OJMO

Open Journal of Mathematical Optimization

Marc Goerigk & Stefan Lendl

Robust Combinatorial Optimization with Locally Budgeted Uncertainty

Volume 2 (2021), article no. 3 (18 pages)

<http://ojmo.centre-mersenne.org/item/OJMO_2021__2__A3_0>

Article submitted on October 6, 2020, revised on April 27, 2021,
accepted on May 6, 2021.

*Open Journal of Mathematical Optimization is member of the
Centre Mersenne for Open Scientific Publishing*
www.centre-mersenne.org

Robust Combinatorial Optimization with Locally Budgeted Uncertainty

Marc Goerigk
Network and Data Science Management, University of Siegen, Siegen, Germany
marc.goerigk@uni-siegen.de

Stefan Lendl
Institute of Discrete Mathematics, Graz University of Technology, Graz, Austria
Department of Operations and Information Systems, University of Graz, Austria
lendl@math.tugraz.at
stefan.lendl@uni-graz.at

Abstract

Budgeted uncertainty sets have been established as a major influence on uncertainty modeling for robust optimization problems. A drawback of such sets is that the budget constraint only restricts the global amount of cost increase that can be distributed by an adversary. Local restrictions, while being important for many applications, cannot be modeled this way.

We introduce a new variant of budgeted uncertainty sets, called locally budgeted uncertainty. In this setting, the uncertain parameters are partitioned, such that a classic budgeted uncertainty set applies to each part of the partition, called region.

In a theoretical analysis, we show that the robust counterpart of such problems for a constant number of regions remains solvable in polynomial time, if the underlying nominal problem can be solved in polynomial time as well. If the number of regions is unbounded, we show that the robust selection problem remains solvable in polynomial time, while also providing hardness results for other combinatorial problems.

In computational experiments using both random and real-world data, we show that using locally budgeted uncertainty sets can have considerable advantages over classic budgeted uncertainty sets.

Digital Object Identifier 10.5802/ojmo.5

Keywords robust optimization, combinatorial optimization, budgeted uncertainty.

Acknowledgments Supported by the Austrian Science Fund (FWF): W1230 and the Deutsche Forschungsgemeinschaft (DFG) through grant GO 2069/1-1. We thank the anonymous referees for helpful comments that improved our paper.

1 Introduction

We consider nominal combinatorial optimization problems of the form

$$\begin{aligned} \min\ & \boldsymbol{c}^t\boldsymbol{x} \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

where $\mathcal{X} \subseteq \{0,1\}^n$ is the set of feasible solutions. For uncertain cost coefficients $\boldsymbol{c} \in \mathcal{U}$, robust optimization approaches have been analyzed. To this end, one assumes that a set $\mathcal{U}$ of possible cost realizations is given by a decision maker or derived from historical data. The set $\mathcal{U}$ is referred to as the uncertainty set. The (min-max) robust counterpart is then to solve

$$\min_{\boldsymbol{x}\in\mathcal{X}} \max_{\boldsymbol{c}\in\mathcal{U}} \boldsymbol{c}^t\boldsymbol{x}.$$

Different possibilities to model the set $\mathcal{U}$ have been proposed. One straight-forward possibility is to use a discrete set of scenarios $\mathcal{U} = \{\boldsymbol{c}^1,\dots,\boldsymbol{c}^N\}$, i.e., to list all possible outcomes explicitly. While this approach is flexible, it usually results in NP-hard robust optimization problems, even if the nominal problem can be solved in polynomial time (see [1, 19, 21] for overviews). Also, implicit descriptions of the uncertainty set can lead to exponential-sized equivalent discrete uncertainty sets.

A popular alternative are budgeted uncertainty sets of the form

$$\mathcal{U} = \left\{ \boldsymbol{c} = \underline{\boldsymbol{c}} + \boldsymbol{\delta} : \delta_i \in [0, d_i] \ \ \forall\, i \in [n], \ \sum_{i \in [n]} \delta_i \le \Gamma \right\} \tag{1}$$

as first introduced in [5, 6]. Here we use the notation $[n] = \{1, \dots, n\}$. For every item $i \in [n]$, we are given a lower bound on the costs $\underline{c}_i$, as well as a possible maximum cost deviation d_i. Additionally, there is a budget Γ on the total increase of costs over the lower bound. Advantages of this set include its intuitive description for a decision maker, and that robust counterparts remain efficiently solvable for nominal problems that can be solved efficiently, even though the budgeted uncertainty set has an exponential number of extreme points. These benefits have lead to a substantial amount of research into robust optimization problems with budgeted uncertainty sets, see, e.g., [3, 7, 10, 11, 18] and many more.

But there are also limitations to this approach, which has lead to the development of alternative uncertainty sets. These include multi-band uncertainty [8], variable budgeted uncertainty [22], and knapsack uncertainty [23].

In the current literature, little attentions has been paid to avoiding the potential problem that the constraint $\sum_{i \in [n]} \delta_i \le \Gamma$ denotes a global budget over all uncertain parameters. For various applications, multiple local budgets are more desirable. As examples, consider multi-period problems, where every period has its own budget limitation, routing problems, where separate budgets apply to geographic regions or types of roads, and portfolio problems, where uncertainty budgets are restricted to asset classes or sectors. The only work we are aware of in this setting is [17], where the uncertain demand for a vehicle routing problem is split into separate regions.

In this paper we introduce a new type of budgeted uncertainty set, where budgets apply locally to their respective regions. These sets are of the form

$$\mathcal{U} = \left\{ \boldsymbol{c} = \underline{\boldsymbol{c}} + \boldsymbol{\delta} : \delta_i \in [0, d_i] \ \ \forall\, i \in [n], \ \sum_{i \in P_j} \delta_i \le \Gamma_j \ \ \forall\, j \in [K] \right\},$$

where $P_1 \cup P_2 \cup \ldots \cup P_K = [n]$ denotes a partition of the items. Each set P_j is called a region. In this approach, every region has a separate budget constraint, which models the local uncertainty. Note that this definition of uncertainty is a generalization of the classic definition (1), which can be recovered by using $K = 1$.

Our contributions are as follows. For min-max problems with locally budgeted uncertainty, we first derive a compact formulation in Section 2.1. Based on this formulation, we then consider the case of a constant number of regions in Section 2.2. We show that the robust problem remains solvable in polynomial time, if it is possible to solve the nominal problem in polynomial time. For an unbounded number of regions, the selection problem remains solvable in polynomial time, while this is not the case for the representative selection problem (see Section 2.3). We conclude that also the spanning tree problem, the s-t-min-cut problem, and the shortest path problem become NP-hard. Additionally, assuming the exponential time hypothesis (ETH), we can exclude the possibility of parameterized algorithms with running time in $O^*(2^{o(K)})$. For the min-knapsack problem, we provide an FPTAS. In Section 3, we present three computational experiments using locally budgeted uncertainty sets. In all experiments, we compare locally budgeted uncertainty to the classic budgeted uncertainty approach. While the first two experiments use randomly generated data, the third experiment is based on real-world data for robust shortest path problems. Section 4 concludes the paper and points out further questions.

2 Theoretical Results

2.1 A Compact Formulation

Let some solution $\boldsymbol{x} \in \mathcal{X}$ be fixed. Its objective value is then determined by solving the adversarial problem

$$\max_{\boldsymbol{c} \in \mathcal{U}} \ \boldsymbol{c}^t \boldsymbol{x},$$

that is, by choosing a scenario $\boldsymbol{c}$ that maximizes the costs of $\boldsymbol{x}$. Using the definition of locally budgeted uncertainty, this is equivalent to solving the following linear program:

$$\max \sum_{i\in[n]} (\underline{c}_i + \delta_i)x_i \tag{2}$$

$$\text{s.t.} \sum_{i\in P_j} \delta_i \leq \Gamma_j \qquad \forall\, j \in [K] \tag{3}$$

$$\delta_i \leq d_i \qquad \forall\, i \in [n] \tag{4}$$

$$\delta_i \geq 0 \qquad \forall\, i \in [n]. \tag{5}$$

By strong duality, we can dualize this linear program to find another linear program with the same optimal objective value. Furthermore, any feasible solution to the dual problem gives an upper bound to the objective value of the primal problem. Using the dual, we hence find the following compact problem formulation for the min-max problem with locally budgeted uncertainty:

$$\min \sum_{j\in[K]} \left(\Gamma_j\pi_j + \sum_{i\in P_j} d_i\rho_i + \sum_{i\in P_j} \underline{c}_i x_i \right) \tag{6}$$

$$\text{s.t. } \pi_j + \rho_i \geq x_i \qquad \forall\, j \in [K], i \in P_j \tag{7}$$

$$\pi_j \geq 0 \qquad \forall\, j \in [K] \tag{8}$$

$$\rho_i \geq 0 \qquad \forall\, i \in [n] \tag{9}$$

$$\boldsymbol{x} \in \mathcal{X}. \tag{10}$$

Recall that $\mathcal{X}$ represents the set of feasible solutions for the underlying combinatorial problem. Variables π_j are the duals of Constraints (3), and variables ρ_i are the duals of Constraints (4).

2.2 Constant Number of Regions

We first consider the case that the number of regions K is a constant value. Note that, in an optimal solution, we can assume that $\rho_i = [x_i - \pi_j]_+$, where $[y]_+ = \max\{0, y\}$ denotes the positive part of y.

