2 Dispatching Problems in Transport Logistics

It is the goal of optimization to adopt a system behavior in order to maximize or minimize an objective function. In economic systems, this objective function refers to revenue maximization. Logistics is a classical area of application for optimization, because it includes multiple complex optimization problems. In transport logistics, these complex problems often refer to tour planning and routing problems. The goal of transport is, to pick up goods at their origin and deliver them to their destination to overcome special distances in order to fulfill the six objectives of logistics: to deliver the right object (1) at the right time (2) to the right place (3) in the right quantity (4) and quality (5) at the right price (6) (Jünemann, 1989, p. 18). Consequently, most logistics problems require multicriteria optimization taking into account several constraints.

This chapter introduces tour planning and routing problems in transport logistics and focuses on the most relevant problem variations and established approaches to solve them. Therefore, the problems are categorized in three groups. Firstly, Section 2.1 presents the well-known Traveling Salesman Problem (TSP) with various constraints which has the objective of determining the shortest route for a single vehicle which includes all stops and fulfills additional constraints. Section 2.2 describes multivehicle variants of the Vehicle Routing Problem (VRP). Its main difference is that not only the shortest tour, but also the optimal allocation of goods to vehicles must be determined. Section 2.3 presents the dynamic variant of the VRP. Since there are numerous domain-dependent variations of each problem and at least as many approaches to solve them, comprehensive overviews are referenced.

M. Gath, *Optimizing Transport Logistics Processes with Multiagent Planning and Control*, DOI 10.1007/978-3-658-14003-8_2

Formal descriptions of mathematical programs are provided as long as they are relevant in subsequent chapters. Section 2.4 concludes the chapter and discusses the limitations of applying classic approaches in dynamic real-world and Industry 4.0 applications.

2.1 Traveling Salesman Problems

The most famous problem in logistics is probably the Traveling Salesman Problem (Flood, 1956), which is NP-complete (Garey, Graham, and Johnson, 1976). The classical TSP can be defined as follows.

Definition 2.1 (Traveling Salesman Problem)**.** *Let S denote a set of stops, which must be visited. Given the costs $c^v_{i,j}$ for traveling from $i \in S$ to $j \in S$ and choosing indicator variables*

$$x_{i,j} = \begin{cases} 1, & \textit{if } (i,j) \textit{ is part of the vehicle's tour} \\ 0, & \textit{otherwise} \end{cases} \tag{2.1}$$

the objective function of the classical TSP is

$$\min \sum_{j \in S} \sum_{i \in S} c_{i,j} \cdot x_{i,j} \tag{2.2}$$

subject to

$$x_{i,j} = \{0,1\} \textit{ for all } i,j \in S \tag{2.3}$$

$$\sum_{i \in S} x_{i,j} = 1 \textit{ for all } j \in S \tag{2.4}$$

$$\sum_{j \in S} x_{i,j} = 1 \textit{ for all } i \in S \tag{2.5}$$

$$\sum_{j \in S} \sum_{i \in S} x_{i,j} \leq |Y| - 1 \textit{ for all } Y \subseteq S. \tag{2.6}$$

The objective function defined by Eq. 2.2 minimizes the overall cost. Eq. 2.4 and Eq. 2.5 ensure that each service request has to be satisfied exactly once. In addition, Eq. 2.6 is a sub-tour elimination constraint which guarantees that all stops are visited in a single and connected tour. In short, the goal is to find the shortest tour of a vehicle, which ensures that all service requests are satisfied.

In most real-world scenarios, time window constraints as well as time consumption at the warehouse/customer have to be considered. The Traveling Salesman Problem with Time Windows (TSPTW) which also involves handling times can be defined as follows.

Definition 2.2 (TSP with Time Windows and Handling Times)**.** *If l_s denotes the latest pickup/delivery time at stop $s \in S$, t_s the time consumption of the loading or unloading process, r_s the release time at s, and $time_{i,j}$ vehicle's time for driving from i to j, then*

$$x_{i,j} = 1 \Rightarrow l_j \geq r_i + t_j + time_{i,j} \tag{2.7}$$

has to be fulfilled.

Let $DEP \subseteq S$ denote a set of depots. In some TSPTW it is relevant that the vehicle starts and ends at a depot $d \in DEP$. This is ensured by the following equations.

$$r_d < \min_j r_{j \in S \setminus DEP} \tag{2.8}$$

and

$$l_d > \max_j r_{j \in S \setminus DEP}. \tag{2.9}$$

In some TSPs capacity constraints have to be considered.

