# Two-Stage Robust Combinatorial Optimization with Priced Scenarios

**Roman Rischke**

**Abstract** Two-stage robust combinatorial optimization is an established methodology for handling combinatorial optimization problems with uncertain input. Without knowing the actual data, a partial solution needs to be fixed in the first stage which is then extended to a feasible solution in the second stage at higher cost once the data is revealed. The overall goal is to construct a solution that is feasible in all scenarios, i.e., robust against uncertainty, and minimizes the worst-case cost. Since considering all possible scenarios usually leads to a robust solution that is too conservative and too expensive, a central question is to decide on a subset of scenarios to be taken into account. Restricting the set of possible scenarios is a common approach, but this usually depends on subjective decision criteria like the willingness to take risks or the expectation on the future. We propose an alternative concept. Instead of restricting the set of scenarios we price all scenarios, which affects the objective function in such a way that we receive a certain scenario-dependent reward that reduces the overall cost. This leads to new two-stage robust optimization problems. We study complexity and devise approximation algorithms for such problems.

## 1 Introduction

Practical applications of combinatorial optimization often require decision making under data uncertainty. Reasons for that are usually measurement errors or simply the impossibility of precisely predicting the future. Data uncertainty in optimization problems is usually represented by a set of possible scenarios, where a scenario is a particular realization of the uncertain input parameters. Two-stage robust combinatorial optimization is an established methodology for handling combinatorial optimization problems with uncertain input. The methodology was introduced by

R. Rischke (✉)
Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany
e-mail: roman.rischke@tu-berlin.de

D. Huisman et al. (eds.), *Operations Research Proceedings 2013*,
Operations Research Proceedings, DOI: 10.1007/978-3-319-07001-8_51,

© Springer International Publishing Switzerland 2014

Dhamdhere et al. [3] and subsequently used by different authors [4–7]. Two-stage robust combinatorial optimization is based on a two-stage decision process that we want to illustrate with the following example.

In a frequently flooded region, high restoration cost needs to be paid if a town is under water. In order to avoid these costs, we want to equip some towns with flood protection systems that prevent any damage in case of a flood event. In the first stage we do not know whether a town will be flood-affected or not. That is, the set of flood-affected towns is uncertain in the first stage. We only have information about possible scenarios, where in this example a scenario is a particular set of affected towns. In the second stage a scenario is revealed to us and we have to pay the cost for the restoration of all flood-affected towns that are not equipped with a protection system. We assume, that the cost for the restoration of a town is much higher than the investment in a flood protection system. Our goal is to find a set of towns that we equip with protection systems in the first stage such that we minimize the total cost (first stage cost and second stage cost) in the worst-case scenario. We remark that the worst-case scenario depends on our first stage decision. This means that we want to solve a min-max problem, where we minimize over our possible first stage choices and maximize over our underlying set of scenarios. We can think of a malign adversary, who, once we have taken the first stage decision, picks a scenario which is worst possible with respect to our decision.

We can extend the described two-stage decision process to a decision process with multiple stages. Optimization problems of that kind are called multi-stage robust (combinatorial) optimization problems. However, in this work we restrict our attention to the two-stage model and refer the interested reader to [1, Chap. 14] and the references therein.

## 2 Priced Scenarios

The goal of robust optimization problems is to construct a solution that is feasible in all scenarios of an underlying scenario set, i.e., robust against uncertainty, and we want to minimize the worst-case cost of the constructed solution. In the above example, feasibility of a solution means that either we have equipped a flood-affected town with a protection system in the first stage or we pay the cost for the restoration of that town in the second stage. If we take all possible scenarios into account, then, by the cost assumption, this will cause us to equip every town with a protection system which is very conservative. Aiming for a more reasonable solution, we firstly have to answer the following central question: *Which scenarios do we take into account?* The more scenarios we take into account the more expensive our robust solution usually is. This is often called “the price of robustness” [2]. Restricting the set of all possible scenarios to a set of reasonable scenarios is a common approach. Note that in robust combinatorial optimization we usually are able to express the set of all possible scenarios in a compact way. In the above example, it is the power set of the set of all considered towns.