▶ **Lemma 1.** *There is an optimal solution to Problem (6–10), where $\pi_j \in \{0,1\}$ for all $j \in [K]$.*

Proof. Let us assume that $\boldsymbol{x} \in \mathcal{X}$ is fixed. Let $X = \{i \in [n] : x_i = 1\}$ denote the set of items taken by solution $\boldsymbol{x}$. The problem then decomposes to:

$$\min_{\boldsymbol{\pi}\geq\mathbf{0}} \sum_{j\in[K]} \left(\Gamma_j\pi_j + \sum_{i\in P_j} d_i[x_i - \pi_j]_+ + \sum_{i\in P_j} \underline{c}_i x_i \right) \tag{11}$$

$$= \sum_{i\in[n]} \underline{c}_i x_i + \sum_{j\in[K]} \min_{\pi_j\geq 0} \left(\Gamma_j\pi_j + \sum_{i\in P_j} d_i[x_i - \pi_j]_+ \right) \tag{12}$$

$$= \sum_{i\in X} \underline{c}_i + \sum_{j\in[K]} \min_{\pi_j\in[0,1]} \left(\Gamma_j\pi_j + \sum_{i\in P_j\cap X} d_i(1 - \pi_j) \right). \tag{13}$$

Note that Equation (13) holds as increasing any variable π_j beyond 1 can never be optimal for $\Gamma_j > 0$. If $\Gamma_j = 0$, then setting $\pi_j = 1$ gives the same value as in (12). We can conclude that there is an optimal solution with $\pi_j \in \{0,1\}$ for all $j \in [K]$. ◀

We can conclude the following result (compare also to [23, Theorem 3]).

▶ **Theorem 2.** *The robust problem with locally budgeted uncertainty (6–10) can be decomposed into 2^K subproblems of nominal type. In particular, if K is a constant and the nominal problem can be solved in polynomial time, Problem (6–10) can be solved in polynomial time as well.*

Proof. By Lemma 1, we can assume every variable π_j to be either 0 or 1. We guess these values. There are K variables π_j, and thus 2^K combinations are possible. For fixed $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, denote by $\Pi \subseteq [K]$ the set of indices j where $\pi_j = 1$. The problem then becomes

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{j\in[K]} \left(\Gamma_j \pi_j + \sum_{i\in P_j} d_i [x_i - \pi_j]_+ + \sum_{i \in P_j} \underline{c}_i x_i \right)$$
$$= \sum_{j\in\Pi} \Gamma_j + \min_{\boldsymbol{x}\in\mathcal{X}} \left(\sum_{j\in\Pi} \sum_{i\in P_j} \underline{c}_i x_i + \sum_{j\in[K]\setminus\Pi} \sum_{i\in P_j} (\underline{c}_i + d_i) x_i \right).$$

This is a problem of nominal type, and the claim follows. ◀

2.3 Unbounded Number of Regions

We now consider the case that the number of regions K is not a constant, but part of the problem input.

2.3.1 Hardness Results

We first consider the representative selection problem, where

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \{0,1\}^n : \sum_{i\in T_\ell} x_i = p_\ell \ \ \forall\, \ell \in [L] \right\}$$

for a partition $T_1 \cup T_2 \cup \ldots \cup T_L = [n]$ and integers p_ℓ for all $\ell \in [L]$ (see, e.g., [12, 16]).

▶ **Theorem 3.** *The robust representative selection problem with locally budgeted uncertainty and arbitrary K is APX-hard, even if $|T_\ell| = 2$, $p_\ell = 1$ for all $\ell \in [L]$, and $|P_j| \le 3$ for all $j \in [K]$.*

Proof. We provide an objective-preserving reduction from an instance of the vertex cover problem, which is APX-hard, even on 3-regular graphs [2, 14].
Given: Graph $G = (V, E)$ 3-regular, $k \in \mathbb{N}$
Question: Does there exist a vertex cover of size less or equal to k, i.e. a set $S \subseteq V$ such that for all $e = \{u, v\} \in E$ it holds that $u \in S$ or $v \in S$, and $|S| \le k$?

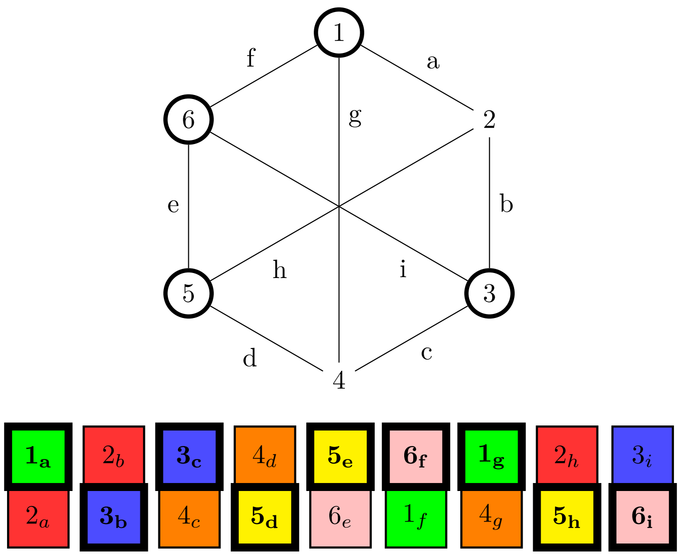


Figure 1 Illustration of the reduction from vertex cover. The big circled vertices correspond to a minimum size vertex cover of the graph. Below we show the instance of the robust representative selection problem corresponding to this graph. Each column corresponds to a partition T_e from which one of the two elements must be selected. The colors correspond to the regions P_v of the instance. The bold elements are an optimal solution corresponding to the shown vertex cover of the graph. Note that only elements in four regions (green, blue, yellow, pink), corresponding to the vertices 1,3,5,6 that are selected.

Given such an instance, we construct an instance of the robust representative selection problem with locally budgeted uncertainty fulfilling the restrictions stated in the theorem. In Figure 1 we illustrate the reduction via an example for a concrete vertex cover instance. Let $L = |E|$ and $n = 2|E|$. For each $e = \{u, v\} \in E$ let u_e, v_e be the two elements of $[n]$ in T_e, which we associate with the two vertices u and v. Note that for every vertex $v \in V$ there exist degree of v many elements that are associated with this vertex, one corresponding to every edge incident to v.

For our partition into regions of the locally budgeted uncertainty set, we use exactly those sets of elements that correspond to the same vertices in V, i.e. we define a region $P_v = \{v_e : e \in E, v \text{ incident to } e\}$ for every $v \in V$. Note that these sets also form a partition of $[n]$, and we have $K = |V|$. We further set $\boldsymbol{\Gamma} = \mathbf{1}$, $\underline{\boldsymbol{c}} = \mathbf{0}$ and $\boldsymbol{d} = \mathbf{1}$.

We show that there is a vertex cover of size at most k if and only if the constructed instance of the robust representative selection problem with locally budgeted uncertainty has a solution with objective value at most k. To see this we first prove the following claim.

▷ Claim 4. Given a feasible solution $\boldsymbol{x}$ of our instance of the robust representative selection problem, the robust objective value is equal to the number of regions P_v in which at least one element is selected, i.e. equal to

$$|\{v \in V : \ \exists\, i \in P_v \text{ with } x_i = 1\}|.$$

Proof of Claim. First observe that this value can be realized by the adversary by selecting for each region P_v with an element $i \in P_v$ such that $x_i = 1$ an arbitrary such element i and set $\delta_i = 1$. It is easy to see that this is a feasible solution for the adversary and the claimed objective value is reached.

To show that the adversary cannot achieve a larger value, we observe that for any $i \in [n]$ with $x_i = 0$, we can assume that $\delta_i = 0$ in an optimal solution. Hence, by the definition of $\mathcal{U}$ we have that $|\{v \in V : \ \exists\, i \in P_v \text{ with } x_i = 1\}|$ is an upper bound on the objective value of the adversary. ◁

Now given a vertex cover S of size k, we construct a solution $\boldsymbol{x}$ by selecting in each T_e the element corresponding to the vertex in the vertex cover. If both vertices incident to e are in S we choose one of the two elements arbitrarily. Since $|S| = k$, we select elements from at most k different regions P_v. Hence, by our claim the objective value of the robust representative selection problem is less or equal to k.

Given a solution $\boldsymbol{x}$ to the robust representative selection problem with objective value k, we know by our claim that the elements selected by $\boldsymbol{x}$ are contained in exactly k different regions $P_{v_1}, \ldots, P_{v_k}$. We define S to be the set corresponding to exactly those k different vertices, i.e. $S = \{v_1, \ldots, v_k\}$. Since for each $T_e, e \in E$ one element is selected by $\boldsymbol{x}$, also for every edge e at least one incident vertex is contained in S, hence S is a vertex cover of size k. ◀

▶ **Corollary 5.** *The robust problem with locally budgeted uncertainty with arbitrary K is APX-hard for the shortest path problem on series-parallel graphs, for the minimum spanning tree problem, and for the s-t-min-cut problem, even if for the regions it holds that $|P_j| \leq 3$ for all $j \in [K]$.*

Proof. The result for the shortest path and minimum spanning tree problem follows directly from Theorem 3. To see this, given an instance of the representative selection problem with $|T_\ell| = 2$ and $p_\ell = 1$ for all $\ell \in [L]$, we construct the graph G with vertex set $V = \{0, 1, \ldots, L\}$ and edge set E consisting of parallel edges e_ℓ^1, e_ℓ^2 connecting vertex $\ell - 1$ with vertex ℓ for all $\ell \in [L]$. Here e_ℓ^1, e_ℓ^2 are in one-to-one correspondence with the two elements in T_ℓ. It is now easy to see that both spanning trees and paths from nodes 0 to L in G are in one-to-one correspondence with feasible solutions to the original representative selection problem. Using the same locally budgeted uncertainty set as for the representative selection problem, we find that objective values of corresponding solutions remain equal.