Definition 2.3 (TSP with Capacity Constraints)**.** *Let CC_s denote the current capacity of the vehicle at stop $s \in S$ and M the maximum capacity of the vehicle, then*

$$CC_s \leq M \textit{ for all } s \in S \tag{2.10}$$

has to be fulfilled.

Of course, there are problems which include time windows, handling times (Eq. 2.7), and capacity constraints (Eq. 2.10) at the same time. In addition, there are TSPs with pickup and delivery requests at the stops. If the transport is offered between a central depot and customers, it is not relevant which kind of service is offered at a stop as long as the maximum allowed capacity of the vehicle is not exceeded at any stop (Eq. 2.10). However, if the problem includes transports between customers in direct tours, the vehicle has to ensure that it visits the pickup stop before the delivery stop. This kind of Traveling Salesman Problem with Pickup and Deliveries (TSPPD) can be defined as follows.

Definition 2.4 (TSP with Pickups and Deliveries)**.** *Stops are a super set of a subset of pickup stops $P \subset S \setminus (D \cup DEP)$ and a super set of a subset of delivery stops $D \subset S \setminus (P \cup DEP)$. Moreover, O denotes a set of orders. An order $o \in O$ contains exactly one pickup stop p_o and one delivery stop d_o. Therefore, it must be ensured that the vehicle visits the pickup stop before the delivery stop by*

$$(x_{i,p_o} = 1 \land x_{i,d_o} = 1) \Rightarrow l_{p_o} < r_{d_o}. \tag{2.11}$$

Definitions 2.1 to 2.4 describe the TSP with frequently considered constraints. Nevertheless, there could be numerous additional constraints, especially in real-world problems. For instance, Jaillet (1988) solves the TSP with probabilistic customer demands and Malandraki and Dial (1996) present a dynamic programming heuristic for solving the time-dependent TSP.

The TSP and its variations are among the most investigated problems in mathematics, Operations Research (OR), and computer science, because the TSP is NP-hard, easy to describe, and hard or even impossible to solve optimally. There are numerous approaches to solve the TSP with various constraints such as genetic algorithms (Grefenstette, Gopal, Rosmaita, and Van Gucht, 1985; Potvin, 1996; Snyder and Daskin, 2006), simulated annealing (Aarts, Korst, and van Laarhoven, 1988; Geng, Chen, Yang, Shi, and Zhao, 2011), tabu-search (Fiechter, 1994; Gendreau, Guertin, Potvin, and Taillard, 1999;

Gendreau, Laporte, and Semet, 1998), ant colony systems (Dorigo and Gambardella, 1997), particle swarm algorithms (Shi, Liang, Lee, Lu, and Wang, 2007), neural networks approaches (Potvin, 1993; Créput and Koukam, 2009), k-opt improvement heuristics (Lin and Kernighan, 1973; Helsgaun, 2009), as well as branch-and-bound algorithms (Padberg and Rinaldi, 1991; Fischetti, Salazar González, and Toth, 1997; Hernández-Pérez and Salazar-González, 2004; Cordeau, Iori, Laporte, and Salazar González, 2010) with several heuristics for upper and lower bounds (Karp and Steele, 1985; Miller and Pekny, 1991; Johnson and McGeoch, 2007), to name but a few. A comprehensive review of the TSP is provided by, e.g., Applegate, Bixby, Chvatal, and Cook (2006) and Cook (2012).

2.2 Vehicle Routing Problems

The Vehicle Routing Problem (VRP) (Dantzig and Ramser, 1959) is a generalization of the TSP which includes multiple vehicles. While the objective of the TSP is to find the shortest route for a single vehicle, the objective of the VRP is to find the optimal allocation of goods to vehicles as well as the shortest route for each vehicle, both of which minimize the total costs. The VRP is an NP-hard problem (Lenstra and Kan, 1981). It is hard to solve even small problems with 25-50 stops optimally (Azi, Gendreau, and Potvin, 2010).

Similar to Definition 2.1 the VRP can be defined as follows.

