 Stochastic or dynamic

3.4.1 Travel times

Fu [41], Schilde et al. [120], Xiang et al. [131]

3.4.2 Requests

Attanasio et al. [5], Beaudry et al. [7], Berbeglia et al. [8], Berbeglia et al. [9], Coslovich et al. [23], Cremers et al. [26], Ho and Haugland [60], Hanne et al. [55], Schilde et al. [119], Hyytiä et al. [63]

3.4.3 User behavior

Deflorio et al. [29], Heilporn et al. [58]

Table 1 Classification of papers extending the standard problem characteristics of Cordeau and Laporte [20]

user ride time constraint is inherent to the DARP and complicates the scheduling subproblem. More specifically, scheduling the start of service in each node as early as possible is not an effective strategy and may cause incorrect infeasibility declarations. It may be necessary to postpone the start of service in pickup nodes in order to decrease the ride time of the user involved. Extensive details on solution approaches for this scheduling subproblem can be found in Section 5.

3.1.2 Mathematical formulations

Cordeau [19] proposes an **arc-based** mixed-binary linear program, shown by equations 1-14. The **three-index** binary decision variable x_{ij}^k indicates whether vehicle k traverses the arc between nodes i and j . Each vehicle route is assumed to start at an origin depot and end at a destination depot (eq. 4, 6). One and the same vehicle should reach and leave corresponding pickup and delivery locations i and $n + i$ (eq. 2, 3, 5), which ensures flow conservation and pairing. Decision variable L_i^k computes the ride time of user i (eq. 9) and cannot exceed the maximum user ride time L (eq. 12). Explicit precedence constraints become redundant if L_i^k is set at least equal to the associated direct ride time. Decision variables B_i^k and Q_i^k track the service start in node i (eq. 7) and the load upon leaving node i (eq. 8), respectively. They should respect the time window of node i (eq. 11) and the capacity Q_k of vehicle k (eq. 13), respectively. The time span between the moment a

vehicle leaves the origin depot and the moment it returns to the destination depot cannot exceed the maximum route duration T_k (eq. 10). A minimum-cost selection of arcs is made (eq. 1), subject to all constraints and full demand satisfaction.

$$\text{Minimize} \quad \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij}^k x_{ij}^k \quad (1)$$

Subject to

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1 \quad \forall i \in P \quad (2)$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \quad \forall i \in P, \forall k \in K \quad (3)$$

$$\sum_{j \in N} x_{0j}^k = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0 \quad \forall i \in P \cup D, \forall k \in K \quad (5)$$

$$\sum_{i \in N} x_{i,2n+1}^k = 1 \quad \forall k \in K \quad (6)$$

$$B_j^k \geq (B_i^k + d_i + t_{ij})x_{ij}^k \quad \forall i \in N, \forall j \in N, \forall k \in K \quad (7)$$

$$Q_j^k \geq (Q_i^k + q_j)x_{ij}^k \quad \forall i \in N, \forall j \in N, \forall k \in K \quad (8)$$

$$L_i^k = B_{i+n}^k - (B_i^k - d_i) \quad \forall i \in P, \forall k \in K \quad (9)$$

$$B_{2n+1}^k - B_0^k \leq T_k \quad \forall k \in K \quad (10)$$

$$e_i \leq B_i^k \leq l_i \quad \forall i \in N, \forall k \in K \quad (11)$$

$$t_{i,i+n} \leq L_i^k \leq L \quad \forall i \in P, \forall k \in K \quad (12)$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q_k, Q_k + q_i\} \quad \forall i \in N, \forall k \in K \quad (13)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i \in N, \forall j \in N, \forall k \in K \quad (14)$$

P = set of pickup nodes, D = set of delivery nodes, N = set of all nodes (including the depot nodes), K = set of vehicles, c_{ij} = cost associated with arc (i,j) , t_{ij} = direct travel time associated with arc (i,j) , d_i = service duration in node i , q_i = net number of users boarding in node i

Since an arc cannot be traversed by multiple vehicles, the vehicle index k can be avoided whenever route duration is unbounded and travel times/costs are vehicle-independent. Different vehicle capacities can still be modeled using artificial pickup and delivery nodes. Røpke et al. [114] present two **two-index** formulations. The first only adapts constraints on the physical route construction. Pairing is ensured by precedence constraints, defining subsets of arcs which cannot jointly be part of a feasible solution. The second two-index formulation of Røpke et al. [114] defines a single decision variable x_{ij} . Time and load restrictions are imposed by infeasible path constraints and rounded capacity constraints. However, eliminating L_i , B_i and Q_i limits possibilities to include objectives related to user quality (see Section 3.3). In addition, the number of constraints grows exponentially with the number of users.

As explained in Section 4, these arc-based formulations are particularly suitable for branch-and-cut approaches. However, **other types** of formulations have been presented as well. Set-partitioning or set-covering formulations [103] are useful for column generation approaches. Such formulations require a set of feasible routes with a known cost. A minimum-cost selection is made among these routes, such that fleet size is respected and each request appears in exactly one (set-partitioning) or at least one (set-covering) route. Constraint programming approaches may invoke a formulation

based on successor variables [10]. A decision variable indicates the immediate successor of each node. A route is represented as a closed circuit, assuming that the origin depot conceptually succeeds the destination depot. Another decision variable registers the vehicle performing service in each node, given that it should belong to the same route as its successor and its corresponding pickup/delivery node. Also the start of service in a node and the load upon leaving this node are tracked by separate decision variables. A feasible set of routes should be constructed subject to all constraints and full demand satisfaction.

3.2 Advanced service design

3.2.1 Heterogeneity

In real-life patient transportation, Parragh [96] observes that users exhibit heterogeneous **physical needs**. Instead of using a normal seat, some users travel in a wheelchair or on a stretcher. Besides, they may require the presence of one [96, 97] or several [76, 134] accompanying persons. This is reflected in the layout of the vehicles, which consists of multiple capacitated resource types, such as staff seats, normal seats, wheelchair spaces and stretchers. If upgrading conditions apply, users can be assigned to a ‘lower’ resource type than requested. Parragh [96] provides the formulation of Cordeau [19] with an additional index, such that vehicle capacity should be respected for each resource type. Any set of eligible resource types may be defined for each user category. Parragh [96] also extends the first formulation of Røpke et al. [114] with a pair of artificial origin and destination depots for each vehicle. Tournament inequalities are defined, based on all paths which start at a particular artificial origin depot and violate a load constraint. Consequently, loads in each route can be checked without using a third index. The second formulation of Røpke et al. [114] is adapted by Braekers et al. [12]. Rounded capacity constraints are formulated for the resource type having the smallest capacity relative to the corresponding demand.

Qu and Bard [106] note that service providers may use **configurable vehicles** to become more demand-responsive. The distribution of total vehicle capacity over resource types is assumed to be adjustable, given a set of possible layouts which are inherent to the vehicle type. Mathematically, an additional index is added to the heterogeneous problem formulation of Parragh [96], indicating which configuration should be selected while traversing an arc. Liu et al. [76], Hanne et al. [55] and Beaudry et al. [7] also include disjunctive multi-dimensional capacities, but only the former provide a formulation. Wong and Bell [129] consider two-dimensional vehicle capacity with two-way substitution, meaning that resources of both types can readily be converted into each other.

In addition to physical needs, users may be heterogeneous with respect to their **medical condition**. Hanne et al. [55] and Beaudry et al. [7] discuss priority constraints for patient transportation between hospital campuses. Emergency requests receive first priority in order to complete their service within a limited time frame. Users may also require isolated transportation because of a contamination risk, which implies that their ride cannot be shared. Afterwards, the vehicle has to return to the depot for cleaning. Consequently, the arc between the user’s pickup and delivery location, as well as the arc from this delivery location to the depot should be part of any feasible solution. The depot may be visited multiple times throughout the day (see Section 3.2.2). Heterogeneity may also give rise to different medical driver qualifications. Xiang et al. [130] assume that a degree of complexity is associated with users and with fixed medical equipment in the vehicle. A driver is only assigned to users and vehicles that do not exceed his qualification level. Similarly, Hanne et al. [55] define tasks which should be performed by staff employees and tasks which may be done by volunteers. Based on real-life observations in patient transportation, Detti et al. [32] and Molenbruch et al. [87] impose incompatibilities among different types of patients and restrictions with respect to the vehicles that are eligible to transport them.

3.2.2 Routing properties

Masson et al. [83] allow **transfers** of users between vehicles. This may increase the productivity of systems with random origins and widespread, clustered destinations (or vice versa). On the other hand, transfers may be impracticable for disabled users, spread delays throughout the network and cause waiting times and discomfort. Assuming fixed transfer points, a formulation for the DARP with transfers is obtained by adding user ride time and route duration constraints to the formulation of Cortés et al. [22] for the PDPTW with transfers. Transfer points are modeled as consecutive arrival and departure nodes. Users may reach and leave a transfer point in different vehicles. Upper and lower bounds on transfer times can be included. Since corresponding origins and destinations are not necessarily in the same route, a three-index decision variable registers which vehicle traverses each arc. Reinhardt et al. [109] apply the transfer principle to organize assistance for people with reduced mobility at an airport.

Furthermore, the **integration** of on-demand transportation and regular public transportation may cause additional operational benefits. Guidelines to determine which requests are eligible for hybrid execution [1, 59] consider distance traveled, handicap of the user and design of the fixed routes. On-demand routes are often created subject to a given use of regular bus services. For example, a user's transfer points are determined in advance. However, an integrated approach is desirable to balance travel times and total distance, as well as to alleviate pressure on the on-demand system. Häll et al. [49] assume that pickup and delivery locations are visited by on-demand vehicles, but the middle section of the ride may involve regular public transportation. Buses have unlimited capacity and drive so frequently that waiting times are neglected. The authors's arc-based mixed-binary linear program includes two binary decision variables to indicate which arcs are traversed by an on-demand vehicle and which regular bus lines are used.

Groups of users formulating a single **multiple-load request** are normally transported together. However, on the one hand, Parragh et al. [102] notice that splitting group requests among multiple rides may be more efficient from an operational point of view, even if the size of the group does not exceed vehicle capacity. Some typical split-delivery properties do not hold in the presence of precedence constraints [91]. For example, there may not exist an optimal solution in which each arc is traversed at most once. This explains why Parragh et al. [102] refrain from an arc-based formulation. They present an alternative mixed-integer program in which nodes are assigned to certain positions in the routes, as well as a set-partitioning formulation. On the other hand, Qu and Bard [107] and Liu et al. [76] propose to reduce the problem size by considering individual requests with similar spatial and temporal characteristics as a group request. From a practical point of view, this simplification seldom affects the solution, whereas the resulting problem is easier to solve.

Various authors consider a **multi-trip** problem variant, motivated by restrictions on staff availability and working times. Parragh et al. [97] assume that accompanying persons (see Section 3.2.1) are limited in number, work mornings or afternoons and should be picked up or delivered at the depot. In addition, drivers take a mandatory lunch break which starts within a prefixed time window. An arc-based mixed-binary linear program requires an artificial noon depot, provided with a time window. Binary decision variables indicate whether a vehicle takes an accompanying person during the morning and/or afternoon, which relates to the assignment of users demanding assistance. Driver assignments are registered as well, since no lunch breaks should be scheduled for unused vehicles. The authors also provide a set covering formulation for the same problem variant. Zhang et al. [134] and Liu et al. [76] include lunch breaks at the depot, both for drivers and full-time accompanying persons. A maximum tour duration requires vehicles to visit the depot even more frequently, e.g. to be cleaned. Liu et al. [76] assure that users may require assistance of multiple accompanying persons, who are limited in number. This introduces an additional manpower planning

problem. Masmoudi et al. [81] combine lunch and coffee breaks with heterogeneous users [96] and present a comprehensive mathematical formulation. Other multi-trip contributions include Beaudry et al. [7], Hanne et al. [55] and Karabuk [69]. The latter two papers only insert lunch breaks in the final scheme, which is a non-integrated approach that is frequently applied in practice. Finally, Braekers and Kovacs [13] consider a multi-period variant involving multiple operating days in order to ensure driver consistency. They propose three-index and four-index formulations in which the total number of drivers serving each user is limited.