To obtain the result for the s-t-min-cut problem, observe that the special instance of the representative selection problem is equivalent to the s-t-cut problem in a graph G where for each part $\ell \in [L]$ of size 2 we add a special vertex v_ℓ in addition to s and t and the path from s via v_ℓ to t. Then s-t-cuts correspond to selecting one of the two edges from each of these paths. ◀

Note that the above reduction does not exclude the possibility of a parameterized algorithm with running time $O^*(2^{o(K)})$, even if we assume the exponential time hypothesis (ETH), since the number of regions K in the reduction cannot be bounded by the solution size k of the vertex cover. In the following we give a direct linear parameterized reduction from 3-SAT to robust representative selection with locally budgeted uncertainty, which shows that the running time of our FPT meta-algorithm obtained in Theorem 2 is essentially tight under ETH. The proofs of the following theorems can be found in Appendix A.

▶ **Theorem 6.** *Assuming ETH, there is no $O^*(2^{o(K)})$ time algorithm for the robust representative selection problem with locally budgeted uncertainty, even if $|T_\ell| \leq 3$ and $p_\ell = 1$ for all $\ell \in [L]$.*

This result is slightly weaker than Theorem 3 in the sense that we cannot assume $|T_\ell| = 2$ but only $|T_\ell| \leq 3$. The existence of a $O^*(2^{o(K)})$ time algorithm for this case is an open problem.

▶ **Theorem 7.** *Let $k = \max_{\ell \in [L]} |T_\ell|$ and $\Delta = \max_{j \in [K]} |P_j|$. Then the robust representative selection problem with locally budgeted uncertainty and arbitrary K inherits inapproximability results from the set cover problem with maximum cardinality of subsets Δ, and maximum number k of subsets containing any element of the ground set.*

Note that the inapproximability of set cover under parameters Δ and k is well-researched (see, e.g., [24]). If $k = \theta(\log \Delta)$, both problems become hard to approximate within $\Omega(\log \Delta / (\log \log \Delta)^2)$.

2.3.2 A Polynomial Time Algorithm for the Selection Problem

While the results from the previous section indicate that the robust counterpart of even simple combinatorial problems becomes hard, we now show that this is not the case for the selection problem, where $\mathcal{X} = \{\boldsymbol{x} \in \{0,1\}^n : \sum_{i \in [n]} x_i = p\}$ for some integer p.

To this end, we use a dynamic program over how many items are taken from every region. Let $n_j = |P_j|$ be the number of items in region $j \in [K]$. If the number of items p_j taken from region P_j is fixed, the robust problem can be decomposed:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{\boldsymbol{c} \in \mathcal{U}} \boldsymbol{c}^t \boldsymbol{x} = \sum_{j \in [K]} \min_{\boldsymbol{x} \in \mathcal{X}_j} \max_{\boldsymbol{c} \in \mathcal{U}_j} \boldsymbol{c}^t \boldsymbol{x} =: \sum_{j \in [K]} f_j(p_j),$$

where $\mathcal{X}_j = \{\boldsymbol{x} \in \{0,1\}^{n_j} : \sum_{i \in P_j} x_i = p_j\}$ and $\mathcal{U}_j = \{\boldsymbol{c} \in \mathbb{R}^{n_j} : c_i = \underline{c}_i + \delta_i,\ \delta_i \in [0, d_i]\ \ \forall\, i \in P_j,\ \sum_{i \in P_j} \delta_i \leq \Gamma_j\}$.

For every $j \in [K]$ and every $p_j \in \{0, 1, \ldots, n_j\}$, the value $f_j(p_j)$ is the solution of a robust selection problem with continuous budgeted uncertainty set. In Theorem 8 we explain how the whole table of these values can be precomputed efficiently.

The robust selection problem with locally budgeted uncertainty thus becomes

$$\min \sum_{j \in [K]} f_j(p_j) \tag{14}$$

$$\text{s.t.} \sum_{j \in [K]} p_j = p \tag{15}$$

$$p_j \in \mathbb{N}_0 \qquad \forall\, j \in [K], \tag{16}$$

where we use $f_j(p_j) = \infty$ if $p_j > n_j$

▶ **Theorem 8.** *The robust selection problem with locally budgeted uncertainty with arbitrary number of regions K can be solved in $O(n \log n + pn)$ time, hence in polynomial time.*

Proof. First we explain how to compute the complete table of values $f_j(p_j)$ for all values $j = 1, \ldots, K$ and $p_j = 0, \ldots, p$. Note that for fixed j and p_j computing $f_j(p_j)$ is equivalent to solving the robust selection problem with continuous budgeted uncertainty. Observe that this problem is equivalent to solving

$$\min_{\boldsymbol{x} \in \mathcal{X}_j} \left(\underline{\boldsymbol{c}}^t \boldsymbol{x} + \min\{\boldsymbol{d}^t \boldsymbol{x}, \Gamma_j\} \right),$$

corresponding to the two cases of $\pi_j = 0$ and $\pi_j = 1$ in formulation (6–10). The optimal solution to this problem can be determined by solving the two instances of the selection problem for parameter p_j with costs $\underline{\boldsymbol{c}}$ and $(\underline{\boldsymbol{c}} + \boldsymbol{d})$ and taking the solution giving the smaller objective value. Hence, for fixed j, all values of f_j can be calculated by sorting the items in P_j once with respect to costs $\underline{\boldsymbol{c}}$, and once with respect to costs $\underline{\boldsymbol{c}} + \boldsymbol{d}$. In total, this requires time $O(\sum_{j \in [K]} n_j \log n_j) = O(\sum_{j \in [K]} n_j \log n) = O(n \log n)$.

We now give a dynamic program solving problems of type (14–16) in general form, based on a given table for the values of values $f_j(p_j)$. This then directly implies our result for the robust selection problem. Let $T(K', p')$

be defined as

$$
\begin{aligned}
T(K',p') := \min \ & \sum_{j\in[K']} f_j(p_j) \\
\text{s.t.} \ & \sum_{j\in[K']} p_j = p' \\
& p_j \in \mathbb{N}_0 \qquad \forall\, j \in [K'].
\end{aligned}
$$

Then problem (14–16) is equivalent to computing $T(K,p)$. It holds that $T(1,p') = f_1(p')$ for all $p' = 0,1,\ldots,p$. It also holds that

$$T(K',p') = \min\left\{T(K'-1,p'-p_{K'}) + f_{K'}(p_{K'}) \colon p_{K'} = 0,1,\ldots,\min\{p',n_{K'}\}\right\}.$$

Hence, calculating entry $T(K',p')$ can be done in $O(n_{K'})$ time, if all preceding entries have already been calculated. In total, this means that $T(K,p)$ can be calculated in $O(\sum_{j\in[K]}\sum_{p'\in[p]} n_j) = O(\sum_{j\in[K]} pn_j) = O(pn)$ time. Hence in total the running time of our algorithm is $O(n\log n + pn)$. ◀

2.3.3 The Knapsack Problem

We now extend the ideas from Section 2.3.2 to the min-knapsack problem, where

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \{0,1\}^n : \sum_{i\in[n]} w_i x_i \geq W \right\}.$$

Let us assume all item weights to be integer. As before with $f_j(p_j)$, we define $g_j(W_j)$ as the optimum objective value on region $j \in [K]$ if items with a total weight at least W_j are packed. For a given region j and budget W_j, we can calculate $g_j(W_j)$ in $O(nW_j)$ using a standard dynamic program for knapsack twice (once for each cost function $\boldsymbol{c}^t\boldsymbol{x} + \Gamma_j$ and $(\boldsymbol{c}+\boldsymbol{d})^t\boldsymbol{x}$). As the dynamic program gives all values for all weights simultaneously, we can calculate all values $g_j(0),\ldots,g_j(W)$ in two runs of the algorithm, which results in a time complexity of $O(nW)$.