Definition 2.5 (Vehicle Routing Problem). *Let V denote a set of vehicles. Given the costs $c_{i,j}^v$ for a vehicle $v \in V$ for traveling from $i \in S$ to $j \in S$ and choosing indicator variables*

$$x_{i,j}^v = \begin{cases} 1, & \text{if } (i,j) \text{ is part of vehicle } v\text{'s tour} \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

the highest priority objective function of a VRP is to minimize the number of used vehicles by

$$\min \sum_{v \in V} 1 \tag{2.13}$$

and second highest priority is to minimize the cost of the vehicles, which commonly depends on the distances driven or travel times of the vehicle, by

$$\min \sum_{v \in V} \sum_{j \in S} \sum_{i \in S} c_{i,j}^{v} \cdot x_{i,j}^{v} \tag{2.14}$$

subject to

$$\sum_{v \in V} \sum_{i \in S} x_{i,j}^{v} = 1 \textit{ for all } j \in S \tag{2.15}$$

$$\sum_{v \in V} \sum_{j \in S} x_{i,j}^{v} = 1 \textit{ for all } i \in S \tag{2.16}$$

$$\sum_{v \in V} x_{i,j}^{v} = \{0, 1\} \textit{ for all } i, j \in S \tag{2.17}$$

$$\sum_{v \in V} \sum_{j \in S} \sum_{i \in S} x_{i,j}^{v} \leq |Y| - 1 \textit{ for all } Y \subseteq S. \tag{2.18}$$

Depending on additional constraints, Eq. 2.7 - Eq. 2.11 must be satisfied for each vehicle $v \in V$ as well.

If the VRP also includes direct tours between customers without any handling operations at a central depot, it is known as the Pickup and Delivery Problem (PDP). While the classical PDP considers the transport of all kinds of goods, the so-called Dial-a-Ride Problem(DARP) (Cordeau and Laporte, 2007) or Handicapped Persons Transportation Problem (Toth and Vigo, 1997) deals with passenger transport and additional objective functions such as minimizing the transport times of passengers.

In unpaired PDPs transported goods are homogeneous and exchangeable. Thus, any item can be delivered to any customer. In paired PDPs every item has a specific sender and recipient. Consequently, the pickup and delivery requests of an order o have to be served by the same vehicle v. This is guaranteed by

$$\sum_{i \in S} x_{i,p_o}^{v} - \sum_{i \in S} x_{i,d_o}^{v} = 0 \text{ for all } i \subseteq S \text{ and } v \in V. \tag{2.19}$$

Problems which contain multiple depots (Renaud, Laporte, and Boctor, 1996) can be considered as a specialization of a PDP described above. In general, the variations of possible constraints of VRPs are similarly diversified as those of the TSP. For instance, there are problems with soft time windows (Balakrishnan, 1993; Taillard, Badeau, Gendreau, Guertin, and Potvin, 1997) and problems with probabilistic customer demands (Gendreau, Laporte, and Séguin, 1996). Especially in real-world problems, the complexity is increased by domain-dependent constraints and requirements. For instance, having a heterogeneous vehicle fleet requires the consideration of the individual properties of the vehicles in the tour planning process. They vary in their capacities. They differ in their costs per kilometer. They have individual working times. Some freight carriers are paid by commission fee while others receive a fixed rate per day. Moreover, loads are generally more individualized than is assumed in scientific investigations. The goods vary not only in their pickup or delivery time window and weight but also in their value, priority, and volume. In addition toll and dangerous goods restrictions as well as customs guidelines on security must be satisfied. An abstract illustration of a solution for a static VRP is shown in Figure 2.1.

Theorem 2.1 (Complexity of the VRP)**.** *Let k denote the number of vehicles minus 1 ($k = |V| - 1$) and n denote the amount of stops which have to be visited ($n = |S|$). Then, the number of possible tours for the VRP is at most $\frac{(n+k)!}{k!} = O(n!(n+k)^k)$.*

Proof. In order to illustrate the complexity and the amount of possible solutions of a general VRP, imagine a bookshelf on which you have to stack a number of books. In addition, the bookshelf can be divided into several sections by separators. In this example, the books refer to the amount of stops (denoted as n) which have to be visited and each section refers to a vehicle. Thus, there are $|V|$ sections and $|V| - 1$ separators (denoted as k) as shown in Figure 2.2. If there is no separator, there is only a single vehicle and there are $n!$ combinations to stack the books. If there is a single separator, there are $n!\binom{n+1}{1}$ combinations (all possible combinations of books which result from

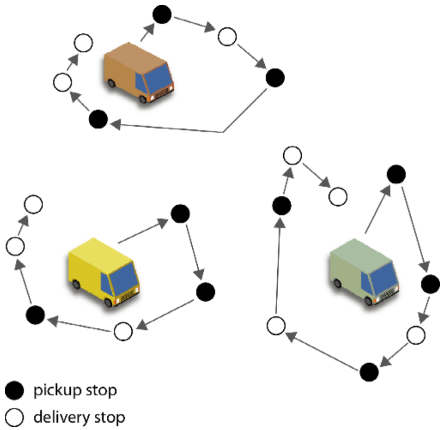

Figure 2.1: An abstract visualization of a solution for a static VRP with pickups and deliveries.