Whereas most authors assume that vehicles start and end their service at a single depot location, the fleet may be allocated between **multiple depots**. For service providers who allow drivers to take their vehicles home, these depots correspond to the home addresses of the drivers. Braekers et al. [12] incorporate multiple depots into the three-index mixed-binary arc-based linear program of Parragh [96] by adapting costs and travel times of arcs which leave or reach the depot. Due to a much larger working area, multiple depots are also common in the context of per-seat on-demand **air transportation** (dial-a-flight system), which is a rather recent development. Flight routes and schedules can be found by modeling the problem as a DARP with specific routing constraints. Regarding the work duration of pilots, Espinoza et al. [37] distinguish between a maximum availability duration and a maximum flight duration. Both capacity and weight limitations are included and a maximum is imposed on the number of intermediate stops for each user.

3.2.3 Specifications of service quality

Most variants of the DARP involve two constraints which determine the service level experienced by users. A **time window** restricts the waiting time a user may face before being picked up or after being delivered. As noted in Jaw et al. [67], a user is typically asked to indicate either a preferred departure time or a preferred arrival time. For an outbound trip to an appointment, it is important not to be late for the appointment, but the user may arrive somewhat earlier than requested. This gives rise to a time window for the destination of an outbound trip. The opposite reasoning applies to an inbound trip, resulting in a time window for the origin node. In both cases, the single time window for each trip is combined with a **maximum user ride time**, which cannot be smaller than the corresponding direct ride time. Since this limits the time a user can spend aboard the vehicle, an additional time window is implicitly constructed for the origin of an outbound trip (resp. the destination of an inbound trip). This time window can be narrowed for reasons of transparency to the user [33]. Paquette et al. [94] review several approaches to express the maximum user ride time itself. Although most authors impose a fixed value for all users, this entails the counterintuitive consequence of a detour being feasible for a short trip and not for a longer one. Alternatively, the maximum user ride time may be defined as a multiple of the corresponding direct ride time or a maximum exceedance of the direct ride time may be imposed. A maximum user ride time may also be enforced indirectly by imposing two time windows for each user, both for the departure and the arrival (e.g. [4, 102, 128]).

Time windows and maximum user ride times can also be seen as **soft constraints** [68, 85], which means that violations are allowed but discouraged through a penalty cost. This approach increases the probability of finding a solution, but may result in users being late for an appointment or unforeseen waiting time for vehicles which are due to leave before the user is available. Analogously, rather than constructing a time window, deviations from the user's preference time can be penalized proportionally [127]. Remarkably, also violations of essential feasibility requirements (e.g. load constraints) are sometimes minimized [84], rather than imposed as hard constraints. Less frequent aspects of service quality are reviewed in Paquette et al. [94]. For example, some service providers avoid **waiting time** being assigned to a vehicle while a user is aboard, except for loading or unloading other users (e.g. [33, 78, 120]).

3.3 Alternative objectives

3.3.1 Single objective

Most solution methods only include a cost-related **operational objective**. Service quality is assumed to be sufficient as soon as the quality constraints (e.g. maximum user ride time) are met. The most common objective is minimizing total distance traveled (see Appendix A). Assuming that the cost and the travel time associated with an arc are proportional to its length, minimizing total routing costs [20] or total vehicle travel time [111] are equivalent objectives. Less common single-objective optimizations consider fleet efficiency, such as minimizing fleet size [34] or maximizing vehicle usage efficiency [110]. Maximizing the vehicle occupancy rate stimulates social contacts among users, rather than enhancing the operational efficiency of the system [45]. Lastly, for private service providers who are not obliged to meet all demand, profit may be maximized based on a revenue associated with each request [102].

In a dynamic environment, which is discussed in Section 3.4.2, different objectives are common. Full demand satisfaction may no longer be attainable when requests are revealed in real time. Therefore, the number of requests served in time may be maximized [5, 9, 132]. With users requesting immediate service, the time needed to complete all available requests or their average completion time may be minimized [39]. Another relevant objective is a minimization of maximum flow time [71], with flow time being the difference between the release of a request and the start of its service.

3.3.2 Multiple objectives

A single operational objective does not provide any incentive to optimize service quality. However, it may be undesirable to ensure just a minimum service level. In addition, this level may be difficult to specify in advance. Therefore, many authors consider multiple objectives, usually being a combination of operational and **quality-related objectives**. In the latter category, the most common objectives (see Appendix A) are minimizing total user ride time and minimizing total user waiting time, caused by deviations from their preference time. Maximum user ride time may be minimized as well [72]. Additional operational objectives relate to a minimization of vehicle idle time [33], total driver wage [130], total route duration [68] and taxi costs to cover a fleet shortage [129]. Variable routing costs may be supplemented with a fixed vehicle activation cost, which contributes to a minimization of fleet size [48, 130]. Whenever the feasibility of a solution is enforced using soft constraints, additional **penalty terms** need to be included in the objective function [68, 84, 85]. They penalize violations of a soft constraint and guide the solution procedure towards the feasible solution area. In models which consider an advanced service design, the objective function may consider user inconvenience caused by the design characteristics (e.g. the number of transfers).

Most problem formulations aggregate the objectives into a single **weighted-sum objective** function. Advanced utility models, e.g. based on the Choquet integral, may be used to consider interactions between multiple objectives [72]. However, computing an at first sight readily interpretable objective value requires preference information (a priori weight choices) and hides trade-offs between different goals. Having insight into these tradeoffs, e.g. the operational cost of implementing a certain quality level, is essential for a service provider to establish a targeted policy. Another indirect strategy defines a **hierarchical objective** function [119, 120], in which an underlying objective is only invoked as a decision criterion whenever two solutions perform equally well in terms of a main objective. Direct multi-objective formulations most frequently adopt the principle of **Pareto dominance** [87, 95, 101]. Their optimum is a set of non-comparable solutions which do not weakly dominate each other. The set does not contain any solution pair for which one solution

is better at one objective and not worse at the other objective. Solution quality of different sets can be compared using Pareto-compliant sorting criteria (e.g. in [101]), such as the hypervolume indicator [136] and multiplicative epsilon indicator [137]. Alternatively, a reference-point method [17] may be used (e.g. in [95]), which sums distances between objective values and an ideal point. Dynamic weights stress the objective for which these distances are large.

3.4 Stochastic or dynamic information

The standard DARP makes two assumptions regarding the nature of the information available to the service provider. First, information is assumed to be static, which implies that all relevant data (e.g. requests, travel times, ...) is known before the route planning process starts. It remains unchanged during the entire time horizon. Second, the data is assumed to be deterministic, meaning that it is not subject to variability or uncertainty. However, both assumptions rarely hold in real-life systems. Most service providers face dynamic changes in inputs and external conditions tend to induce stochasticity into the system. This section presents different causes that complicate the availability of information and explains how the subject has been addressed in the literature.

3.4.1 Travel times

Fu [41] argues that in an urban environment, a system's reliability can be increased considering **stochastic** and **time-dependent** travel times. They allow to account for traffic congestion and avoid that delays are accumulated. A normal distribution is assumed for the travel times on each arc. The average travel time varies with the precise departure time, whereas the corresponding standard deviation is assumed constant. A route is considered as a sequence of schedule blocks [67] with zero variance at the start of each block. The expected start of service in a node can be computed recursively and should respect the time constraints with a given probability. Routes being feasible in a deterministic context may be rejected if they exhibit a large variance, which increases fleet requirements. To estimate the travel times, Fu and Teply [43] suggest three approaches, based on zones, distance and an artificial neural network. Xiang et al. [131] include stochastic travel and service times in a dynamic problem context. Vehicle breakdowns may occur. Schilde et al. [120] model accidents as gradually expanding and shrinking circles, causing congestion on the arcs they cover. Travel time consists of an average and a stochastic influence of accidents, both being time-dependent. Vehicles always use the shortest-distance path.

3.4.2 Requests

Most authors assume that all requests are known in advance, such that static routes and schedules can be constructed. In the **dynamic** problem variant, additional information may be revealed during the planning phase or even during the execution phase. The most studied case involves (part of) all requests being received in real time. These users either follow the usual reservation principle or ask for immediate service, in which case a maximum position shift may be imposed to respect the order of booking [104]. The service provider should be able to decide instantaneously whether an additional request can be inserted [5]. For this purpose, Berbeglia et al. [9] present a constraint satisfaction problem formulation which can be used to prove the infeasibility of a problem. Hyytiä et al. [63] point out the risk of congestive collapse when the capacity of the control policy is exceeded, which suddenly causes an unacceptably high rejection rate. Specific problem contexts involving additional requests may be considered. For example, Hanne et al. [55] and Beaudry et al. [7] study transportation systems in a hospital context, where emergency requests should be serviced within a very limited time frame. Coslovich et al. [23] focus on unexpected users asking for service during the stop of a vehicle. Cremers et al. [26] consider subcontracting requests to taxi services during

peak moments. The taxis are cheaper when booked one day in advance, but some requests are only revealed at the beginning of the operation day. Apart from additional requests, several unexpected events related to users or vehicles may be taken into account, including user no-shows, cancellations of requests, changes of requests, vehicle breakdowns and traffic jams [36, 52]. Particularly the latter two may have a considerable operational impact [131]. For a rather recent literature review on the dynamic DARP, the reader is referred to Berbeglia et al. [8].

Despite technological advances, such as vehicle localization systems or increased processing power, responding to new information remains a **time-critical** task. Therefore, most authors focus on repairing existing solutions, rather than repeatedly applying static solution methods. In general, such repair heuristics first seek for a feasible solution once new information is revealed. Next, continuous optimization is performed until the next event occurs. Parallel computation may be applied [5]. Parallelization strategies differ in whether control is executed by a single processor or distributed, whether new best solutions are communicated to other processors or not, and whether search parameters and initial solutions differ or not. Nevertheless, a problem's real-time nature may heavily affect the efficiency with which it can be solved. This is reflected in the **competitive ratio**, being the worst-case proportion between an algorithmic result and the corresponding static optimum. Minimizing the time interval needed to complete all available requests, Feuerstein and Stougie [39] and Ascheuer et al. [3] compute lower bounds on this competitive ratio. Lipmann et al. [75] include incomplete ride information, meaning that destinations only become known when their corresponding origin is visited. Feuerstein and Stougie [39] also find lower bounds on the competitive ratio for a minimization of the *average* completion time. Minimizing the maximum flow time, Krumke et al. [71] show that a solution method for a single vehicle with unit capacity cannot be competitive. Yi and Tian [132] maximize the number of requests for which service starts within a fixed time period after their release. They provide lower bounds for the single-vehicle case with either unit capacity or infinite capacity. Yi et al. [133] add restricted information and a finite capacity to the work of Yi and Tian [132].

Apart from the dynamic DARP, problem variants with a limited availability of information may involve known requests with a **stochastic** or **probabilistic** nature. Schilde et al. [119] observe that some users, such as patients in a hospital, may be unable to specify their return time in advance. Rather than considering such inbound trips as dynamic requests, a statistical distribution can be used to anticipate possible inbound trips at various times. Ho and Haugland [60] consider requests that are served with a given probability. In real-life cases, such a problem arises when fixed routes are executed on a regular basis, but users are absent with a known probability. For example, elderly people may not feel fit enough to go to the daycare center on a particular day. In this case, the order of the remaining nodes in the route remains unchanged. The authors construct routes such that expected costs are minimized, using a recourse function that takes into account the skipping probabilities.

3.4.3 User behavior

Reliability may also be impacted by **stochastic user behavior**. Heilporn et al. [58] consider users showing up late at their pickup location. In this case, the on-demand vehicle leaves at the scheduled time and a taxi is called in, causing a cost which exceeds the savings of skipping the corresponding delivery node. An arc-based mixed-binary linear program for this problem variant includes an expected delay cost. The probability of being late decreases as a user is visited later in his pickup time window and thus also depends on the probability of skipping preceding nodes. Some users may be scheduled early in their time window in order to serve the majority of users as late as possible. This may generate considerable savings over a deterministic optimum with expected delay costs, even if the scheduling procedure is adapted. Deflorio et al. [29] discuss lateness of both users and

drivers, the latter due to time-dependent variability of travel times and unforeseen waiting times. Decreasing the variance on how long drivers decide to wait for late users increases the number of requests met.

