Managing Disruptions in Last Mile Distribution

Astrid M. C. Dettenbach  
RWTH Aachen University  
dettenbach@dpor.rwth-aachen.de

Sarah Ubber  
RWTH Aachen University  
ubber@dpor.rwth-aachen.de

Abstract

This paper considers disruption management in last mile distribution of delivery items. The delivery area is split into districts. Districts, and tours within them, are fixed over a long period of time. The sequence of each tour, which starts and ends at the depot, is given. Delivery items arrive daily at the depot in predetermined sequences. While minor failures in the delivery process can be dealt with quickly, major failures (mechanical failure, staff shortage) can have severe impacts on the initially planned operational schedule. Vehicle operators are highly familiar with their own district only, and changes to the ordinary schedule should be kept low to avoid major rearrangement of the tours. In this contribution we present a reassignment problem that determines a back-up plan in the case of disruptions before delivery starts. This paper is motivated by the introduction of an electric vehicle fleet into the last mile.

1. Introduction

Every day, a vehicle operator delivers items in predesigned districts. Districts are of a tactical nature and are fixed for several weeks or months. The plans are revised only once or twice a year to respond to changes in demand. One vehicle and its operator are assigned to one specific district. The static delivery plan supports the learning effect for vehicle operators and reduces the time needed for delivery. However, while executing an operational plan, disruptions may occur. Disruptions can be caused by different external and internal factors, such as shortage of staff, mechanical failures, or power outages. The last two factors gain in importance because we consider an electric vehicle fleet. Electric power vehicles have not been tested before in a large-scale delivery vehicle fleet. Experiences with respect to reliability are missing, and a stable delivery cannot be guaranteed. Especially in the introduction phase of the new technology, operational failures can occur. For example it might happen that the battery has not been charged over night or the vehicle does not start. While minor failures can be resolved quickly, major failures cause severe troubles in executing the original operational plan.

In this contribution we develop a decision making process for managing disruptions in delivery systems, e.g. postal services in the last mile. We identify potential factors of disruptions. Our focus lies on handling disruptions before delivery starts that result in a lack of resources, with one district not being servable. A back-up plan is developed based on a reassignment problem for disruption. The main objective is to minimize operational costs while serving the total demand, including the demand of the disrupted district. Limiting factors are working time and vehicle load capacity. To avoid sorting failure and to profit from learning effects by using the same sequence every day, the back-up plan’s sequence should not differ significantly from the sequence of the presorted delivery items. One so called back-up district is defined, which is divided into sections. Whenever a disruption occurs, the sections of the back-up district are reassigned to the same surrounding districts. Due to this procedure we can still profit from learning effects.

The paper is organized as follows. An overview of disruption management in the field of routing problems is given in Section 2. In Section 3 we describe the problem, while in Section 4 a mathematical model is presented. We show and discuss numerical results in Section 5. A summary and conclusion is given in Section 6.

2. Routing problems: Overview of disruption management

The capacitated vehicle routing (CVRP) and the capacitated arc routing (CARP) problems are widely studied problems in the literature. The classical problem formulation considers a depot where items are stored in order to be distributed to their final customer. A finite set of costumers with known demand is usually assumed. A finite set of vehicles with a given maximum vehicle load capacity is available to transport the delivery items to the customers. Tours start and end at the depot. The CVRP are node orientated and the CARP are arc orientated models.
Routing models decide upon the assignment of a customer to tours and the sequence that customers are served in. The vehicle schedules need to meet problem customized constraints, such as working time, capacity limitation of vehicles, and time windows. A lot of research on vehicle and arc routing problems has been done. Toth and Vigo [1] give an overview of the VRP and solution methods. Another overview is given in [2]. In their contribution a master schedule is given. Master schedules are often put in place and describe the daily or weekly routes for a longer period of time. Wøhlk [3] gives an overview of the capacitated arc routing in recent years and a general overview of arc routing is given in [4].