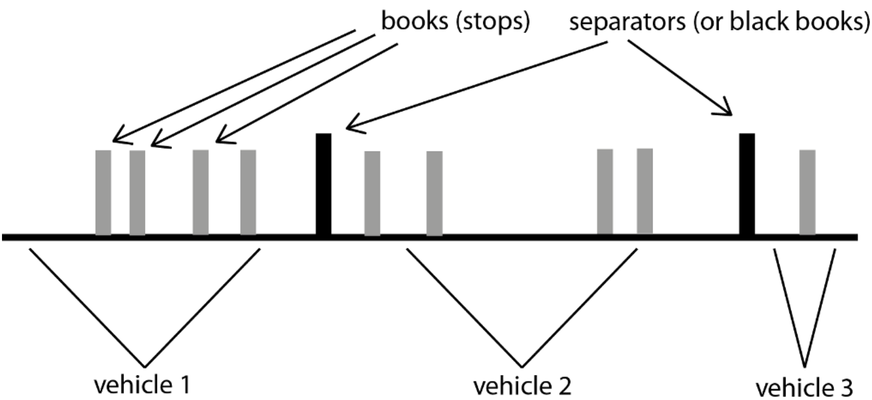

Figure 2.2: The figure shows an illustration of a VRP with 3 vehicles and 9 stops applied to the *bookshelf* example. There are 1,320 possibilities for assigning the books to the sections divided by the separators. In other words, there are $\binom{9+3}{3}$ combinations to put the separators for each of the 3! permutations of the books positions.

Table 2.1: The table shows the number of possible combinations for very small-size general VRPs. Note, that in problems with orders having a pickup and delivery, e.g., 10 stops are required to transport 5 orders only.

vehicles / stops	1	2	3	4	5
1	1	2	3	4	5
2	2	6	12	20	30
3	6	24	60	120	210
4	24	120	360	840	1,680
5	120	720	2,520	6,720	15,120
6	720	5,040	20,160	60,480	151,200
7	5,040	40,320	181,440	604,800	1,663,200
8	40,320	362,880	1,814,400	6,652,800	19,958,400
9	362,880	3,628,800	19,958,400	79,833,600	$\approx 2.59 \cdot 10^8$
10	3,628,800	39,916,800	$\approx 2.39 \cdot 10^8$	$\approx 1.03 \cdot 10^9$	$\approx 3.63 \cdot 10^9$
11	39,916,800	$\approx 4.79 \cdot 10^8$	$\approx 3.11 \cdot 10^9$	$\approx 1.45 \cdot 10^{10}$	$\approx 5.44 \cdot 10^{10}$
12	$\approx 4.79 \cdot 10^8$	$\approx 6.22 \cdot 10^9$	$\approx 4.35 \cdot 10^{10}$	$\approx 2.17 \cdot 10^{11}$	$\approx 8.71 \cdot 10^{11}$
13	$\approx 6.22 \cdot 10^9$	$\approx 8.71 \cdot 10^{11}$	$\approx 6.53 \cdot 10^{11}$	$\approx 3,48 \cdot 10^{12}$	$\approx 1.48 \cdot 10^{13}$
14	$\approx 8.71 \cdot 10^{11}$	$\approx 1.30 \cdot 10^{12}$	$\approx 1.04 \cdot 10^{13}$	$\approx 5.92 \cdot 10^{13}$	$\approx 2.66 \cdot 10^{14}$

different locations of the separators). If there are two separators, there are $n!\binom{n+2}{2}$ combinations. Consequently, with k separators there are $n!\binom{n+k}{k}$ combinations. The value $n!\binom{n+k}{k}$ can be transformed by

$$n!\binom{n+k}{k} = n!\frac{(n+k)!}{n!k!} = \frac{(n+k)!}{k!} \tag{2.20}$$

due to the factorial equation of binomial coefficients $\binom{n'}{k} = \frac{n'!}{k!(n'-k)!}$. To follow the division of $k!$, the separators can be shown as black books which are interchangeable (in contrast to the other books which are unique). Due to $\binom{n'}{k} \leq n'^k$, the effort is $O(n!(n+k)^k)$. Table 2.1 depicts the numbers of possible assignments for very small-size VRPs computed by Eq. 2.20. $\square$