3.5 Discussion and research opportunities

The literature analysis in the preceding sections reveals a trend of including more **real-life problem characteristics** in models on the DARP. Variants considering heterogeneity, more complex routing properties and dynamic or stochastic information have gained interest in recent years, as demonstrated by the number of contributions in these categories. Although it is encouraging that authors base their work on a practical operating context, some **challenges** should be pointed out as well.

First, recent surveys on service quality and user satisfaction [93, 94] still indicate an insufficient match with the actual needs and concerns expressed by users. Consulting users, service providers and other stakeholders remains advisable to ensure the practical applicability of theoretical research. Second, the research domain lacks unification, since the integration of problem characteristics from different categories has been overlooked. For example, no research has been performed on the realistic combination of stochastic travel times and configurable vehicles, despite all efforts to investigate both characteristics separately. The extensive overview table in Appendix A allows the interested reader to identify similar gaps in the current state of the art. Third, from a managerial point of view, it is insufficient to include real-life problem characteristics without analyzing their effect on operational costs and service quality. For example, a service provider is interested in a quantitative analysis on the benefits of purchasing configurable vehicles before actually doing the investment. However, such insights are currently unknown for many problem variants, although at least a lower bound on the impact of real-life characteristics can be found (e.g. [83]). Fourth, a trend of solving the DARP in combination with related problems has recently arisen. Instead of considering requests and time preferences as an input to the routing problem, Coppi et al. [18] claim that mutual benefits may be obtained by integrating health care scheduling and vehicle routing. They propose to schedule patient treatments in a hospital in such a manner that the costs of transporting the patients to the hospital is minimized. In other words, the overall costs of the health care provider and the dial-a-ride provider can be decreased if their operational activities are planned jointly. Li et al. [73] integrate the on-demand transportation of people and parcels and propose an adapted scheduling procedure to avoid that slack time is scheduled while people are aboard the vehicle. Santos and Xavier [117] discuss a dynamic integration of the DARP and ride sharing by private car owners who are willing to deviate from their intended routes. Such integrated approaches inspired on real-life examples are interesting topics to address in future research.

4 Solution methods

As a second contribution, this work identifies and classifies all **solution methods** applied in the literature to date. Section 4.1 provides an overview of different exact solution methods. However, due to the complexity of the problem, research has mainly been directed at the approximate solution methods discussed in Section 4.2. Table 2 outlines the detailed classification scheme of all approaches as proposed in this work. This table also indicates the references associated with each category. Papers delivering multiple contributions may appear in several categories. Furthermore, common benchmark instances on which the solution methods have been applied are presented in Section 4.3. Section 4.4 includes a summarizing discussion and identifies opportunities for future research.

4.1 Exact methods

4.1.1 Branch-and-cut

Braekers et al. [12], Cordeau [19], Häll et al. [49], Heilporn et al. [58], Liu et al. [76], Parragh et al. [101], Parragh [96], Røpke et al. [114]

4.1.2 Column generation

Garaix et al. [45], Parragh et al. [97]

4.1.3 Branch-(cut-)and-price

Gschwind and Irnich [47], Parragh et al. [102], Qu and Bard [107]

4.1.4 Dynamic programming

Desrosiers et al. [30], Häme [52], Häme and Hakula [53], Häme and Hakula [54], Lois et al. [77], Psaraftis [104], Psaraftis [105], Ziliaskopoulos and Kozanidis [135]

4.2 Approximate methods

4.2.1 Classical heuristics

Borndörfer et al. [11], Coslovich et al. [23], Diana and Dessouky [33], Ioachim et al. [65], Jaw et al. [67], Karabuk [69], Luo and Schonfeld [79], Teodorovic and Radivojevic [126], Toth and Vigo [127]

4.2.2 Metaheuristics based on local search

Attanasio et al. [5], Baugh et al. [6], Beaudry et al. [7], Braekers et al. [12], Braekers and Kovacs [13], Cordeau and Laporte [20], Chassaing et al. [15], Detti et al. [32], Guerriero et al. [48], Häll and Peterson [51], Healy and Moll [57], Ho and Haugland [60], Kirchler and Wolfler Calvo [70], Lehuédé et al. [72], Li et al. [73], Masmoudi et al. [81], Masson et al. [83], Mauri and Lorena [84], Melachrinoudis et al. [85], Molenbruch et al. [87], Muelas et al. [88], Muelas et al. [89], Neven et al. [90], Paquette et al. [95], Parragh et al. [100], Parragh et al. [102], Qu and Bard [106], Ritzinger et al. [111], Schilde et al. [119], Schilde et al. [120], Toth and Vigo [127], Wolfler Calvo and Touati-Moungla [128], Xiang et al. [130], Xiang et al. [131]

4.2.3 Metaheuristics based on population search

Atahran et al. [4], Chevrier et al. [16], Cubillos et al. [27], Cubillos et al. [28], Hanne et al. [55], Jørgensen et al. [68], Masmoudi et al. [80], Parragh et al. [101], Rekiek et al. [110], Zhang et al. [134]

4.2.4 Matheuristics

Berbeglia et al. [9], Crawford et al. [24], Crawford et al. [25], Espinoza et al. [37], Jain and Van Hentenryck [66], Oberscheider and Hirsch [92], Parragh and Schmid [103], Ritzinger et al. [111]

4.2.5 Simulation studies

Deflorio et al. [29], Fu [42], Häll et al. [50], Hyytiä et al. [64], Quadrifoglio et al. [108], Schilde et al. [119], Schilde et al. [120], Shen and Quadrifoglio [123], Shen and Quadrifoglio [124]

Table 2 Classification of papers with respect to the type of solution method applied

4.1 Exact methods

4.1.1 Branch-and-cut

Cordeau [19] applies a **branch-and-cut** algorithm to an arc-based mixed-binary linear program. Preprocessing steps include time window tightening, elimination of arcs which violate physical or time-related requirements, as well as variable fixing due to incompatible requests. The formulation is strengthened using valid inequalities related to time and load variables, subtour elimination, rounded capacity, precedence, generalized order, order matching and infeasible paths. Røpke et al. [114] add inequalities related to fork structures and reachability. An initial pool of inequalities is generated and the LP relaxation of the formulation is solved iteratively. If the root node solution is infeasible, additional cuts are generated using separation heuristics and branching is performed on the variable x_{ij}^k (aggregated or not) to obtain integer values. Parragh [96] and Braekers et al. [12] adapt rounded capacity and strengthened capacity inequalities to the heterogeneous problem variant, defining them for each resource type. Häll et al. [49] discuss which arc eliminations are

(in)valid with integration of regular public transportation. Based on their rich problem definition, Liu et al. [76] incorporate configurable vehicles into the capacity inequalities. Given a minimal capacity requirement of each resource to serve a subset of nodes, dynamic programming determines the minimum fleet size to serve that subset. Liu et al. [76] also define path elimination constraints for paths within a trip and for two consecutive trips of a vehicle or an accompanying person. New valid inequalities impose a lower bound on the total number of trips and force certain users to be served either before or after lunch. Finally, the epsilon constraint technique of Parragh et al. [101] and the L-shaped algorithm of Heilporn et al. [58] are applications of the branch-and-cut algorithm in a bi-objective and stochastic problem context, respectively.

4.1.2 Column generation

Column generation is a particularly useful technique to solve large-scale instances, since the problem is split into a master problem and a subproblem. The master problem consists of solving a set partitioning/covering formulation, making use of a subset of promising routes. With multiple vehicles types, a different subset is assumed for each type [45, 97]. The subproblem consists of generating additional routes with a negative reduced cost, such that they may improve the objective value when added to the master problem. Although the subproblem may be approached in a heuristic (e.g. [18]) or metaheuristic (e.g. [103], see Section 4.2.4) manner, the discussion below focusses on exact techniques. Garaix et al. [45] and Parragh et al. [97] represent the subproblem as an elementary shortest path problem with additional constraints, which can be solved using a labeling method based on dynamic programming. In addition to the cost upon reaching a node and a reference to its predecessor, other data should be stored to identify dominated labels [113]. This includes time and load information, supplemented with information related to specific problem characteristics. Labels can also be eliminated if users aboard the vehicle cannot be delivered in a feasible manner. These procedures adopt a different definition of maximum user ride time, such that this constraint is implicitly included in the time windows of users. Besides, Parragh et al. [97] only launch the exact procedure if six pricing heuristics fail, two of which simplify the exact algorithm.

4.1.3 Branch-(cut)-and-price

The **branch-and-cut-and-price** algorithm of Gschwind and Irnich [47] has obtained the most efficient exact solutions to date for the common benchmark instances of the standard problem (see Section 4.3). Column generation is integrated into a branch-and-cut algorithm, based on the observation that most variables in the solution are nonbasic. The branch-and-cut algorithm includes classical preprocessing steps and cutting planes (valid inequalities [19, 114]). Compared with earlier column generation algorithms, however, new dominance rules are developed to handle both time windows and maximum user ride time constraints in the subproblem. Two pricing heuristics, operating on a reduced network and neglecting ride times, are tried before invoking the exact method. Qu and Bard [107] present a branch-and-cut-and-price algorithm which takes into account configurable vehicles. Parragh et al. [102] design a **branch-and-price** algorithm, embedding column generation into a branch-and-bound algorithm, for a problem variant with split requests and profits. In the subproblem, positive reduced profit columns are identified using a classical labeling algorithm [113].

4.1.4 Dynamic programming

Pure **dynamic programming** algorithms were mainly studied in early contributions on single-vehicle problem variants (e.g. [30, 104, 105]). In spite of few direct applications in practice, they may be used to optimize pairs of routes in a multi-vehicle problem [77, 135] and to prove whether

or not an additional request can feasibly be inserted in a route [54]. For example, Häme and Hakula [54] use dynamic programming to solve a constraint programming formulation based on successor variables [10]. Häme and Hakula [53] maximize the number of users served, investigating unassigned nodes in a specific order. They also propose a restricted variant in which backtracks are limited. The advanced insertion algorithm of Häme [52] considers all feasible insertion positions for a certain user and preserves all resulting partial routes after each iteration, which is only possible in the presence of small time windows. A restricted variant selects partial routes which maximize the available time slack, such that future insertions are facilitated. Some aforementioned methods have heuristic variants in the multi-vehicle context [53, 54]. For each route, the single-vehicle algorithm is executed on a set of unassigned requests and demand satisfaction is maximized.

4.2 Approximate methods

Since the DARP is an NP-hard problem [6], optimal solutions are not expected to be found in polynomial time. Therefore, most problems of realistic size are solved using the approximate solution methods discussed in this section. First, Section 4.2.1 provides an overview of the two most common types of classical heuristics, being insertion heuristics and cluster-first-route-second heuristics. Due to their inability to escape from local optima, research attention for such classical heuristics has steadily declined during the past decade. Nowadays, these classical heuristics are usually incorporated into metaheuristic frameworks, e.g. to generate an initial solution. This study provides an extensive discussion on metaheuristic solution methods, based on local search strategies (Section 4.2.2) or on population strategies (Section 4.2.3). Section 4.2.4 delves into matheuristics, which represent a recent trend of embedding exact mathematical programming models in a (meta)heuristic algorithm. Finally, Section 4.2.5 focuses on simulation studies.

4.2.1 Classical heuristics

The sequential **insertion heuristic** of Jaw et al. [67] sorts users according to their earliest pickup time. In each iteration, the first-sorted user is inserted at the best feasible position in the first route for which a feasible insertion is found. However, this position is not necessarily globally optimal. Diana and Dessouky [33] address this myopic behavior in two ways. First, they use a parallel insertion strategy in which the first-sorted user is inserted at the best feasible position over all routes. A seed request is chosen for each route, based on geographic decentralization. Second, a regret criterion prioritizes requests whose non-immediate insertion may harm the eventual solution quality. Toth and Vigo [127] initialize routes with a difficult request in terms of user requirements and spatial or temporal situation. Furthermore, they include a dynamic component which takes into account decentralization from initial requests in other routes. Generally, the insertion order of the remaining users may be tailored to the specific problem context (e.g. [109, 129]) or be based on time windows or randomness. If requests cannot be inserted due to earlier myopic assignments, time-overlapping users may be temporally removed and relocated to another route [78, 116]. Finally, insertion heuristics are particularly useful in a dynamic problem context, since they can add new requests without requiring re-computation of the complete solution (e.g. [23]). Luo and Schonfeld [79] find that a rolling horizon insertion strategy, which postpones the insertion of requests in the distant future, increases flexibility and reduces fleet requirements.

