On Scenario Aggregation to Approximate Robust Combinatorial Optimization Problems

André Chassein[1] and Marc Goerigk[2]

[1]Fachbereich Mathematik, University of Kaiserslautern, Germany
[2]Department of Management Science, Lancaster University, United Kingdom

Abstract

As most robust combinatorial min-max and min-max regret problems with discrete uncertainty sets are NP-hard, research in approximation algorithm and approximability bounds has been a fruitful area of recent work. A simple and well-known approximation algorithm is the midpoint method, where one takes the average over all scenarios, and solves a problem of nominal type. Despite its simplicity, this method still gives the best-known bound on a wide range of problems, such as robust shortest path or robust assignment problems.

In this paper, we present a simple extension of the midpoint method based on scenario aggregation, which improves the current best K-approximation result to an (εK)-approximation for any desired $\varepsilon > 0$. Our method can be applied to min-max as well as min-max regret problems.

Keywords: robust combinatorial optimization; approximation algorithms; scenario aggregation; min-max optimization; min-max regret optimization

1 Introduction

We consider uncertain combinatorial optimization problems of the form

$$\min \left\{ c^t x : x \in \mathcal{X} \subseteq \{0,1\}^n \right\} \tag{P}$$

where $\mathcal{X}$ is the set of feasible solutions, and c is an uncertain objective function that comes from some uncertainty set $\mathcal{U}$. Two frequently studied approaches to reformulate such an uncertain problem to a robust counterpart are min-max optimization

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

	Problem	LB	UB	FPTAS for fixed K
Min-Max	Shortest Path	$\Omega(\log^{1-\varepsilon} K)$	K	✓ [KZ07]
	Spanning Tree	$\Omega(\log^{1-\varepsilon} K)$	$\mathcal{O}(\log^2 K)$	✓ [ABV07]
	s-t Cut	$\Omega(\log^{1-\varepsilon} K)$	K	
	Assignment	$\Omega(\log^{1-\varepsilon} K)$	K	
	Selection	$\Omega(1)$	$\mathcal{O}(\frac{\log K}{\log\log K})$	✓ [KZ07]
	Knapsack	$\Omega(1)$	-	✓ [ABV07]
Regret	Shortest Path	$\Omega(\log^{1-\varepsilon} K)$	K	✓ [KZ07]
	Spanning Tree	$\Omega(\log^{1-\varepsilon} K)$	K	✓ [ABV07]
	s-t Cut	$\Omega(\log^{1-\varepsilon} K)$	K	
	Assignment	$\Omega(\log^{1-\varepsilon} K)$	K	
	Selection	$\Omega(1)$	K	✓ [KZ07]
	Knapsack	not approx.	not approx.	

Table 1: Current best known approximation guarantees (UB) for unbounded number of scenarios K, and best known inapproximability results (LB) (see [KZ16]).

and min-max regret optimization

$$\min_{x\in\mathcal{X}} \max_{c\in\mathcal{U}} \left(c^t x - opt(c)\right)$$

where $opt(c) = \min_{y\in\mathcal{X}} c^t y$ is used as an additional normalization term. Both types of problems have received significant attention in the research literature, see, e.g., the surveys [KY97, ABV09, KZ16] on this topic. In this paper we focus on the case of discrete uncertainty, i.e., the uncertainty set is of the form $\mathcal{U} = \{c_1, \dots, c_K\} \subset \mathbb{R}^n$.

For most combinatorial problems where the deterministic version can be solved in polynomial time (e.g., shortest path, spanning tree, selection), both robust counterparts turn out to be NP-hard. Therefore, the approximability of such problems has been analyzed (see, e.g., [ABV07]).

A popular approximation algorithm due to its generality and simplicity is the midpoint method (see, e.g., [CG15]). The idea is to define a new scenario $\hat{c} := \frac{1}{K}\sum_{i\in[K]} c_i$, which is the average of all scenarios in the uncertainty set, and to solve a nominal problem with respect to these costs. This method is known to be a K-approximation algorithm for both min-max and min-max regret optimization. In the case of interval uncertainty, this approach even gives a 2-approximation [KZ06, Con12]. Quite surprisingly, this is still the best known approximation guarantee for several problems, see Table 1. In column "FPTAS for fixed K", we denote if a fully polynomial-time approximation scheme (FPTAS) is known for the problem with fixed number of scenarios K.