Optimal approaches to solve the VRP often apply *branch-and-bound* techniques, which are limited to solving small problems due to the high computational complexity. For instance, Ropke, Cordeau, and

Laporte (2007) present an exact solution method for PDPs containing 96 orders. The solver of Azi et al. (2010) computes optimal solutions for VRPs with 25-50 stops. Over the few last decades, numerous efficient heuristics and meta-heuristics have been developed for solving large VRPs and PDPs such as tabu-searches (Garcia, Potvin, and Rousseau, 1994; Rochat and Taillard, 1995; Taillard et al., 1997; Cordeau and Laporte, 2003), genetic algorithms (Potvin and Bengio, 1996; Tan, Lee, Ou, and Lee, 2001; Berger and Barkaoui, 2004; Bräysy, Dullaert, and Gendreau, 2004; Pankratz, 2005), large neighborhood searches (Shaw, 1998; Ribeiro and Laporte, 2012), simulated annealing (Bent and Hentenryck, 2006), and ant systems (Gambardella, Éric Taillard, and Agazzi, 1999; Barán and Schaerer, 2003; Gajpal and Abad, 2009), to name but a few. Comprehensive and recommendable overviews of solution methods for solving multiple variations of VRPs are provided by, e.g., Bräysy and Gendreau (2005a), Bräysy and Gendreau (2005b), Golden, Raghavan, and Wasil (2008), and Parragh, Doerner, and Hartl (2008a,b).

An early publication by Clarke and Wright (1964) applies the so-called *savings heuristic*. The algorithm assigns each order to a vehicle which must only transport the respective order. In following iterations, the tours of two vehicles are merged to a single tour as long as all constraints are satisfied. This process is continued until the maximum capacity of a vehicle is reached. A frequently applied and established heuristic for the construction of tours is the *sequential insertion heuristic I1* developed by Solomon (1987) which is based on the savings heuristic. Solomon's algorithm or a variation of it is applied by several of meta-heuristics referenced above and also by numerous of the multiagent-based approaches which are described in Section 3.4. Solomon's algorithm initially creates a new tour containing only a high constrained order. Using a constrained order (instead of a less constrained one) increases the probability of inserting a higher number of less constrained orders in the following steps. Next, for each not yet inserted order the minimum cost and optimal position in the constructed tour is computed. The order with the least cost is inserted at the respective position. This process is continued until the orders' constraints prohibit inserting any additional order. The goal of the sequential insertion heuristic I1 is to minimize the number of vehicles required to transport all orders by satisfying all constraints. In addition, Solomon (1987) presents the *sequential insertion heuristic I2* to minimize the distances driven and the *sequential insertion heuristic I3* to minimize customers' waiting times of customers. Several authors, such as Dullaert and Bräysy (2003), Potvin and Rousseau (1993), and Balakrishnan (1993), adapt the insertion algorithm, e.g., to allow for parallel insertion, to consider soft time windows, or to increase performance.

As the tour's efficiency highly depends on the sequence in which orders are inserted in the tours, the algorithms are neither optimal nor complete. To reduce this dependency, k-Opt improvement techniques (Lin, 1965a) are applied such as the 2-Opt or 3-Opt operators (Solomon and Desrosiers, 1988; Savelsbergh, 1992), the 2-Opt* operator (Potvin and Rousseau, 1995), or the CROSS-Exchange operator (Taillard et al., 1997). For example, the tabu-search approach of Garcia et al. (1994) applies the I1 heuristic of Solomon (1987) to compute an initial solution, which is further optimized by k-opt improvement techniques.

2.3 Dynamic Vehicle Routing Problems

The classical VRP described above is a static and deterministic problem. All orders, vehicles, constraints, and the distance matrix are exactly known in advance and do not change. However, most real-world applications in transport logistics refer to dynamic problems in which information is revealed or updated during operations after an initial solution is computed. Thus, the planning horizon is generally unbounded, and scheduling and planning requires a periodical or continuous reoptimization to react to new or changing information and unexpected events. An example for solving a dynamic routing problem is illustrated in Figure 2.3.