The robust min-knapsack problem with locally budgeted uncertainty then becomes

$$
\begin{aligned}
\min \ & \sum_{j\in[K]} g_j(W_j) && (17)\\
\text{s.t.} \ & \sum_{j\in[K]} W_j = W && (18)\\
& W_j \in \mathbb{N}_0 \qquad \forall\, j \in [K]. && (19)
\end{aligned}
$$

▶ **Theorem 9.** *The robust min-knapsack problem with locally budgeted uncertainty can be solved in $O(Wn+W^2K)$ time. It is therefore weakly NP-hard, as already the nominal knapsack problem is NP-hard.*

Proof. We use a similar dynamic program as in Section 2.3.2. Let $T(K',W')$ be the optimal objective value of problem (17–19) using only the first K' regions and a budget of at most W'. In every iteration of the dynamic program, we calculate

$$T(K',W') = \min\{T(K'-1,W'-W_{K'}) + g_{K'}(W_{K'}) : W_{K'} = 0,\ldots,W'\}.$$

To calculate one such value, we hence need $O(W)$ steps, which results in an overall time complexity of $O(W^2K)$. In combination with the time required to precompute the values $g_j(W_j)$, the claim follows. ◀

If we replace the lower bound on the total weight with an upper bound in the definition of $\mathcal{X}$ and the objective is to maximize profits $\sum_{i\in[n]} p_i x_i$ with a profit vector $\boldsymbol{p}$ affected by locally budgeted uncertainty (note that deviations are downwards in this case), we obtain the classic max-knapsack problem. It is straightforward to see that the dynamic program stated above also solves this variant, since for exact computation max-knapsack and min-knapsack are equivalent.

This is not the case if we consider approximation algorithms (see e.g. [20]). Already for the classic budgeted uncertainty ($K = 1$) the robust max-knapsack problem cannot be approximated, unless $P = NP$. This can be seen as follows. Given an arbitrary instance $(\boldsymbol{p}', \boldsymbol{w}', W')$ of the max-knapsack problem and let $P \in \mathbb{N}$, we construct an instance of the robust max-knapsack problem by setting $\boldsymbol{p} = \boldsymbol{d} = \boldsymbol{p}'$, $\boldsymbol{w} = \boldsymbol{w}'$, $W = W'$ and $\Gamma = P - 1$. Then the original max-knapsack problem has a solution with profit $\geq P$ iff the robust variant has a solution with objective value > 0.

For the robust min-knapsack problem, on the other hand, we obtain an FPTAS even for an arbitrary number of regions K part of the input. This is explained in the following.

The first ingredient is an FPTAS for the classic min-knapsack problem with parametric weight bound W, i.e. an algorithm computing a table $\widetilde{g}(\widetilde{W})$ for polynomially many values $\widetilde{W}$ such that for each W there exists a $\widetilde{W} \geq W$ with a corresponding solution such that $\widetilde{g}(\widetilde{W})$ is an $(1+\epsilon)$ approximation to the optimal solution. Note that this table $\widetilde{g}$ can be extended into a function $\widetilde{g}(W)$ for arbitrary values W by setting $\widetilde{g}(W) = \widetilde{g}(\widetilde{W})$ for the smallest existing weight $\widetilde{W}$ such that $\widetilde{W} \geq W$.

Such an FPTAS can be obtained by modifying the ideas in [13] for the multi-objective min-knapsack problem. The main idea is to partition the cost space into $u+2$ intervals $\{0\}, \{1\}, (1, (1+\epsilon)^{1/n}], \ldots, ((1+\epsilon)^{(u-1)/n}, (1+\epsilon)^{u/n}]$ with $u = \lceil n \log_{1+\epsilon} UB \rceil$ for an upper bound UB. Then we run a classic dynamic programming scheme that finds the maximum weight $W(v)$ for each possible cost value v, but instead of considering all integer costs v we consider only the set $\widetilde{V}$ consisting of the upper endpoints of these intervals as possible cost values and let $\widetilde{W}(v)$ be the corresponding dynamic programming function. Using similar arguments as in [13] we obtain that for every possible entry $W(v)$ there exists a $\widetilde{v} \in \widetilde{V}$ such that $\widetilde{W}(\widetilde{v}) \geq W(v)$ and $\widetilde{v} \leq (1+\epsilon)v$.

Using the table $\widetilde{W}(v)$ we can compute $\widetilde{g}(W)$ efficiently. Given W, let $v = g(W) = g(W(v))$. It follows that $W(v) \geq W$. For $W(v)$ there exists a $\widetilde{W}(\widetilde{v}) \geq W(v)$ such that $\widetilde{v} \leq (1+\epsilon)v$. We assign $\widetilde{g}(W) := \widetilde{v}$. Then it holds that

$$\widetilde{g}(W) \leq \widetilde{g}(\widetilde{W}(\widetilde{v})) = \widetilde{v} \leq (1+\epsilon)v = (1+\epsilon)g(W).$$

In summary we obtain the following lemma.

▶ **Lemma 10.** *There exists an FPTAS for the parametric min-knapsack problem with a weight table $\widetilde{W}$ of polynomial size, i.e. an algorithm computing a table $\widetilde{g}(\widetilde{W})$ for polynomially many values $\widetilde{W}$ such that for each W there exists a $\widetilde{W} \geq W$ with a corresponding solution such that $\widetilde{g}(\widetilde{W}) \leq (1+\epsilon)g(W)$.*

The second ingredient is a known FPTAS for the min-multiple choice knapsack problem [15]. The basis idea is that each region $j \in [K]$ is seen as an item, and for each item there are polynomially many variants of which one must be chosen, depending on how much weight we wish to invest into the region.

We are now able to prove the following result.

▶ **Theorem 11.** *There exists an FPTAS for the robust min-knapsack problem with locally budgeted uncertainty and arbitrary number of regions K.*

Proof. Let $\widetilde{g}_j^{\underline{\boldsymbol{c}}}(W)$ and $\widetilde{g}_j^{\overline{\boldsymbol{c}}}(W)$ be the functions obtained by running the parametric FPTAS for the weight-parametric min-knapsack instances $g_j^{\underline{\boldsymbol{c}}}(W)$ and $g_j^{\overline{\boldsymbol{c}}}(W)$ given by region j with cost functions $\underline{\boldsymbol{c}}$ and $\overline{\boldsymbol{c}} = \underline{\boldsymbol{c}} + \boldsymbol{d}$, giving $(1+\epsilon_1)$-approximations for arbitrary chosen $\epsilon_1 > 0$. Based on those instances we define functions $g_j(W) = \min\{g_j^{\overline{\boldsymbol{c}}}(W), g_j^{\underline{\boldsymbol{c}}}(W) + \Gamma_j\}$ and $\widetilde{g}_j(W) = \min\{\widetilde{g}_j^{\overline{\boldsymbol{c}}}(W), \widetilde{g}_j^{\underline{\boldsymbol{c}}}(W) + \Gamma_j\}$. Note that

$$\widetilde{g}_j(W) \leq \min\{(1+\epsilon_1)g_j^{\overline{\boldsymbol{c}}}(W), (1+\epsilon_1)g_j^{\underline{\boldsymbol{c}}}(W) + \Gamma_j\} \leq (1+\epsilon_1)\min\{g_j^{\overline{\boldsymbol{c}}}(W), g_j^{\underline{\boldsymbol{c}}}(W) + \Gamma_j\} = (1+\epsilon_1)g_j(W).$$

Then the robust min-knapsack problem is equivalent to

$$\min \sum_{j \in [K]} g_j(W_j) \tag{20}$$

$$\text{s.t.} \sum_{j \in [K]} W_j \geq W \tag{21}$$

$$W_j \geq 0 \qquad \forall\, j \in [K]. \tag{22}$$

If we replace g_j by its approximation $\widetilde{g}_j$, this is an instance of min-multiple choice knapsack for which an FPTAS exists [15]. More precisely, from the FPTAS there are polynomially many values $\widetilde{W}_j^1, \ldots, \widetilde{W}_j^{r(j)}$ for each $j \in [K]$

that define the weight table, including the zero weight. The problem we consider is thus

$$\min \sum_{j\in[K]} \sum_{i\in[r(j)]} \tilde{g}_j(\widetilde{W}_j^i) x_{ji} \tag{23}$$

$$\text{s.t.} \sum_{i\in[r(j)]} x_{ji} = 1 \qquad \forall\, j \in [K] \tag{24}$$

$$\sum_{j\in[K]} \sum_{i\in[r(j)]} \widetilde{W}_j^i x_{ji} \geq W \tag{25}$$

$$x_{ji} \in \{0,1\} \qquad \forall\, j \in [K], i \in [r(j)]. \tag{26}$$

Let $\boldsymbol{x}^*$ be an optimal solution to this problem, and let $\widetilde{W}_j^* = \sum_{i\in[r(j)]} \widetilde{W}_j^i x_{ji}^*$ be the corresponding optimal choice of weights per region j. Let $\widetilde{W}_j$ for $j \in [K]$ be the solution obtained by running the FPTAS to obtain a $(1+\epsilon_2)$ approximation for this instance of the min-multiple choice knapsack problem and $\text{ALG} = \sum_{j\in[K]} g_j(\widetilde{W}_j)$ the corresponding objective value. Let W_j^* for $j \in [K]$ be the optimal solution for the original problem (20–22).

