On the Complexity of Recoverable Robust Optimization in the Polynomial Hierarchy

Christoph Grüne
Department of Computer Science, RWTH Aachen University, Germany

Lasse Wulf
Section of Algorithms, Logic and Graphs, Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

Recoverable robust optimization is a popular multi-stage approach, in which it is possible to adjust a first-stage solution after the uncertain cost scenario is revealed. We consider recoverable robust optimization in combination with discrete budgeted uncertainty. In this setting, it seems plausible that many problems become Σ_3^p-complete and therefore it is impossible to find compact IP formulations of them (unless the unlikely conjecture $NP = \Sigma_3^p$ holds). Even though this seems plausible, few concrete results of this kind are known. In this paper, we fill that gap of knowledge. We consider recoverable robust optimization for the nominal problems of Sat, 3Sat, vertex cover, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median, independent set, clique, subset sum, knapsack, partition, scheduling, Hamiltonian path/cycle (directed/undirected), TSP, k-directed disjoint path ($k \geq 2$), and Steiner tree. We show that for each of these problems, and for each of three widely used distance measures, the recoverable robust problem becomes Σ_3^p-complete. Concretely, we show that all these problems share a certain abstract property and prove that this property implies that their robust recoverable counterpart is Σ_3^p-complete. This reveals the insight that all the above problems are Σ_3^p-complete "for the same reason". Our result extends a recent framework by Grüne and Wulf.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Complexity, Robust Optimization, Recoverable Robust Optimization, Two-Stage Problems, Polynomial Hierarchy, Sigma 2, Sigma 3

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.52

Related Version *Full Version*: `https://arxiv.org/abs/2411.18590` [19]

Funding *Christoph Grüne*: Funded by the German Research Foundation (DFG) – GRK 2236/2. *Lasse Wulf*: Funded by the Carlsberg Foundation CF21-0302 "Graph Algorithms with Geometric Applications".

1 Introduction

Real-world decision makers are faced with a large degree of uncertainty when making important decisions in economics, planning, and operations research. The successful area of *Robust optimization* [1, 14, 28] has been developed as one possible way to deal with these uncertainties. However, sometimes the classical robust optimization approach has turned out to be too conservative. For this reason, *Recoverable robust optimization* was introduced by Liebchen, Lübbecke, Möhring and Stiller [31]. Initially motivated by train scheduling problems, it has since then found wide-spread application in practice and in the analysis of standard optimization problems. For instance, it has been successfully applied to combinatorial problems such as s-t-path [4, 23, 24], matching [11], scheduling [3], spanning tree [21, 22], knapsack [6, 7, 8, 29], and TSP [9, 15]. Recoverable robust optimization follows a two-step robust optimization approach. In contrast to classic robust optimization, where a first-stage solution cannot be changed, in recoverable robust optimization the decision


© Christoph Grüne and Lasse Wulf;
licensed under Creative Commons License CC-BY 4.0

maker is allowed to incorporate a limited recovery action after the underlying uncertainty is revealed. Mathematically, this is described by the expression

$$\min_{S_1} \quad \max_{c_2 \in C} \quad \min_{S_2, \mathrm{dist}(S_1,S_2) \leq \kappa} c_1(S_1) + c_2(S_2).$$

Here, S_1 denotes the first-stage solution, S_2 denotes the second-stage solution, c_1 denotes the first-stage cost function, C denotes the set of uncertain scenarios, and *dist* is some distance function. A more formal definition will be given later.

In this paper, we make the standard assumption that the underlying uncertainty for the robust problem is given as a discrete budgeted uncertainty set, also denoted as discrete Γ-uncertainty [2].

The study of Σ_3^p-complete problems. Recoverable robust problems are described by min-max-min formulations. From a complexity-theoretic point of view, such min-max-min problems are often even harder than NP-complete, namely they are often complete for the third stage of the so-called *polynomial hierarchy* [33]. A problem complete for the k-th stage of the hierarchy is called Σ_k^p-complete. (In this paper, we are concerned mostly with the case of $k = 3$). The theoretical study of Σ_k^p-complete problems is important: If a problem is found to be Σ_3^p-complete, it means that, under some basic complexity-theoretic assumptions[1], it is not possible to find an integer programming formulation of the problem of polynomial size [34] (also called a *compact* model). This means that no matter how cleverly a decision maker tries to design their integer programming model, it must inherently have a huge number of constraints and/or variables, and may be much harder to solve than even NP-complete problems.

