A Robust Approach to the Capacitated Vehicle Routing Problem with Uncertain Costs

Lars Eufinger
Deutsche Bahn AG, Poststraße 20, 60329 Frankfurt a. Main, Germany,
lars.eufinger@deutschebahn.com

Jannis Kurtz
Department for Mathematics, RWTH Aachen University, Pontdriesch 12-14, 52062 Aachen, Germany
kurtz@mathc.rwth-aachen.de

Christoph Buchheim
Department for Mathematics, TU Dortmund University, Vogelpothsweg 87, 44227 Dortmund, Germany,
christoph.buchheim@math.tu-dortmund.de

Uwe Clausen
Institute of Transport Logistics, TU Dortmund University, Leonhard-Euler-Straße 2, 44227 Dortmund, Germany,
uwe.clausen@iml.fraunhofer.de

We investigate a robust approach for solving the Capacitated Vehicle Routing Problem (CVRP) with uncertain travel times. It is based on the concept of K-adaptability, which allows to calculate a set of k feasible solutions in a preprocessing phase before the scenario is revealed. Once a scenario occurs, the corresponding best solution may be picked out of the set of candidates. The aim is to determine the k candidates by hedging against the worst-case scenario, as it is common in robust optimization. This idea leads to a min-max-min problem.

In this paper, we propose an oracle-based algorithm for solving the resulting min-max-min CVRP, calling an exact algorithm for the deterministic problem in each iteration. Moreover, we adjust this approach such that also heuristics for the CVRP can be used. In this way, we derive a heuristic algorithm for the min-max-min problem, which turns out to yield good solutions in a short running time. All algorithms are tested on standard benchmark instances of the CVRP.

Key words: Capacitated Vehicle Routing Problem; Robust Optimization; K-Adaptability

1. Introduction.

The *Vehicle Routing Problem* (VRP) is one of the most widely studied optimization problems in the field of Operations Research. Dantzig and Ramser (1959) introduced the *Truck Dispatching Problem*, modeling how a fleet of homogeneous gasoline delivery trucks could serve the demands of a number of stations within a minimum traveled distance, starting from a central hub. Five years later, Clarke and Wright (1964) developed an iterative heuristic for this problem. Since Christofides (1976) the problem is known as *Vehicle Routing Problem.* The VRP is commonly encountered in the domain of logistics and transport: the question is how to serve a set of customers, which are located geographically distributed around a central depot, using a fleet of trucks. While already in Dantzig and Ramser (1959) capacities on the trucks have been considered, the problem including capacaties is nowadays called *Capacitated Vehicle Routing Problem* (CVRP), to distinguish it from the problem variant without capacity constraints, now called VRP.

Several surveys on solution methods for the VRP and the CVRP have been published by Toth and Vigo (2001, 2014), Baldacci et al. (2012, 2010), Laporte (2009), Golden et al. (2008) and Cordeau et al. (2007). Although the VRP in its deterministic version is extensively studied in the literature, it is still not possible to solve the VRP for large size instances arising in practical applications. Therefore, also many heuristic algorithms were developed to compute feasible solutions for the VRP, without a guarantee for the quality of the solution (Cordeau et al. (2002), Kumar and Panneerselvam (2012), Laporte et al. (2014), Toth and Vigo (2014)).

In practical applications of the VRP, many parameters may be unknown in advance or disturbances can occur. For example, unknown traffic situations, the resulting unknown costs or unknown future demands of customers have to be considered during the optimization process, in order to find a solution which works well under all possible (or at least probable) realizations of the uncertain parameters. Usually, it is not enough to consider an average traffic situation, since an optimal solution for this scenario can perform very badly in other traffic situations. Furthermore, a solution which satisfies certain demands need not be feasible for all realizations of future demands. To address such problems, besides other approaches, robust optimization has been proposed as a paradigm. Here, some (not necessarily finite) *uncertainty set* U is given that contains all possible *scenarios*, i.e. all outcomes of the uncertain parameters that should be taken into account. The aim is then to find a solution which is feasible for all scenarios in U and which is worst-case optimal, i.e. the worst objective value over all scenarios in U is optimized. The robust optimization approach has been studied intensively for different classes of uncertainty sets U and for several combinatorial optimization problems; see Aissi et al. (2009), Ben-Tal and Nemirovski (1998, 1999), Bertsimas and Sim (2004), Kouvelis and Yu (1996), Sim (2004).

One well-known drawback of robust solutions is that, since all possible scenarios in U are assumed to be equiprobable and only the worst case is relevant, they often perform badly in most of the scenarios. Another drawback is that the calculated robust solution afterwards is static and has to be used in any scenario. At the other extreme, computing a different optimal solution from scratch every time a new scenario materializes is too time-consuming for the CVRP. In this paper, we investigate an intermediate approach to tackle these drawbacks: we propose to solve Vehicle Routing Problems by an extended model called *min-max-min robustness*, which aims to calculate k different solutions for the underlying problem in a preprocessing step such that the best of them can be chosen each time a new scenario occurs. At the same time, the model uses the robust paradigm to hedge against uncertainty by calculating worst-case optimal solutions. To this end, the uncertainty set is covered by k solutions such that, for each scenario, the best of the k solutions can be used.

In practice, such a framework is reasonable whenever the computation of an optimal solution from scratch, after the scenario is known, is too expensive. In the case of the CVRP, the problem is generally too hard to be solved in real-time, while after solving the min-max-min model, the selection of the best out of the optimal k precomputed solutions can be performed very quickly. Moreover, in situations where solutions have to be implemented repeatedly by a human user, e.g., a truck driver, it is preferable to have a small set of potential solutions instead of choosing a completely new solution each time, from an exponential set of feasible solutions.

The min-max-min approach was investigated by Buchheim and Kurtz (2017) and is based on the concept of K-adaptability for robust two stage problems, which was introduced by Bertsimas and Caramanis (2010) and already studied for binary problems by Hanasusanto et al. (2015). The idea of the approach is to calculate k second-stage solutions already in the first stage and then pick the best of it in the second stage after the scenario is known. The resulting problem of finding the best set of k solutions can be modeled as a min-max-min optimization problem. Clearly, this approach is less conservative than the standard robust approach: since we cover the uncertainty by k solutions instead of one, we usually obtain solutions which are not worse (and often much better) than the classical robust optimal solution. In Buchheim and Kurtz (2017) the authors derived an oracle based polynomial time algorithm to solve the min-max-min problem for large enough k and showed that it works very well on random knapsack instances.

1.1. Related Literature

Only few publications consider robust versions of the VRP. A variant with time windows and uncertain travel times has been studied by Agra et al. (2013). Here, among others, the authors apply the framework of adjustable robustness to the problem. Gounaris et al. (2013), Sungur et al. (2008), Lee et al. (2012) study the CVRP with uncertain demands. Wohlgemuth et al. (2012) studied the Pickup and Delivery Problem (PDP), a variant of the VRP, with variable demand and transport conditions. The dynamic PDP is modeled as a multi-stage mixed integer problem and solved via tabu search. In Solano-Charris et al. (2015) the authors study the robust CVRP with discrete uncertainty. The authors propose a mixed integer linear program for this problem and several heuristics based on greedy algorithms or local search algorithms, besides others. The classical VRP with time windows and travel time uncertainty is studied in Braaten et al. (2017). The authors consider budgeted uncertainty sets and derive a heuristic based on adaptive large neighborhood search. A robust version of the VRP with multiple deliverymen was studied in De La Vega et al. (2017). A first robust bi-objective approach for the VRP is studied by Solano-Charris et al. (2016), where the worst total cost of traversed arcs and the maximum total unmet demand over all scenarios are minimized. An overview about different robust VRP models with different sources of uncertainty can be found in Ordóñez (2014). Stochastic versions of the CVRP are studied in Jaillet et al. (2016), Rostami et al. (2017), Sun (2014).

1.2. Contribution and Overview

In this paper, we investigate the potential of the new min-max-min-robust approach for the CVRP. As mentioned above, in Buchheim and Kurtz (2017) an algorithm has been proposed to solve the min-max-min problem for any combinatorial problem, given by an algorithm for the deterministic version. We apply the latter algorithm to the CVRP and run it on several benchmark instances. Furthermore, we adjusted the algorithm to make use of heuristics for the CVRP and derive a heuristic algorithm for the min-max-min problem by this idea.

It turns out that the adjusted algorithm, which combines exact and heuristic algorithms for the CVRP, does not yield an improvement of the running time. On the other hand, the heuristic version of the algorithm is very fast and calculates solutions which are not too far from optimal in most of the instances. Furthermore, the number of solutions calculated by the heuristic is very low on average, making it even more useful for practical applications.

In Section 2, we will define the CVRP and give an overview over the exact and heuristic methods which we used in our computations. In Section 3, we recall the min-max-min robust approach and introduce the aforementioned algorithms. We present the results of our computations for the min-max-min robust CVRP in Section 4, followed by a conclusion.