Then, by combining the approximation guarantees, we get

$$\begin{aligned}
\text{ALG} &= \sum_{j\in[K]} g_j(\widetilde{W}_j) \leq \sum_{j\in[K]} \tilde{g}_j(\widetilde{W}_j) \leq (1+\epsilon_2) \sum_{j\in[K]} \tilde{g}_j(\widetilde{W}_j^*) \\
&\leq (1+\epsilon_2) \sum_{j\in[K]} \tilde{g}_j(W_j^*) \leq (1+\epsilon_2) \sum_{j\in[K]} (1+\epsilon_1) g_j(W_j^*) \\
&= (1+\epsilon_1)(1+\epsilon_2)\text{OPT},
\end{aligned}$$

where OPT is the objective value of an optimal solution to our instance of the robust min-knapsack problem. ◀

2.4 A Variant Based on Different Types of Budgeted Uncertainty Sets

Different types of classic budgeted uncertainty sets have been considered in the literature. Instead of the definition used in this paper, where

$$\mathcal{U} = \left\{ \boldsymbol{c} = \underline{\boldsymbol{c}} + \boldsymbol{\delta} : \delta_i \in [0, d_i] \;\; \forall\, i \in [n],\; \sum_{i\in[n]} \delta_i \leq \Gamma \right\},$$

it is possible to consider the variant

$$\mathcal{U} = \left\{ \boldsymbol{c} : c_i = \underline{c}_i + d_i z_i, z_i \in [0,1] \;\; \forall\, i \in [n],\; \sum_{i\in[n]} z_i \leq \Gamma \right\},$$

which results in the following variant of locally budgeted uncertainty sets

$$\mathcal{U} = \left\{ \boldsymbol{c} : c_i = \underline{c}_i + d_i z_i, z_i \in [0,1] \;\; \forall\, i \in [n],\; \sum_{i\in P_j} z_i \leq \Gamma_j \;\; \forall\, j \in [K] \right\}.$$

Using the same techniques as applied in Section 2.1 we obtain the following compact MIP formulation for this variant of the problem:

$$\min \sum_{j\in[K]} \left(\Gamma_j \pi_j + \sum_{i\in P_j} \rho_i + \sum_{i\in P_j} \underline{c}_i x_i \right) \tag{27}$$

$$\text{s.t. } \pi_j + \rho_i \geq d_i x_i \qquad \forall\, j \in [K], i \in P_j \tag{28}$$

$$\pi_j \geq 0 \qquad \forall\, j \in [K] \tag{29}$$

$$\rho_i \geq 0 \qquad \forall\, i \in [n] \tag{30}$$

$$\boldsymbol{x} \in \mathcal{X}. \tag{31}$$

Based on this formulation, the theoretical results obtained can be extended to this case. In Theorem 2, this means we need to solve $O(n^K)$ instead of $O(2^K)$ subproblems. The hardness results from Section 2.3.1 hold as well, as we used $d_i = 1$ in the uncertainty sets we constructed for the reductions, in which case both variants of uncertainty coincide. The algorithms presented in Sections 2.3.2 and 2.3.3 can be adapted as well, as they are based on decomposing the robust problem into subproblems that remain tractable (instead of solving two subproblems, we now need to solve $O(n)$ of them).

3 Experiments

3.1 Overview

We first present an experiment related to computation times when solving robust problems with locally budgeted uncertainty. We then present three more experiments to quantify differences between ”classic” budgeted uncertainty sets and the locally budgeted uncertainty sets proposed in this paper. Experiments 1 to 3 use randomly generated data for the uncertain selection problem, while Experiment 4 is based on real-world data. In Experiment 2, we assume that the uncertainty set is locally budgeted, and consider the benefit of using this information instead of using a classic budgeted set. In the next experiment, the actual regions are not known to the imagined decision maker. Instead, only sampled scenarios are provided. We analyze the differences between solutions based on classic and locally budgeted uncertainty sets fitted to the data. Finally, in Experiment 4, we consider the differences between solutions based on classic and locally budgeted uncertainty sets fitted to real-world data for shortest path problems, where nothing is known about the underlying distribution.

3.2 Experiment 1

3.2.1 Setup

We compare the computation times when solving robust selection problems using a general MIP solver for the mixed-integer programming formulation (6–10) with the polynomial-time dynamic program from Section 2.3.2. Recall that formulation (6–10) has $2n + K$ variables and $n + 1$ constraints, while the dynamic program takes $O(n \log n + pn)$ time.

We create 100 instances for each $n \in \{2^8, 2^9 \ldots, 2^{15}\}$. Each instance with n items is solved for each $K \in \{2^0, 2^1, \ldots, 2^{\log(n)-1}\}$. Item costs $\underline{c}_i$ and d_i are chosen independently and uniformly from $\{10, 11, \ldots, 49\}$. We further set $p = n/2$ and $\Gamma_j = 10|P_j|$, where regions are of equal size, i.e., $|P_j| = n/K$.

Instances are solved using the dynamic program (DP), which has been implemented in C++, and by the mixed-integer programming formulation (IP), for which CPLEX version 12.8. was used. In formulation (6–10) we forced variables π_j to be binary, as the use of continuous variables resulted in significantly higher computation times for $K = 2^0$.

Experiments were conducted on a virtual server with Intel Xeon Gold 5220 CPU running at 2.20GHz. Of ten available threads, every solution process was allowed to use only one.

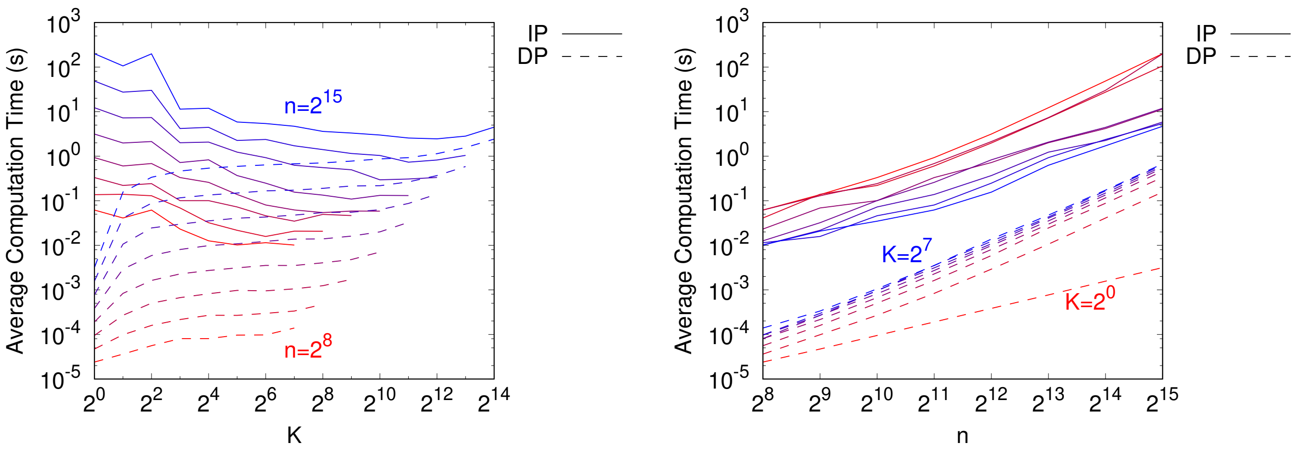
3.2.2 Results

We present average computation times over the 100 instances of each type in Figure 2.

In Figure 2a, each line represents a fixed value of n for varying values of K. As can be expected, the largest problems ($n = 2^{15}$) have the highest computation times. The solid lines indicate computation times when using IP, while the dashed lines give the computation times when using DP. Note that higher values of n mean that also K can be increased to a higher value, which gives lines of different lengths.

It can be seen that computation times increase in K for DP, which may be surprising from our $O(n \log n + pn)$ complexity estimate, which does not contain K. But K has an impact on the size of the entry table $T(K', p')$, which becomes dominated by $O(np)$ in our estimate. Furthermore, a larger value of K tends to decrease the computation times for IP, with the hardest problems being those that have $K = 2^0$. This means that solving robust problems with locally budgeted uncertainty is even easier than solving the classic budgeted uncertainty variant when using IP for the selection instances considered here. This trend does not continue for large values of K, as can be seen, e.g., for $n = 2^{15}$ in the top right of the plot. Overall, computation times using DP are several orders of magnitude smaller for smaller values of n or K than when using IP, and also remain smaller for larger values of n or K.

In Figure 2b, we show the same computation times, but with n on the horizontal axis. Now every line corresponds to a choice of K, where we omit values for $K > 2^7$ for better legibility. Note that computation



(a) Computation times for varying K.

(b) Computation times for varying n.

Figure 2 Experiment 1: Computation times.

times for DP are almost linear in the log-log-plot, which indicates monomial growth (in our choice $p = n/2$, the runtime complexity becomes $O(n^2)$). All lines are almost parallel, which means that they have a similar power term. Only for $K = 2^0$ we can observe a different slope. In this case, the complexity of DP becomes $O(n \log n)$, as we only need to sort items twice. Furthermore, it is possible to observe the same trend from Figure 2a regarding K, where larger values of K mean that DP tends to increase and IP tends to decrease. The asymptotic trend of IP in n cannot be reliably estimated from Figure 2b, but explorative experiments indicate that it does not remain linear in n, and computation times of DP remain below those of IP.

3.3 Experiment 2

3.3.1 Setup

In this experiment, we focus on randomly generated selection problems. We fix $n = 30$. Given the number of regions K, we distribute items into the K regions as uniformly as possible. For every item, we generate $\underline{c}_i$ and d_i independently and uniformly from $\{10, \dots, 49\}$. We set $\Gamma_j = 10|P_j|$ and use $K = 2, 3, 4, 5$. We generate 10,000 instances using the same random seed for each K (i.e., cost coefficients of these instances are the same for each K). We consider all values $p = 1, \dots, 29$.