In this paper a simple improvement of the midpoint approach is presented, where the basic idea is to aggregate scenarios not into a single scenario, but into a sufficiently small set of scenarios instead. We show that if the min-max problem for a constant number of scenarios is sufficiently approximable, then there is a polynomial-time εK-approximation for any constant $\varepsilon > 0$. With a slight modification, this also holds for min-max regret. This result hence improves all entries of Table 1 where the best-known approximation is K and the column "FPTAS for fixed K" is checked. Interestingly, this also leads to the first-ever approximation algorithm for min-max knapsack problems with unbounded K.

Note that this method is not a PTAS. While a PTAS exists for most problems when K is fixed, it has exponential runtime in K. Our approach remains polynomial in K, but does not give a constant approximation guarantee.

The remainder of this paper is structured as follows. In Section 2 we present our improved approximation algorithm in the case of min-max robustness, and discuss its application to min-max regret in Section 3. We describe two computational experiments for the proposed aggregation scheme in Section 4 before we conclude the paper in Section 5.

2 Min-Max Approximation

In this section, we show how to improve the K-approximation algorithm for the min-max problem to a εK-approximation algorithm for any constant $\varepsilon > 0$, if a 2-approximation is available for a fixed number of scenarios. The basic idea is the following. Let us assume we have $K = 16$ scenarios. Solving the robust problem with all 16 scenarios would yield a 1-approximation (i.e., an optimal solution). Solving the problem with only one aggregated scenario gives a 16-approximation. We show that intermediate scenario aggregations also yield intermediate approximation guarantees (see Figure 1).

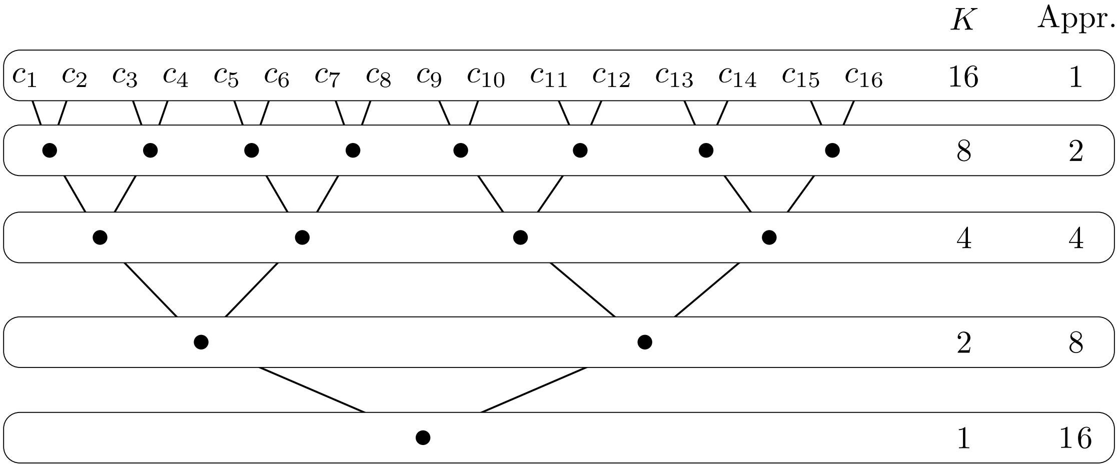

K
Appr.
c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8 c_9 c_{10} c_{11} c_{12} c_{13} c_{14} c_{15} c_{16}
16
1
8
2
4
4
2
8
1
16

Figure 1: Basic aggregation scheme.

Now let us assume we would like to have a $K/2$-approximation algorithm. We could aggregate to two scenarios, and solve the resulting problem. However, solving a min-max problem with two scenarios is usually already NP-hard. Hence, we aggregate to four scenarios instead (which would give a $K/4$-approximation if solved exactly, which is more than we need), and solve this problem with an algorithm that guarantees a

2-approximation. In total, this method then yields a $2 \cdot K/4 = K/2$-approximation. In the following, we explain the details of this procedure.

For simplicity, we assume $K = 2^k$ here, but our results readily extend to any K. If K is not equal to 2^k for some integer k, we set $k' = \lceil \log_2 K \rceil$ and create an uncertainty set $\mathcal{U}'$ with $2^{k'}$ scenarios by copying any scenarios from $\mathcal{U}$ until the required size is reached. Note that $|\mathcal{U}'| \leq 2|\mathcal{U}|$, and this does not change the corresponding robust optimization problem. We then apply our analysis to the uncertainty set $\mathcal{U}'$.