2. The Capacitated Vehicle Routing Problem.

In the literature, many different variants of the Vehicle Routing Problem were presented (Laporte (1992), Toth and Vigo (2014)). In this paper, we study the *Capacitated Vehicle Routing Problem* and address it by the min-max-min robust approach. In the following assume $G=(V,A)$ to be a directed complete graph with nodes $V=\{0\}\cup V_C$ and arcs $A=\{(i,j)\mid i,j\in V, i\neq j\}$. Node 0 represents the *depot* and the nodes in V_C represent the *customers*. In the following, let $n:=|A|$. Each customer $i\in V_C$ has a positive demand $d_i\in\mathbb{R}_+$, while we set $d_0=0$. Furthermore, we have a set of *vehicles* $=\{1,\dots,|V_C|\}$ where each vehicle has the same *capacity* $Q\in\mathbb{R}_+$ and a cost function $c\colon A\to\mathbb{R}$ on the arcs of G, which can be interpreted as traveling times. A *tour* $T\subset A$ in G is a directed cycle in G which traverses the depot. This can be interpreted as a tour of a vehicle which starts from the depot and supplies a subset of customers before it returns to the depot. The cost of a tour T is defined by

$$c(T) := \sum_{a \in T} c(a)$$

and the demand of a tour is the sum over all demands of the customers which are served by the tour, i.e.,

$$d(T) := \sum_{(v,w) \in T} d_v.$$

The Capacitated Vehicle Routing Problem is now to find a set of at most $|V_C|$ tours which minimize the total cost such that the sum of all demands on each tour does not exceed the capacity of the vehicle. Formally we define the problem as follows.

Problem 1 (CVRP). Let $G = (V, A)$ be a complete directed graph and c and d defined like above. Find a set of tours $T_1, \ldots, T_m \subset A$ with $d(T_i) \leq Q$ for $i = 1, \ldots, m$ and $m \leq |V_C|$ such that the total costs

$$c(T_1, \ldots, T_m) := c(T_1) + \ldots + c(T_m)$$

are minimized and each customer $i \in V_C$ is traversed by exactly one of the tours.

Note that we allow an unlimited number of vehicles in our problem formulation. In the literature, instead of fixing the fleet size, often a fixed cost on each vehicle is assumed. While fixed vehicle costs may be easily implemented in our model, we will omit them in this paper for sake of simplicity. We assume that the graph is directed. This is due to the fact that in real-world applications the travel time in one direction can be different from the travel into the other direction in some of the possible traffic scenarios.

Several methods and algorithms were presented to solve the Capacitated Vehicle Routing Problem (Cordeau et al. (2007), Golden et al. (2008), Laporte (2009), Toth and Vigo (2014)). The common approaches use integer programming formulations of the CVRP and then apply classical optimization techniques such as Branch-and-Cut or Branch-and-Cut-and-Price (Poggi and Uchoa (2014)). The two-index commodity flow formulation was introduced in Gavish and Graves (1982). Letchford and Salazar-González (2006) compare two-index, three-index and multi-commodity flow formulations, besides others. Letchford and Salazar-González (2015) strengthened the latter formulations by adding additional constraints and, besides giving theoretical relations between some of the formulations, compare the resulting lower bounds given by the continuous relaxations. In our computations in Section 4 we will use a strengthened three-index commodity flow formulation studied by Letchford and Salazar-González (2015). Nevertheless, the robust min-max-min approach presented in Section 3 can be combined with any exact algorithm or formulation for the CVRP. Moreover, also CVRP variants on undirected graphs can be addressed by our approach.

In the formulation used in this paper, we identify each arc $a = (i,j) \in A$ with a variable $x_{ij} \in \{0,1\}$ which has value one if the arc is contained in any of the tours and zero otherwise. Additionally we add variables $f_{ij}^u \geq 0$ for each customer $u \in V_C$ and each arc $(i,j) \in A$ which take the value 1 if and only if a vehicle traverses the arc (i,j) on the way from the depot to customer u. The formulation is then given by

$$
\begin{aligned}
\min \quad & c^\top x && (1)\\
s.t. \quad & \sum_{j\in V, j\neq i} x_{ij} = \sum_{j\in V, j\neq i} x_{ji} = 1 \ \ \forall\, i \in V_C && (2)\\
& \sum_{j\in V, j\neq 0} f_{0j}^u = \sum_{j\in V, j\neq u} f_{ju}^u = 1 \ \ \forall\, u \in V_C && (3)\\
& \sum_{j\in V, j\neq 0} f_{j0}^u = \sum_{j\in V, j\neq u} f_{uj}^u = 0 \ \ \forall\, u \in V_C && (4)\\
& \sum_{j\in V, j\neq i} f_{ij}^u = \sum_{j\in V, j\neq i} f_{ji}^u \ \ \forall\, u, i \in V_C : i \neq u && (5)\\
& f_{ij}^u \leq x_{ij} \ \forall i, j, u \in V_C : i \neq j && (6)\\
& \sum_{u\in V_C\setminus\{i,j\}} d_u f_{ij}^u \leq (Q - d_i - d_j)\, x_{ij} \ \forall\, i, j \in V_C : i \neq j && (7)\\
& \sum_{i\in V_C\setminus\{u\}} d_i \left(\sum_{j\in V, j\neq i} f_{ij}^u + \sum_{j\in V, j\neq u} f_{uj}^i \right) \leq Q - d_u \ \forall\, u \in V_C && (8)\\
& x_{ij} \in \{0,1\} \ \ i, j \in V && (9)\\
& f_{ij}^u \in \mathbb{R}_+ \ \forall\, i, j, u \in V_C : \ i \neq j, && (10)
\end{aligned}
$$

Here the constraints (2) ensure that each customer is visited exactly once on a tour. The constraints (3) – (6) ensure that $f_{ij}^u = 1$ if and only if customer u is visited after arc (i,j) is traversed on the same tour. Note that the latter constraints ensure as well that such a tour always starts and ends in the depot node 0. By adding the constraints (7) and (8) we ensure that the tours satisfy the capacity restriction. The latter constraints were devised by Letchford and Salazar-González (2015) also to strengthen the continuous relaxation.

Since the Vehicle Routing Problem is an NP-hard problem (Lenstra and Rinnooy Kan (1981)) and very hard to solve exactly in high dimensions, often heuristic algorithms are used to calculate feasible solutions. One of the most used heuristics for the VRP is the Clarke-Wright algorithm (Clarke and Wright (1964)), also known as the *savings* algorithm. Given an instance of Problem 1, the Clarke-Wright algorithm starts with $|V_C|$ tours where on each tour one single customer is visited. These routes are then merged, this process is guided by the savings criterion. Let c_{ij} be the costs of arc (i,j). Then the initial solution has total costs of $\sum_{i\in V_C} 2c_{0i}$. For each pair of customers (i,j) the (asymmetric) savings value s_{ij} is defined by

$$s_{ij} = (c_{0i} + c_{i0} + c_{0j} + c_{j0}) - (c_{0i} + c_{ij} + c_{j0}),$$

which is the decrease in the objective function obtained by merging the two initial routes servicing nodes i and j. Note that in the asymmetric case we get two different savings values s_{ij} and s_{ji} for each pair of customers (i,j). The Clarke-Wright algorithm uses these savings values to determine the order in which the initial routes are merged to form larger routes. For the CVRP routes are only merged if the vehicle capacity of the resulting route is not exceeded. The latter merging process is performed iteratively until no more routes can be merged without leading to infeasible solutions or the merging would result in higher costs. For the computations in Section 4, a variant of the Clarke-Wright algorithm is used. Yellow (1970) introduced the parameter λ (called the route shape parameter) which controls the relative importance of the direct arc between the two customers in the savings computation. The modified parameter-dependent savings formula $\bar{s}_{ij} = c_{0i} - \lambda c_{ij} + c_{0j}$ yields better overall results.

To improve feasible solutions of the Vehicle Routing Problem generated by the Clarke-Wright algorithm, local search operators can be applied. For the VRP, the neighborhoods are typically defined in terms of a heuristic operation, where a node in a given solution is moved to a new position, or several arcs are removed and replaced with new arcs. In other words, given a solution and a heuristic operation, the neighborhood consists of all solutions that can be created by applying the heuristic operation to the current solution. The six different local search operators are shown in Figure 1. For each operator, a diagram illustrating the change to the solution is shown and described briefly. Local search has proven to be very effective for the VRP and this technique is used in many VRP metaheuristics.

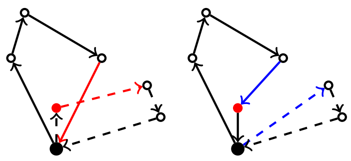
3. Min-max-min Robustness for the CVRP.