Each instance is solved exactly, using the compact formulation for locally budgeted uncertainty. Additionally, we solve each instance using the classic budgeted uncertainty approach, by ignoring the partition into regions and using $\Gamma = \sum_{j\in[K]} \Gamma_j = 10n$. We measure the robust objective value of both solutions with respect to the locally budgeted uncertainty set.

By this setup, we already know that the approach using the locally budgeted uncertainty set must lead to better objective values, as it takes into account the actual uncertainty set that is used to evaluate a solution. The question we answer here is how much we lose by ignoring such local information. As discussed in Section 1, local uncertainty naturally arises in some practical applications. Our experiment simulates the effect of using classic budgeted uncertainty in this case.

3.3.2 Results

In Figure 3 we show the ratio of average objective values between the solution found by the model using classic budgeted uncertainty, and by the model using locally budgeted uncertainty, for different values of p. The higher the ratio, the higher are the additional costs that arise by ignoring the locally budgeted uncertainty structure. The data in Figure 3 is not distorted by outliers.

Note that for small ($p = 1$) and large ($p = 29$) values of p, the ratio is close to one. Hence, the local information does not matter in this setting. The best choice is to buy the one item i where $\underline{c}_i + d_i$ is smallest (or to avoid the one item where this value is largest, respectively). For values of p between these two extremes, solutions differ. The region of values for p where there is a difference between solutions based on classic and locally budgeted uncertainty increases with K. For $K = 2$ and $p = 11$, the average cost difference is 15.6%, while this increases to 17.9% for $K = 5$ and $p = 10$.

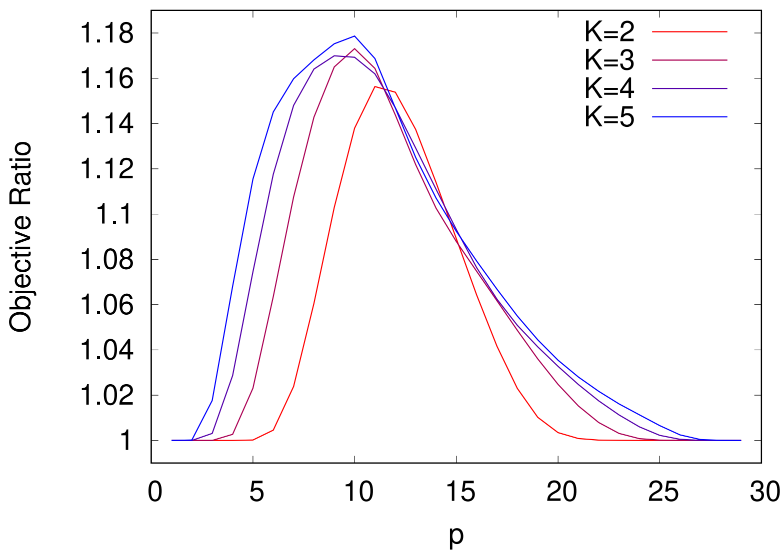


Figure 3 Experiment 2: Ratio of average objective values of solutions based on classic and locally and budgeted uncertainty.

3.4 Experiment 3

3.4.1 Setup

In the previous experiment we considered the effect if the decision maker knows the parameters of a locally budgeted uncertainty set, but chooses to ignore these and use a classic budgeted uncertainty instead. In practice, an uncertainty set is usually not given, but needs to be derived from data.

Hence in this second experiment, we build locally budgeted uncertainty sets in the same way as before, but then sample N scenarios from the set. To create a sample scenario $\boldsymbol{c}^k$ with $k \in [N]$, we choose a random value γ_j from $\{0, \ldots, \Gamma_j\}$ uniformly and distribute $\min\{\gamma_j, \sum_{i \in P_j} d_i\}$ many unit cost increases to all items $i \in P_j$. We do so iteratively, i.e., we first begin with $\boldsymbol{c}^k = \underline{\boldsymbol{c}}$, and then repeatedly choose an item from P_j where $c_i^k \leq \underline{c}_i + d_i - 1$ at random, and increase this item's costs by one.

Having constructed N scenarios, we then fit suitable classic and locally budgeted uncertainty sets. The focus of this experiment is to derive the regions from the data. We therefore assume that $\underline{c}_i$ and d_i are given for each item.

To estimate the underlying partition into regions, we can assume that in a sufficiently large sample N, two items from different regions are not correlated. As each budget constraint applies locally, correlation can only be found within regions. Based on this idea, we calculate the correlation matrix using the available sample data. We then consider two items to be connected, if the absolute value of correlation is above a certain threshold (in this experiment, we used 0.3). Each connected component then forms its own region. Note that this way, the number of regions $\widetilde{K}$ we use is not prescribed, but estimated from the data.

The classic budgeted uncertainty set uses $\widetilde{\Gamma} = \max_{k \in [N]} \left(\sum_{i \in [n]} c_i^k - \underline{c}_i \right)$ as an estimate for the uncertainty budget. For each region, we estimate $\widetilde{\Gamma}_j$ in the same way.

We use $n = 30$, $p = 10$, $K = 2, \ldots, 5$ and vary the sample size N from 10 to 10,000. As before, for each parameter combination, we construct and solve 10,000 instances.

3.4.2 Results

Our results are summarized in Figure 4. On the horizontal axis, we denote the sample size N (note the logarithmic scale). On the vertical axis, we show average objective values with respect to the original, unknown locally budgeted uncertainty set.

The solid lines indicate the objective value of the solutions based on fitted classic budgeted uncertainty sets, while the dashed lines represent locally budgeted uncertainty sets. The dotted lines indicate the optimal objective value, if the actual uncertainty set were known.

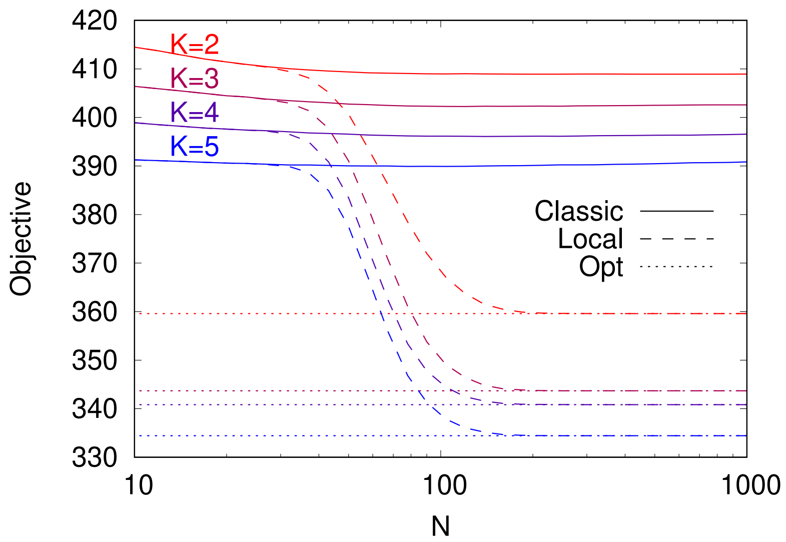


Figure 4 Experiment 3: Average objective values of solutions based on locally and classic budgeted uncertainty.

First note that the larger the number of regions K, the smaller become objective values overall. For the classic budgeted uncertainty set, the decrease in objective value with increasing sample size N is small, the line is mostly horizontal. This is different for the locally budgeted uncertainty set, where a significant decrease can be observed after the sample size reaches a certain threshold. This begins at around $N = 30$, and is completed at approximately $N = 110$. We find that even if the locally budgeted uncertainty set is not given explicitly, it is possible to take significant advantage of this model by identifying the corresponding structure in the data.

3.5 Experiment 4

3.5.1 Setup

While the previous experiments used artificial data that is based on an underlying locally budgeted uncertainty set, we now consider real-world data, where no such underlying structure is known. The data we use was first introduced in [9]. It consists of a graph modeling the city of Chicago with 538 nodes and 1308 edges, and 4363 snapshots of traffic speed for each edge over 46 days. Figure 5 shows the structure of the graph.

The data is prepared in the same way as in [9]. We use each traffic speed snapshot as a scenario. Of the 4363 scenarios, we use 75% for training our models, and 25% for evaluation. We sample 200 random s-t pairs and calculate a shortest path for each pair using each of our models.

The classic budgeted model is trained on the data in the same way as in Experiment 2 (see Section 3.4.1), where $\underline{c}_i$ and d_i are estimated from the data. To model locally budgeted uncertainty sets, we create regions by using edge sequences between any two crossings in one direction. In Figure 5, we show three such regions in red as an illustration. In total, this results in 546 regions.

We control the degree of conservatism of our two approaches by multiplying the estimated $\widetilde{\Gamma}$ value (or $\widetilde{\Gamma}_j$ values, respectively) with a budget factor f. We use all values of f from 0 to 0.5 in step size 0.002.

For each value of f and each model, we solve the 200 shortest path problems and evaluate the path choices in-sample and out-of-sample. We then calculate the average of the average path length and the worst-case path length over the two scenario sets.

3.5.2 Results

We show our in-sample results in Figure 6a and the out-of-sample results in Figure 6b. On the horizontal axis is the average travel time (in minutes), and the vertical axis is the average worst-case travel time. The results of both models are shown as a line, starting with $f = 0$ in the top left, and moving to the right with increasing value of f.