No matter how well tours are planned, they underlie the risk of disruption or operational failures. Measures need to be undertaken to reduce the negative effects on the overall system, e.g. to assure the running operations and to reduce delays. To cope with uncertainties in distribution systems, different approaches have been developed ([5]). One approach type considers the possibility of disruptions during the planning phase of the master schedule. The planning takes place in advance. The outcome of the “in advance planning” is an optimal operational plan that includes the identified risks of disruption based on estimation of risk probability. The second approach type manages disruptions at the time of occurrence and creates a revised plan. This approach is part of disruption management, which dynamically revises original plans based on deviation costs, and of real-time rescheduling. Actual rescheduling problems also incorporate deviation costs ([6]). Not many contributions exist in this field. Dynamic vehicle routing is a closely related problem. A recent paper of [7] gives a general overview of dynamic vehicle routing problems, also in the case of vehicle breakdown. Vehicle breakdown introduces some kind of dynamism into the model. Mu et al. [8] consider the disrupted capacitated vehicle routing problem with vehicle breakdown (DCVRP-B) and develop a heuristic to solve the model. In their setting, a vehicle breaks down during execution, and a revised plan needs to be created quickly. Also related to our problem are the contributions of [9, 10, 11, 12, 13]. All these papers have in common that other scheduled vehicles or empty vehicles need to be (re)scheduled to service the demands of the disabled vehicle.

Approaches classified as “in advance planning models” are stochastic models, robust optimization, and contingency planning. Stochastic models incorporate the uncertainty in the environment via stochastic processes. The models aim to create a master schedule that is optimal in terms of average output. Ideally, it can produce a plan for every possible scenario. Uncertain factors can, for example, be service times and the demand or number of customers. Laporte et al. [14] introduce a capacitated arc routing model with stochastic demand and present an adaptive large neighborhood search heuristic to solve the stochastic model. Their contribution is motivated by garbage collection and the uncertainty in demand for waste collection. Disruptions occur if realized demand exceeds the vehicle capacity. Lei et al. [15] developed a two-stage solution method for the vehicle routing and districting problem with stochastic demands in delivery areas.

Robust optimization aims at reducing uncertainty by introducing some measures of robustness. Kouvelis and Yu [16] minimize, for example, the maximum deviation from optimality in all scenarios. In contrast to stochastic models, no detailed distribution function is needed, but a set of scenarios must be specified. A plan also acceptable for the worst case scenario is generated by robust optimization models. Tajik et al. [17] present a pick-up and delivery model with a robust component. Sungur et al. [18] developed a robust optimization approach for the vehicle routing problem with stochastic demand.

Managing risks is also part of contingency planning. The first step of the planning consists of the identification of sources of risk and their effects on the operational processes. After that the potential risks are analyzed. One issue here is to analyze the probability of disruption occurrence. Risk management is handled in the next planning phase. Different scenarios and solutions for handling disruptions are developed and implemented. Overall, whenever a disruption occurs, a pre-defined sequence of actions is used. This approach is considered in this contribution.

3. Development of a back-up plan

We consider delivery systems such as the postal service in the last mile. As indicated before, a master schedule is given in our contribution. Stochastic and robust optimization models can be used to create an operational delivery plan, in this case the given master schedule, which is robust to different scenarios. However, using stochastic models or robust optimization methods cannot eliminate the risk of severe disruptions that interfere with the completion of the routes as planned. Therefore, we do not focus on those approaches but on contingency planning for managing risks.

We developed an overall disruption management procedure, regardless of the reason for the disruption. This procedure is presented in Figure 1. We introduced a routine check before delivery starts. Different sources of risk which can cause a disruption are checked. Potential sources of risk are power outages, vehicle breakdowns, and staff shortage. If a problem is noticed before or during
delivery, measures are undertaken to fix the problem. The measures depend on the specific problem occurring.

If it is not possible for the problem to be fixed by any measure, the tour is cancelled, and the back-up plan developed in advance is put in place. If the back-up plan is valid in terms of constraints, the sections of the back-up district are reassigned. Otherwise the tour is cancelled.