In two-stage robust combinatorial optimization, the *discrete scenario approach* (see [3, 8]) and the *$\Gamma$-scenario approach* (see [2, 4]) have become the two main approaches for representing the scenario set in the input. In the discrete scenario approach, all scenarios that we want to take into account are explicitly given as part of the input. This approach is appropriate for problems where the number of possible scenarios is manageable. In the $\Gamma$-scenario approach, we implicitly describe the scenario set by a parameter $\Gamma$ which is part of the input. Let us illustrate this by the above example. Suppose, we only want to be robust against situations where at most $\Gamma$ many towns are flood-affected. One reason for that might be that we expect the number of affected towns to be no greater than $\Gamma$, though we do not know the exact set of affected towns. In this case, we only need the set of all towns and the parameter $\Gamma$ to describe our scenario set. This approach allows us to consider an exponential number of scenarios without listing them all as in the discrete scenario approach.

Both approaches enable us to restrict the set of all possible scenarios, but restricting the scenario set usually depends on subjective decision criteria like the willingness to take risks or the expectation on the future. We propose an alternative approach. Instead of restricting the scenario set we price all scenarios. That is, we define a function that assigns a nonnegative price to each scenario of the unrestricted scenario set. Those prices affect the objective function and lead to new two-stage robust combinatorial optimization problems. We want to illustrate this new approach by the introductory example.

In the new approach, we consider the unrestricted scenario set. In other words, our scenario set is the power set of the set of all considered towns. We extend our example by an insurance company. With this insurance company we negotiate in advance a scenario-dependent price which is paid out to us in the second stage. That is, we agree a price for each scenario that we get paid if the scenario materializes. Our new goal is to find a set of towns that we equip with protection systems in the first stage such that we minimize the balance (first stage cost and second stage cost minus insurance payout) in the worst-case scenario. We assume the insurance premium (fee paid by us to the insurer) to be constant and therefore we can ignore it in the objective function. Again, we want to solve a min-max problem, but now we have a different objective function and we do not have to restrict the scenario set.

Before we give an overview of our results with the new approach, we observe that the new approach generalizes both the discrete scenario approach and the $\Gamma$-scenario approach. We can set the price of the scenarios that we want to "exclude" from the scenario set to infinity and thus the adversary has no incentive to choose those scenarios.

## 3 Results

In this section we give an overview of our results with the new approach. We study complexity and approximation algorithms for a generalization of the afore-mentioned example. This more general problem is called *two-stage robust weighted disjoint* *hitting set problem*. The deterministic version of that problem is a special case of many different combinatorial optimization problems, e.g. the *set cover problem* and the *Steiner tree problem*. This also holds for the two-stage robust versions of those problems. Let us first describe the deterministic weighted disjoint hitting set problem (WDHS problem). We are given a set of $n$ elements $E := \{e_1, \dots, e_n\}$, a collection $\mathscr{M} := \{M_1, \dots, M_m\}$ of $m$ pairwise disjoint subsets of $E$, i.e., $M_i \cap M_j = \emptyset$ for all $i, j \in \{1, \dots, m\}$ with $i \neq j$, and a cost function $c \colon E \to \mathbb{N}$. A feasible solution for the WDHS problem is a set $F \subseteq E$ that has at least one element in common with every set $M \in \mathscr{M}$, i.e., $|F \cap M| \geq 1$ for all $M \in \mathscr{M}$. Thus, the set of feasible solutions can be defined as $\mathscr{F} := \{F \subseteq E \mid \forall M \in \mathscr{M} : |F \cap M| \geq 1\}$. The goal is to find a feasible solution $F \in \mathscr{F}$ that minimizes the total cost $f(c, F) := \sum_{e \in F} c(e)$. The WDHS problem can be solved in polynomial time by selecting the cheapest element out of each set $M \in \mathscr{M}$.