A **cluster-first route-second** heuristic solves the problem in two separate phases. Ioachim et al. [65] first use column generation to construct clusters, solving a set partitioning master problem. In the subproblem, being a constrained pickup and delivery shortest path problem, clusters with a minimal reduced cost are generated using dynamic programming. However, they are based on a simplified network that only consists of arcs between neighboring requests, having similarities

in terms of time, space and direction. In a second phase, a similar column generation algorithm is applied to chain these clusters. Borndörfer et al. [11] generate clusters by complete enumeration. Since possible combinations are numerous, efficient feasible routes are selected heuristically and a branch-and-cut approach is used to solve the resulting set partitioning problem. A distinction is made between different vehicle types. Karabuk [69] develops a nested column generation approach which integrates the clustering and routing decisions, rather than composing routes after the construction of the clusters. The author formulates a set partitioning problem of combining complete routes. The simultaneous construction of these routes is done solving a network flow model which is based on the entire set of clusters, provided that the lengths of user or driver rides remain reasonable.

4.2.2 Metaheuristics based on local search

The **tabu search** (TS) framework investigates moves that consist of changing a single solution attribute. In each iteration, the most improving or least deteriorating neighbor is selected, which allows to escape from local optima. Cycling is avoided using a short-term memory (tabu list). Cordeau and Laporte [20] relocate a single request to another route according to the critical vertex rule, which first inserts the node having the tightest time window. Infeasible intermediate solutions and long-term repetitive insertions are penalized in an evaluation function with dynamic weights. An aspiration criterion allows tabu moves which improve the best-found solution containing that tabu attribute. Additional intra-route local search is performed after a given number of iterations. Various authors [7, 32, 60, 85, 95] apply a comparable tabu search strategy, often adapted to a richer problem context. Attanasio et al. [5] present a parallel implementation in a dynamic context. Wolfler Calvo and Touati-Moungla [128] and Kirchler and Wolfler Calvo [70] suggest granular TS. Based on the cost of feasible request combinations, an assignment problem is solved to obtain clusters of close requests. A granular neighborhood is defined using reduced cost information. The authors require intermediate feasibility and ensure diversification through a variable length of the tabu list and an adaptive granular threshold. Except for short running times, however, classical TS is preferred because of its more extensive search area. Toth and Vigo [127] implement tabu thresholding, which is a variant which requires no memory structure. Cycling is avoided by splitting the neighborhood into subsets and selecting the most improving or least deteriorating neighbor within a single subset. The relocate operator is supplemented with exchange and chain operators. Finally, TS constitutes the improvement phase in the hybrid greedy randomized adaptive search (hybrid GRASP) of Guerriero et al. [48].

The **variable neighborhood search** (VNS) framework exploits the insight that neighborhoods are defined with respect to a particular operator. Switching between multiple operators may allow to escape from local optima. Parragh et al. [100] alternate between three inter-route operator types with different sizes, being the exchange operator, chain operator and zero-split operator. The latter selects a natural sequence of nodes, meaning that the vehicle is empty at the start and at the end of this sequence. As in Cordeau and Laporte [20], a dynamic evaluation function penalizes infeasible intermediate solutions. Promising solutions are subjected to additional (intra-route) local search, which is a common intensification technique in (meta)heuristic approaches. Additional local search is also applied on a small percentage of other solutions to foster diversification. Muelas et al. [88] propose additional relocate and exchange neighborhoods and sort their operators according to past performance. Their distributed variant [89] solves large-scale applications containing up to 16,000 requests by creating independent and equally sized subsets. Detti et al. [32] pay specific attention to operators that perturbate slightly infeasible solutions in order to attain feasibility. Schilde et al. [119, 120] adapt VNS to a problem with stochastic requests, using a simple average indicator to handle different future scenarios. In Parragh et al. [102], a pool of unserved (unprofitable) requests is introduced and intermediate load violations are prohibited, since requests can be split. Local search

may rejoin split requests. In a bi-objective context, Parragh et al. [101] apply iterated VNS for multiple weight combinations using a weighted-sum objective function, whereas Molenbruch et al. [87] integrate a variable neighborhood descent approach into a multi-directional local search framework.

The **large neighborhood search** (LNS) framework first removes a considerable percentage of requests, after which an attempt is made to insert these requests in a better way. Typical destroy (ruin) operators [115, 121] are random removal, worst removal, sequential removal, route removal and related removal (also referred to as Shaw [122] removal). Common repair (recreate) operators [115, 121] are random insertion, greedy insertion, k -regret insertion, most-constrained-first insertion and space-time-related insertion. Operators are usually tuned and evaluated for a specific problem variant (e.g. [51]) and may even be tailored to this context. For example, Masson et al. [83] develop operators which take into account transfers, whereas Lehuédé et al. [72] exploit the specific context of common destination locations and Braekers and Kovacs [13] create a multi-period variant with driver consistency. Ritzinger et al. [111] propose block-based operators, which remove zero-split sequences [100]. These sequences are either reinserted as a whole or decomposed in individual requests. In a multi-objective context, Lehuédé et al. [72] include parameterized repair operators, which reduce computational efforts by focusing on a single objective. Furthermore, the basic LNS framework is often extended. Qu and Bard [106], Masson et al. [83], Li et al. [73] and Masmoudi et al. [81] implement adaptive LNS [115], in which the probability of selecting an operator depends on its past performance. Qu and Bard [106] also suggest a multi-start strategy and solve a maximum diversity problem to decide which initial solutions are actually used.

The **deterministic annealing** (DA) framework accepts deteriorations which are smaller than a gradually lowered threshold. Braekers et al. [12] apply relocate, exchange, 2-opt*, r -4-opt and route elimination operators in a random order. The r -4-opt operator simplifies 4-opt [74] by selecting successive arcs only. Preliminary feasibility checks reduce the invocations of a scheduling procedure. Braekers et al. [12] obtain the best-known pure local search solutions for the common benchmark instances of the single-depot and multi-depot problem (see Section 4.3). Neven et al. [90] use deterministic annealing of Braekers et al. [12] to predict resource requirements for round-the-clock transportation in Flanders. Molenbruch et al. [86] apply the same algorithm to analyze the effect of service level variations on the operational costs for service providers. A distinction is made between various operating circumstances, including the size of the service provider, traffic conditions in the service area and heterogeneity of users. **Simulated annealing** (SA) accepts deteriorating solutions according to a stochastic process. It is usually combined with characteristics from other metaheuristic frameworks. For example, Baugh et al. [6] and Mauri and Lorena [84] apply an acceptance criterion based on SA, but also encourage diversification by integrating a tabu list and three different neighborhoods (intra-route relocation, inter-route relocation and exchange), respectively. Masmoudi et al. [81] integrate a population-based component into DA and SA by means of a hybrid bee algorithm, which outperforms the results of Braekers et al. [12] for the multi-depot DARP.

Similarly, **evolutionary local search** is a hybridization of the **iterated local search** metaheuristic, starting each iteration with a population of perturbed solutions. All are subjected to local search and the best result serves as the new incumbent solution. Chassaing et al. [15] invoke six local search operators, having dynamically adapted activation probabilities that stimulate convergence. Their clone detection principle avoids that previously obtained solutions are visited too often. This approach improves the results of Braekers et al. [12] for some instances of the standard DARP.

Upon reaching a local optimum, Xiang et al. [130, 131] diversify their search using a **secondary objective**. Local search relocates one or two users or performs exchanges, alternatingly minimizing the primary and secondary objective. These objectives should ideally guide the search towards the

same global optimum, following alternative paths. The sacrificing principle of Healy and Moll [57] is a similar approach, but the deterioration of the first objective value is bounded.

4.2.3 Metaheuristics based on a population

Genetic algorithms (GA) combine desirable characteristics of two parent solutions, making use of a specific encoding scheme and crossover technique. A grouping GA (e.g. [55, 68, 110]) uses an encoding scheme which clusters users into routes and requires a separate heuristic to determine the order of service within each route. Jørgensen et al. [68] perform crossovers by randomly selecting user-vehicle combinations from both parents. Rekiek et al. [110] insert clusters from the first parent into the second and correct for double requests. Other GA (e.g. [4, 16, 27, 28, 134]) use a variety of encoding schemes in which both the clustering of users and their order in the route is determined. However, such representations make it even more difficult to define tailored crossover operators which strategically combine good solution characteristics. General crossovers are usually invoked, such as the uniform crossover, one-point crossover, two-point crossover and partially matched crossover. To reduce the number of infeasible solutions found, Cubillos et al. [28] propose preprocessing techniques, a list of incompatible users and the use of a precedence table. Zhang et al. [134], who consider a multi-trip problem variant, combine trips (segments starting and ending at the depot) from both parents. If trips cannot be inserted, the remaining requests are added individually using k -regret insertion. All aforementioned authors include sporadic mutations which foster diversification. Parent selection may be based on a tournament principle or (in multi-objective approaches) crowding distance.

Hybrid GA, also referred to as memetic algorithms, extend an evolutionary framework with local search. For example, Chevrier et al. [16] add a 2-opt operator and Zhang et al. [134] apply relocate and exchange operators after each crossover. The highly effective hybrid GA of Masmoudi et al. [80] includes two crossover operators, based on a sequencing strategy using a one-point crossover and a merge strategy selecting individual genes from the parent solutions, an additional local search phase and four different mutation operators. This approach outperforms the results of both Chassaing et al. [15] and Braekers et al. [12].

Parragh et al. [101] and Molenbruch et al. [87] invoke **path relinking** (PR) to intensify their approximation of the Pareto frontier in a bi-objective problem context. PR is a stepwise conversion between two solutions in the solution set. By combining desirable characteristics of both, new non-comparable or even dominating solutions might be discovered. Mapping the routes from both solutions is done randomly, according to the number of identical requests or according to the number of transformations needed. During the actual procedure, a random weighted-sum function determines the most appropriate move.

4.2.4 Matheuristics

Hybrid solution approaches that embed exact mathematical programming models in a (meta)heuristic algorithm are defined as **matheuristics**. Archetti and Speranza [2] describe them as an emerging trend in vehicle routing. Matheuristics combine the advantages of exploiting the specific problem structure and using high-performance mathematical solvers. They can be classified into three classes, being decomposition approaches, improvement heuristics and column generation approaches [2]. Despite their excellent performances, matheuristics are relatively scarce in the literature on the DARP.

In the class of **decomposition approaches**, Oberscheider and Hirsch [92] first perform an exact optimization of users that can be combined in the same vehicle, taking into account time

preferences, ride times and combination constraints. Based on the resulting clusters, a tabu search strategy is applied to optimize the routing problem. The current approach does not allow intermediate reclustering, which would be useful to broaden the search space. The class of **improvement heuristics** is represented by hybrid large neighborhood search (hybrid LNS) algorithms. Jain and Van Hentenryck [66] apply constraint programming to complete a partial solution in which random users have been removed. The reinsertion order is based on the remaining number of feasible vehicles and insertion points, as well as on the insertion cost. At the end of each iteration, non-improving complete solutions may also be accepted. Ritzinger et al. [111] perform insertions using a dynamic programming algorithm, which adopts a giant tour representation to deal with multiple vehicles. A restricted version selects the most promising states throughout the procedure. The search strategy in Espinoza et al. [38] can be described as hybrid adaptive LNS. Various route removal operators are applied, based on similarity criteria and selected according to dynamic probabilities. The associated requests are restructured using integer programming. Since changes involve a limited number of routes, this approach is particularly suitable for parallel execution. In the class of **column generation** (CG) approaches, Parragh and Schmid [103] identify feasible routes with a negative reduced cost in a non-exhaustive manner. Variable neighborhood search (VNS) is performed on individual routes in the solution. After a given number of CG iterations, large neighborhood search (LNS) is applied on the complete solution, invoking classical operators [115]. All newly generated routes may be used in a subsequent execution of the CG procedure. Crawford et al. [24, 25] implement hybrid ant colony optimization (hybrid ACO) to select columns which represent clusters of requests. Guided by a short-term memory (pheromone trail) regarding the profitability of columns, ants select columns until a complete solution is reached. Constraint programming is integrated into the selection phase, such that the number of possible additional columns is reduced through the identification of constraint violations.