Let any partition of $[K]$ into sets S_j, $j \in [K/2]$, with cardinality 2 be given. For each $S_j = \{j_1, j_2\}$, set $\overline{c}_j = \frac{1}{2}(c_{j_1} + c_{j_2})$, i.e., $\overline{c}_j$ is the midpoint scenario of scenario set S_j.

Lemma 1. *Let $\overline{x}$ be an optimal solution for the min-max problem with scenario set $\overline{\mathcal{U}} = \{\overline{c}_1, \ldots, \overline{c}_{K/2}\}$. Then, $\overline{x}$ is a 2-approximation for the min-max problem with scenario set $\mathcal{U}$.*

Proof. Let $\overline{x}$ be an optimal solution for $\overline{\mathcal{U}}$, and x^* an optimal solution for $\mathcal{U}$. Let $i^* = \text{argmax}_{i \in [K]} c_i^t \overline{x}$ be the index of the worst-case scenario in $\mathcal{U}$ with respect to $\overline{x}$, and choose j^* such that $i^* \in S_{j^*} = \{j_1^*, j_2^*\}$. Then

$$\begin{aligned} \max_{i \in [K]} c_i^t \overline{x} = c_{i^*}^t \overline{x} \leq c_{j_1^*}^t \overline{x} + c_{j_2^*}^t \overline{x} \leq 2 \max_{j \in [K/2]} \left(\frac{c_{j_1} + c_{j_2}}{2} \right)^t \overline{x} \\ = 2 \max_{j \in [K/2]} \overline{c}_j^t \overline{x} \leq 2 \max_{j \in [K/2]} \overline{c}_j^t x^* \leq 2 \max_{i \in [K]} c_i^t x^* \end{aligned}$$

□

We repeatedly apply Lemma 1 to reduce the number of scenarios. Denote by $\overline{\mathcal{U}}(k)$ the original scenario set $\mathcal{U}$ containing all scenarios. After the first level of aggregation we end up with scenario set $\overline{\mathcal{U}}(k-1)$ containing 2^{k-1} scenarios. Repeating the aggregation process we create sets $\overline{\mathcal{U}}(\ell)$ for ℓ from k to 0.

Corollary 2. *Applying Lemma 1 repeatedly, we get a scenario set $\overline{\mathcal{U}}(\ell)$ with 2^ℓ scenarios such that solving the min-max problem with respect to $\overline{\mathcal{U}}(\ell)$ gives a $2^{k-\ell} = K/2^\ell$-approximation for the min-max problem with scenario set $\mathcal{U}$.*

We present an instance where the approximation guarantee obtained in Corollary 2 is tight for the min-max shortest path problem. Let $K = 2^k$ be the number of scenarios and 2^ℓ the number of scenarios that are used in the aggregation. Consider an instance of the shortest path problem with 2 disjoint paths of length K. The top path is divided into 2^ℓ blocks of $r := 2^{k-\ell}$ edges. All edges in the i^{th} block have cost 1 in scenario $(i-1) \cdot r + 1$ and cost 0 in all other scenarios. Hence, the objective value of the top path is equal to r. The cost structure for the bottom path is different: The i^{th} edge of the bottom path has cost 1 in the i^{th} scenario and cost 0 in all other scenarios. Hence, the objective value of the bottom path is 1. Consider the aggregation schema as in Figure 1. For both paths it holds that after the aggregation the cost of an edge of the i^{th} block has cost $\frac{1}{r}$ in the i^{th} aggregated scenario and 0 in all other aggregated scenarios. Hence, both paths are identical with respect to the aggregated scenarios and the optimal solution of

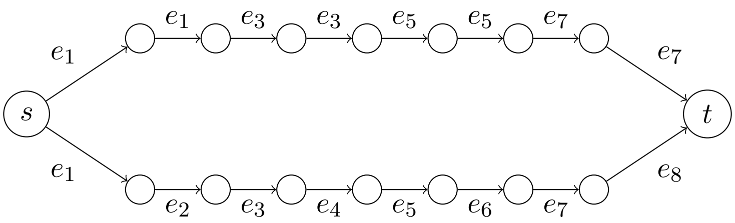


Figure 2: An instance of the min-max shortest path problem with 8 scenarios. An edge with vector e_i has cost 1 in the i^{th} scenario and 0 in all other scenarios. For this instance the approximation guarantee of Corollary 2 is tight if we aggregate to 4 scenarios.

the aggregated problem may consist of the top instead of the bottom path. This leads to a gap of $2^{k-\ell} = K/2^\ell$. In Figure 2, we present this worst case instance for $K = 8$ and $\ell = 2$.