In the classical robust optimization approach, a so called *uncertainty set* $U \subset \mathbb{R}^n$ is given which contains the random parameters with a high probability. In this paper we assume that only the costs of the CVRP are uncertain such that we can formulate the CVRP as a linear problem of the form

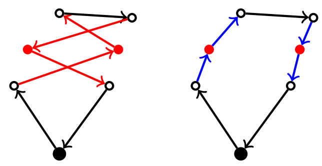
$$\min_{x \in X} c^\top x \tag{M}$$

where $X \subseteq \{0,1\}^n$ is the set of incidence vectors of all feasible solutions and the set U contains all relevant cost vectors c. In case the CVRP is modeled using auxiliary variables as in Section 2, the set X contains the projections of all feasible solutions of this formulation onto the subspace of arc variables, i.e., every element of X corresponds to a subset of the complete graph on V. Though considering this specific formulation in the remainder of this paper, any other representation of the feasible arc set X of the CVRP may be used. The robust counterpart is then given by the min-max problem

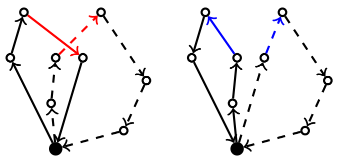
$$\min_{x \in X} \max_{c \in U} c^\top x. \tag{M^2}$$



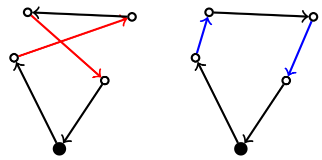
(a) **One-Point move:** Relocate an existing node into a new position.



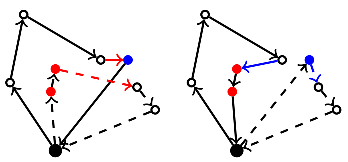
(b) **Two-Point move:** Swap the position of two nodes.



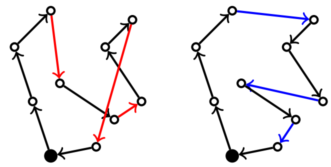
(c) **Inter-route Two-Opt move:** Remove one arc from two different routes and replace them with two new arcs.



(d) **Intra-route Two-Opt move:** Remove two arcs from a single route and replace them with two new arcs.



(e) **Three-Point move:** Swap the position of a pair of adjacent nodes with the position of a third node.



(f) **Three-Opt Move:** Remove three arcs from a route and replace them with three new arcs

Figure 1 **Local Search Operators (Groër (2008))**

The latter problem has been studied intensively in the robust optimization literature for different classes of uncertainty sets U. The two main classes which are studied are finite and convex sets. If $|U|$ is finite we say that U is a discrete uncertainty set. In the convex case, mostly polyhedral or ellipsoidal sets are studied. One well-studied subclass of the polyhedral uncertainty sets are the budgeted uncertainty sets introduced by Bertsimas and Sim (2004). For each parameter i, given an interval $[l_i, u_i]$ in which the parameter can vary, it is very unlikely in practice that all parameters differ from their nominal value at the same time. To this end the authors introduce a fixed parameter Γ which gives the maximal number of parameters which can differ from their nominal values. This set can be modeled by a polyhedron (see Section 4). It was shown by Bertsimas and Sim (2003) that the min-max problem (M^2) with budgeted uncertainty can be solved by solving $n+1$ instances of the underlying deterministic problem. Therefore it remains tractable whenever the underlying problem is tractable. In contrast to this, it was shown that Problem (M^2) is *NP*-hard for several tractable combinatorial problems under general discrete, polyhedral and ellipsoidal uncertainty (Aissi et al. (2009), Sim (2004)). Moreover, Problem (M^2) is rarely used for practical applications since it can produce solutions which can be far from optimal in many scenarios; see Bertsimas and Sim (2004) and Example 1 below. To tackle the latter problems, several extensions of the robust approach have been introduced; see e.g. Ben-Tal et al. (2004), Fischetti and Monaci (2009), Liebchen et al. (2009), Schöbel (2014).

One important class of problems investigated in the literature are two-stage robust problems. Here some decisions have to be made in a first stage before the uncertain scenario is known. Then, after a scenario is revealed, the second stage decisions have to be made. In general, this idea can be modeled by min-max-min problems. The latter problems are known to be theoretically and practically hard to solve in general (Ben-Tal et al. (2004), Kasperski and Zieliński (2017)). Therefore the idea of K-adatability was first proposed by Bertsimas and Caramanis (2010) to approximate robust two-stage problems. The main idea was to calculate k second-stage decisions in the first stage before the scenario is known. Then afterwards the best of the k pre-calculated solutions can be chosen. Hanasusanto et al. (2015) applied this approach to binary problems with cost uncertainty. A special case of the problem considered by Hanasusanto et al. (2015) is the so called *min-max-min robustness*, further investigated by Buchheim and Kurtz (2017). Here the authors propose to calculate k feasible solutions for the underlying combinatorial problem by solving the problem

$$\min_{x^{(1)},\dots,x^{(k)}\in X} \max_{c\in U} \min_{i=1,\dots,k} c^\top x^{(i)}. \tag{M^3}$$

The idea of the approach is thus to calculate k different solutions $x^{(1)},\dots,x^{(k)}$, instead of one. The aim is then to determine these k solutions such that the worst case over all scenarios is optimized, where we assume that we can choose the best of the k solutions for each scenario independently. Regarding a practical application, after calculating the optimal set of k solutions once, each time a scenario occurs the best of the k solutions for this scenario can be chosen by comparing the k objective values. Therefore Problem (M^3) only has to be solved once and the calculated solutions hedge against the uncertainty in a robust way. Clearly this approach is less conservative than the standard min-max approach since we can always add the optimal min-max solution to our set of k solutions and therefore the optimal value of (M^3) must be at least as good as the optimal value of (M^2). Note that, if we add the optimal min-max solution to the set of solutions of the min-max-min problem, in each possible scenario we can choose a solution from our optimal set which is at least as good as the min-max solution in this scenario. Nevertheless in general this may not be the case. In the following example we create an instance for which (M^3) has a strictly better optimal value than Problem (M^2).

Example 1. Consider a CVRP instance on the graph in Figure 2. The capacity of the vehicles is defined such that all pairs of customers can be served, but not all customers together, i.e. an optimal solution will use exactly two vehicles. The costs on the arcs are the same in both directions and we assume budgeted uncertainty with $\Gamma = 1$. The related interval $[l_i, u_i]$ for the possible costs on each arc is given in Figure 2.

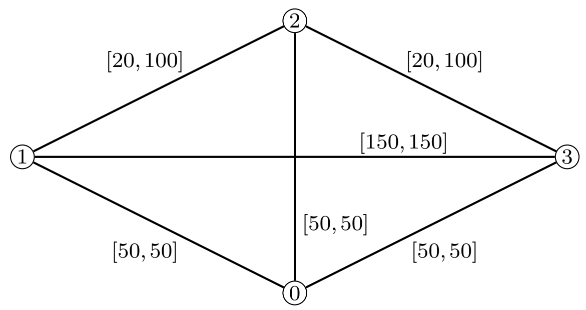


Figure 2

Then one of the optimal min-max solutions with optimal value 300 is the solution where customers 2 and 3 are served by one vehicle and customer 1 by the other vehicle. The worst case scenario is the one where arc $(1,2)$ has costs 20 and the arc $(2,3)$ has costs 100. The other optimal solution is the solution where customers 1 and 2 are served by one vehicle and customer 3 by the other. Here the worst case scenario is the one where arc $(1,2)$ has costs 100 and the arc $(2,3)$ has costs 20. If we consider the min-max-min problem with $k=2$, the optimal solution consists of the both latter solutions but has an optimal value of 260 since the worst case scenario has costs 60 on both arcs $(1,2)$ and $(2,3)$. Note that the absolute deviation from the min-max optimal value can be arbitrarily high if we increase the upper bound for the costs on the arcs $(1,2)$ and $(2,3)$.

A practical motivation for Problem (M^3) is a supplier which has to deliver goods to the same customers every day. Depending on the traffic situation every day the company wants to find the best solution to serve all customers with the available fleet of vehicles, i.e. it has to solve a vehicle routing problem with uncertain travel times. This problem is known to be hard to solve and in high dimension it cannot be solved in appropriate time every morning when the actual traffic scenario is known. Instead, Problem (M^3) only has to be solved once in a potentially expensive preprocessing. Afterwards the best of the k solutions can be chosen every morning depending on the current or expected traffic situation. This can be done easily by comparing the objective values of the calculated solutions for the selected traffic situation. Therefore, even if a very short time-window is given by a company to calculate a solution for a given scenario, the latter approach can be used instead of heuristic algorithms.

In Hanasusanto et al. (2015) it was shown that the optimal value of Problem (M^3) is the same for all $k \geq n$. Based on the latter result, Buchheim and Kurtz (2017) showed that the Problem (M^3) can be solved in oracle-polynomial time if the uncertainty set U is convex and if $k \geq n$. More precisely, the authors present a polynomial time algorithm which calls an oracle for the deterministic problem at most polynomially many times. In particular, any algorithm or formulation which solves

Problem (M) can be used to implement the oracle. Note that an optimal solution for the problem with $k = n$ may not consist of n pairwise different solutions and can therefore be optimal even for the problem with a lower k (see Section 4). Nevertheless, the following example shows that there exist instances of the uncertain CVRP with budgeted uncertainty for which each optimal solution of Problem (M^3) consists of $\mathcal{O}(n)$ pairwise different solutions.