There are two approaches that can be used to determine the back-up district. The first approach defines exactly one district as the back-up district that is reassigned regardless of the vehicle/driver that fails. This is shown in Figure 2. In the case of a vehicle breakdown/illness, the vehicle/driver of the back-up district is reallocated to the disrupted district, and the delivery items of the back-up district are reassigned.

A second approach is that the disrupted district is split up and its delivery items are reassigned. In that case, the back-up district represents the disrupted district. For each district a back-up plan needs to be developed.

Both approaches have good arguments in their favor. Considering learning aspects, we can assume that the drivers need to memorize the back-up plan for each district if the second approach is used. Whereas, if the same back-up plan is always put in place, the drivers need to learn only one back-up plan. However, in the case of a sick driver, the driver of the back-up district must know all the other districts. But not only learning effects affect the decision on the action to be taken. In addition, other incentives have to be considered, such as minimizing the total additional time for servicing all segments. By determining only one back-up district, this can be taken into account. Therefore, our aim in this contribution is to determine a back-up district that is used independently from the district causing the disruption. Nevertheless, our model can also be used for the second approach by splitting up the disrupted district and defining that one as the back-up district.

The main objective of the reassignment problem is to minimize operational costs while serving the total demand, including the district that is causing the disruption. Because of working time and capacity limitations, it is not possible for a vehicle from another district to perform the entire service in the back-up district. For this reason, the back-up district is divided
into sections that are reassigned to surrounding districts, where a section is a cluster of segments, and a segment is the smallest unit that a district consists of. Segments are clustered in districts by CVRP/CARP methods. As we strongly insist on maintaining the original sequence of the tour, we decided to use a node-based formulation. The segments of a section are served in the original sequence. To avoid manual sorting and driving failures, we assume that only one section at most can be reassigned to a district. The vehicle operator, therefore, does not deviate more than once from the original tour. Figure 3 exemplifies the creation of a back-up plan. One depot and four districts are given. The district in the center of the figure on the left side is chosen as the back-up district. The reassignment decision is shown on the right side of the figure. The back-up district has been divided into three sections that are served by the surrounding districts in the original sequence.

Our model presented in the next section answers the following questions:
- Which district is to be chosen as back-up district?
- Which segments of the back-up district are to be assigned to which other district?
- Where is the section inserted into the tour of the district taking over that section?

4. Model

In this section we formulate a model which determines a back-up district \( j \in J \) and reassigns the segments \( i \in I \) of the back-up district to the surrounding districts \( j' \in J \), where \( j' \neq j \). We assume that a disruption occurs in only one district at a time. Each district has a maximum volume and a maximum distance limitation. As each vehicle is assigned to one fixed district \( j \), the capacity limitation of the district \( \text{MaxVol}_j \) and \( \text{MaxD}_j \) are derived from the vehicle used initially. We assume that the capacity of the vehicle of the back-up district is large enough to service each district. We also assume that tours will experience minor changes. If some segments need to be reassigned, the segments are inserted into one district in such a way that the sequences of the initial tours are kept to a large extent. No vehicle operator is allowed to leave his original tour more than once in order to service additional segments.

We used the following sets, parameters, and decision variables:

- \( I \): Set of segments \( i = \{0, \ldots, n\} \)
- \( J \): Set of districts \( j = \{1, \ldots, m\} \)
- \( D_j \): Total sum of the distances in district \( j \)
- \( \text{MaxD}_j \): Total maximum distance in district \( j \)
- \( \text{Vol}_j \): Total sum of the package volumes in district \( j \)
- \( \text{MaxVol}_j \): Total maximum volume in district \( j \)
- \( G_j \): Total sum of the working time in district \( j \)
- \( \text{MaxG}_j \): Total maximum working time in district \( j \)
- \( d_{ij} \): Distance from segment \( i \) to segment \( i' \)
- \( d_{ij} \): Distance from the predecessor \((\neq 0)\) of segment \( i \) to segment \( i' \)
- \( s_{ij} \): Time from segment \( i \) to segment \( i' \)
- \( s_{ij} \): Time from the predecessor \((\neq 0)\) of segment \( i \) to segment \( i' \)
- \( t_{ij} \): Original sequence: segment \( i' \) is originally a direct successor of segment \( i \)
- \( f_{ij} \): Original sequence: segment \( i' \) is originally a successor of segment \( i \)
- \( b_i \): Original district of segment \( i \in I \setminus \{0\} \)
- \( z_{ij} \): Segment \( i \) is originally assigned to district \( j \)
- \( \text{vol}_i \): Package volume of segment \( i \)
- \( g_i \): Working time of segment \( i \)
- \( stH_j \): Time from the depot 0 to district \( j \)
- \( stR_j \): Time from district \( j \) to the depot 0
\[dH_j; \quad \text{Distance from the depot 0 to district } j\]
\[dR_j; \quad \text{Distance from district } j \text{ to the depot 0}\]
\[M \quad \text{Big number}\]

\[y_j = \begin{cases} 
1, & \text{District } j \text{ is the back-up district} \\
0, & \text{Otherwise}
\end{cases}
\]
\[z_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ reassigned to district } j \\
0, & \text{Otherwise}
\end{cases}
\]
\[v_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ has predecessors that are} \\
0, & \text{reassigned to district } j \\
\end{cases}
\]
\[n_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ has successors that are} \\
0, & \text{reassigned to district } j \\
\end{cases}
\]
\[v_{zi_{ij}} = \begin{cases} 
1, & \text{Segment } i \text{ is reassigned to district } j \\
0, & \text{and has predecessors which are also} \\
\end{cases}
\]
\[n_{zi_{ij}} = \begin{cases} 
1, & \text{Segment } i \text{ is reassigned to district } j \\
0, & \text{and has successors which are also} \\
\end{cases}
\]
\[rn_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ inserted behind} \\
0, & \text{segment } i' \text{ in district } j \\
\end{cases}
\]
\[rv_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ inserted in front of} \\
0, & \text{segment } i' \text{ in district } j \\
\end{cases}
\]
\[l_{ij} = \begin{cases} 
1, & \text{Segment } i \text{ and segment } i' \text{ are} \\
0, & \text{reassigned to district } j \\
\end{cases}
\]

\[(IP) \min z = \sum_{i,i'j,j,i\neq i'\neq j}^{rn_{ij}, st_{ij}} + \sum_{i,i'j,j,i\neq i'\neq j}^{rv_{ij} \cdot (st_{ij} - st_{ij})} - \sum_{i,i\neq j}^{rn_{ij} \cdot stH_j} - \sum_{i,i\neq j}^{rv_{ij} \cdot stR_j} \\
\text{s.d.} \quad \sum_j y_j = 1 \quad (1)
\]
\[\sum_{j' \neq j} z_{ij} = y_j \quad \forall i \in I, j \in J, b_i = j, i \neq 0 \quad (2)
\]
\[M \cdot v_{ij} \geq \sum_{i' \neq i, i' \neq 0}^{f_{ij} \cdot z_{ij}} \quad \forall i \in I, j \in J, i \neq 0 \quad (3)
\]
\[v_{ij} \leq \sum_{i' \neq i, i' \neq 0}^{f_{ij} \cdot z_{ij}} \quad \forall i \in I, j \in J, i \neq 0 \quad (4)
\]
\[M \cdot n_{ij} \geq \sum_{i' \neq i, i' \neq 0}^{f_{ij} \cdot z_{ij}} \quad \forall i \in I, j \in J, i \neq 0 \quad (5)
\]
\[n_{ij} \leq \sum_{i' \neq i, i' \neq 0}^{f_{ij} \cdot z_{ij}} \quad \forall i \in I, j \in J, i \neq 0 \quad (6)
\]
\[v_{ij} + n_{ij} - z_{ij} \leq 1 \quad \forall i \in I, j \in J, i \neq 0 \quad (7)
\]