Lemma 3. *A solution that is an α-approximation for $\overline{\mathcal{U}}(\ell)$ is also an $(\alpha K/2^\ell)$-approximation for $\mathcal{U}$.*

Proof. Analogously to the proof of Lemma 1. □

We can now state the main result of this section.

Theorem 4. *Let a constant $0 < \varepsilon \leq 1$ be given, and let K be unbounded. If there exists a 2-approximation algorithm for the min-max problem with a fixed number of scenarios, there exists a polynomial-time algorithm that gives an (εK)-approximation for the min-max problem.*

Proof. Let $\varepsilon \leq 1$ be constant. We choose $\ell' := \lceil \log \frac{1}{\varepsilon} + 1 \rceil$. As K is unbounded, we can assume that $K \geq \ell'$. According to Corollary 2 we construct the set $\overline{\mathcal{U}}(\ell')$ with $2^{\ell'}$ scenarios. Using the 2-approximation algorithm for the min-max problem with a fixed number of scenarios, we find a 2-approximation for $\overline{\mathcal{U}}(\ell')$. Using Lemma 1, we conclude that the solution is a $2 \cdot 2^{k-\ell'} = 2K/2^{\lceil \log \frac{1}{\varepsilon} + 1 \rceil} \leq K/\frac{1}{\varepsilon} = \varepsilon K$-approximation. Note that the running time of this procedure is polynomial since the value of ϵ and, therefore, also ℓ', is fixed. □

Corollary 5. *Let a constant $0 < \varepsilon \leq 1$ be given, and let K be unbounded. Then there exists a polynomial-time (εK)-approximation algorithm for the min-max shortest path, spanning tree, selection, and knapsack problem.*

As examples, let us consider the min-max selection and shortest path problems. For both problems we need a 2-approximation algorithm for the min-max problem with a fixed number of scenarios.

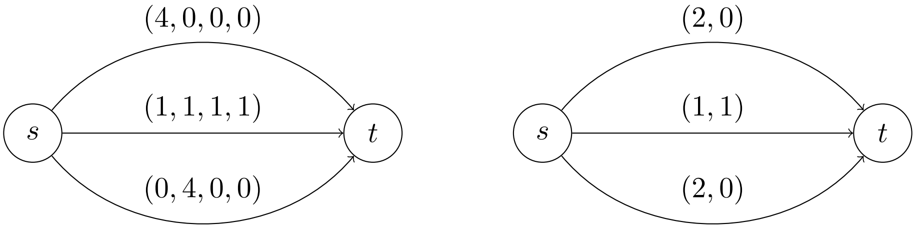
- For selection, there exists an FPTAS that finds for a fixed number of scenarios $\tilde{K}$ a $(1 + \tilde{\varepsilon})$-approximation with running time $\mathcal{O}\left(\frac{p^{\tilde{K}} n}{\tilde{\varepsilon}^{\tilde{K}-1}}\right)$ (see [KZ07]). Hence, using Theorem 4, an (εK)-approximation is possible in $\mathcal{O}\left(p^{\frac{4}{\varepsilon}}n\right)$ by aggregating to $\tilde{K} = 2^{\lceil \log \frac{1}{\varepsilon}+1\rceil}$ scenarios, and approximating the resulting problem with a factor of 2, i.e., choosing $\tilde{\varepsilon} = 1$.

- For shortest path, there exists an FPTAS that finds for a fixed number of scenarios $\tilde{K}$ a $(1+\tilde{\varepsilon})$-approximation with running time in $\mathcal{O}\left(\frac{mn^{\tilde{K}}}{\tilde{\varepsilon}^{\tilde{K}-1}}\right)$, which makes it possible to find an (εK)-approximation in time $\mathcal{O}\left(mn^{\frac{4}{\varepsilon}}\right)$ by aggregating to $\tilde{K} = 2^{\lceil \log \frac{1}{\varepsilon}+1\rceil}$ scenarios, and approximating the resulting problem with a factor of 2, i.e. choosing $\tilde{\varepsilon} = 1$.

3 Min-Max Regret Approximation

To translate the results obtained for min-max to min-max regret problems we need to modify the aggregation procedure, as the following example shows.