In general, dynamic events can be categorized in three different classes:

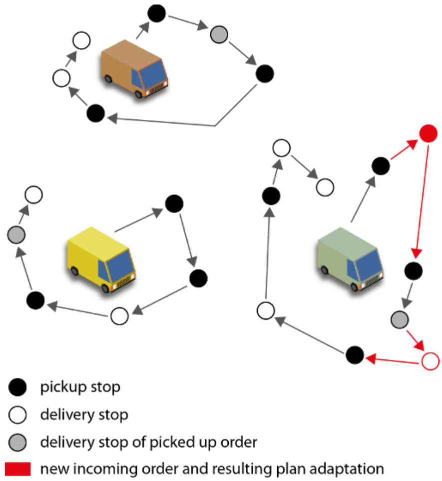

Figure 2.3: A solution for the dynamic VRP with pickup and deliveries.

1. dynamically incoming or outgoing entities, e.g., orders or vehicles;
2. dynamically occurring unexpected events;
3. new or changing information transmitted dynamically.

Since customer demands are not fully known in advance, most investigations only focus on dynamically incoming orders which have to be integrated into existing tours. However, real-world planning and control often need to consider some of the following dynamic events as well:

- The amount, type, and properties of the vehicle fleet change during operations. This is particularly relevant in domains with an order situation that varies daily or seasonally.
- The exact amount of goods which must be picked up at a customer location could be significantly larger than estimated. If the plan is not adapted, loading all pallets might result in exceeding the maximum capacity of the vehicle during the rest of the trip. Also a breakdown or a delay in the manufacturing process could reduce

the real demand. Thus, available capacities are wasted if the system is not re-optimized.

- Other properties of the goods may change. For instance, there is increasing customer demand in courier- and express services to change the pickup or delivery address dynamically (Deutsche Post AG, 2012, p. 63). Also time windows should be adapted dynamically to offer more flexible and thus customer-friendly services.
- Delay in an incoming goods department means further delays on the overall tour plan.
- The traveling time may vary, e.g., because of traffic congestions.

While this list includes general aspects only, further domain specific demands have to be considered in real-world operations.

In most dynamic problems, some information is already known in advance, while additional information becomes known during the operation. In order to measure the dynamics of a problem, Lund, Oli, and Rygaard (1996) introduced the *Degree of Dynamism*.

Definition 2.6 (Degree of Dynamism)**.** *Let n_{con} denote to number of dynamically appearing events, e.g., the number of available orders, and let n_{total} denote the total number of events, e.g., the overall amount of orders. The Degree of Dynamism is defined by*

$$dod = \frac{n_{con}}{n_{total}}. \tag{2.21}$$

The higher the *dod* value is, the more dynamic is the system. The *dod* assumes that each dynamic event during operation occurs evenly distributed. However, this depends on the application domain. For instance, 90% of all orders might come in during the rush hour after the operation has started. Therefore, Larsen, Madsen, and Solomon (2002) introduced the *Effective Degree of Dynamism*.

Definition 2.7 (Effective Degree of Dynamism)**.** *The planning horizon is limited by T. Let $t \in \mathbb{R}$ with $0 \leq t \leq T$ define the point of time* *an event occurs during operation. The Effective Degree of Dynamism is defined by*

$$edod = \frac{\sum_{i=1}^{n_{con}} \left(\frac{t_i}{T}\right)}{n_{total}}. \tag{2.22}$$

Thus, the *edod* value can also take into account rush hours and peak buying times.

Solving dynamic transport problems instead of static problems increases the flexibility and robustness in real-world transport processes and improves the customer service quality. Moreover, adapting to the current situation creates more optimization potential than advanced planning with estimated ranges of variable values. Thus, the overall efficiency increases and the operational costs decrease. Lack of information, unavailability of data, and uncertainty in the environment are handled dynamically. However, the computational complexity is further increased by a high degree of dynamics. Comprehensive surveys of dynamic VRPs are provided by, e.g., Ghiani, Guerriero, Laporte, and Musmanno (2003), Berbeglia, Cordeau, and Laporte (2010), and Pillac, Gendreau, Guret, and Medaglia (2013).

2.3.1 Static Approaches for Dynamic Problems

One of the first approach for solving dynamic VRPs was designed by Psaraftis (1980). His approach make use of a static and optimal dynamic programming solver which performs immediate replanning whenever a dynamic event appears. Thus, the dynamic problem is split up into multiple static problems. Each static sub-problem is solved optimally and all available information is taken into account. However, the optimal dynamic programming solver has a computational complexity of $O(n^2 3^n)$. Consequently, the complexity of VRPs limits the approach to solving only very small problems.