Constraint (1) ensures that exactly one district \(j \in J\) is selected as a back-up district. Each segment \(i\), which is originally serviced by the back-up district \(j\), has to be reassigned to another district \(j' \neq j\). This is guaranteed by constraint (2). Constraints (3), (4), (5), (6), and (7) ensure that segments are assigned to other districts in the predetermined sequence. Constraints (3) and (4) check for a segment \(i\) if there are predecessors \(i'\) of \(i\) that have been reassigned to \(j\). In this case, the decision variable \(v_{ij}\) is fixed to one. Constraints (5) and (6) check for a segment \(i\) if there are successors \(i'\) of \(i\) that have been reassigned to \(j\). In this case, the decision variable \(n_{ij}\) is fixed to one. If segment \(i\) has a predecessor and a successor that have been reassigned to \(j\), \(i\) has to be assigned to \(j\), too. Due to this, the predetermined sequence is kept. This is ensured in constraint (7).

If a section beginning at segment \(i\) and ending at segment \(i'\) is reassigned to a certain district \(j\), the driver of the district has to leave the predetermined sequence at a certain segment \(k\) to service the new section. After servicing the new section, the driver returns to the segment \(k'\), which is usually serviced directly after \(k\). The model decides about the segments \(k\) and \(k'\) the section should
be inserted between. Only the first segment \(i\) and the last segment \(i'\) of a section have to be connected to a segment \(k\) and a segment \(k'\). The decision variables \(r_{ij}k\) and \(rv_{ij}k\) have to be fixed to one if \(i\) is in inserted behind \(k\) and \(i'\) in front of \(k'\). Constraints (8) – (17) ensure exactly that the new section is inserted correctly into the new district.

This involves two issues. First, one must guarantee that the new section is inserted into a district \(j\). Second, the section needs to be inserted between two segments \(k\) and \(k'\) of district \(j\) that are in initial sequence.

\[
vz_{ij} \leq v_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{8}
\]

\[
vz_{ij} \leq z_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{9}
\]

\[
z_{ij} + v_{ij} - vz_{ij} \leq 1 \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{10}
\]

\[
nz_{ij} \leq n_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{11}
\]

\[
nz_{ij} \leq z_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{12}
\]

\[
z_{ij} + n_{ij} - nz_{ij} \leq 1 \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{13}
\]

\[
\sum_{i \neq i', j = 1} r_{ij}^0 = z_{ij} - v_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{14}
\]

\[
\sum_{i \neq i', j = 1} \sum_{j \neq j'} \sum_{j \neq j'} r_{ij}^0 = z_{ij} - n_{ij} \quad \forall i \in I, j \in J, \quad i \neq 0 \tag{15}
\]

Constraints (8), (9), and (10) determine the first segment of the new section. It is checked for each segment \(i\) whether \(i\) has been reassigned to district \(j\) and whether it has a predecessor that has also been reassigned to district \(j\). If that is the case, the decision variable \(vz_{ij}\) is fixed to one. If \(i\) is the first segment of the new section, it follows that \(z_{ij} - vz_{ij} = 1\). To determine the last segment of the new section, an equal procedure is used (constraints (11), (12), and (13)). If \(i\) is the last segment of the new section, it satisfies the condition \(z_{ij} - nz_{ij} = 1\).