Example 6. *Consider the min-max regret shortest path instance shown in Figure 3a. There are four scenarios. An optimal solution is to take the path in the middle with a regret of* 1. *If we aggregate the first two and the last two scenarios, we arrive at the instance shown in Figure 3b. Here, all three paths are optimal with a perceived regret of* 1 *(the top and bottom part have a regret of* 1 *in the first scenario, while the middle path has a regret of* 1 *in the second scenario). Without loss of generality we can hence assume that an optimization algorithm will return the top path, but its true regret is* 4. *Hence, in this example, we obtain only a* 4*-approximation and not a* 2*-approximation as in the case of the min-max objective function.*



(a) Instance with 4 scenarios.

(b) Instance with aggregated scenarios

Figure 3: An example instance where aggregating from four to two scenarios leads to a 4-approximation for min-max regret.

Instead of simply aggregating pairs $S_j = \{j_1, j_2\}$ of scenarios and solving a min-max regret problem on this new scenario set, we consider the following problem

$$\min_{x\in\mathcal{X}} \max_{j\in[K/2]} \bar{c}_j^t x - \frac{1}{2}opt(c_{j_1}) - \frac{1}{2}opt(c_{j_2}) \tag{*}$$

Note that we do not use the objective $\overline{c}_j^t - opt(\overline{c}_j)$, as would be usual for min-max regret. Further generalizing, we call a problem

$$\min_{x\in\mathcal{X}} \max_{i\in[K]} c_i^t x - d(c_i)$$

with arbitrary d a generalized min-max regret problem.

Lemma 7. *Solving the generalized min-max regret problem (*) on $\overline{\mathcal{U}} = \{\overline{c}_1, \ldots, \overline{c}_{K/2}\}$ is a 2-approximation for $\mathcal{U}$.*

Proof. Let $\overline{x}$ be optimal for problem (*), and x^* optimal for the original problem with uncertainty set $\mathcal{U}$. Again, denote by $i^* = \text{argmax}_{i\in[K]}\, c_i^t\overline{x} - opt(c_i)$ and choose j^* such that $i^* \in S_{j^*} = \{j_1^*, j_2^*\}$. Then

$$\begin{aligned}
\max_{i\in[K]} \left(c_i^t\overline{x} - opt(c_i)\right) &= c_{i^*}^t\overline{x} - opt(c_{i^*}) \\
&\le c_{j_1^*}^t\overline{x} - opt(c_{j_1^*}) + c_{j_2^*}^t\overline{x} - opt(c_{j_2^*}) \\
&\le \max_{j\in[K/2]} \left((c_{j_1} + c_{j_2})^t\overline{x} - opt(c_{j_1}) - opt(c_{j_2})\right) \\
&= 2 \max_{j\in[K/2]} \left(\overline{c}_j^t\overline{x} - \frac{1}{2}opt(c_{j_1}) - \frac{1}{2}opt(c_{j_2})\right) \\
&\le 2 \max_{j\in[K/2]} \left(\overline{c}_j^t x^* - \frac{1}{2}opt(c_{j_1}) - \frac{1}{2}opt(c_{j_2})\right) \\
&= \max_{j\in[K/2]} \left(c_{j_1}^t x^* - opt(c_{j_1}) + c_{j_2}^t x^* - opt(c_{j_2})\right) \\
&\le 2\max_{i\in[K]} \left(c_i^t x^* - opt(c_i)\right)
\end{aligned}$$

□

Note that the arguments used in the proof of Lemma 7 can be generalized to the case where scenarios are aggregated repeatedly, i.e., we aggregate to sets S_j with more than two elements. Similar to Corollary 2 we obtain:

Corollary 8. *Given an aggregated scenario set $\overline{\mathcal{U}}(\ell)$ where each of the 2^ℓ scenarios is given as $\overline{c}_j := \frac{1}{2^{k-\ell}}\sum_{s\in S_j} c_s$. The optimal solution of the generalized min-max regret problem*

$$\min_{x\in\mathcal{X}} \max_{j\in[2^\ell]} \overline{c}_j x - \frac{1}{2^{k-\ell}} \sum_{s\in S_j} opt(c_k)$$

yields an $(K/2^\ell)$-approximation of the min-max regret problem with scenario set $\mathcal{U}$.