Example 2. We consider a complete directed graph with depot node 0 and customer nodes $1, \dots, s$. We assume that the total demand of all customers does not exceed the capacity of the vehicles, i.e. all customers can be served by one vehicle. On this graph we define a budgeted uncertainty set

$$U^\Gamma := \left\{ c \in \mathbb{R}^A \mid c_a = l_a + \delta_a(u_a - l_a),\ \sum_{a \in A} \delta_a \leq \Gamma \right\},$$

where $l = 0$, $u_a = 1$ if a is an arc adjacent to the depot and $u_a = 0$ otherwise. Furthermore $\Gamma = 2s - 2$. In other words, only the arcs which leave the depot or which arrive at the depot have a positive deviation of 1, and in the budgeted uncertainty set, all of these arcs except for 2 of them can be selected to achieve this deviation. Then, for any combination of two customers i,j, an optimal solution of Problem (M^3) must contain one tour through all customers which uses the arcs $(0, i)$ and $(j, 0)$ and one tour through all customers which uses the arcs $(0, j)$ and $(i, 0)$. Indeed, if any of these tours is missing, consider the scenario where all depot-arcs have a deviation except the two respective arcs. Since no tour exists which uses these arcs, the objective value must be at least 1, whereas if all of these tours are contained in the solution we have an objective value of 0, which is optimal. Therefore an optimal solution consists of

$$2\binom{|V_C|}{2} = |V_C|(|V_C| - 1) = |A|$$

pairwise different solutions.

For the case $k \geq n$, Buchheim and Kurtz (2017) provide Algorithm 1 below to solve the min-max-min problem. To show that the algorithm is correct, the authors first prove that Problem (M^3) is equivalent to the convex problem

$$\min_{x \in \operatorname{conv}(X)} \max_{c \in U} c^\top x \tag{11}$$

if $k \geq n$. Problem (11) can be equivalently written as

$$\min_{\substack{y = \sum_{x \in X} \lambda_x x \\ \sum_{x \in X} \lambda_x = 1 \\ \lambda_x \geq 0\ \forall x \in X}} \max_{c \in U} c^\top y\,. \tag{12}$$

The idea of Algorithm 1 is based on the idea of column-generation, i.e. it starts with only a subset of variables λ_x and iteratively adds new variables. The variable which is added in each iteration is the one which has the largest impact on the optimal value. More formally, in iteration $i+1$ we calculate an optimal solution (z^*, c^*) of problem

$$\begin{aligned} \max \quad & z \\ s.t. \quad & c^\top x_j \geq z \ \forall j = 0, \dots, i \\ & z \in \mathbb{R}, \ c \in U \end{aligned} \tag{13}$$

which is the dual of the restricted Problem (12) which only contains the variables λ_{x_j} for $j = 0, \dots, i$. For this optimal solution in Step 5 we search for the most violated constraint in the latter dual problem, i.e. the variable in the primal problem with the largest improvement on the optimal value. The latter task can be done by solving the deterministic CVRP with cost function c^* by any exact algorithm or formulation for the CVRP.

Algorithm 1 Algorithm to solve Problem (M^3) for $k \geq n$

Input: convex $U \subset \mathbb{R}^n$, $X \subseteq \{0,1\}^n$

Output: optimal solution of Problem (M^3)

1: $i := 0$
2: choose any $x_0^* \in X$
3: **repeat**
4: calculate optimal solution (z^*, c^*) of

$$\max \{z \mid c^\top x_j^* \geq z \ \forall j = 0, \dots, i, \ z \in \mathbb{R}, \ c \in U\}$$

5: calculate optimal solution x_{i+1}^* of

$$\min_{x \in X} (c^*)^\top x$$

6: $i := i+1$
7: **until** $(c^*)^\top x_i^* \geq z^*$
8: **return** $\{x_j^* \mid (c^*)^\top x_j^* = z^*\}$

Note that the dual Problem (13) that has to be solved in Step 4 depends on the uncertainty set U. For polyhedral or ellipsoidal uncertainty sets this is a continuous linear or quadratic problem, respectively. Both can be solved by the latest versions of optimization software like IBM ILOG CPLEX (2014). Therefore Algorithm 1 can be implemented for the CVRP and any exact algorithm for the deterministic CVRP can be used in Step 5. Note that we exclude all redundant solutions in the return statement. In Buchheim and Kurtz (2017) the authors applied the latter algorithm to the knapsack problem and showed that on random instances the algorithm calculates significantly less than n solutions in general. Nevertheless, it may happen that the number of solutions returned by the algorithm is larger than n. In this case, for the subset of solutions returned by the algorithm, it suffices to eliminate those having zero dual variables in Step 4. However, it never happened in our computations presented in Section 4 that more than n solutions were returned by the algorithm.

Note that, using Problem (11), we can easily derive a lower bound for the exact optimal value of the min-max-min problem by relaxing the integer variables in the classical min-max problem. Indeed, let X be given by a linear formulation, e.g. the three-index commodity flow formulation presented in Section 2, such that $X = \left\{x \in \{0,1\}^d \mid Ax \leq b\right\}$ for a matrix A and a vector b in appropriate dimension. Then

$$\min_{x \in X_r} \max_{c \in U} c^\top x \tag{14}$$

with $X_r = \left\{x \in [0,1]^d \mid Ax \leq b\right\}$ is a lower bound for Problem (11). By dualizing the inner maximum expression, the latter problem becomes a linear problem if U is a polyhedron or a quadratic problem if U is an ellipsoid, respectively.

In the following section, we will apply Algorithm 1 to the vehicle routing problem defined in Section 2. The vehicle routing problem is already hard to solve in its deterministic version, which has to be done in Step 5, and therefore several heuristic algorithms have been developed in the literature (see Section 2). To make use of these algorithms, we adjusted Algorithm 1 as follows. Instead of solving the deterministic problem exactly in Step 5 we calculate a feasible solution by using a heuristic algorithm (see Section 2) as long as the stopping criterion of the loop is not satisfied. If the heuristic algorithm does not return an improving solution anymore, we switch back to the exact algorithm. In the following we call this procedure Algorithm 2. Note that the result is still a provably optimal solution to Problem (M^3).

Several of the vehicle routing instances given in Diaz (2012) were not solved to optimality yet even in the deterministic case where no uncertainty is considered. Clearly for these instances we are also not able to solve Problem (M^3) exactly by the algorithms above. Since for many of the latter instances heuristic solutions can be calculated very quickly, we propose a heuristic variant of the algorithm which only uses a heuristic algorithm for the deterministic problem instead of the exact algorithm in Algorithm 1. In the following, we call this procedure Algorithm 3. Algorithm 3 calculates a feasible solution for Problem (M^3) but without any guarantee for the quality of the objective value. Nevertheless in our computations it turned out that the heuristic variant produces good solutions and runs very fast, as shown in the next section.

4. Computations.

In the following, we present our computational results on Problem (M^3) for the CVRP. As mentioned before, this problem is very hard to solve in practice even in its deterministic version. Since

Algorithm 1 has to solve the deterministic problem several times, clearly it is at most as efficient as the algorithm for the underlying deterministic version. In particular, it cannot be expected that the instances which have not been solved to optimality yet even in the deterministic version can be solved by our algorithm.

For the computations below, we implemented the original Algorithm 1 introduced by Buchheim and Kurtz (2017) and the two variants of it discussed in Section 3, namely Algorithm 2 and Algorithm 3. For the exact CVRP oracle we implemented the strengthened three-index commodity flow formulation presented in Section 2 in CPLEX 12.6. The dual problem in Step 4 was also implemented in CPLEX 12.6. As a heuristic algorithm for the deterministic problem we used the open source library VRPH (Groër (2012)). Here 3 initial solutions are calculated via the modified Clarke-Wright algorithm described in Section 2 using different values of the shape parameter λ. The best solution is then used as a starting solution for the simulated annealing heuristic. For the simulated annealing, the operations One-Point, Two-Point and Two-Opt move are used; see Fig. 1. After the simulated annealing stage, a tour cleanup is performed. During the cleanup, each tour is improved individually via local search if possible. The local search operators used in the cleanup are One-Point, Two-Point, Two-Opt, Three-Opt and Three-Point move. Note that during the simulated annealing stage the local search operators can be inter-route and intra-route operators, but during the cleanup stage the used operators can be intra-route only. All computations were calculated on a cluster of 64-bit Intel(R) Xeon(R) E5-2670 processors running at 2.60 GHz with 20MB cache.