Motivated by this approach, other authors such as Savelsbergh and Sol (1998), Yang, Jaillet, and Mahmassani (2004), Montemanni, Gambardella, Rizzoli, and Donati (2005), and Chen and Xu (2006) followed the general idea of splitting up the dynamic problem into several time episodes, time slices, or time epochs. Next, a static solver computes solutions for each of these sub-problems independently. However, also the application of meta-heuristics and non-optimal approaches has significant drawbacks. Even marginally relevant events cause a complete recomputation of a new NP-hard problem from scratch. Especially with larger problems, the resulting response times are unacceptable for dispatchers and operators in real-world applications. Another disadvantage is that even small improvements could result in complete reallocations of shipments to vehicles, because each problem is solved independently. This complicates the handling process and could confuse the operators. To avoid this problem Montemanni et al. (2005) facilitate the reassignment of orders by applying a *Least Commitment Strategy* which assigns orders at the latest possible time. This is when a vehicle is idle or a later transport would violate time constraints. However, applying a *Least Commitment Strategy* decreases the service quality through late deliveries and is inefficient in scenarios where shipments can be consolidated in economical tours. Instead, this strategy could successfully solve the dynamic Stacker Cane Problem (SCP) (Frederickson, Hecht, and Kim, 1976). The dynamic SCP is a specialization of the dynamic PDP in which vehicles have a limited capacity of exactly one. However, it is not possible to react to other dynamically appearing events on the way to the next stop.

2.3.2 Continuous Reoptimization

Another approach to solve dynamic problems is to first compute a solution for the static problem and continuously update this solution whenever a dynamic event appears. Gendreau et al. (1999) developed a tabu-search algorithm which uses a continuous reoptimization of tours to solve the dynamic VRP. They adapted the method of Rochat and Taillard (1995) who implemented the so-called *adaptive memory*, a temporary storage space for best solutions already found. Similar to genetic algorithms, a fitness function evaluates the efficiency of each tour. The tours of the adaptive memory are adapted or combined and the resulting new route is added to the adaptive memory. As the size of the storage is bounded, only best solutions remain saved. Gendreau et al. (1999) handle the dynamics by firstly solving a static problem. If a dynamic event appears, all tours which are saved in the adaptive memory are updated and the optimization process is continued. However, a *least commitment strategy* is applied to assign the orders to the vehicles. Thus, the approach has all the disadvantages of *least commitment strategies* described above. Other authors have implemented similar tabu-search and large neighborhood search algorithms for solving dynamic VRPs such as Kergosien, Lent, Piton, and Billaut (2011), Mitrovic-Minic and Laporte (2004), Ichoua, Gendreau, and Potvin (2006), as well as Gendreau, Guertin, Potvin, and Séguin (2006).

Other authors applied one of the approaches described in Section 2.2 to initially solve the static problem. Next, dynamically incoming orders are integrated into existing tours by insertion heuristics, by k-opt improvement techniques, and by partial modifications of tours like Jih and Hsu (1999), Montané and ao (2006), Hanshar and Ombuki-Berman (2007), Beaudry, Laporte, Melo, and Nickel (2010). As these approaches have an initial population which is optimized by mutation and combination techniques, they are also called genetic algorithms. Unfortunately, in most cases the approaches consider only the integration of new orders or vehicles and neglect other dynamic events.

2.3.3 Anticipation and Stochastic Approaches

Stochastic approaches process historical data to anticipate future events. On the one hand, in static stochastic problems anticipations can be used to find adequate a-priori solutions which are only slightly modified during operations. In general, these adaptations are performed manually. For example, problems can include stochastic customer demands, which are estimated on the basis of a probability distribution, while the exact demand is revealed during operations. On the other hand, predictions and the consideration of future situations and events improve the proactive behavior of the system and reduce the required reactiveness in dynamic stochastic problems. Especially in domains with fix customers and recurring events, companies benefit from anticipations, e.g., of dynamically incoming orders. This increases the overall efficiency of the planning process.

Ichoua et al. (2006) consider probabilistic information about future events in the tour construction process. Therefore, they split up the service area into multiple clusters. In order to optimize the allocation of vehicles to these clusters, they consider not only the current status and information, e.g., about the actual distribution of vehicles and the current order situation, but also the probability of future incoming orders in the decision making process. Thus, the allocation is continuously optimized to shorten the reaction time and to increase the resource utilization to satisfy all service requests in each cluster.