Similar to min-max, we can use a 2-approximation for the generalized min-max regret problem with a fixed number of scenarios to obtain an (ϵK)-approximation for the min-max regret problem. Using the same arguments as in the proof of Theorem 4 we obtain:

Theorem 9. *Let a constant $0 < \varepsilon \leq 1$ be given, and let K be unbounded. If there exists a 2-approximation algorithm for the generalized min-max regret problem with a fixed number of scenarios, then there exists a polynomial-time algorithm that gives an (εK)-approximation for the min-max regret problem.*

Note that in our construction of the generalized problem, we have $d(c_i) \leq opt(c_i)$. Hence, we can use the same proof as in [ABV07] using the FPTAS for multi-objective spanning tree to show that there is an FPTAS for our generalized min-max regret spanning tree problem. The same approach applies to the min-max regret selection problem.

Furthermore, we can modify any generalized min-max regret shortest path problem by adding an edge from s to t with costs $d(c_i)$ in scenario i. We create an additional scenario where the costs of each edge is 0, and the costs of the new edge is a sufficiently large value M. As $d(c_i) \leq opt(c_i)$, we can then solve a classic min-max regret problem on this instance, giving the same objective value as before. Hence, the FPTAS for min-max regret shortest path (see [ABV07]) can also be applied to our generalized problem.

Corollary 10. *Let a constant $0 < \varepsilon \leq 1$ be given, and let K be unbounded. Then there exists a polynomial-time (εK)-approximation algorithm for the min-max regret shortest path, spanning tree, and selection problem.*

4 Computational Experiments

In this section, we present two experiments to test the proposed aggregation method. In both experiments the underlying problem is a robust shortest path problem with a discrete uncertainty set. The underlying graph of the shortest path problem is a complete layered graph with 10 layers and width 4. The scenario set consists of randomly generated scenarios. The cost of each edge is chosen uniformly in $[0, 1]$ for each scenario.

In the first experiment, we investigate the influence of the aggregation scheme on solution quality and computation time. We start by solving the robust problem exactly using an IP formulation which contains one additional constraint per scenario. In the next step, we aggregate two consecutive scenarios. This reduces the number of constraints in the IP formulation. We solve the reduced IP formulation and repeat the aggregation process until we end up with a single scenario. To make the aggregation scheme work properly we have to ensure at the beginning of the aggregation that the number of scenarios is a power of two. This is done by duplicating some of the scenarios. For example the scenario set $\{c_1, c_2, c_3, c_4, c_5\}$ is replaced with the scenario multiset $\{c_1, c_1, c_2, c_2, c_3, c_3, c_4, c_5\}$. We evaluate each computed solution using the original set of scenarios and divide the objective value of the solution by the objective value of the optimal solution. The results are shown in Figure 4. Additionally, we measure the time which is necessary to solve the resulting IP formulations see Figure 5. All results are averaged over 10,000 experiments. The IP formulation are solved with CPLEX v12.6 using a desktop computer with 8 GB RAM and a $i5$ quad-core processor with 3.2 GHz.

The aggregation level 0 corresponds to the original set of scenarios, i.e. solving the original problem to optimality. The aggregation level 1 means that the number of scenarios is halved, etc. An increasing number of aggregation reduces the solution quality.

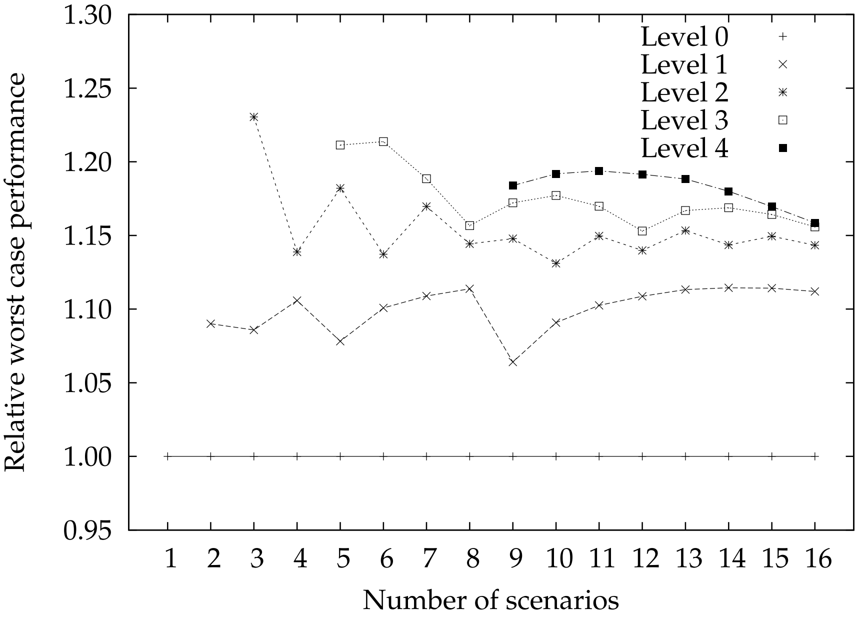


Figure 4: The horizontal axis gives the number of scenarios. The relative worst case performance of the different solutions for the different levels of aggregation is shown on the vertical axis.