For our computations, we chose several instances from Diaz (2012). For each such instance we created 10 ellipsoidal and 10 budgeted uncertainty sets. To this end, we chose the Euclidean distances between the coordinates of the nodes as the mean cost vector $\bar{c}$ (i.e. the center of the ellipsoid or the lower interval limit l for the budgeted uncertainty set, respectively). If no coordinates are defined, we chose the arc-weights which are given by the instance. In the following, n is the number of arcs in the underlying graph of the CVRP instance, i.e. $n = |V|(|V|-1)$.

An ellipsoidal uncertainty set

$$U^E := \left\{ c \in \mathbb{R}^n \mid (c-\bar{c})^\top \Sigma^{-1} (c-\bar{c}) \leq \Omega^2 \right\} ,$$

where $\Sigma^{-1} \in \mathbb{Q}^{n \times n}$ is a symmetric positive semidefinite matrix and $\Omega \in \mathbb{R}$, can be seen as the quantile of a multivariate normal distribution with mean $\bar{c}$ and covariance matrix Σ. More formally the ellipsoid

$$U^E := \left\{ c \in \mathbb{R}^n \mid (c-\bar{c})^\top \Sigma^{-1} (c-\bar{c}) \leq \chi_n^2(\alpha) \right\} , \tag{15}$$

where $\chi_n^2(\alpha)$ is the α-quantile of the n-dimensional χ^2-distribution, contains $(1-\alpha)100\%$ of the probability. For our experiments, we constructed ellipsoidal uncertainty instances given by the 90%

and the 95% quantile of a normal distribution with random covariance matrix. The construction of the covariance matrix is done as follows: First, to obtain a random positive definite correlation matrix, we calculate a random orthonormal basis by the Gram-Schmidt procedure. More precisely, for each $i = 1, \ldots, n$ we create a vector v_i with random entries in $[-1, 1]$. Then we project v_i to the complement of the space spanned by the vectors $v_1, \ldots, v_{i-1}$ and normalize it. Afterwards we calculate random values $\lambda_i \in (0, 1)$ for each $i = 1, \ldots, n$. Then

$$\tilde{\Sigma} = \sum_{i=1}^{n} \lambda_i v_i v_i^\top$$

is a positive definite matrix. Since the covariances between arc i and j should depend on the distances $\bar{c}_i$ and $\bar{c}_j$ we have to scale the latter matrix. To obtain variances and covariances which can vary between zero and at most 5% of the distances, we define the matrix Σ by

$$\Sigma_{ij} = \tilde{\Sigma}_{ij} 0.05^2 \bar{c}_i \bar{c}_j .$$

Using this covariance matrix, we consider the related ellipsoids (15) for $\alpha = 0.9$ and $\alpha = 0.95$.

The budgeted uncertainty sets

$$U^\Gamma := \left\{ c \in \mathbb{R}^n \mid c_i = l_i + \delta_i (u_i - l_i),\ \sum_{i=1}^{n} \delta_i \leq \Gamma,\ \delta \in [0, 1]^n \right\},$$

where Γ is a given parameter and l is the mean cost vector described above, were created as follows: the deviations $(u_i - l_i)$ were chosen randomly between 0 and $0.05\sqrt{\chi_n^2(0.9)}\bar{c}_i$ for all $i = 1, \ldots, n$ to obtain instances which have similar size to the ellipsoids constructed above. Each instance has been solved for the two values of Γ from the set

$$\{0.15(2|V_C|), 0.5(2|V_C|)\},$$

rounded down if fractional. The latter choice is motivated by the fact that each feasible solution of Problem (M^3) uses at most $2|V_C|$ arcs in the graph. For $\Gamma \geq 2|V_C|$ our computations showed that often only one solution is needed to solve Problem (M^3) exactly while the instances for small Γ were more difficult to solve; in the former case, the solution agrees with the optimal solution of Problem (M^2).

An alternative to our approach would be to wait for the traffic scenario to materialize, e.g. considering the traffic in the morning, and afterwards solve the CVRP by a fast heuristic algorithm as described in Section 2. To compare the min-max-min approach to the latter procedure, for each instance we generated 100 random scenarios in U and compared the solution calculated by the heuristic of Groër (2012) to the best of our k solutions. More precisely, for each random scenario we applied the heuristic of Groër (2012) and compared the costs of the calculated solution to the costs of the best solution contained in the exact min-max-min solution. The average percental difference over all 100 random scenarios of all 10 instances is given in the tables below in column Δ_h. A positive value of Δ_h indicates that the min-max-min approach yields better solutions on average.

Since we naturally derive the ellipsoids as quantiles of normal distributions, to create the random scenarios for the ellipsoid instances we sample random vectors y out of a normal distribution with mean $\bar{c}$ and covariance matrix Σ given by the corresponding ellipsoid. To obtain y we calculated n independent standard normal distributed values $z_1, \ldots, z_n$ by the Box-Muller method and set

$$y := \Sigma z + \bar{c}.$$

To create the random scenarios for the budgeted uncertainty sets we first create n equally distributed random numbers $y_1, \ldots, y_n$ in $[0, \Gamma]$ and define $y_0 := 0$. Assume the numbers are given in increasing order. We then define $\delta_i := y_i - y_{i-1}$. If $\delta \leq \mathbf{1}$ is not true we start the procedure again. The random scenario is then given by c with

$$c_i = l_i + \delta_i (u_i - l_i).$$

Since the exact Algorithm 1 could not solve all instances for which the heuristic Algorithm 3 calculated a feasible solution, we compare the quality of the heuristic solution to a lower bound of the exact min-max-min problem. To calculate this lower bound, we solve Problem (14) using CPLEX 12.6; see Section 3.

					Algorithm 1					Algorithm 2					
Inst.	$\|V_C\|$	$\|A\|$	α	Δ_h	$\|X^*\|$	iter	t_{tot}	t_{dual}	t_{comb}	$\|X^*\|$	iter$_e$	iter$_h$	t_{tot}	t_{dual}	t_{comb}
eil7	6	42	90%	1.3	6.2	7.2	0.5	0.1	0.4	7.3	3.4	6.2	0.7	0.2	0.6
eil7	6	42	95%	1.2	6.4	7.5	0.5	0.1	0.4	6.9	3.8	5.4	0.7	0.2	0.5
eil13	12	156	90%	0.3	19.9	22.3	39507.1	2.5	39504.1	25.3	13.5	19.5	28815.6	4.6	28810.3
eil13	12	156	95%	0.3	19.3	21.2	40482.7	2.4	40479.8	25.5	11.5	20.6	28221.8	4.9	28216.2
P-n16-k8	15	240	90%	0.9	12.7	14.0	1099.5	7.9	1090.6	16.7	11.1	8.5	1002.7	12.0	989.5
P-n16-k8	15	240	95%	0.9	11.4	13.3	985.7	7.7	977.0	13.8	9.8	8.8	922.4	12.6	908.5
gr17	16	272	90%	8.6	5.1	6.1	635.8	3.5	631.2	5.0	4.4	4.8	589.6	5.9	582.4
gr17	16	272	95%	9.0	5.0	6.0	651.4	3.4	647.0	4.7	4.2	5.8	567.0	7.2	558.5

Table 1 Results of Algorithm 1 and 2 for CVRP with ellipsoidal uncertainty.

In the Tables 1, 2, 3, and 4 we list the computational results for a selection of instances. For each combination of $|V_C|$ and α or $|V_C|$ and Γ, respectively, we show the average over all 10 instances of the following numbers (from left to right): the average difference (in percent) over 100 random scenarios of the heuristic of Groër (2012) to the best solution contained in the optimal solution of the min-max-min problem; the number of solutions in the computed set X^*; the number of major iterations; the run-times used by the two oracles (t_{dual} for the dual problem in Step 4 and t_{comb} for solving the deterministic Vehicle Routing Problem (M) in Step 5) and the total run-time t_{tot} of the algorithm for an instance. Furthermore for Algorithm 2 where we use both the exact and the heuristic oracle, we show the number of calls of the exact oracle in column iter_e and the number of calls of the heuristic oracle in column iter_h. Table 3 and 4 which show the results for Algorithm 3, which only uses the heuristic algorithm, includes the columns Δ_e and Δ_{lb}. The column Δ_e shows the difference (in percent) of the objective value of the solution of Algorithm 3 to the exact optimal value if the latter is known. In column Δ_{lb} we show the difference (in percent) of the objective value of the solution of Algorithm 3 to the lower bound calculated by Problem (14). All times are given in CPU seconds and all numbers are rounded to one decimal.