Even though this fact makes the study of Σ_3^p-complete problems compelling, surprisingly few results relevant to the area of min-max-min combinatorial optimization were known until recently. While the usual approach to prove Σ_k^p-completeness (or NP-completeness) is to formulate a new proof for each single problem, a recent paper by Grüne & Wulf [18], extending earlier ideas by Johannes [26] breaks with this approach. Instead, it is shown that there exists a large 'base list' of problems (called *SSP-NP-complete* problems in [18]), including many classic problems like independent set, vertex cover, knapsack, TSP, etc. Grüne and Wulf show that for each problem from the base list, some corresponding min-max version is Σ_2^p-complete, and some corresponding min-max-min version is Σ_3^p-complete. This approach has three main advantages: A1.) It uncovers a large number of previously unknown Σ_k^p-complete problems. A2.) It reveals the theoretically interesting insight, that for all these problems the Σ_k^p-completeness follows from essentially the same argument. A3.) It can simplify future proofs, since heuristically it seems to be true that for a new problem it is often easier to show that the nominal problem belongs to the list of SSP-NP-complete problems, than to find a Σ_k^p-completeness proof from scratch.

The main goal of the current paper is to extend the framework of Grüne & Wulf [18] also to the setting of recoverable optimization, since this setting was not considered in the original paper. It turns out that compared to [18] more complicated assumptions are needed. We introduce a set of sufficient conditions for some nominal problem, which imply that the recoverable robust version of that problem is Σ_3^p-complete.

[1] More specifically, we assume here that $\Sigma_3^p \neq$ NP. This is a similar assumption to the famous P $\neq$ NP assumption, and it is believed to be similarly unlikely. However, the true status of the conjecture is not known. More details can be found in [34].

Our results. We consider recoverable robust optimization for the following nominal problems: Sat, 3Sat, vertex cover, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median, independent set, clique, subset sum, knapsack, partition, scheduling, Hamiltonian path/cycle (directed/undirected), TSP, k-directed disjoint path ($k \geq 2$), and Steiner tree. In addition we consider the three most popular distance measures *dist* used in recoverable robust optimization: The κ-addition distance, the κ-deletion distance, and the Hamming distance. (A formal definition of these distance measures is given in Section 4.1)

We show that for every combination of the above problems with any of the three distance measures, the recoverable robust problem (with discrete budgeted uncertainty) is Σ_3^p-complete. More generally, we identify an abstract property of all our studied problems, and prove that this abstract property already implies that the recoverable robust problem becomes Σ_3^p-complete. This answers a question asked by Goerigk, Lendl and Wulf [16]. As a consequence, the advantages A1–A3 as explained above also apply to recoverable robust problems. We remark that Σ_3^p-completeness was already known in the case of clique/independent set, TSP or shortest path combined with the Hamming distance [16]. It was also already known for shortest path in combination with all three distance measures [23] and for several optimization problems [17] for so-called xor-dependencies and Γ-set scenarios. Hence our work is an extension of these results.

1.1 Related Work

Recoverable robustness concepts were analyzed for a variety of different standard optimization problems. Büsing [4] analyzed the recoverable shortest path problem with discrete budgeted uncertainty, in which adding at most κ elements to the second stage solution are allowed. This analysis was lately continued by Jackiewicz, Kasperski, and Zieliński [24] for several different graph classes and for interval budgeted uncertainty. Furthermore, recoverable robust knapsack was analyzed by Büsing, Koster, and Kutschka [8] for discrete scenarios and in which adding at most κ elements and deleting at most ℓ elements are allowed. Büsing, Koster, and Kutschka [7] additionally analyzed the Γ-scenario case (discrete budgeted uncertainty) while allowing at most ℓ elements to be deleted. This work was also continued by Büsing, Goderbauer, Koster, and Kutschka [6]. Further classical optimization problems that were studied in the recoverable robustness context are matching by Dourado, Meierling, Penso, Rautenbach, Protti, and de Almeida [11] and spanning tree under interval cost uncertainty by Hradovich, Kasperski, and Zieliński [21, 22]. Additionally, Lendl, Peis, and Timmermans [30] examined matroidal problems. One variant of independent set for recoverable robustness with a commitment property was analyzed by Hommelsheim, Megow, Muluk, and Peis [20]. Beside these problems, recoverable robust selection was studied by Kasperski and Zieliński [27], Chassein, Goerigk, Kasperski, and Zieliński [10], and Goerigk, Lendl, and Wulf [15]. Moreover, Lachmann, Lendl, and Woeginger [29] developed a linear time algorithm for the recoverable Γ-robust knapsack problem. The recoverable robust assignment problem was also investigated by Fischer, Hartmann, Lendl, and Woeginger [12].

Approximation results for recoverable robust problems were also achieved. For example, the recoverable traveling salesman problem was studied by Chassein and Goerigk [9], as well as Goerigk, Lendl, and Wulf [15], where the latter showed a 4-approximation algorithm. Beyond that Bold and Goerigk [3] presented a 2-approximation for the recoverable robust single machine scheduling problem under interval uncertainty.