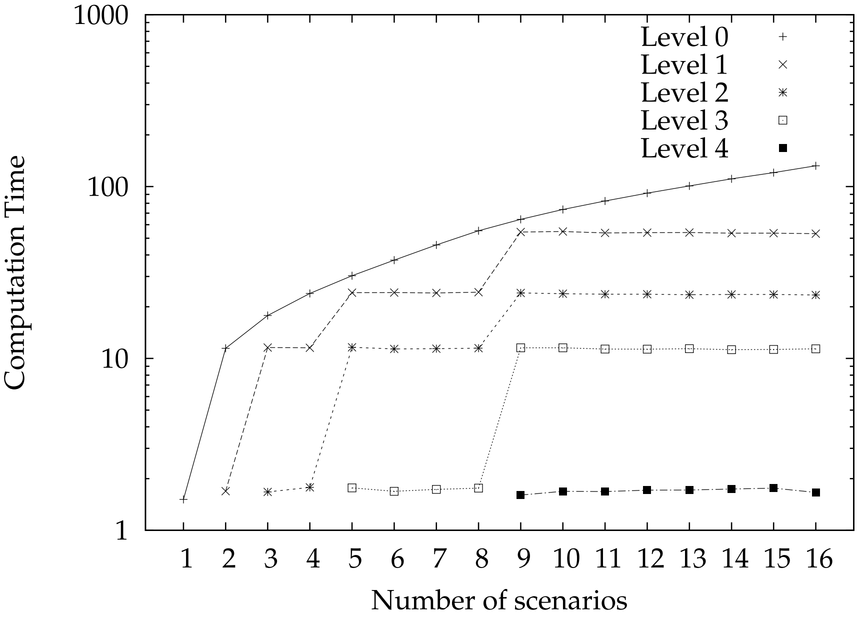


Figure 5: The horizontal axis gives the number of scenarios. The computation times (in ms) for the different levels of aggregation is shown on the vertical axis in logarithmic scale.

But each additional aggregation level has a smaller impact. Note also that the relative performance of the aggregated solutions is far below the theoretical performance guarantee. The solution time depends only on the number of scenarios which remains after the aggregation. Recall the duplication of scenarios which increases the number of scenarios to the next power of two.

In the proposed aggregation scheme (see Figure 1) always two consecutive scenarios are aggregated. This rule is arbitrary. In the second experiment, we test if a more sophisticated aggregation rule can lead to an improved solution quality. For the improved aggregation scheme we aggregate similar scenarios. To this end, we computed a minimal cost perfect matching between the different scenarios. We set the cost of matching scenario i with scenario j to the Euclidean distance of scenario i and j. The optimal matching produces a list of pairs of scenarios. We sort this list with respect to the Euclidean distance between the scenarios of the pair. We then iteratively aggregate the matched pairs of scenarios, beginning with the pair which has the smallest Euclidean distance, until all matched pairs are aggregated. Then we recompute the perfect matching and repeat the aggregation process. This is repeated until all scenarios are aggregated to a single scenario. To model the naive approach we iteratively aggregate two consecutive scenarios until we have halved the number of scenarios. Then we repeat the aggregation process from the beginning, until we end up with a single scenario. For the experiment we used a robust shortest path problem with 16 scenarios. The results of the experiment, averaged over 10,000 instances, are shown in Figure 6.

It can be seen that the more involved aggregation rule based on scenario similarity gives indeed better results for intermediate aggregation levels. For full or no aggregation, the used aggregation rule is of course irrelevant.

5 Conclusions

The midpoint method is a central approximation algorithm in robust optimization. Despite its simplicity, it has been the best-known method for several classic combinatorial problems. In this paper, we presented a simple variant of the method, where the uncertainty set is not aggregated to a single scenario, but to a sufficiently small set of scenarios instead. This reduced scenario set is then approximated using, e.g., an FPTAS for discrete uncertainty of constant size. Our approach can be used to find polynomial time (εK)-approximations for any constant $\varepsilon \in (0, 1]$, thus improving several currently known best approximability results.