					Algorithm 1					Algorithm 2					
Inst.	$\lvert V_C\rvert$	$\lvert A\rvert$	Γ	Δ_h	$\lvert X^*\rvert$	iter	t_{tot}	t_{dual}	t_{comb}	$\lvert X^*\rvert$	iter_e	iter_h	t_{tot}	t_{dual}	t_{comb}
eil7	6	42	1	0.7	4.1	5.4	0.3	0.0	0.3	3.9	2.5	3.8	0.3	0.0	0.3
eil7	6	42	6	1.2	10.6	14.6	0.7	0.0	0.7	10.6	7.5	8.7	1.0	0.0	0.9
eil13	12	156	3	-0.1	11.3	16.0	22488.0	0.0	22487.8	11.8	10.2	12.2	17006.5	0.0	17006.3
eil13	12	156	12	-0.1	19.2	34.0	58712.9	0.0	58712.7	19.1	20.0	28.5	42853.2	0.0	42852.9
P-n16-k8	15	240	4	0.4	7.4	10.9	399.4	0.0	399.0	7.0	7.3	8.5	372.6	0.0	372.2
P-n16-k8	15	240	15	0.7	13.8	25.9	1353.8	0.0	1353.4	13.7	21.2	16.0	1326.7	0.0	1326.2
gr17	16	272	4	8.0	8.1	10.9	2224.8	0.0	2224.3	7.9	7.7	7.6	2100.3	0.0	2099.7
gr17	16	272	16	8.8	20.9	37.1	51411.8	0.0	51411.2	20.9	31.7	19.0	51424.2	0.0	51423.6

Table 2 Results of Algorithm 1 and 2 for CVRP with budgeted uncertainty.

In Table 1 we show the results of Algorithm 1 and Algorithm 2 for ellipsoidal uncertainty sets. Interestingly, the number of calculated solutions and the number of iterations remains the same or even decreases for the larger α, i.e. the case where we have a larger uncertainty set. The total run-time as well as the number of solutions does not necessarily increase with the dimension. Instance gr17 could be solved very fast compared to the instance eil13, since the latter is much harder to solve by our formulation in its deterministic version. Most of the run-time is used by the deterministic oracle, while the dual oracle could be solved in a few seconds on average for all instances. Note that the combination of the exact oracle and the heuristic oracle in Algorithm 2 does only improve the run-time significantly for instance eil13. For all other instances the effect is very small. This is possibly due to the fact that the constraints in the dual problem calculated by the heuristic algorithm in each iteration are not as strong as the exact ones and the resulting higher number of iterations increases the run-time. The average percental improvement of using the min-max-min approach instead of solving a heuristic for an upcoming scenario is always positive. While for most of the instances the improvement is around 1%, for instance gr17 it is even more than 8%. Nevertheless it turned out that for instances with more than 16 customers we could not solve all configurations for the 10 ellipsoidal instances within days.

				Algorithm 3						
Inst.	$\|V_C\|$	$\|A\|$	α	Δ_e	Δ_{lb}	$\|X^*\|$	iter	t_{tot}	t_{dual}	t_{comb}
eil7	6	42	90%	1.7	12.2	3.4	4.6	0.2	0.1	0.1
eil7	6	42	95%	1.5	12.0	3.3	4.5	0.2	0.1	0.1
eil13	12	156	90%	1.4	21.2	6.7	8.3	2.5	1.1	0.9
eil13	12	156	95%	0.7	20.3	7.3	9.4	3.0	1.4	1.1
P-n16-k8	15	240	90%	1.8	11.4	4.3	6.0	5.0	3.2	0.9
P-n16-k8	15	240	95%	1.1	10.6	4.4	7.2	6.4	4.4	1.0
gr17	16	272	90%	6.9	11.7	2.2	5.0	6.8	4.4	1.2
gr17	16	272	95%	6.5	11.2	2.5	4.7	5.2	3.0	1.2
P-n20-k2	19	380	90%	-	7.1	5.7	7.6	15.6	10.7	2.0
P-n20-k2	19	380	95%	-	7.1	6.0	7.5	15.9	11.1	2.0
gr21	20	420	90%	-	11.0	5.5	7.6	18.0	11.5	2.8
gr21	20	420	95%	-	10.8	5.9	8.0	19.0	12.2	3.0
P-n23-k8	22	506	90%	-	11.0	4.8	7.7	32.4	25.2	2.1
P-n23-k8	22	506	95%	-	11.3	4.4	6.5	29.8	23.1	1.7

Table 3 Results of Algorithm 3 for CVRP with ellipsoidal uncertainty.

The results for budgeted uncertainty in Table 2 are very similar. The main difference is that here for larger Γ the number of calculated solutions and the number of iterations increases. Again instance eil13 is the hardest to solve. Interestingly, instance gr17 is much harder to solve here than in the ellipsoidal case. The combination of the exact and heuristic oracles again only has a significant impact on the run-time for instance eil13. The average percental improvement of using the min-max-min approach instead of solving a heuristic for an upcoming scenario is always positive except for instance eil13 where it is -0.1%. Again for instance gr17 the improvement is the largest and more than 8%. Nevertheless it turned out that for instances with more than 16 customers we could not solve all configurations for the 10 budgeted uncertainty instances within days.

The results for Algorithm 3 are very positive. For ellipsoidal uncertainty sets the total run-time and the number of calculated solutions is very low compared to the exact versions above. On average the total run-time never exceeded 33 seconds while the number of calculated solution was never larger than 8 on average, although we could solve instances with up to 22 customers, i.e. a dimension of 506. Interestingly here the run-time of the dual problem is significantly larger than the run-time of the heuristic for the large instances. The number of calculated solutions remains nearly the same for the different values of α. The difference to the exact optimal value of the min-max-min problem is never higher than 7% on average and is even around 1% for most of the instances. The percental difference compared to the lower bound is always around 10% (except for instance eil13) even for the instances which we could not solve exactly. This gives rise to the conjecture that the percental difference to the exact optimal value does not increase significantly for larger instances. The reason why we could not solve larger instances for ellipsoidal uncertainty was the fact that creating the instances took days for instances with more than 22 customers.

For budgeted uncertainty sets the total run-time and the number of calculated solutions is very low compared to the exact versions above. While the run-time is even much lower than for ellipsoidal

				Algorithm 3						
Inst.	$\|V_C\|$	$\|A\|$	Γ	Δ_e	Δ_{lb}	$\|X^*\|$	iter	t_{tot}	t_{dual}	t_{comb}
eil7	6	42	1	1.6	12.1	2.4	3.4	0.1	0.0	0.1
eil7	6	42	6	3.3	13.0	5.2	6.8	0.2	0.0	0.2
eil13	12	156	3	3.0	22.9	4.7	7.0	1.0	0.0	0.8
eil13	12	156	12	5.7	25.6	7.5	10.5	1.4	0.0	1.2
P-n16-k8	15	240	4	3.7	13.2	3.7	6.0	1.1	0.0	0.7
P-n16-k8	15	240	15	7.1	16.9	4.8	7.4	1.3	0.0	0.9
gr17	16	272	4	7.8	14.5	4.4	6.8	2.2	0.0	1.6
gr17	16	272	16	8.6	15.5	7.8	10.4	2.9	0.0	2.3
P-n20-k2	19	380	5	-	8.8	8.0	12.4	4.4	0.0	3.1
P-n20-k2	19	380	19	-	7.7	14.9	22.9	6.8	0.0	5.3
gr21	20	420	6	-	14.0	6.9	9.5	4.4	0.0	2.8
gr21	20	420	20	-	14.4	12.1	15.9	6.3	0.0	4.7
P-n23-k8	22	506	6	-	19.7	4.2	6.0	4.2	0.0	1.5
P-n23-k8	22	506	22	-	23.9	6.8	10.3	5.2	0.0	2.5
A-n34-k5	33	1122	9	-	14.6	1.0	2.1	28.1	0.0	1.7
A-n34-k5	33	1122	33	-	14.6	1.0	2.1	28.1	0.0	1.7
P-n40-k5	39	1560	11	-	9.6	22.4	35.5	96.2	0.0	28.9
P-n40-k5	39	1560	39	-	-	-	-	-	-	-
A-n45-k6	44	1980	13	-	17.2	18.8	25.9	161.9	0.0	26.1
A-n45-k6	44	1980	44	-	-	-	-	-	-	-

Table 4 Results of Algorithm 3 for CVRP with budgeted uncertainty.

uncertainty for the large instances, the number of calculated solutions is about the same for most of the instances. On average the total run-time never exceeded 162 seconds while the number of calculated solution was never larger than 23 on average, although we could solve instances with up to 44 customers, i.e. a dimension of 1980. Here in contrast to the ellipsoidal case the run-time of the dual problem is not larger than a tenth of a second while the run-time of the heuristic is a bit larger. As in the exact case the number of calculated solutions increases for larger Γ. The difference to the exact optimal value of the min-max-min problem is never higher than 9% on average and is even around 5% for most of the instances. The percental difference compared to the lower bound is around 15% for most of the instances even for the instances which we could not solve exactly. So similar to the ellipsoidal case the exact optimal value possibly does not increase significantly for larger instances.

In Fig. 3, we present all solutions of an exact optimal solution of Problem (M^3) for a selected ellipsoidal instance of P-$n16$-$k8$. Since we assumed a directed graph which can have different costs on arcs (i,j) and (j,i) for any customers $i,j \in V_C$ it can happen that in one optimal solution one set of tours can occur more than once but at least one tour is oriented in a different direction (see solutions 3 and 9 in Fig. 3).