Constraint (14) ensures that the first segment of the new section is inserted behind exactly one segment \(i'\) of district \(j\). The last segment of the new section has to be inserted in front of exactly one segment \(i'\) of district \(j\). This is guaranteed by constraint (15). Constraint (16) ensures that the predefined sequence of district \(j\) is not changed:

If the new section is inserted behind segment \(i\), the new section has to be inserted in front of segment \(i'\), which is usually serviced directly after \(i\). For two segments \(i\) and \(i'\), which are not in sequence in the original tour plan, i.e. \(t_{ij} = 0\), it has to be ensured that the new section can be connected with \(i\) or \(i'\), but cannot be connected with both segments at the same time. This is guaranteed by constraint (17). Constraints (18), (22), and (23) ensure that capacity conditions are satisfied.

\[
D_j + \sum_{i \neq i', j = 1} t_{ij} \cdot d_{ij} \cdot l_{ij} + \sum_{i \neq i', j = 1} \sum_{j \neq j'} \sum_{j \neq j'} r_{ij}^0 \cdot d_{ij} \cdot l_{ij} + \sum_{i \neq i', j = 1} \sum_{j \neq j'} \sum_{j \neq j'} rv_{ij}^0 \cdot d_{ij} \cdot l_{ij}
\]

\[
- \sum_{i \neq i', j = 1} rm_{ij} \cdot d_{ij} \cdot l_{ij} - \sum_{i \neq i', j = 1} rv_{ij}^0 \cdot d_{ij} \cdot l_{ij} \leq MaxD_j \quad \forall j \in J \tag{18}
\]

Constraint (18) ensures that the length of the delivery tour including the distances to and from the depot is less than the maximum distance \(MaxD_j\) of district \(j\). The length of the delivery tour consists of the length of the original district \(j\) plus extra distances. Extra distances include the ones to and from the new section and the total distances in the section itself minus the original distance between the point where the vehicle operator leaves the original tour and the point where he reenters it. This is described by two parts again. One part considers distances not including the depot. The last two terms consider distances from and to the depot only.

\[
l_{ij} \leq l_{ij} \quad \forall i \in I, j \in J \tag{19}
\]

\[
l_{ij} \leq l_{ij} \quad \forall i \in I, j \in J \tag{20}
\]

\[
z_{ij} + z_{ij} - l_{ij} \leq 1 \quad \forall i \in I, j \in J \tag{21}
\]

Constraints (19), (20), and (21) linearize the term \(z_{ij} \cdot l_{ij}\). The auxiliary variable \(l_{ij}\) indicates whether segments \(i\) and \(i'\) are both reassigned to a district \(j\). It is used in constraints (18) and (22).

Constraint (22) is similar to constraint (18), but it considers time instead of distances. It guarantees that the original working time plus working time and extra time for additional and dropped distances for the reassigned segments does not exceed the maximum working time \(MaxG_j\).
\[ G_j + \sum_i g_{ij}z_{ij} + \sum_{i', \delta_i = 1} t_{i'j} \cdot s_{i'j} \cdot l_{i'j} \]
\[ + \sum_{i', \delta_i = 1} r_{i'j} \cdot s_{i'j} \cdot l_{i'j} \]
\[ - \sum_{i', \delta_i = 1} r_{i'j} \cdot s_{i'j} \cdot l_{i'j} \]
\[ - \sum_{i, \delta_i = 0} r_{i_0j} \cdot s_{i_0j} \leq \text{MaxVol}_j \quad \forall j \in J \] (22)
\[ \text{Vol}_j + \sum_i \text{vol}_i z_{ij} \leq \text{MaxVol}_j \quad \forall j \in J \] (23)

Constraint (23) ensures that the transported volume is less than the maximum volume MaxVol\(_j\) of district \(j\).

\( y_{ij}, z_{ij}, v_{ij}, n_{ij}, v_{z_{ij}}, n_{z_{ij}} \in \{0, 1\} \)
\( \forall i, i' \in I, j \in J, i \neq 0 \) (24)
\( r_{i_0j}, r_{i'j} \in \{0, 1\} \)
\( \forall i, i' \in I, j \in J, i \neq 0, \quad \delta_{ij} = 0, \delta_{i'j} = 1 \)
\( l_{i_0j} \in \{0, 1\} \)
\( \forall i, i' \in I, j \in J, i \neq 0 \) (25)

Constraints (24), (25), and (26) state the binary constraints of the decision variables.