Our results hold for any aggregation scheme. However, for practical purposes, aggregating similar scenarios is reasonable, so as to preserve the structure of the uncertainty set as far as possible. To quantify this effect we considered a computational experiment using random shortest path instances. Our results indicate that approximation guarantees are considerably smaller on these instances than the theoretical bounds suggest, and that aggregating similar scenarios does indeed improve the quality of solutions.

Another frequently considered approximation algorithm for min-max problems is the element-wise worst-case approach (see [ABV09]). That is, when two scenarios $c_i =$

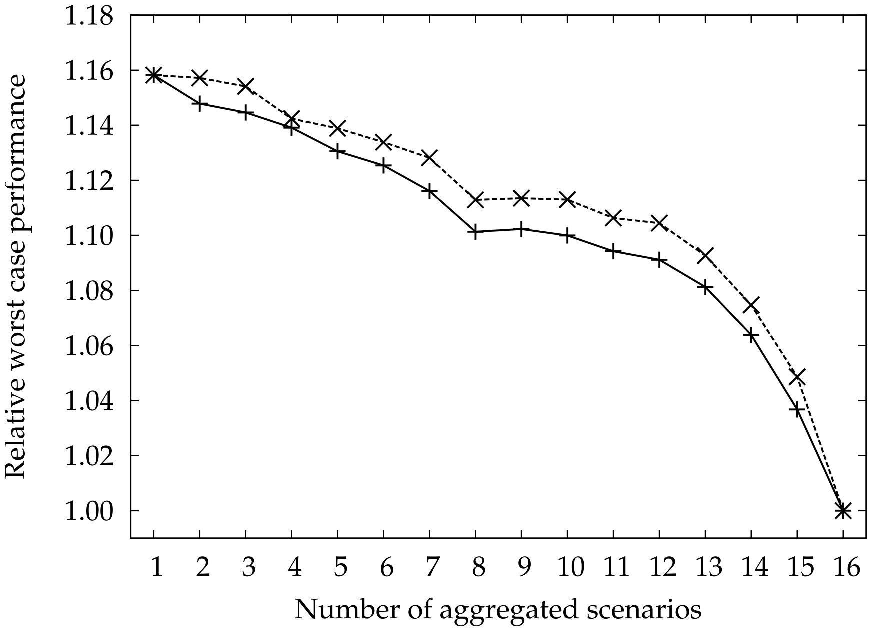


Figure 6: The horizontal axis gives the number of scenarios that remain after each aggregation. The relative worst case performance of the different solutions for the different levels of aggregation is shown on the vertical axis. The straight line shows the performance when aggregating similar scenarios and the dashed line shows the performance of the consecutive aggregation scheme.

$(c_{i1},\ldots,c_{in})$ and $c_j = (c_{j1},\ldots,c_{jn})$ are aggregated, the resulting scenario is given as $\overline{c} = (\max\{c_{i1},c_{j1}\},\ldots,\max\{c_{in},c_{jn}\})$. With an analogous analysis as in Section 2, one can also show that the same aggregation approach can be used in this setting.

References

[ABV07] H. Aissi, C. Bazgan, and D. Vanderpooten. Approximation of min–max and min–max regret versions of some combinatorial optimization problems. *European Journal of Operational Research*, 179(2):281 – 290, 2007.

[ABV09] H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427 – 438, 2009.

[CG15] A. Chassein and M. Goerigk. A new bound for the midpoint solution in minmax regret optimization with an application to the robust shortest path problem. *European Journal of Operational Research*, 244(3):739–747, 2015.

[Con12] E. Conde. On a constant factor approximation for minmax regret problems using a symmetry point scenario. *European Journal of Operational Research*, 219(2):452–457, 2012.

[KY97] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.

[KZ06] A. Kasperski and P. Zieliński. An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Information Processing Letters*, 97(5):177–180, 2006.

[KZ07] A. Kasperski and P. Zieliński. Approximation of min-max (regret) combinatorial optimization problems under discrete scenario representation. Technical report, Institute of Mathematics and Computer Science, Wroclaw University of Science and Technology, PRE 7, 2007.

[KZ16] A. Kasperski and P. 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.