Apart from the aforementioned classification of matheuristics, solution approaches may combine full versions of exact and (meta)heuristic algorithms. This is particularly beneficial in a dynamic problem context. Berbeglia et al. [9] combine tabu search (TS) [20] and constraint programming (CP) [10] to determine whether a newly received request can be accepted. It is rejected if CP proves that the insertion is infeasible or both methods cannot find a feasible insertion within a certain time frame. Since CP performs well in the presence of tight constraints and TS easily discovers insertions if constraints are weak, the strengths of both methods are combined. After the insertion of a request, TS keeps running continuously to improve the unexecuted part of the solution.

4.2.5 Simulation studies

Several authors design **simulation** systems to gain a better understanding of dial-a-ride systems. They mostly analyze the impact of parameter changes concerning the design of the system and the way it is operated. Fu [42] quantifies the benefits of technological enhancements in automatic vehicle location (AVL) systems, which allow service providers to track and redirect vehicles. Such systems particularly increase productivity and reliability in the presence of uncertain information, such as stochastic travel times or service durations. Schilde et al. [119], who assume that inbound requests may be predicted through stochastic information from historical data, develop a simulation framework to continuously keep track of requests and to provide information on newly arising requests during the execution. In Schilde et al. [120], this framework is extended to handle stochastic time-dependent travel times. Deflorio et al. [29] report considerable effects on operational efficiency and service quality caused by the punctuality of users (resp. drivers), as well as the patience they exhibit while waiting for drivers (resp. users) being late. Quadrifoglio et al. [108] analyze the effect of splitting a service area into zones, which are assigned to different service providers. Real-life spatial and temporal demand distributions with two user types and potential no-shows are used. Shen and Quadrifoglio [124] add different time distributions for requests in high-density areas.

More centralized strategies are found to reduce fleet size and empty mileage, but cause larger deviations from user time preferences. Shen and Quadrioglio [123] model inter-zone trips as two intra-zonal requests, imposing transfers at specific locations. These transfers enhance the number of user trips per vehicle revenue hour without substantially deteriorating service quality. Hyytiä et al. [64] compare different route selection policies in a system with stochastic requests, simulating both random and clustered demand. Routes are approximated as M/M/1 queues and route selections are modeled as a Markov decision process. Häll et al. [50] develop a graphical user interface for a simulation system (DARS) which integrates multiple solution methods. By means of example, they assess the importance of user-related costs (waiting time and excess ride time), given two kinds of users and partly dynamic requests. More details on agent-based simulation approaches in demand-responsive transportation can be found in a recent review by Ronald et al. [112].

4.3 Benchmark data

Since many contributions in the literature are based on practical problems, algorithms are often tested on case-specific **real-life instances**. Such an approach is useful to demonstrate the practical applicability of solution methods, but causes difficulties in comparing their efficiency. Consequently, two sets of **artificial benchmark data** have been proposed to perform computational tests. This section discusses the characteristics of these instances and the efficiency of various algorithms in solving them.

The data set introduced by Cordeau [19] and extended by Røpke et al. [114] consists of 42 instances, including 16 to 96 requests. An instance is structured as follows. Half of all requests are outbound, whereas the others represent inbound requests. Origins and destinations are randomly and independently generated in a square region $[-10, 10]^2$. The depot is located in the center. The data set can be subdivided into two groups of 21 instances. The *a*-instances consider a single user to be picked up or delivered at each location, whereas the *b*-instances assume that users may travel in group. A uniform distribution is used to determine the number of users picked up or delivered at each location and the corresponding service duration. Time windows of 15 minutes are created for destinations of an outbound request, as well as for origins of an inbound request. Travel times equal Euclidean distances. A fixed maximum user ride time of 30 minutes and a fixed service duration of 3 minutes are taken into account. Vehicle capacity is fixed at 3 customers. The maximum route duration is dependent on the instance and ranges from 240 to 720 minutes. Several authors extended this data set to richer problem variants, including heterogeneous customers and multiple vehicle types [96], multiple depots [12] and breaks [81]. Berbeglia et al. [9, 10] assume that part of the requests are dynamically revealed. Parragh et al. [102] use the *b*-instances to test the effect of split requests.

The data set of Cordeau and Laporte [20] consists of 20 instances, containing between 24 and 144 requests. The requests are generated in a comparable manner as described before, but clustered around a number of seed points. The coordinates of the depot are the averages over those of the seed points. A single user is picked up or delivered at each location. Half of the instances have time windows of 30 minutes, whereas wider time windows of 60 minutes are defined for the other half. A fixed maximum user ride time of 90 minutes and a fixed service duration of 10 minutes are imposed. Vehicle capacity is fixed at 6 customers and the maximum route duration equals 480 minutes for all instances. Richer problem characteristics have also been introduced into this data set, such as heterogeneity, multiple depots and breaks [81] and the combination of parcels and people [73]. Ho and Haugland [60] define probabilities with which certain requests may be submitted and Masson et al. [83] extend the instances with one or several transfer points.

Exact solution approaches are usually tested on the data of Røpke et al. [114]. To date, the most efficient results have been obtained by the branch-and-cut-and-price method of Gschwind and Irnich [47], solving all instances to optimality. Approximate solution methods are often tested on both artificial data sets. State-of-the-art heuristics are the hybrid genetic algorithm of Masmoudi et al. [80] and the evolutionary local search of Chassaing et al. [15]. Note that such efficient combinations of local search and population characteristics are a recent development. Other good results have been obtained by solution methods based on pure local search, such as the deterministic annealing metaheuristic of Braekers et al. [12].

4.4 Discussion and research opportunities

Over three decades of research on the DARP has resulted in the development of numerous solution methods. Initially, **exact** solutions were obtained through dynamic programming. The use of this technique is restricted to small instances, preferably involving a single vehicle. During the past decade, branch-and-cut approaches on efficient two-index formulations have been proposed [114]. Formulations and valid inequalities have been extended to various rich problem variants [12, 76]. Because of their efficiency, column generation and branch-(and-cut)-and-price (e.g. [47]) approaches have recently gained attention. Since these solution approaches currently obtain the fastest exact solutions for artificial instances with up to 96 requests and 8 vehicles, their application to rich problem variants and larger instances is an important direction for future research.

The NP-hardness of the DARP explains why most authors develop **approximate** solution methods. Initially, a variety of **classical heuristics** was presented in the literature, mainly focussing on standard problem characteristics. Insertion heuristics and cluster-first-route-second heuristics were the most common types. The former are also useful to update solutions in a dynamic setting. The further development of approximate solution methods was characterized by two major shifts of attention. The first one occurred when Cordeau and Laporte [20] proposed an efficient implementation of tabu search. Since then, authors have designed metaheuristics for various problem variants. The efficiency of such a metaheuristic framework is not an intrinsic feature, but depends on its specific implementation. Numerous design choices, such as the representation of the solution, the definition of neighborhoods and the balance between intensification and diversification should be tailored to the specific problem context. Nevertheless, **metaheuristics based on local search** (e.g. VNS [100] and DA [12]) most often obtained good results within realistic computation times. They typically use classical neighborhood types, such as relocating or exchanging one or several users between routes, supplemented with intra-route optimization. However, solutions methods for rich problem variants benefit from additional operators that take advantage of the specific structure. For example, this is demonstrated by the transfer-related destroy and repair operators in Masson et al. [83]. The further development of such operators for other rich problem variants may be addressed in future work. A second, more recent shift of attention is the integration of multiple types of solution approaches. Several state-of-the-art metaheuristics benefit from **integrating** a population concept into local search approaches [15, 81] or vice versa [80]. Implementations of pure population-based metaheuristics proved less effective in the past, as it is difficult to define crossover operators which combine desirable characteristics of two parent solutions in a strategic manner. Yet, combining multiple types of crossovers and mutations in a genetic algorithm with additional local search operators can be highly effective [80]. Besides, hybridizations of exact and approximate solution methods have been proposed in the literature. Results of the first implementations, such as the hybrid column generation approach of Parragh and Schmid [103], show that this is a promising and relatively unexplored field that offers interesting opportunities for future research.

5 Scheduling procedures

This section compares various procedures for solving the scheduling subproblem, which consists of determining the start time of the service at each node in a route, given all time-related constraints and the order by which the nodes are visited. Finding a correct time schedule is an essential condition to prove the feasibility of a route. Most of the scheduling procedures discussed below have a general applicability, rather than being restricted to a specific problem variant or solution method.

Most frequently used is the eight-step scheduling procedure of Cordeau and Laporte [20], adopting the forward time slack principle [118] to successively minimize violations of time windows, route duration and violations of maximum user ride time. Since the last two phases do not influence violations in the previous phase(s), it may also be used in algorithms that should maintain feasibility at all times. First, the start of service in each node is initialized at the maximum of (1) the lower time window bound and (2) the earliest time by which the node can be reached. Second, the start of service in the origin depot is delayed as much as possible without causing additional time window violations, such that total route duration is minimized. The service starts in all following nodes are updated, which never increases user ride times. Third, an analogous shift is considered for all pickup nodes, since delaying the start of service in a pickup node may eliminate a violation of maximum user ride time. All pickup nodes are investigated one by one, using a forward loop through the route. The service start in a pickup node is delayed as much as possible, without causing additional time window violations, maximum user ride time violations or route duration increases. The starts of service in all following nodes are updated after each shift. Parragh et al. [101] observe that reducing a user's ride time (violation) may increase - but not violate - the ride time of other users aboard the vehicle. To approximate the minimal total user ride time, they propose a more conservative computation of the forward time slack, even though this modification causes incorrect infeasibility declarations. If minimizing total user ride time is not an objective and all ride times are feasible, the third phase may be skipped to reduce computation time [100]. Molenbruch et al. [87] propose a scheduling heuristic that minimizes total user ride time according to a different strategy. Starting from a schedule with minimal ride times for the given time windows, potential travel time shortages are eliminated while keeping ride time increases as limited as possible. A variant imposing constraints on the allocation of waiting time is presented.

The aforementioned scheduling procedures based on the forward time slack principle have $O(n^2)$ complexity. However, Hunsaker and Savelsbergh [62] argue that the earliest feasible schedule for a DARP with maximum waiting times in each node can be determined in linear time. After initiating arrivals and departures at any node to the earliest possible time, a backward pass through the route postpones the pickup time of users whose maximum ride time is violated. If there is still any ride time violation left, waiting time is shifted. Both Tang et al. [125] and Haugland and Ho [56] detect incorrect infeasibility declarations. First, no waiting time is modeled at the last node in the route. Second, the backward pass corrects departures in pickup nodes without delaying the corresponding destination. This may allow larger waiting time shifts, but increases complexity to $O(n^2)$ [125]. Tests on a related problem show that this worst-case complexity is rarely met [46]. Haugland and Ho [56] propose a correction which results in a $O(n \log(n))$ complexity. Relaxing all upper time window bounds, they minimize the departure times at nodes for which the lower time window bound is a binding constraint. Firat and Woeginger [40] solve a shortest path problem vertex-weighted interval graph. Their formulation of linear inequalities uses a single decision variable, indicating waiting time before leaving a node. All constraints are expressed as a difference constraint system. A feasible schedule exists if the graph does not include negative-weight cycles. A feasible schedule is obtained with linear complexity, as confirmed by computational tests in Chassaing et al. [15].