5. Case study and numerical results

We used a dataset from a city in Germany, including information about districts, vehicle fleet, and tours. For each district, the total working time and the assigned vehicle are given. Maximum vehicle load capacity (total parcel volume and weight) and the maximum vehicle range are known. Tours contain information about the sequence of segments in which letters and parcels are delivered. For each segment, the average amount of delivery items is known as well as the working time. We calculated the distance matrix in meters and seconds delivery items is known as well as the working time.

Sequence of segments in which letters and parcels are run.

The program ran on a PC with two IntelXeon processors running at 3.1 GHz and 128 GB of RAM.

The model formulation of the test instance has \(3n^2m + 5nm + m\) variables and \(m(5n^2 + 6n + 8(n - 1) + 3) + 1\) constraints. A time limit of 24 hours was set for each run. Our instance with \(m=13\) districts and \(n=671\) segments was not solved to optimality.

Tables 1, 2, and 3 present computational results.
### Table 1. Results for the entire model with a set time limit of 24h

<table>
<thead>
<tr>
<th>j</th>
<th>Number of segments in j</th>
<th>Best solution</th>
<th>LP bound</th>
<th>Computational time [sec]</th>
<th>Reassignment to districts</th>
<th>Number of segments in a section</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>51</td>
<td>890</td>
<td>-4194</td>
<td>–</td>
<td>1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13</td>
<td>2, 14, 1, 23, 1, 1, 1, 1, 2, 4</td>
</tr>
</tbody>
</table>

### Table 2. Results for fixed back-up districts with a set time limit of 24h

<table>
<thead>
<tr>
<th>j</th>
<th>Number of segments in j</th>
<th>Best solution</th>
<th>LP bound</th>
<th>Computational time [sec]</th>
<th>Reassignment to districts</th>
<th>Number of segments in a section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55</td>
<td>-800</td>
<td>-1216</td>
<td>–</td>
<td>2, 3, 4, 7, 8, 9, 10, 12, 13</td>
<td>2, 4, 2, 2, 18, 10, 14, 1, 2</td>
</tr>
<tr>
<td>2</td>
<td>55</td>
<td>-650</td>
<td>-1291</td>
<td>–</td>
<td>1, 3, 4, 5, 6, 7, 8, 9, 12, 13</td>
<td>3, 9, 1, 2, 1, 1, 1, 22, 2, 13</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>-1260</td>
<td>85617</td>
<td>4, 7, 8, 9, 10, 11, 12, 13</td>
<td>3, 14, 2, 9, 4, 2, 16, 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>53</td>
<td>-410</td>
<td>36150</td>
<td>3, 7, 9, 10, 12</td>
<td>2, 14, 22, 9, 6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>49</td>
<td>220</td>
<td>-38</td>
<td>–</td>
<td>2, 3, 10</td>
<td>19, 23, 7</td>
</tr>
<tr>
<td>6</td>
<td>51</td>
<td>-560</td>
<td>-1210</td>
<td>–</td>
<td>1, 2, 3, 4, 7, 8, 9, 10, 11, 13</td>
<td>3, 1, 9, 1, 3, 1, 3, 16, 14</td>
</tr>
<tr>
<td>7</td>
<td>46</td>
<td>-370</td>
<td>8850</td>
<td>3, 4, 10, 12</td>
<td>22, 1, 15, 8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>46</td>
<td>-490</td>
<td>10950</td>
<td>1, 3, 9, 10, 12</td>
<td>12, 1, 12, 11, 8, 2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>52</td>
<td>-80</td>
<td>1863</td>
<td>3, 10, 11, 12</td>
<td>11, 21, 9, 11</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>58</td>
<td>-810</td>
<td>37329</td>
<td>3, 4, 9, 11, 12</td>
<td>21, 11, 12, 1, 13</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>45</td>
<td>10</td>
<td>5293</td>
<td>9, 10, 13</td>
<td>18, 19, 8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>54</td>
<td>-1680</td>
<td>-2016</td>
<td>–</td>
<td>1, 2, 3, 4, 5, 7, 8, 9, 10, 11</td>
<td>8, 4, 4, 11, 1, 1, 6, 12, 4, 3</td>
</tr>
<tr>
<td>13</td>
<td>55</td>
<td>-170</td>
<td>9524</td>
<td>3, 9, 11</td>
<td>12, 14, 29</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3. Results with a set time limit of 72h