Based on this, we can describe the two-stage robust WDHS problem. As in the deterministic version, we are given the set $E$, the collection $\mathscr{M}$ and the cost function $c$ as defined above. Additionally, we are given a vector $\lambda := \left(\lambda_{e_1}, \dots, \lambda_{e_n}\right)^{\mathrm{T}} \in \mathbb{Q}^n$ with $\lambda_e \geq 1$ for all $e \in E$ and a scenario set $\mathscr{S}$, where a scenario $S$ is a subset of $\mathscr{M}$. That means that every scenario $S \in \mathscr{S}$ defines a set of feasible solutions $\mathscr{F}^S := \{F \subseteq E \mid \forall M \in S : |F \cap M| \geq 1\}$. In the following, a set $M \in \mathscr{M}$ is called *active* in scenario $S$ if $M \in S$. In the first stage we do not know which scenario $S \in \mathscr{S}$ will materialize in the second stage, but we already can buy elements $e \in E$ in order to “hit" sets. A set $M \in \mathscr{M}$ is *hit* if we buy at least one element of the set $M$. In the first stage, the cost of an element $e \in E$ is $c(e)$. If we hit a set $M \in \mathscr{M}$ already in the first stage, we do not have to hit $M$ in the second stage in case $M$ is active in the realized scenario. In the second stage a scenario $S \in \mathscr{S}$ is revealed to us and we need to hit all sets $M \in S$ that were not already hit in the first stage. Hitting a set in the second stage is costlier than in the first stage. Every element $e \in E$ has its own given inflation factor $\lambda_e \geq 1$ and costs in the second stage $\lambda_e c(e)$. Let us formulate the goal. We buy a set of elements $F_1 \subseteq E$ already in the first stage and pay $f(c, F_1) := \sum_{e \in F_1} c(e)$. In the second stage we augment the set $F_1$ by buying an additional set of elements $F_S \subseteq E$, where $S$ is the realized scenario, and we pay $f(\lambda c, F_S) := \sum_{e \in F_S} \lambda_e c(e)$. Our solution $(F_1, F_S)$ is feasible in the scenario $S$ if $F_1 \cup F_S \in \mathscr{F}^S$. The goal is to find a set $F_1$ and sets $F_S$, $S \in \mathscr{S}$, such that we minimize the total cost in the worst-case scenario. That is, we want to find a solution for the following min-max problem:

$$\min \left\{ f(c, F_1) + \max_{S \in \mathscr{S}} \{f(\lambda c, F_S)\} \mid \forall S \in \mathscr{S} : F_1 \cup F_S \in \mathscr{F}^S \right\}. \tag{1}$$

Following the discrete scenario approach, we can show that the two-stage robust WDHS problem is *NP*-hard, even if we are given only two scenarios. This is shown by a reduction from the *NP*-complete decision problem *minimum knapsack*. Ideas from the reduction can be used to formulate the problem as a dynamic program that can be solved in pseudo-polynomial time if the cardinality of the scenario set is constantly bounded from above. By using known methods from two-stage stochastic programming [9], we can show that there is a 2-approximation algorithm for the two-stage robust WDHS problem with discrete scenarios.

However, if we follow the $\Gamma$-scenario approach, the two-stage robust WDHS problem can be solved in polynomial time. In this case the scenario set is defined by $\mathscr{S} := \{S \subseteq \mathscr{M} : |S| \leq \Gamma\}$, where $\Gamma \in \{1, \dots, m\}$ is part of the input. We obtain this result by narrowing down the solution space and greedily selecting sets $M \in \mathscr{M}$ to be hit already in the first stage.

Using the new approach, we slightly need to modify the objective function in (1). We need to incorporate the given price function $p \colon \mathscr{S} \to \mathbb{Q}^+$ that reduces the second stage cost as described in Sect. 2. This leads to the following min-max problem:

$$\min \left\{ f(c, F_1) + \max_{S \in \mathscr{S}} \{f(\lambda c, F_S) - p(S)\} \mid \forall S \in \mathscr{S} : F_1 \cup F_S \in \mathscr{F}^S \right\}, \quad (2)$$

where $\mathscr{S} := 2^{\mathscr{M}}$. In the following, we consider two different price functions $p$. To define them we need the following additional input. For each set $M \in \mathscr{M}$ we are given a price $\gamma_M \in \mathbb{Q}^+$. Based on this, we say that the price function $p$ is *sum-based* if $p(S) := \sum_{M \in S} \gamma_M$ and we called it *extremum-based* if $p(S) := \max_{M \in S} \gamma_M$ for all $S \in \mathscr{S}$. To explain the following results we introduce the values $\alpha_M := \min_{e \in M} c(e)$ and $\beta_M := \min_{e \in M} \lambda_e c(e)$ for all $M \in \mathscr{M}$.