Finally, some specific contributions to the scheduling subproblem are presented. The heuristic of Jaw et al. [67] divides routes into schedule blocks, being continuous active periods. It relies on the simplifying assumption that no waiting time is scheduled while users are aboard the vehicle. This approach is particularly useful with nonconstant travel times [41, 120], as blocks can be scheduled independently. Braekers et al. [12] introduce preliminary checks which may prove infeasibility of a route before any scheduling procedure is executed. They compute the earliest arrival and latest departure in each node on the basis of travel times, time windows and service durations, not taking into account user ride times. Finally, specific procedures have been presented for certain problem variants. Masson et al. [82] consider route interdependencies due to transfers and show that evaluating feasibility is equivalent to checking the consistency of a simple temporal problem. Zhang et al. [134] deal with scheduling for the multi-trip problem variant. Garaix et al. [44] approach scheduling as a multi-objective problem in which nodes are connected by multiple Pareto-optimal paths. For example, costs and travel times should be balanced when congestion or multimodal transportation are taken into account. The scheduling subproblem becomes an NP-hard fixed-sequence arc selection problem, which is solved using dynamic programming. Gschwind [46] generalizes the forward time slack principle and two of the aforementioned scheduling procedures [40, 125] to a problem variant with both minimum and maximum ride time constraints.

6 Conclusion

Recent literature on dial-a-problems takes into account increasingly more **real-life problem characteristics** that generalize the standard definition. This work presents a thorough classification of all existing problem variants. Variants that involve heterogeneity, more complex routing properties and stochastic or dynamic information have gained interest in recent years. Although this encouraging trend enhances the practical applicability of solution methods, surveys among users still reveal an insufficient match between theoretical problem characteristics and their actual concerns in daily practice. This is due to the fact that authors fail to combine real-life characteristics from different categories. From a managerial point of view, research should to a larger extent focus on analyzing the impact of particular problem characteristics on the service provider's operational efficiency and the service quality offered to users. An extensive discussion of future research opportunities related to the definition of problem variants can be found in Section 3.5. For example, a new perspective in patient transportation consists of combining health care scheduling and vehicle routing, rather than considering user requests as an input to the routing problem.

From an algorithmic point of view, the wide range of **solution methods** proposed in the literature is reviewed in a structured manner. In the past decade, exact solutions for relatively small artificial instances have been obtained using branch-and-cut, column generation or branch-(and-cut)-and-price approaches. Due to the NP-hardness of the problem, approximate methods have been invoked to solve larger problems with diverse characteristics. In particular, metaheuristics based on local search have obtained good results within realistic computation times. Their performance on rich problem variants can still be improved using operators that exploit the specific problem structure. In that respect, it may also be useful to investigate whether certain search operators and algorithmic components perform better in a particular problem context. Furthermore, hybridization is a recent trend to improve the performance of solution methods. For example, combinations of local search and population strategies, as well as combinations of exact and approximate methods constitute a promising and relatively unexplored research area. More interesting topics for future work are discussed in Section 4.4.

Solution methods are often tested on real-life instances. To compare them among each other, two sets of artificial benchmark data are commonly used for a limited number of problem variants.

For other variants, either no artificial benchmark data exist or data have been created to the sole discretion of authors. A more general applicability is desirable to compare solution techniques for all rich problem variants. In addition, the number of users and vehicles in the artificial instances is relatively small in comparison with real-life systems and some reasonable assumptions are ignored, such as the fact that users normally request corresponding inbound and outbound trips on the same day. In that respect, computational analyses should devote more attention to the robustness of the solution methods, verifying whether they produce equally efficient solutions if different operational characteristics apply.

Finally, the interested reader is referred to the extensive **overview table** in Appendix A. This overview table provides full details on the problem characteristics and solution methods applied in each of the papers discussed in this work. Column 1 contains the references to these papers in alphabetic order. Columns 2-24 correspond to the problem characteristics discussed in Section 3. They indicate whether a certain problem characteristic is studied in the corresponding paper. More specifically, columns 2-6 refer to the characteristics of the standard DARP (Section 3.1), columns 7-13 to heterogeneity (Section 3.2.1), columns 14-18 to more complex routing properties (Section 3.2.2) and columns 19-24 to stochastic or dynamic information (Section 3.4). Column 25 gives more details on the solution method applied (Section 4). Columns 26-32 categorize the objectives, whereas Column 33 enumerates all components taken into account in the objective function. Columns 34-37 mention the data sets used to test the corresponding solution method (Section 4.3). Finally, an explanation of the abbreviations used can be found below the table.

Acknowledgements This work is supported by the Interuniversity Attraction Poles Programme, initiated by the Belgian Science Policy Office. (Research Project COMEX - Combinatorial Optimization: Metaheuristics & Exact Methods)

Appendix A: Overview table

Reference	Problem characteristics								Approach	Objectives				Data																												
	User time preference	Max. user ride time	Service duration	Max. route duration	Max. vehicle capacity	Heterogeneous users	Upgrading conditions	Accompanying person		Configurable vehicles	Priority requests	Isolated transportation	Driver qualifications	Transfers	Integration	Split requests	Breaks	Multiple depots	Time-dep. travel times	Stochastic travel times	Dynamic context	Stochastic requests	Probabilistic requests	Stochastic user behavior	Efficiency of routing	User quality	Constraint violations	Fleet size or fleet cost	Requests satisfied	Other objectives	Multi-objective strategy	Based on C&L (2003)	Based on C (2006)	Other artificial	Real-life							
Atahran et al. [4]	x	x	x																														x	x								
Attanasio et al. [5]	x	x	x	x	x														x															x								
Baugh et al. [6]	x				x																														x							
Beaudry et al. [7]	x	x	x	x		x	x	x	x	x						x	x			x																x						
Berbeglia et al. [9]	x	x	x	x															x																x	x						
Borndörfer et al. [11]	x	x	x	x	x	x	x																													x						
Braekers et al. [12]	x	x	x	x	x	x	x	x																												x						
Braekers and Kovacs [13]	x	x	x	x	x	x	x	x																												x						
Chassaing et al. [15]	x	x	x	x	x																															x	x					
Chevrier et al. [16]	x	x	x	x																																	x					
Coppi et al. [18]	x	x	x	x																																	x					
Cordeau and Laporte [20]	x	x	x	x	x																																x	x				
Cordeau [19]	x	x	x	x	x																																	x				
Coslovich et al. [23]	x	x																																				x				
Crawford et al. [24, 25]	x	x	x	x	x																																					
Cremers et al. [26]	x	x			x																																	x				
Cubillos et al. [27]	x	x			x																																	x				
Cubillos et al. [28]	x				x																																	x				
Desrosiers et al. [30]	x	x	x																																			x				
Dessouky et al. [31]	x	x	x	x	x	x																																x				
Detti et al. [32]	x	x	x	x		x																																x	x			
Diana and Dessouky [33]	x	x	x																																				x			
Espinoza et al. [37, 38]	x	x	x	x																																			x			
Fu [41]	x	x			x																																		x			
Garaix et al. [45]	x	x	x	x																																			x	x	x	x

Continued on next page

Reference	Problem characteristics							Approach	Objectives				Data	
	User time preference Max. user ride time Service duration Max. route duration Max. vehicle capacity	Heterogeneous users Upgrading conditions Accompanying person Configurable vehicles Priority requests Isolated transportation Driver qualifications	Transfers Integration Split requests Breaks Multiple depots	Time-dep. travel times Stochastic travel times Dynamic context Stochastic requests Probabilistic requests Stochastic user behavior					Efficiency of routing User quality Constraint violations Fleet size or fleet cost Requests satisfied Other objectives Multi-objective strategy				Based on C&L (2003) Based on C (2006) Other artificial Real-life	
Gschwind and Irnich [47]	x x x x							BCP	x			Routing costs		x
Guerriero et al. [48]	x x x x x							H-GRASP	x			Routing costs	x	x x
Häll et al. [49]	x x x x x	x						BC	x			Routing costs		x
Häll and Peterson [51]	x x x x	x x						LNS	x x			Routing costs, vehicle waiting time, user ride time, user waiting time		x
Häme [52]	x x x							DP	x x			Route duration, user waiting time, user ride time		x
Häme and Hakula [53]	x x x x							DP		x		Users served		x
Häme and Hakula [54]	x x x x							DP			x	Users served		x
Hanne et al. [55]	x x x x	x x x x x						IH, GA	x x x			Total lateness, total earliness, vehicle travel time, total user ride time		x
Healy and Moll [57]								SOF	x			Routing costs (first); routing costs over neighborhood size (second)		x
Heilporn et al. [58]	x x x x x							BC	x x			Routing costs, expected delays		x
Ho and Haugland [60]	x x x x x							TS	x			Expected routing costs	x	
Hu and Chang [61]	x x x x							BP	x			Routing costs		x
Ioachim et al. [65]	x x x x	x						CG	x			Distance		x
Jain and Van Hentenryck [66]	x x x x x							H-LNS	x			Routing costs	x	
Jaw et al. [67]	x x x x							IH	x x		x	Deviation from user preference time, user ride time, active vehicle time, vehicle slack time, system workload		x x
Jørgensen et al. [68]	x x x x x							GA	x x x			Vehicle travel time, user ride time, user waiting time, route duration, time window violations, maximum user ride time violations, maximum route duration violations	x	
Karabuk [69]	x x x x	x x x						CG	x		x	Users served, distance		x
Kirchler and Wolfler Calvo [70]	x x x x x							TS	x x		x	Routing costs, user ride time, user waiting time aboard, route duration, early arrivals, users served	x	
Lehuédé et al. [72]	x x x x x							LNS	x x	x	x	Maximum user ride time, user ride time, user waiting time, vehicle travel time, fleet size		x
Li et al. [73]	x x x x							ALNS	x		x	Profit from parcels, profit from user rides, routing costs, discount for detours	x	x
Liu et al. [76]	x x x x	x x x						BC	x			Vehicle travel time		x
Luo and Schonfeld [78]	x x x x							IH	x x			Vehicle travel time, user ride time		x x
Luo and Schonfeld [79]	x x x x							IH	x x			Vehicle travel time, user ride time		x
Masmoudi et al. [80]	x x x x x	x x x						H-GA	x			Routing costs	x	x
Masmoudi et al. [81]	x x x x x	x x x						ALNS, H-BA	x			Routing costs	x	x

Continued on next page

Reference	Problem characteristics							Approach	Objectives				Data	
	User time preference Max. user ride time Service duration Max. route duration Max. vehicle capacity	Heterogeneous users Upgrading conditions Accompanying person Configurable vehicles Priority requests Isolated transportation Driver qualifications	Transfers Integration Split requests Breaks Multiple depots	Time-dep. travel times Stochastic travel times Dynamic context Stochastic requests Probabilistic requests Stochastic user behavior					Efficiency of routing User quality Constraint violations Fleet size or fleet cost Requests satisfied Other objectives Multi-objective strategy				Based on C&L (2003) Based on C (2006) Other artificial Real-life	
Masson et al. [83] Mauri and Lorena [84]	x x x x x x x x x x		x					ALNS SA	x x x x x	Distance Distance, fleet size, route duration, user ride time, waiting time, maximum route duration violations, maximum user ride time violations, maximum waiting time violations, vehicle capacity violations, time window violations	x x	x		
Melachrinoudis et al. [85]	x					x		TS	x x x	Routing costs, time window violations, user ride time		x		
Molenbruch et al. [86] Molenbruch et al. [87]	x x x x x x x x x x	x x x						DA MDLS	x x x	Distance, user ride time	x x x			
Muelas et al. [88] Muelas et al. [89] Neven et al. [90] Paquette et al. [95]	x x x x x x x x x x x x x x x x x					x		VNS VNS DA TS	x x x x x x	Routing costs Routing costs Fleet size, routing costs Routing costs, user waiting time, user ride time		x x x x		
Parragh et al. [101] Parragh et al. [100] Parragh [96] Parragh et al. [97] Parragh et al. [102] Parragh and Schmid [103] Psaraftis [104]	x x							IVNS VNS BC CG BP, VNS H-CG, LNS DP	x x x x x x x x x	Routing costs, user ride time Routing costs Routing costs, vehicle waiting time Routing costs Profits Routing costs Vehicle travel time, user waiting time, user ride time	x x x x x x x x	x x x		
Psaraftis [105] Qu and Bard [106] Reinhardt et al. [109] Rekiek et al. [110]	x x x x x x x x x x x x x	x x x				x		DP ALNS SA GA	x x x x x x x	Vehicle travel time Fleet size, distance, user ride time Users served, user ride time Vehicle efficiency (= exploitation of vehicle capacity)		x x x x x x		
Ritzinger et al. [111] Røpke et al. [114] Rubinstein et al. [116] Santos and Xavier [117] Schilde et al. [119]	x x x x x x x x x x x x x x x x x x x							H-LNS BC IH GRASP SVNS	x x x x x x x x x x x	Total vehicle travel time Routing costs Time window violations Users served, cost payed by all users Time window violations, fleet size, route duration	x x x x x x x x	x		
Schilde et al. [120]	x x x x							SVNS	x x x x	Time window violations, user ride time violations, fleet size, route duration		x		
Toth and Vigo [127]	x x x x	x x						TT	x x	Routing costs, deviation from user preference time		x		