Additionally to the experiments presented above, we compared our approach to a simple heuristic approach for solving Problem (M^3). The idea is to choose n non-dominated scenarios $c_1,\ldots,c_n \in U$ and to solve the deterministic CVRP with objective vector c_i for each i. Clearly, the set of the n resulting solutions forms a feasible solution for the min-max-min Problem (M^3). We generated

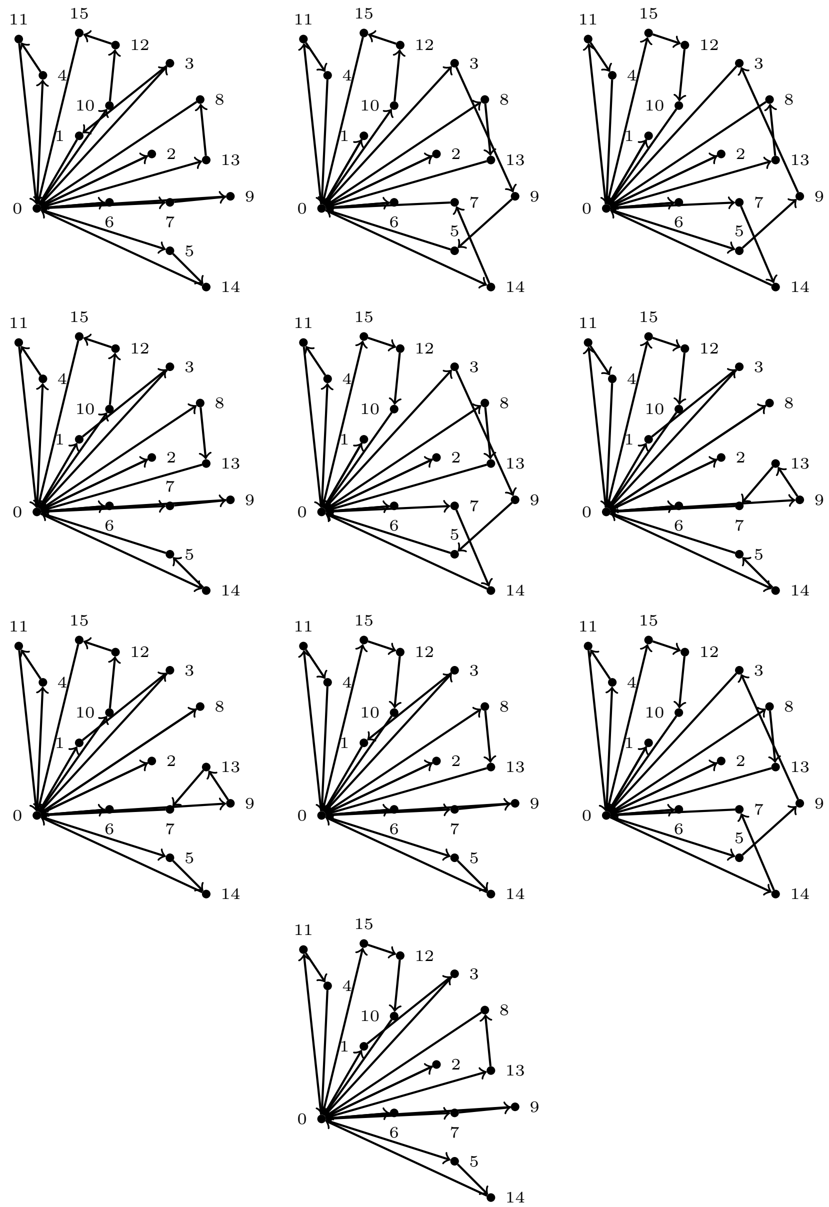


Figure 3 The optimal solution of Problem (M^3) for instance P-n16-k8 with an ellipsoidal uncertainty set.

the non-dominated scenarios by the following procedure: we calculate a random vector $\lambda \in (0,1]^n$, where each entry λ_i is chosen randomly in the interval $(0,1]$. Then we optimize over our uncertainty set U in direction λ and choose the optimal solution as the non-dominated scenario.

In Figures 4 and 5 we show the evolution of the objective value of the latter heuristic in terms of the number of calculated non-dominated scenarios, for two given instances. The values in the figures are the percental differences of the min-max-min objective value of the actual feasible solution compared to the optimal value of the min-max-min problem. The figures show that the objective value improves very fast at the beginning, while after around $\frac{1}{4}n$ scenarios the improvement is

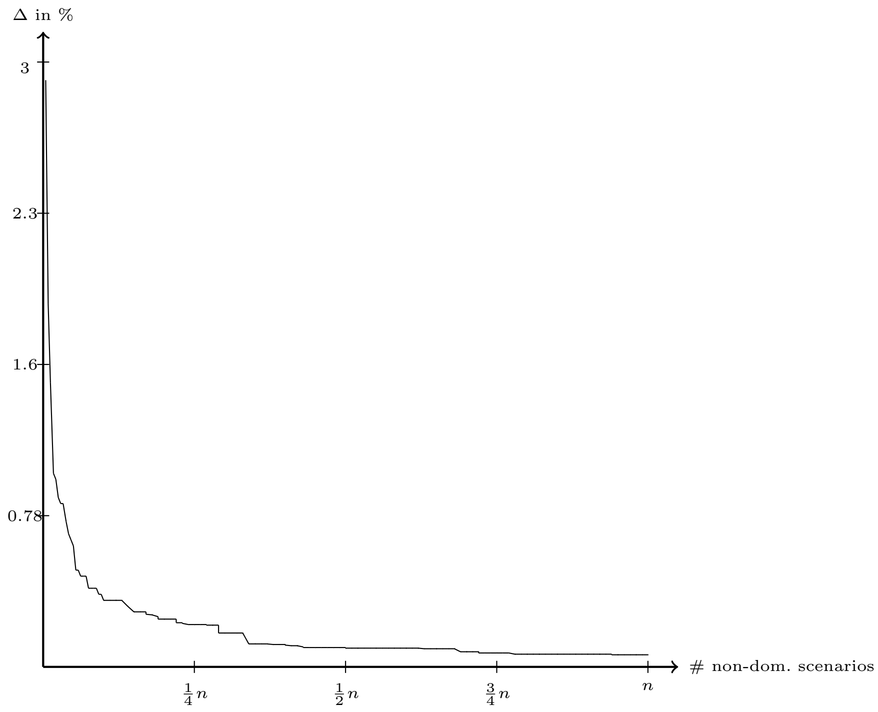


Figure 4 **Percental difference of the heuristic value to the optimal value in terms of the number of generated non-dominated scenarios, for an ellipsoidal instance of** P-$n16$-$k8$ **with** $\alpha = 0.95$.

small. Interestingly, for budgeted uncertainty sets the average difference does not fall below 2% even for n scenarios, while for ellipsoidal uncertainty sets it is very close to 0% then.

In terms of solution quality, this heuristic approach thus seems to be promising. However, the catch of this approach is its running time: for each of the generated scenarios, a deterministic instance of the CVRP has to be solved. In comparison, for Algorithm 1, we usually need to solve much fewer instances of the deterministic CVRP, so that our exact approach turns out to be significantly faster than the heuristic based on non-dominated scenarios. For this reason, we do not include a detailed comparison of the two approaches.

5. Conclusion.

In this paper we use a robust approach investigated in Buchheim and Kurtz (2017) to solve the Capacitated Vehicle Routing Problem with uncertain travel times. We calculate up to k feasible solutions which are optimal for the min-max-min problem and therefore hedge against the uncertain travel times of the vehicles in a robust way. We implemented the algorithm proposed in Buchheim and Kurtz (2017) and derived a heuristic algorithm for the problem by replacing the exact oracle, used in each iteration, by a heuristic oracle. Our results on several benchmark instances for different kinds of uncertainty sets look very promising. By using heuristic algorithms for the deterministic CVRP, we could not speed up the exact algorithm for the min-max-min problem, but the heuristic