Note that for $f = 0$, the classic and the locally budgeted approach result in the same solution, that is, they only optimize for best-case travel times. In an ideal trade-off between average and worst-case travel time, we

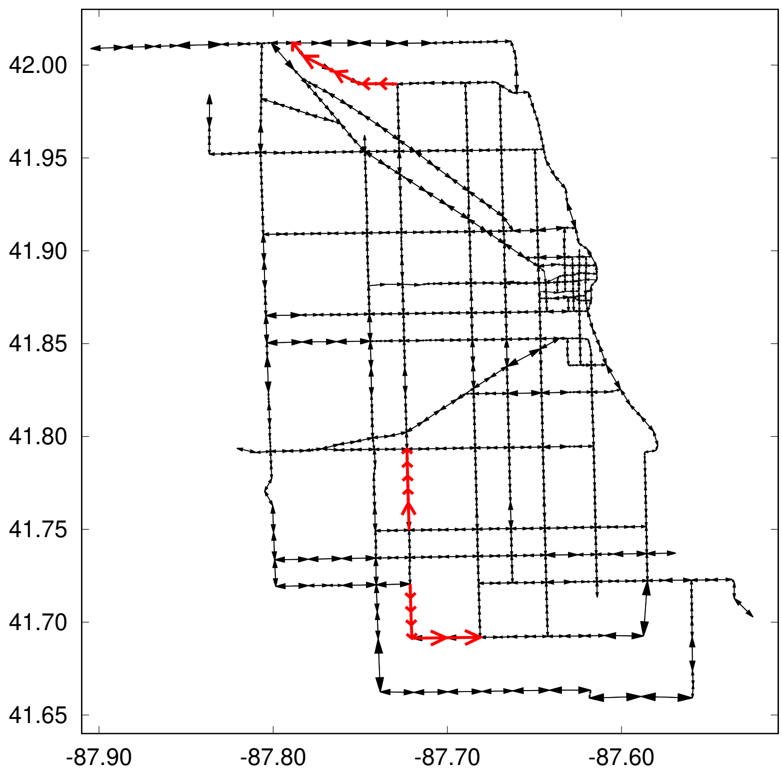
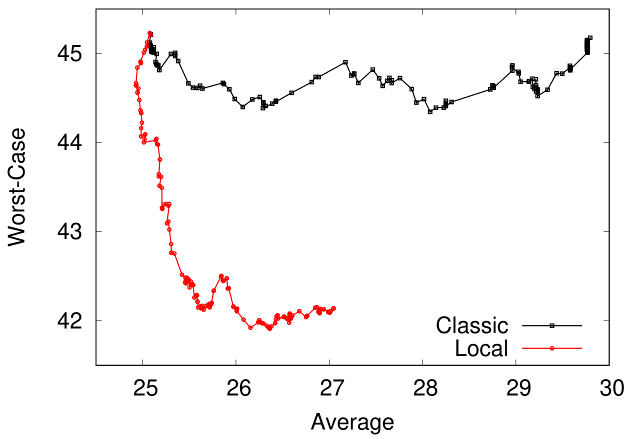
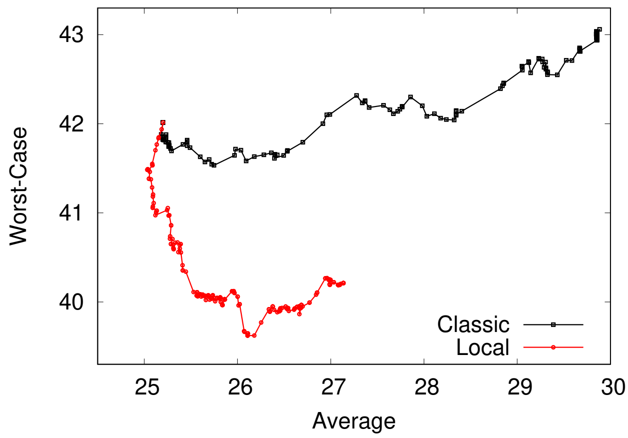


Figure 5 Experiment 4: Chicago graph with three regions highlighted.



(a) In-sample results.



(b) Out-of-sample results.

Figure 6 Experiment 4: Average versus average worst-case objective value for different budget factors.

would expect the lines to reach from the top left corner (low average time, high worst-case time) to the bottom right corner (high average time, low worst-case time).

In Figure 6a we can see that for the classic approach, no such trade-off can be reached. With increasing budget factor, we increase the average travel time, but do not decrease the worst-case travel times. From the perspective of Pareto optimality, most of the budget factors result in dominated solutions. The locally budgeted uncertainty set, on the other hand, gives a trade-off with increasing budget factor and considerably outperforms the solutions found by the locally budgeted uncertainty set. In the out-of-sample results (Figure 6b), the classic approach performs even worse, with the curve leading upwards to the top right. The locally budgeted solutions retain a trade-off between average and worst-case time.

Overall, we see that it is possible to model the discrete real-world scenarios more accurately using the locally budgeted uncertainty approach, while with the classic budgeted approach, it is not possible to capture the underlying data.

4 Conclusions and Further Research

In this paper we studied a generalization of budgeted uncertainty sets, where there is a separate uncertainty budget for different regions of items. We showed that for a constant number of regions K, the robust counterpart remains polynomially solvable if the nominal problem is solvable in polynomial time. For unbounded values of K, we show that the robust selection problem can still be solved in polynomial time, while this is not the case for the representative selection problem, even if only one item is chosen from each partition. This extends to other combinatorial problems that include the representative selection problem as a special case. Table 1 gives an overview to these results. In addition, we show that no parameterized algorithms with running time in $O^*(2^{o(K)})$ exist. To the best of our knowledge, robust optimization problems have not been considered from the perspective of fixed parameter tractability so far. We also presented an FPTAS for the robust min-knapsack problem (the max-knapsack problem is not approximable even for classic budgeted uncertainty).

Table 1 Overview of complexity results from this paper.

Problem	$K = O(1)$	$K = O(n)$
Unconstrained	P	P
Selection	P	P
Min/Max-Knapsack	weakly NPH	weakly NPH
Representative Selection	P	strongly NPH
Spanning Tree	P	strongly NPH
s-t-min-cut	P	strongly NPH
Shortest Path	P	strongly NPH

In computational experiments we showed that solving robust selection problems with locally budgeted uncertainty is even easier than with classic budgeted uncertainty when using the standard MIP formulation. The dynamic program introduced in this paper outperforms the MIP solver, especially for small numbers of regions, where the dynamic program becomes faster by multiple orders of magnitude. In experiments concerning the quality of solutions, we found that the added flexibility in the modeling of uncertainty allows for better-performing solutions both on random and on real-world data sets.

Furthermore, the locally budgeted uncertainty set proposed in this paper can be seen in the light of data-driven robust optimization (see, e.g., [4, 9]), where the aim is to find the most suitable uncertainty set to describe given data. Using local budgets extends the capabilities of classic budgeted uncertainty models, and thus gives more degrees of freedom to describe the data.

From a theoretical perspective, further investigations into the parameterized running time of our meta-algorithm obtained in Theorem 2 are of interest. Note that a minor modification of the proof of Theorem 6 implies that no $O^*((\sqrt{2}-\epsilon)^K)$ algorithm for the robust representative selection problem with locally budgeted uncertainty exists, unless the strong exponential time hypothesis (SETH) fails. We conjecture that there are combinatorial optimization problems for which the constant 2 can be improved. Whether this can be done in a meta-algorithm or only for specific combinatorial optimization problems is another interesting open problem.

Also for the second variant of locally budgeted uncertainty as mentioned in Section 2.4, interesting open questions remain. It is not known whether a fixed-parameter tractable algorithm exists for this case, as the decomposition proposed here results in $O(n^K)$ many subproblems. Also, it is unknown if this slight change in the definition of the uncertainty set leads to W[1]-hardness for the problems which were previously NP-hard in Table 1. It is also possible to study further variants based on this model, for instance by dropping the condition that the regions P_j must form a partition. Our positive results cannot be directly translated into this setting, since now there are arbitrary dependencies between the regions and our methods strongly rely on the fact that the uncertainty of one region is independent of the uncertainty of another.

A Additional Proofs

Proof of Theorem 6. We reduce from an instance of the well known 3-SAT problem.
Given: A formula φ in 3-CNF with variables $x_1, \ldots, x_{\tilde{n}}$ and $\widetilde{m}$ clauses, i.e.

$$\varphi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge \cdots \wedge (l_{\tilde{m},1} \vee l_{\tilde{m},2} \vee l_{\tilde{m},3}), \qquad \text{where the } l \text{ are literals of the variables } \boldsymbol{x}.$$

$$\varphi = (\bar{x}_1 \lor x_2 \lor \bar{x}_4) \land (x_2 \lor \bar{x}_3 \lor x_4) \land (x_1 \lor \bar{x}_2 \lor x_3)$$
$$x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1$$

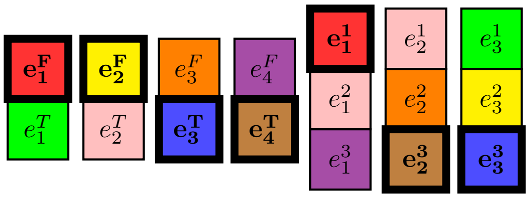


Figure 7 Illustration of the reduction from 3-SAT. Above there is an example of a 3-SAT formula with a feasible assignment. Below we show the instance of the robust representative selection problem corresponding to this instance. Each column corresponds to a partition from which one of the two elements must be selected. The colors correspond to the regions of the instance. The bold elements are an optimal solution corresponding to the shown variable assignment. Note that elements in exactly 4 regions (red, yellow, blue, brown), corresponding to the shown feasible variable assignment, are selected.