Continued on next page

Reference	Problem characteristics							Approach	Objectives				Data	
	User time preference Max. user ride time Service duration Max. route duration Max. vehicle capacity	Heterogeneous users Upgrading conditions Accompanying person Configurable vehicles Priority requests Isolated transportation Driver qualifications	Transfers Integration Split requests Breaks Multiple depots	Time-dep. travel times Stochastic travel times Dynamic context Stochastic requests Probabilistic requests Stochastic user behavior					Efficiency of routing User quality Constraint violations Fleet size or fleet cost Requests satisfied Other objectives Multi-objective strategy				Based on C&L (2003) Based on C (2006) Other artificial Real-life	
Wolfler Calvo and Touati-Moungla [128]	x	x		x			TS	x	x	x	Users served, fleet size, user ride time		x	
Wong and Bell [129]	x	x	x				IH	x	x		Vehicle travel time, user ride time, taxi costs		x	
Xiang et al. [130]	x	x	x			x	SOF	x	x		Routing costs, driver wage (first); empty distance, wage for driving an empty vehicle or waiting, user ride time (second)		x	
Xiang et al. [131]	x	x	x	x			SOF	x	x		Routing costs, driver wage (first); empty distance, wage for driving an empty vehicle or waiting, user ride time (second)		x	
Zhang et al. [134]	x	x	x				H-GA	x		x	Requests served (first), total vehicle travel time (second)	x	x	x
Ziliaskopoulos and Kozanidis [135]	x	x	x				DP	x			Vehicle travel time	x		x

Solution methods:

ACO = ant colony optimization, (A)LNS = (adaptive) large neighborhood search, BA = bee algorithm, BC = branch-and-cut, BCP = branch-and-cut-and-price, BP = branch-and-price, CG = column generation, CP = constraint programming, DA = deterministic annealing, DP = dynamic programming, ELS = evolutionary local search, FL = fuzzy logic, GA = genetic algorithm, H = hybrid, IH = insertion heuristic, (I/S)VNS = (iterated/stochastic) variable neighborhood search, MDLS = multi-directional local search, SA = simulated annealing, SOF = secondary objective function, TS = tabu search, TT = tabu thresholding

References

1. Aldaihani, M., Dessouky, M. M., 2003. Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering* 45 (1), 75–96.
2. Archetti, C., Speranza, M. G., 2014. A survey on mathheuristics for routing problems. *EURO Journal on Computational Optimization* 2 (4), 223–246.
3. Ascheuer, N., Krumke, S. O., Rambau, J., 2000. Online dial-a-ride problems: Minimizing the completion time. *Lecture Notes in Computer Science* 1770, 639–650.
4. Atahran, A., Lenté, C., T'kindt, V., 2014. A multicriteria dial-a-ride problem with an ecological measure and heterogeneous vehicles. *Journal of Multi-Criteria Decision Analysis* 21 (5-6), 279–298.
5. Attanasio, A., Cordeau, J.-F., Ghiani, G., Laporte, G., 2004. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing* 30 (3), 377–387.
6. Baugh, J. W., Kakivaya, G. K. R., Stone, J. R., 1998. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization* 30 (2), 91–123.
7. Beaudry, A., Laporte, G., Melo, T., Nickel, S., 2010. Dynamic transportation of patients in hospitals. *OR Spectrum* 32 (1), 77–107.
8. Berbeglia, G., Cordeau, J.-F., Laporte, G., 2010. Dynamic pickup and delivery problems. *European Journal of Operational Research* 202 (1), 8–15.
9. Berbeglia, G., Cordeau, J.-F., Laporte, G., 2012. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing* 24 (3), 343–355.
10. Berbeglia, G., Pesant, G., Rousseau, L.-M., 2011. Checking the feasibility of dial-a-ride instances using constraint programming. *Transportation Science* 45 (3), 399–412.
11. Borndörfer, R., Grötschel, M., Klostermeier, F., Küttner, C., 1999. *Telebus Berlin: Vehicle scheduling in a dial-a-ride system*. *Lecture Notes in Economics and Mathematical Systems* 471, 391–422.
12. Braekers, K., Caris, A., Janssens, G. K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological* 67, 166–186.
13. Braekers, K., Kovacs, A., 2016. A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological* 94, 355–377.
14. Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A. A., 2014. Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)* 47 (2), 32.
15. Chassaing, M., Duhamel, C., Lacomme, P., 2016. An ELS-based approach with dynamic probabilities management in local search for the dial-a-ride problem. *Engineering Applications of Artificial Intelligence* 48, 119–133.
16. Chevrier, R., Liefoghe, A., Jourdan, L., Dhaenens, C., 2012. Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. *Applied Soft Computing* 12 (4), 1247–1258.
17. Clímaco, J. C. N., Craveirinha, J. M. F., Pascoal, M. M. B., 2006. An automated reference point-like approach for multicriteria shortest path problems. *Journal of Systems Science and Systems Engineering* 15 (3), 314–329.
18. Coppi, A., Detti, P., Raffaelli, J., 2013. A planning and routing model for patient transportation in health care. *Electronic Notes in Discrete Mathematics* 41, 125–132.
19. Cordeau, J.-F., 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54 (3), 573–586.
20. Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological* 37 (6), 579–594.
21. Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: Models and algorithms. *Annals of Operations Research* 153 (1), 29–46.
22. Cortés, C. E., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research* 200 (3), 711–724.
23. Coslovich, L., Pesenti, R., Ukovich, W., 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research* 175 (3), 1605–1615.

24. Crawford, B., Castro, C., Monfroy, E., 2007. Solving dial-a-ride problems with a low-level hybridization of ants and constraint programming. *Lecture Notes in Computer Science* 4528, 317–327.
25. Crawford, B., Castro, C., Monfroy, E., Cubillos, C., 2007. Decomposition approach to solve dial-a-ride problems using ant computing and constraint programming. *Lecture Notes in Computer Science* 4729, 448–457.
26. Cremers, M. L. A. G., Klein Haneveld, W. K., van der Vlerk, M. H., 2009. A two-stage model for a day-ahead paratransit planning problem. *Mathematical Methods of Operations Research* 69 (2), 323–341.
27. Cubillos, C., Rodríguez, N., Crawford, B., 2007. A study on genetic algorithms for the DARP problem. *Lecture Notes in Computer Science* 4527, 498–507.
28. Cubillos, C., Urra, E., Rodríguez, N., 2009. Application of genetic algorithms for the DARPTW problem. *International Journal of Computers Communications & Control* 4 (2), 127–136.
29. Deflorio, F. P., Dalla Chiara, B., Murro, A., 2002. Simulation and performance of DRTS in a realistic environment. In: *Proceedings of the 9th Meeting EWGT on Intermodality, Sustainability and Intelligent Transportation Systems and 13th Mini EURO Conference on Handling Uncertainty in the Analysis of Traffic and Transportation Systems*, Bari, Italy.
30. Desrosiers, J., Dumas, Y., Soumis, F., 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences* 6 (3-4), 301–325.
31. Dessouky, M., Rahimi, M., Weidner, M., 2003. Jointly optimizing cost, service, and environmental performance in demand-responsive transit scheduling. *Transportation Research Part D: Transport and Environment* 8 (6), 433–465.
32. Detti, P., Papalini, F., de Lara, G. Z. M., 2016. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega* (forthcoming).
33. Diana, M., Dessouky, M. M., 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological* 38 (6), 539–557.
34. Diana, M., Dessouky, M. M., Xia, N., 2006. A model for the fleet sizing of demand responsive transportation services with time windows. *Transportation Research Part B: Methodological* 40 (8), 651–666.
35. Doerner, K. F., Salazar-González, J.-J., 2014. Pickup-and-delivery problems for people transportation. In: Toth, P., Vigo, D. (Eds.), *Vehicle Routing: Problems, Methods, and Applications*. pp. 193–212.
36. Donoso, M., Sandoval, D., Cubillos, C., 2009. SIM-MADARP: An agent-based tool for dial-a-ride simulation. *Lecture Notes in Artificial Intelligence* 5925, 191–199.
37. Espinoza, D., Garcia, R., Goycoolea, M., Nemhauser, G. L., Savelsbergh, M. W., 2008. Per-seat, on-demand air transportation part i: Problem description and an integer multicommodity flow model. *Transportation Science* 42 (3), 263–278.
38. Espinoza, D., Garcia, R., Goycoolea, M., Nemhauser, G. L., Savelsbergh, M. W., 2008. Per-seat, on-demand air transportation part ii: Parallel local search. *Transportation Science* 42 (3), 279–291.
39. Feuerstein, E., Stougie, L., 2001. On-line single-server dial-a-ride problems. *Theoretical Computer Science* 268 (1), 91–105.
40. Firat, M., Woeginger, G. J., 2011. Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh. *Operations Research Letters* 39 (1), 32–35.
41. Fu, L., 2002. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research Part B: Methodological* 36 (6), 485–506.
42. Fu, L., 2002. A simulation model for evaluating advanced dial-a-ride paratransit systems. *Transportation Research Part A: Policy and Practice* 36 (4), 291–307.

43. Fu, L., Teply, S., 1999. On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. *Computer-Aided Civil and Infrastructure Engineering* 14 (5), 309–319.
44. Garaix, T., Artigues, C., Feillet, D., Josselin, D., 2010. Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operational Research* 204 (1), 62–75.
45. Garaix, T., Artigues, C., Feillet, D., Josselin, D., 2011. Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation. *Computers & Operations Research* 38 (10), 1435–1442.
46. Gschwind, T., 2015. Route feasibility testing and forward time slack for the synchronized pickup and delivery problem. Tech. rep.
47. Gschwind, T., Irnich, S., 2014. Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Science* 49 (2), 335–354.
48. Guerriero, F., Bruni, M. E., Greco, F., 2013. A hybrid greedy randomized adaptive search heuristic to solve the dial-a-ride problem. *Asia-Pacific Journal of Operational Research* 30 (1), 1–17.
49. Häll, C. H., Andersson, H., Lundgren, J. T., Värbrand, P., 2009. The integrated dial-a-ride problem. *Public Transport* 1 (1), 39–54.
50. Häll, C. H., Högberg, M., Lundgren, J. T., 2012. A modeling system for simulation of dial-a-ride services. *Public Transport* 4 (1), 17–37.
51. Häll, C. H., Peterson, A., 2013. Improving paratransit scheduling using ruin and recreate methods. *Transportation Planning and Technology* 36 (4), 377–393.
52. Häme, L., 2011. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. *European Journal of Operational Research* 209 (1), 11–22.
53. Häme, L., Hakula, H., 2013. Routing by ranking: A link analysis method for the constrained dial-a-ride problem. *Operations Research Letters* 41 (6), 664–669.
54. Häme, L., Hakula, H., 2014. A maximum cluster algorithm for checking the feasibility of dial-a-ride instances. *Transportation Science* 49 (2), 295–310.
55. Hanne, T., Melo, T., Nickel, S., 2009. Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces* 39 (3), 241–255.
56. Haugland, D., Ho, S. C., 2010. Feasibility testing for dial-a-ride problems. *Lecture Notes in Computer Science* 6124, 170–179.
57. Healy, P., Moll, R., 1995. A new extension of local search applied to the dial-a-ride problem. *European Journal of Operational Research* 83 (1), 83–104.
58. Heilporn, G., Cordeau, J.-F., Laporte, G., 2011. An integer L-shaped algorithm for the dial-a-ride problem with stochastic customer delays. *Discrete Applied Mathematics* 159 (9), 883–895.
59. Hickman, M. D., Blume, K. L., 2001. An investigation of integrated transit service. Tech. rep.
60. Ho, S. C., Haugland, D., 2011. Local search heuristics for the probabilistic dial-a-ride problem. *OR Spectrum* 33 (4), 961–988.
61. Hu, T.-Y., Chang, C.-P., 2015. A revised branch-and-price algorithm for dial-a-ride problems with the consideration of time-dependent travel cost. *Journal of Advanced Transportation* 49 (6), 700–723.
62. Hunsaker, B., Savelsbergh, M., 2002. Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters* 30 (3), 169–173.
63. Hyytiä, E., Penttinen, A., Sulonen, R., 2010. Congestive collapse and its avoidance in a dynamic dial-a-ride system with time windows. *Lecture Notes in Computer Science* 6148, 397–408.
64. Hyytiä, E., Penttinen, A., Sulonen, R., 2012. Non-myopic vehicle and route selection in dynamic DARP with travel time and workload objectives. *Computers & Operations Research* 39 (12), 3021–3030.
65. Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M. M., Villeneuve, D., 1995. A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science* 29 (1),