<table>
<thead>
<tr>
<th>j</th>
<th>Number of segments in j</th>
<th>Best solution</th>
<th>LP bound</th>
<th>Computational time [sec]</th>
<th>Reassignment to districts</th>
<th>Number of segments in a section</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>54</td>
<td>-1770</td>
<td>–</td>
<td>227684</td>
<td>1, 2, 3, 4, 7, 8, 9, 10, 11</td>
<td>8, 4, 6, 12, 1, 1, 8, 11, 3</td>
</tr>
<tr>
<td>6</td>
<td>51</td>
<td>710</td>
<td>-4134</td>
<td>–</td>
<td>1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13</td>
<td>2, 14, 1, 23, 1, 1, 1, 1, 2, 4</td>
</tr>
</tbody>
</table>

of the best solution and LP bound was only marginal. The assignment decision is basically the same, but the sections are inserted at different positions.

We now want to provide some general insights into the reallocation decision. We therefore had a closer look at the instances that were solved to optimality. If possible in terms of constraints, segments are assigned to neighboring districts or to districts whose paths from or back to the depot cross the area of the back-up district. Districts where such an assignment is not possible are districts which are most often located at the edge of the delivery area (with a small number of neighboring districts and/or crossing paths).

Figures 4 and 5 show an example solution for nominating district 7 as back-up district. For the sake of simplicity, we chose that district because sections...
are reassigned to four districts only. Figure 4 gives an overview of the delivery area and shows district 7 as back-up district. That tour, as well as all other tours, starts and ends in the depot on the left side of the figure. The lines indicate the sequence of delivery. Nodes represent the representative node of the segment. The back-up districted is split into four sections. The reassignment is shown in Figure 5. Tours 3 and 10 integrate the additional segments at the beginning of the tour. Tour 12 serves the extra segments in-between its own delivery sequence. Tour 4 serves one segment on the way back to the district. The thin line represents the back-up district. Lines with arrows show the section that is reassigned to another district. This district is represented by a medium thick line.

6. Summary and conclusion

In this contribution we developed a contingency plan for disruptions in last mile distribution. We identified potential sources of risks and developed procedures that define the course of action that is taken in the case of disruption. If the problem cannot be fixed by any measure, the tour is canceled, and a back-up plan is put in place. We introduced a model that determines a back-up plan and presented first results. The entire model was not solved to optimality. A reduced model with a fixed back-up district was tested and results were compared. To validate the observed insights and to generate more general ones, we have to create more datasets and run extensive numerical computations, which will be done in the near future.

If, in practice, the decision-maker defines back-up districts in a tactical/strategic way, long computational
times might be accepted. However, if the decision-maker needs a fast solution with updated package load and maximum vehicle range in an emergency situation, other solution methods or model formulations that find optimal or near optimal solutions in a short computational time need to be developed. Using heuristics might be suitable for finding good solutions in a short time.

Our model can easily be extended to a setting of vehicle failure during delivery. The model needs to be adapted to an open vehicle routing problem, as vehicles are already on tour when disruption occurs. Some vehicles need to interrupt their own delivery for a short time to load the undelivered items at the vehicle breakdown point. In delivery-only systems the later the disruption occurs the more easily a revised plan can be found, because available vehicle capacity increases over time. Limiting factors are distance and the working time of drivers. This is one scenario which will be looked at in the near future. Another extension we will examine is that of scenarios with more than one disrupted district. To handle this, a more flexible reassignment is necessary.

Acknowledgment

This work is supported by the Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (FKZ: 16EM1032).

References