Question: Is there an assignment for $\boldsymbol{x}$ such that φ is true?

The exponential time hypothesis (ETH) implies that there does not exist an algorithm to decide 3-SAT with running time $2^{o(n)}$, and is widely believed [25]. We define an instance of our robust problem on the element set $[2\widetilde{n} + 3\widetilde{m}]$ in the following way. In Figure 7 we illustrate the reduction via an example for a 3-SAT instance. The partition of $[2\widetilde{n} + 3\widetilde{m}]$ for the representative selection problem consists of $\widetilde{n} + \widetilde{m}$ parts, one for each variable and clause. For each $i \in [\widetilde{n}]$ the set T_i consists of two elements, an element e_i^T and an element e_i^F. These parts are the variable gadgets and selecting e_i^T or e_i^F corresponds to setting x_i to true or false respectively. For each clause we create a part consisting of exactly three elements, i.e. for each $j \in [\widetilde{m}]$ the set $T_{\tilde{n}+j}$ consists of the elements e_j^1, e_j^2 and e_j^3.

Using the locally budgeted uncertainty set and cost structure we will enforce that e_j^i can only be selected without inducing additional cost, if the selection in the variable gadget corresponding to the variable of literal $l_{j,i}$ corresponds to $l_{j,i}$ being true. To this aim, we define the partition for the locally budgeted uncertainty set consisting of $K = 2\widetilde{n}$ regions P_i^T and P_i^F for each $i \in [\widetilde{n}]$. Formally, for each $i \in [\widetilde{n}]$ we set $P_i^T := \{e_i^T\} \cup \{e_j^\ell \colon l_{j,\ell} \text{ is true if } x_i \text{ is true}\}$ and $P_i^F := \{e_i^F\} \cup \{e_j^\ell \colon l_{j,\ell} \text{ is true if } x_i \text{ is false}\}$. Selecting an element inside region P_i^T or P_i^F corresponds to setting the variable x_i to true or false respectively. We set $\Gamma_i^T = \Gamma_i^F = 1$ for all $i \in [\widetilde{n}]$ and the costs to $\underline{\boldsymbol{c}} = \mathbf{0}$ and $\boldsymbol{d} = \mathbf{1}$. In a similar way as in the proof of Theorem 3, one can prove the following claim.

▷ Claim 12. Given a feasible solution $\boldsymbol{x}$ of our instance of the robust representative selection problem, the robust objective value is equal to the number of regions in which at least one element is selected by $\boldsymbol{x}$.

Based on this, we show that φ is feasible, if and only if the objective value of our instance is $\widetilde{n}$.

Given a feasible assignment for φ we select in each variable gadget the corresponding element. Then, since φ is true, for each clause $j \in [\widetilde{m}]$ there is at least one literal $l_{j,i}$ which is true. We select the corresponding element e_j^i in $T_{\tilde{n}+j}$. Observe that this selection selects elements from exactly $\widetilde{n}$ regions. For each $i \in [\widetilde{n}]$ if x_i is true only elements from P_i^T and none from P_i^F are selected and if x_i is false only elements from P_i^F and none from P_i^T are selected. Hence, by our claim the objective value of this selection is $\widetilde{n}$.

For the other direction first observe that the objective value of our instance cannot be smaller than $\widetilde{n}$, since in every variable gadget one element must be chosen and each element e_i^v has its exclusive region P_i^v. Now assume that there is a selection with robust objective value $\widetilde{n}$. Then by our claim for every $i \in [\widetilde{n}]$ in exactly one of the two regions P_i^T and P_i^F an element of T_i is selected. Hence, the truth assignment to x_i induced by the selection in the variable gadget satisfies all the clauses, since otherwise in one of the clause gadgets we would have to select an element inside an additional region. ◀

Proof of Theorem 7. We use an objective-preserving reduction from the set cover problem.
Given: A ground set V with $|V| = \widetilde{n}$, and a set of subsets $V_s \subseteq V$, $s \in S$ with $|S| = \widetilde{m}$.
Question: Does there exist a set cover of size less or equal to k, i.e., a set $C \subseteq S$ with $|C| \leq k$ such that $\cup_{s \in C} V_s = V$?

We set $L = \widetilde{n}$ and $K = \widetilde{m}$. For each $v \in V$ and each $s \in S$ with $v \in V_s$, we add an element (v, s) to set T_v. Each such element (v, s) belongs to a region P_s. We set $\Gamma_s = 1$ for all $s \in S$ and the costs to $\underline{\boldsymbol{c}} = \mathbf{0}$ and $\boldsymbol{d} = \mathbf{1}$. Finally, we set $p_v = 1$ for all sets T_v.

As an example, let us assume we have $\widetilde{n} = 4$, $\widetilde{m} = 3$, and $V_1 = \{1, 2\}$, $V_2 = \{2, 3\}$, $V_3 = \{3, 4\}$. Then Table 2 illustrates the construction.

By choosing one item from each set T_v, we determine by which set V_s we intend to cover it. It can be easily seen that a set cover of size k exists if and only if the robust representative selection problem with locally budgeted uncertainty has an objective value at most k. Hence, the reduction is cost-preserving, and the claim follows. ◀

Table 2 Example construction for the proof of Theorem 7.

	V_1	V_2	V_3	
1	x			T_1
2	x	x		T_2
3		x	x	T_3
4			x	T_4
	P_1	P_2	P_3	

References

1 Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *Eur. J. Oper. Res.*, 197(2):427–438, 2009.

2 Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.

3 Artur Alves Pessoa, Luigi Di Puglia Pugliese, Francesca Guerriero, and Michael Poss. Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66(2):98–111, 2015.

4 Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Math. Program.*, 167(2):235–292, 2018.

5 Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1):49–71, 2003.

6 Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, 2004.

7 Marin Bougeret, Artur Alves Pessoa, and Michael Poss. Robust scheduling with budgeted uncertainty. *Discrete Appl. Math.*, 261:93–107, 2019.

8 Christina Büsing and Fabio D'andreagiovanni. New results about multi-band uncertainty in robust optimization. In *International Symposium on Experimental Algorithms*, pages 63–74. Springer, 2012.

9 André Chassein, Trivikram Dokka, and Marc Goerigk. Algorithms and uncertainty sets for data-driven robust shortest path problems. *Eur. J. Oper. Res.*, 274(2):671–686, 2019.

10 André Chassein, Marc Goerigk, Adam Kasperski, and Paweł Zieliński. On recoverable and two-stage robust selection problems with budgeted uncertainty. *Eur. J. Oper. Res.*, 265(2):423–436, 2018.

11 André Chassein, Marc Goerigk, Jannis Kurtz, and Michael Poss. Faster algorithms for min-max-min robustness for combinatorial problems with budgeted uncertainty. *Eur. J. Oper. Res.*, 279(2):308–319, 2019.

12 Vladimir G. Deineko and Gerhard J. Woeginger. Complexity and in-approximability of a selection problem in robust optimization. *4OR*, 11(3):249–252, 2013.

13 Thomas Erlebach, Hans Kellerer, and Ulrich Pferschy. Approximating multiobjective knapsack problems. *Manage. Sci.*, 48(12):1603–1612, 2002.

14 Michael R. Garey and David S. Johnson. *Computers and intractability*, volume 174 of *A Series of Books in the mathematical Sciences*. W. H. Freeman and Company, 1979.

15 Georgii V. Gens and Eugenii V. Levner. Computational complexity of approximation algorithms for combinatorial problems. In *International Symposium on Mathematical Foundations of Computer Science*, pages 292–300. Springer, 1979.

16 Marc Goerigk, Stefan Lendl, and Lasse Wulf. Recoverable Robust Representatives Selection Problems with Discrete Budgeted Uncertainty. `https://arxiv.org/abs/2008.12727`, 2020.

17 Chrysanthos E. Gounaris, Panagiotis P. Repoussis, Christos D. Tarantilis, Wolfram Wiesemann, and Christodoulos A. Floudas. An adaptive memory programming framework for the robust capacitated vehicle routing problem. *Transportation Science*, 50(4):1239–1260, 2016.

18 Christoph Hansknecht, Alexander Richter, and Sebastian Stiller. Fast robust shortest path computations. In *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

19 Adam Kasperski and Paweł Zieliński. Robust discrete optimization under discrete and interval uncertainty: A survey. In *Robustness analysis in decision aiding, optimization, and analytics*, pages 113–143. Springer, 2016.

20 Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.

21 Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer, 2013.

22 Michael Poss. Robust combinatorial optimization with variable budgeted uncertainty. *4OR*, 11(1):75–92, 2013.

23 Michael Poss. Robust combinatorial optimization with knapsack uncertainty. *Discrete Optimization*, 27:88–102, 2018.

24 Rishi Saket and Maxim Sviridenko. New and improved bounds for the minimum set cover problem. In *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, pages 288–300. Springer, 2012.

25 Gerhard J. Woeginger. Exact algorithms for NP-hard problems: A survey. In *Combinatorial optimization – Eureka, you shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–207. Springer, 2003.