- 63–78.
66. Jain, S., Van Hentenryck, P., 2011. Large neighborhood search for dial-a-ride problems. *Lecture Notes in Computer Science* 6876, 400–413.
 67. Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., Wilson, N. H. M., 1986. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological* 20, 243–257.
 68. Jørgensen, R. M., Larsen, J., Bergvinsdottir, K. B., 2007. Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society* 58 (10), 1321–1331.
 69. Karabuk, S., 2009. A nested decomposition approach for solving the paratransit vehicle scheduling problem. *Transportation Research Part B: Methodological* 43 (4), 448–465.
 70. Kirchler, D., Wolfler Calvo, R., 2013. A granular tabu search algorithm for the dial-a-ride problem. *Transportation Research Part B: Methodological* 56, 120–135.
 71. Krumke, S. O., de Paepe, W. E., Poensgen, D., Lipmann, M., Marchetti-Spaccamela, A., Stougie, L., 2006. On minimizing the maximum flow time in the online dial-a-ride problem. *Lecture Notes in Computer Science* 3879, 258–259.
 72. Lehuédé, F., Masson, R., Parragh, S. N., Péton, O., Tricoire, F., 2013. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society* 65 (7), 983–1000.
 73. Li, B., Krushinsky, D., Van Woensel, T., Reijers, H. A., 2016. An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research* 66, 170–180.
 74. Lin, S., Kernighan, B. W., 1973. An effective heuristic algorithm for the traveling salesman problem. *Operations Research* 21 (2), 498–516.
 75. Lipmann, M., Lu, X., de Paepe, W. E., Sitters, R. A., Stougie, L., 2004. On-line dial-a-ride problems under a restricted information model. *Algorithmica* 40 (4), 319–329.
 76. Liu, M., Luo, Z., Lim, A., 2015. A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological* 81 (1), 267–288.
 77. Lois, A., Ziliaskopoulos, A., Aifantopoulou, G., 2007. A very large scale neighborhood heuristic algorithm for the multivehicle dial a ride with time windows. In: *Proceedings of 87th Transportation Research Board Annual Meeting, Washington DC, USA*.
 78. Luo, Y., Schonfeld, P., 2007. A rejected-reinsertion heuristic for the static dial-a-ride problem. *Transportation Research Part B: Methodological* 41 (7), 736–755.
 79. Luo, Y., Schonfeld, P., 2011. Online rejected-reinsertion heuristics for dynamic multivehicle dial-a-ride problem. *Transportation Research Record: Journal of the Transportation Research Board* 2218 (1), 59–67.
 80. Masmoudi, M. A., Braekers, K., Masmoudi, M., Dammak, A., 2017. A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Computers & Operations Research* 81, 1–13.
 81. Masmoudi, M. A., Hosny, M., Braekers, K., Dammak, A., 2016. Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem. *Transportation Research Part E: Logistics and Transportation Review* 96, 60–80.
 82. Masson, R., Lehuédé, F., Péton, O., 2012. Simple temporal problems in route scheduling for the dial-a-ride problem with transfers. *Lecture Notes in Computer Science* 7298, 275–291.
 83. Masson, R., Lehuédé, F., Péton, O., 2014. The dial-a-ride problem with transfers. *Computers & Operations Research* 41, 12–23.
 84. Mauri, G., Lorena, L., 2006. A multiobjective model and simulated annealing approach for a dial-a-ride problem. In: *Workshop dos Cursos de Computação*.
 85. Melachrinoudis, E., Ilhan, A. B., Min, H., 2007. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research* 34 (3), 742–759.
 86. Molenbruch, Y., Braekers, K., Caris, A., 2017. Operational effects of service level variations for the dial-a-ride problem. *Central European Journal of Operations Research* 25 (1), 71–90.
 87. Molenbruch, Y., Braekers, K., Caris, A., Vanden Berghe, G., 2017. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Opera-*

- tions Research 77, 58–71.
88. Muelas, S., LaTorre, A., Peña, J.-M., 2013. A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city. *Expert Systems with Applications* 40 (14), 5516–5531.
 89. Muelas, S., LaTorre, A., Peña, J.-M., 2015. A distributed vns algorithm for optimizing dial-a-ride problems in large-scale scenarios. *Transportation Research Part C: Emerging Technologies* 54, 110–130.
 90. Neven, A., Braekers, K., Declercq, K., Bellemans, T., Janssens, D., Wets, G., 2014. Methodology to optimize resource requirements of a demand-responsive transport system for persons with disabilities: a case study in Flanders. In: *TRB 2014: 93th Annual Meeting of the Transportation Research Board*, Washington D.C.
 91. Nowak, M., Ergun, Ö., White, C. C., 2008. Pickup and delivery with split loads. *Transportation Science* 42 (1), 32–43.
 92. Oberscheider, M., Hirsch, P., 2016. Analysis of the impact of different service levels on the workload of an ambulance service provider. *BMC Health Services Research* 16 (1), 487.
 93. Paquette, J., Bellavance, F., Cordeau, J.-F., Laporte, G., 2012. Measuring quality of service in dial-a-ride operations: the case of a Canadian city. *Transportation* 39 (3), 539–564.
 94. Paquette, J., Cordeau, J.-F., Laporte, G., 2009. Quality of service in dial-a-ride operations. *Computers & Industrial Engineering* 56 (4), 1721–1734.
 95. Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M. M. B., 2013. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological* 52, 1–16.
 96. Parragh, S. N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies* 19 (5), 912–930.
 97. Parragh, S. N., Cordeau, J.-F., Doerner, K. F., Hartl, R. F., 2012. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum* 34 (3), 593–633.
 98. Parragh, S. N., Doerner, K. F., Hartl, R. F., 2008. A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft* 58 (2), 81–117.
 99. Parragh, S. N., Doerner, K. F., Hartl, R. F., 2010. Demand responsive transportation. *Wiley Encyclopedia of Operations Research and Management Science*, 1–9.
 100. Parragh, S. N., Doerner, K. F., Hartl, R. F., 2010. Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research* 37 (6), 1129–1138.
 101. Parragh, S. N., Doerner, K. F., Hartl, R. F., Gandibleux, X., 2009. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks* 54 (4), 227–242.
 102. Parragh, S. N., Pinho de Sousa, J., Almada-Lobo, B., 2014. The dial-a-ride problem with split requests and profits. *Transportation Science* 49 (2), 311–334.
 103. Parragh, S. N., Schmid, V., 2013. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research* 40 (1), 490–497.
 104. Psaraftis, H. N., 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* 14 (2), 130–154.
 105. Psaraftis, H. N., 1983. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17 (3), 351–357.
 106. Qu, Y., Bard, J. F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies* 32, 1–20.
 107. Qu, Y., Bard, J. F., 2014. A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science* 49 (2), 254–270.
 108. Quadrifoglio, L., Dessouky, M. M., Ordóñez, F., 2008. A simulation study of demand responsive transit system design. *Transportation Research Part A: Policy and Practice* 42 (4),

- 718–737.
109. Reinhardt, L. B., Clausen, T., Pisinger, D., 2013. Synchronized dial-a-ride transportation of disabled passengers at airports. *European Journal of Operational Research* 225 (1), 106–117.
 110. Rekiek, B., Delchambre, A., Saleh, H. A., 2006. Handicapped person transportation: An application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence* 19 (5), 511–520.
 111. Ritzinger, U., Puchinger, J., Hartl, R. F., 2016. Dynamic programming based metaheuristics for the dial-a-ride problem. *Annals of Operations Research* 236 (2), 341–358.
 112. Ronald, N., Thompson, R., Winter, S., 2015. Simulating demand-responsive transportation: a review of agent-based approaches. *Transport Reviews* 35 (4), 404–421.
 113. Røpke, S., Cordeau, J.-F., 2009. Branch-and-cut-and-price for the pickup and delivery problem with time windows. *Transportation Science* 43 (3), 267–286.
 114. Røpke, 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.
 115. Røpke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40 (4), 455–472.
 116. Rubinstein, Z. B., Smith, S. F., Barbulescu, L., 2012. Incremental management of oversubscribed vehicle schedules in dynamic dial-a-ride problems. In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. pp. 1809–1815.
 117. Santos, D. O., Xavier, E. C., 2015. Taxi and ride sharing: a dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications* 42 (19), 6728–6737.
 118. Savelsbergh, M. W., 1992. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing* 4 (2), 146–154.
 119. Schilde, M., Doerner, K. F., Hartl, R. F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research* 38 (12), 1719–1730.
 120. 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 Journal of Operational Research* 238 (1), 18–30.
 121. Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* 159 (2), 139–171.
 122. Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science* 1520, 417–431.
 123. Shen, C.-W., Quadrioglio, L., 2012. Evaluation of zoning design with transfers for paratransit services. *Transportation Research Record: Journal of the Transportation Research Board* 2277 (1), 82–89.
 124. Shen, C.-W., Quadrioglio, L., 2013. Evaluating centralized versus decentralized zoning strategies for metropolitan ADA paratransit services. *Journal of Transportation Engineering* 139, 524–532.
 125. Tang, J., Kong, Y., Lau, H., Ip, A. W. H., 2010. A note on “Efficient feasibility testing for dial-a-ride problems”. *Operations Research Letters* 38 (5), 405–407.
 126. Teodorovic, D., Radivojevic, G., 2000. A fuzzy logic approach to dynamic dial-a-ride problem. *Fuzzy Sets and Systems* 116 (1), 23–33.
 127. Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31 (1), 60–71.
 128. Wolfler Calvo, R., Touati-Moungla, N., 2011. A matheuristic for the dial-a-ride problem. *Lecture Notes in Computer Science* 6101, 450–463.
 129. Wong, K. I., Bell, M. G. H., 2006. Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research* 13 (3), 195–208.

130. Xiang, Z., Chu, C., Chen, H., 2006. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research* 174 (2), 1117–1139.
131. Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research* 185 (2), 534–551.
132. Yi, F., Tian, L., 2005. On the online dial-a-ride problem with time-windows. *Lecture Notes in Computer Science* 3521, 85–94.
133. Yi, F., Xu, Y., Xin, C., 2006. Online dial-a-ride problem with time-windows under a restricted information model. *Lecture Notes in Computer Science* 4041, 22–31.
134. Zhang, Z., Liu, M., Lim, A., 2015. A memetic algorithm for the patient transportation problem. *Omega* 54, 60–71.
135. Ziliaskopoulos, A., Kozanidis, G., 2006. Dynamic programming strategies for the dial a ride problem with time window constraints. In: *TRB 2006: 85th Annual Meeting of the Transportation Research Board*, Washington D.C.
136. Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3 (4), 257–271.
137. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Grunert da Fonseca, V., 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7 (2), 117–132.