Similarly, other authors apply anticipations in combination with different *waiting strategies* such as Mitrovic-Minic and Laporte (2004), Branke, Middendorf, Noeth, and Dessouky (2005), Ichoua et al. (2006), Thomas (2007), and Pureza and Laporte (2008). If a vehicle has finished the handling operations at a certain stop, it remains idle instead of immediately continuing the tour. The idea is that in anticipation of new incoming orders in nearby districts, the reaction time, the overall required time, and the distances driven can be reduced. Although the authors prove the advantages of *waiting strategies* in their investigated domains, the benefit highly depends on the quality of anticipation and the structure of the problem. In most real-world cases it is unacceptable to wait at specific stops or to increase the idle times of vehicles if it is not guaranteed that new orders will appear. In addition, waiting requires enough time to visit the next stop on time and probably reduces the service quality, because goods are delivered at the latest possible time.

2.4 Summary and Conclusion

This chapter introduced general as well as the most relevant dispatching problems in transport logistics such as TSP, VRP, and dynamic VRP including numerous variations with diversified constraints and requirements. Since route and tour planning is a classic application domain of optimization algorithms, the general problems with standard constraints have intensively been investigated.

Operations Research (OR) and mathematical approaches such as dynamic programming clearly dominated the research in logistics planning and scheduling at the end of the 20th century. However, especially in the last few decades numerous efficient algorithms have been developed for the transportation domain, extending these approaches with artificial intelligence. Therefore, numerous heuristics and meta-heuristics, such as simulated annealing, tabu-search, neural networks, ant systems, and genetic algorithms, extend and combine OR methods to both reduce the search space and to accelerate the search. Moreover, they increase the quality of solutions for large problems. These mainly centralized solvers have been extensively evaluated and also been applied in professional software systems for transport planning and scheduling.

Motivated by technological advances such as the Global Positioning System (GPS), smartphones, Geographical Information Systems (GIS), and cheap communication technologies which cover larger areas, research started to focus on the dynamic VRP around the year 2000. However, most of the approaches developed so far limit their examinations of dynamics to continuously incoming orders. Other dynamic events, such as dynamically changing addresses and order information, delays at incoming goods departments, and breakdowns of vehicles, are often neglected.

In order to handle the dynamics, a general approach presented in Section 2.3.1 is to split up the dynamic problem in a sequence of multiple static problems. However, the main drawback is that this results in long reaction times for dispatchers and freight carriers, due to the high computational complexity of the static problems. Especially in dynamic problems, online and short-term decision-making is essential. In addition, freight carriers are confused if allocations change significantly, which frequently happens if solutions are recomputed from scratch in each time episode. These drawbacks reduce the usability of these systems and preclude their application in real-world operation.

Therefore, Section 2.3.2 discusses approaches which continuously re-optimize a solution in case of dynamically appearing events. Many of the methods presented have a restricted proactive behavior, because they apply a *least commitment strategy*. This *wait and go* strategy impedes the consolidation of goods to build more economical loads, but might be successfully applied for problems which only include full-truckload storage. K-opt improvement techniques in combination with other meta-heuristics, such as simulating annealing or large neighborhood searches, are often applied and result in adequate solutions. Nevertheless, similar to most other centralized approaches, there is a lack of individualization: The solutions fail to be easily adapted and modified by additional domain dependent constraints because of the high complexity of the model and/or of the equations. In general, they even neglect considering a heterogeneous fleet. In addition, most approaches for dynamic problems are evaluated on benchmarks. Thus, they only use Euclidian distances between stops and locations to compute the required distance-matrix as input. They ignore the problem and the computational effort of recurring shortest-path computations. Especially in real-world operation, efficient shortest-path computations are essential (cf. Section 4.5).

Stochastic approaches presented in Section 2.3.3 are promising methods to further extend and improve other solvers. In problems with a limited degree of dynamics which include multiple periodical incoming orders on a regular basis, they can identify patterns of dynamic events. Thus, anticipations are applied to transform the dynamic problem into a stochastic static one, whose solution has merely (and often manually) been adapted during operations.

In conclusion, classical centralized planning and control is limited in dynamic, complex, and customized logistics processes due to the requirements of flexibility and adaptability to changing environmental influences and individual processes. Autonomous logistic processes and multiagent systems overcome these essential drawbacks.

http://www.springer.com/978-3-658-14002-1

Optimizing Transport Logistics Processes with
Multiagent Planning and Control
Gath, M.
2016, XXIII, 279 p. 59 illus., Softcover
ISBN: 978-3-658-14002-1