Let us first consider the case that we are given a sum-based price function $p$. In this case the two-stage robust WDHS problem (as defined in (2)) can be solved in polynomial time. Let us examine why this is the case and therefor we let $\mathscr{M}' \subseteq \mathscr{M}$ be the collection of all sets that we hit already in the first stage. The collection $\mathscr{M}'$ depends on $F_1$. First of all, we observe that the adversary will confront us with the worst-case scenario $S^* := \{M \in \mathscr{M} \setminus \mathscr{M}' \mid \beta_M - \gamma_M > 0\}$. Thus we pay $\sum_{M \in \mathscr{M}'} \alpha_M$ in the first stage and $\sum_{M \in S^*} (\beta_M - \gamma_M)$ in the second stage. It is not hard to see that we minimize that payment if we hit all sets $M \in \mathscr{M}$ (at minimum cost) already in the first stage that fulfill $\alpha_M \leq \beta_M - \gamma_M$. This is in line with what intuition tells us.

However, the situation changes drastically if we consider the case that we are given an extremum-based price function $p$. We can show that the corresponding two-stage robust WDHS problem is *NP*-hard. This is also shown by a reduction from the *NP*-complete decision problem minimum knapsack. We can also show that the problem can be formulated as a dynamic program which can be transformed into an FPTAS by using [10].

## 4 Conclusion

Pricing scenarios instead of restricting the set of scenarios is an alternative way for dealing with two-stage robust optimization problems. In this work we have motivated and introduced the new approach and we have studied complexity and approximation algorithms for the two-stage robust WDHS problem. In particular, we have seen that the complexity significantly depends on the pricing method. In this work we have presented our first results with the new approach and we hope to foster further research in this fascinating area.

**Acknowledgments** I thank my advisors, Rolf H. Möhring and Sebastian Stiller, for all their support and suggestions. I also thank Wiebke Höhn, Daniela Luft and Roland Pörner for their helpful comments on this paper.

## References

1. Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization. Princeton series in applied mathematics*. Princeton, N.J.: Princeton University Press.
2. Bertsimas, D. J., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*(1), 35–53.
3. Dhamdhere, K., Goyal, V., Ravi, R., & Singh, M. (2005). How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)* (pp. 367–378).
4. Feige, U., Jain, K., Mahdian, M., & Mirrokni, V. (2007). Robust combinatorial optimization with exponential scenarios. In D.P. Williamson & M. Fischetti (Eds.), *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO 2007)*. Lecture notes in computer science, Vol. 4513 (pp. 439–453).
5. Golovin, D., Goyal, V., & Ravi, R. (2006). Pay today for a rainy day: Improved approximation algorithms for demand-robust min-cut and shortest path problems. In B. Durand & W. Thomas (Eds.), *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS 2006)*. Lecture notes in computer science, Vol. 3884 (pp. 206–217).
6. Gupta, A., Nagarajan, V., & Ravi, R. (2010). Thresholded covering algorithms for robust and max-min optimization. In S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, P.G. Spirakis (Eds.), *Proceedings of the 37th International Colloquium (ICALP 2010)*. Lecture notes in computer science, Vol. 6198 (pp. 262–274).
7. Khandekar, R., Kortsarz, G., Mirrokni, V., & Salavatipour, M. R. (2008). Two-stage robust network design with exponential scenarios. In D. Halperin & K. Mehlhorn (Eds.), *Proceedings of the 16th Annual European Symposium (ESA 2008)*. Lecture notes in computer science, Vol. 5193 (pp. 589–600).
8. Kouvelis, P., & Yu, G. (1997). *Robust discrete optimization and its applications. Nonconvex optimization and its applications*, Vol. 14. Dordrecht: Kluwer Academic Publishers.
9. Shmoys, D. B., & Swamy, C. (2006). An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *Journal of the ACM*, *53*(6), 978–1012.
10. Woeginger, G. J. (2001). *When does a dynamic programming formulation guarantee the existence of an FPTAS?*. Electronic Colloquium on Computational Complexity, Report No. 84.