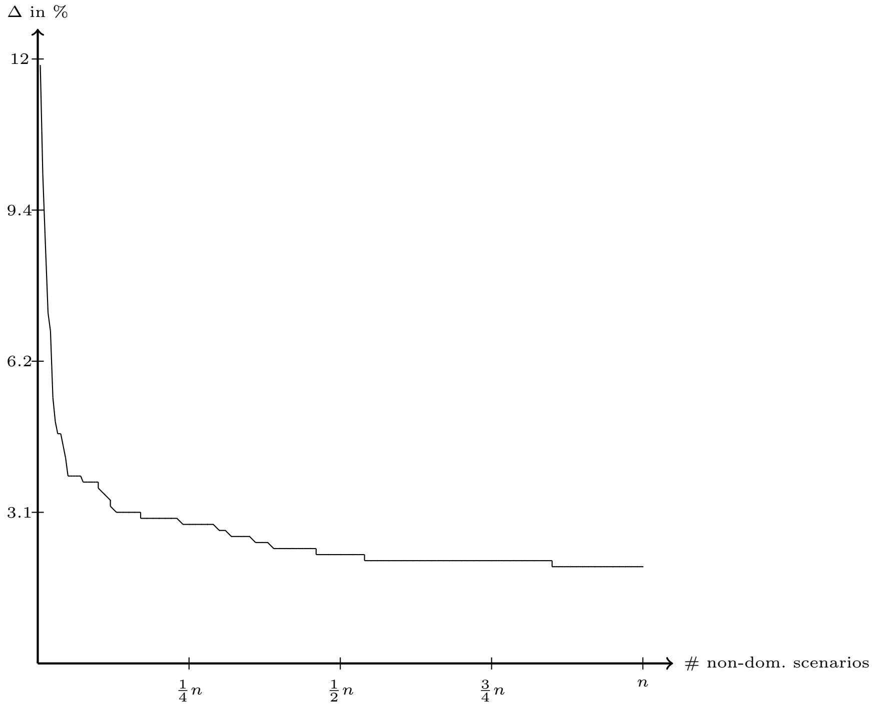


Figure 5 Percental difference of the heuristic value to the optimal value in terms of the number of generated non-dominated scenarios, for a budgeted instance of P-$n16$-$k8$ with $\Gamma = 15$.

variant of the algorithm runs very fast and could also solve larger instances. Furthermore an optimal solution of the min-max-min problem is on average around 1% better than using a heuristic algorithm each time a scenario comes up, even though it chooses the solution from a small set of candidates.

Several directions for future research can be identified. First, it would be promising to use different exact and heuristic algorithms for the deterministic CVRP in each iteration of Algorithm 1. Second, since often pairs of different solutions differ only in one tour (see Figure 3), one could try to adjust the idea of the min-max-min model and only calculate a set of tours instead of complete solutions. These tours could then be combined to a feasible solution by the user each time a scenario occurs. This would require however to define specific rules for the feasible combination of tours.

It also seems promising to adapt the min-max-min robust approach to other logistical problems, such as pickup and delivery problems or even hub location problems. Finally, Vehicle Routing Problems with time windows or uncertain demands could be investigated. However, these versions of the VRP are more challenging than the considered CVRP with uncertain travel times: as the uncertainty could now affect the feasibility of the solutions, these problems would require an adaptation of the min-max-min robust approach and, in particular, of Algorithm 1.

References

Agra A, Christiansen M, Figueiredo R, Hvattum LM, Poss M, Requejo C (2013) The robust vehicle routing problem with time windows. *Computers & Operations Research* 40(3):856–866.

Aissi H, Bazgan C, Vanderpooten D (2009) Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research* 197(2):427–438.

Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* 218(1):1–6.

Baldacci R, Toth P, Vigo D (2010) Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research* 175(1):213–245.

Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A (2004) Adjustable robust solutions of uncertain linear programs. *Mathematical Programming* 99(2):351–376.

Ben-Tal A, Nemirovski A (1998) Robust convex optimization. *Mathematics of Operations Research* 23(4):769–805.

Ben-Tal A, Nemirovski A (1999) Robust solutions of uncertain linear programs. *Operations Research Letters* 25(1):1–13.

Bertsimas D, Caramanis C (2010) Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control* 55(12):2751–2766.

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. *Mathematical Programming* 98(1–3):49–71.

Bertsimas D, Sim M (2004) The price of robustness. *Operations Research* 52(1):35–53.

Braaten S, Gjønnes O, Hvattum LM, Tirado G (2017) Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications* 77:136–147.

Buchheim C, Kurtz J (2017) Min–max–min robust combinatorial optimization. *Mathematical Programming* 163(1):1–23.

Christofides N (1976) The vehicle routing problem. *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle* 10(V1):55–70.

Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4):568–581.

Cordeau JF, Gendreau M, Laporte G, Potvin JY, Semet F (2002) A guide to vehicle routing heuristics. *The Journal of the Operational Research Society* 53(5):512–522.

Cordeau JF, Laporte G, Savelsbergh MW, Vigo D (2007) Chapter 6: Vehicle routing. Barnhart C, Laporte G, eds., *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, 367–428 (Elsevier).

Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Science* 6(1):80–91.

De La Vega J, Munari P, Morabito R (2017) Robust optimization for the vehicle routing problem with multiple deliverymen. *Central European Journal of Operations Research* 1–32.

Diaz BD (2012) Vrp web. http://neo.lcc.uma.es/vrp/.

Fischetti M, Monaci M (2009) Light robustness. Ahuja RK, Möhring RH, Zaroliagis CD, eds., *Robust and online large-scale optimization*, 61–84 (Springer).

Gavish B, Graves SC (1982) Scheduling and routing in transportation and distribution systems: formulations and new relaxations. *Rochester: Graduate School of Management, University of Rochester* .

Golden B, Raghavan S, Wasil EA (2008) *The vehicle routing problem : latest advances and new challenges.* Operations research/Computer science interfaces series, 43 (Springer).

Gounaris CE, Wiesemann W, Floudas CA (2013) The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research* 61(3):677–693.

Groër C (2008) *Parallel and serial algorithms for vehicle routing problems.* Ph.D. thesis.

Groër C (2012) VRPH. *URL http://www.coin-or.org/projects/VRPH* .

Hanasusanto GA, Kuhn D, Wiesemann W (2015) K-adaptability in two-stage robust binary programming. *Operations Research* 63(4):877–891.

IBM ILOG CPLEX (2014) 12.6.

Jaillet P, Qi J, Sim M (2016) Routing optimization under uncertainty. *Operations Research* 64(1):186–200.

Kasperski A, Zieliński P (2017) Robust two-stage network problems. Dörner KF, Ljubic I, Pflug G, Tragler G, eds., *Operations Research Proceedings 2015*, 35–40 (Springer).

Kouvelis P, Yu G (1996) *Robust Discrete Optimization and Its Applications* (Springer).

Kumar S, Panneerselvam R (2012) A survey on the vehicle routing problem and its variants. *Intelligent Information Management* 4(3):66–74.

Laporte G (1992) The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3):345–358.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Science* 43(4):408–416.

Laporte G, Ropke S, Vidal T (2014) Chapter 4: Heuristics for the vehicle routing problem. Toth P, Vigo D, eds., *Vehicle Routing: Problems, Methods, and Applications*, volume 18, 87–116 (SIAM Philadelphia).

Lee C, Lee K, Park S (2012) Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society* 63(9):1294–1306.

Lenstra JK, Rinnooy Kan AHG (1981) Complexity of vehicle routing and scheduling problems. *Networks* 11(2):221–227.

Letchford AN, Salazar-González JJ (2006) Projection results for vehicle routing. *Mathematical Programming* 105(2):251–274.

Letchford AN, Salazar-González JJ (2015) Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research* 244(3):730–738.

Liebchen C, Lübbecke M, Möhring R, Stiller S (2009) The concept of recoverable robustness, linear programming recovery, and railway applications. Ahuja RK, Mhring RH, Zaroliagis CD, eds., *Robust and online large-scale optimization*, 1–27 (Springer).

Ordóñez F (2014) Robust vehicle routing. Hasenbein JJ, Gray P, eds., *Risk and Optimization in an Uncertain World*, 153–178 (INFORMS).

Poggi M, Uchoa E (2014) Chapter 3: New exact algorithms for the capacitated vehicle routing problem. Toth P, Vigo D, eds., *Vehicle Routing: Problems, Methods, and Applications*, volume 18, 59–86 (SIAM Philadelphia).

Rostami B, Desaulniers G, Errico F, Lodi A (2017) The vehicle routing problem with stochastic and correlated travel times. Technical report.

Schöbel A (2014) Generalized light robustness and the trade-off between robustness and nominal quality. *Mathematical Methods of Operations Research* 80(2):161–191.

Sim M (2004) *Robust optimization*. Ph.D. thesis, Massachusetts Institute of Technology.

Solano-Charris E, Prins C, Santos AC (2015) Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios. *Applied Soft Computing* 32:518–531.

Solano-Charris EL, Prins C, Santos AC (2016) Solving the bi-objective robust vehicle routing problem with uncertain costs and demands. *RAIRO-Operations Research* 50(4-5):689–714.

Sun L (2014) A new robust optimization model for the vehicle routing problem with stochastic demands. *Journal of Interdisciplinary Mathematics* 17(3):287–309.

Sungur I, Ordónez F, Dessouky M (2008) A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions* 40(5):509–523.

Toth P, Vigo D, eds. (2001) *The Vehicle Routing Problem* (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics).

Toth P, Vigo D, eds. (2014) *Vehicle routing: problems, methods, and applications* (Siam).

Wohlgemuth S, Oloruntoba R, Clausen U (2012) Dynamic vehicle routing with anticipation in disaster relief. *Socio-Economic Planning Sciences; Special Issue: Disaster Planning and Logistics: Part 2* 46(4):261–271.

Yellow PC (1970) A computational modification to the savings method of vehicle scheduling. *Journal of the Operational Research Society* 21(2):281–283.