Closely related to our work are the complexity studies by Goerigk, Lendl, and Wulf [16] who analyze the problems independent set, traveling salesman and vertex cover and obtain Σ_3^p-completeness for the three problems. At the same time, Grüne [17] introduced a gadget reduction framework to derive Σ_3^p-completeness for various different optimization problems by reusing already known reductions for recoverable robust problems with so-called xor-dependency scenarios and Γ-set scenarios. Additionally, Jackiewicz, Kasperski, and Zieliński [23] show that the shortest path problem with discrete budgeted interval uncertainty is Σ_3^p-complete.

2 Preliminaries

A *language* is a set $L \subseteq \{0,1\}^*$. A language L is contained in Σ_k^p iff there exists some polynomial-time computable function V (verifier), and $m_1, m_2, \ldots, m_k = |w|^{O(1)}$ such that for all $w \in \{0,1\}^*$

$$w \in L \;\Leftrightarrow\; \exists y_1 \in \{0,1\}^{m_1}\ \forall y_2 \in \{0,1\}^{m_2} \ldots\ Q y_k \in \{0,1\}^{m_k} : V(w, y_1, y_2, \ldots, y_k) = 1,$$

where $Q = \exists$, if k is odd, and $Q = \forall$, if k even. An introduction to the polynomial hierarchy and the classes Σ_k^p can be found in the book by Papadimitriou [32] or in the article by Jeroslow [25]. An introduction specifically in the context of bilevel optimization can be found in the article of Woeginger [34]. A *many-one-reduction* or *Karp-reduction* from a language L to a language L' is a map $f : \{0,1\}^* \to \{0,1\}^*$ such that $w \in L$ iff $f(w) \in L'$ for all $w \in \{0,1\}^*$. A language L is Σ_k^p-hard, if every $L' \in \Sigma_k^p$ can be reduced to L with a polynomial-time many-one reduction. If L is both Σ_k^p-hard and contained in Σ_k^p, it is Σ_k^p-complete.

For some cost function $c : U \to \mathbb{R}$, and some subset $U' \subseteq U$, we define the cost of the subset U' as $c(U') := \sum_{u \in U'} c(u)$. For a map $f : A \to B$ and some subset $A' \subseteq A$, we define the image of the subset A' as $f(A') = \{f(a) : a \in A'\}$. We defer the proofs marked with $(\star)$ to the full version of the paper [19].

3 Framework

Because this work builds upon the framework developed by Grüne and Wulf, it is essential to reintroduce its key concepts. A more comprehensive explanation of these concepts and their underlying motivation can be found in the original paper [18]. Grüne and Wulf begin by providing a precise definition of their primary focus: linear optimization problems, or in short LOP problems. An example of an LOP problem is vertex cover.

▶ **Definition 1** (Linear Optimization Problem, from [18]). *A linear optimization problem (or in short LOP problem)* Π *is a tuple* $(\mathcal{I}, \mathcal{U}, \mathcal{F}, d, t)$*, such that*

- $\mathcal{I} \subseteq \{0,1\}^*$ *is a language. We call* $\mathcal{I}$ *the set of instances of* Π*.*
- *For each instance* $I \in \mathcal{I}$*, there is some*
  - *a set* $\mathcal{U}(I)$ *with* $|\mathcal{U}(I)| = |I|^{O(1)}$ *that we call the universe associated to the instance* I*.*
  - *set* $\mathcal{F}(I) \subseteq 2^{\mathcal{U}(I)}$ *that we call the feasible solution set associated to the instance* I*.*
  - *function* $d^{(I)} : \mathcal{U}(I) \to \mathbb{Z}$ *mapping each universe element* e *to its costs* $d^{(I)}(e)$*.*
  - *threshold* $t^{(I)} \in \mathbb{Z}$*.*

For $I \in \mathcal{I}$*, we define the solution set* $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : d^{(I)}(S) \leq t^{(I)}\}$ *as the set of feasible solutions below the cost threshold. The instance* I *is a Yes-instance, if and only if* $\mathcal{S}(I) \neq \emptyset$*. We say that an LOP is contained in NP, if it can be checked in polynomial time in* $|I|$ *whether some proposed set* $F \subseteq \mathcal{U}(I)$ *is a solution, i.e.* $F \in \mathcal{S}(I)$*.*

Vertex Cover
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all vertex covers of G.
Solution set: The set of all vertex covers of G of size at most k.

It turns out that often the mathematical discussion is a lot clearer, when one omits the concepts $\mathcal{F}, d^{(I)}$, and $t^{(I)}$, since for the abstract proof of the theorems only $\mathcal{I}, \mathcal{U}, \mathcal{S}$ are important. This leads to the following abstraction from the concept of an LOP problem:

▶ **Definition 2** (Subset Search Problem (SSP), from [18]). *A subset search problem (or short SSP problem)* Π *is a tuple* $(\mathcal{I}, \mathcal{U}, \mathcal{S})$*, such that*

- $\mathcal{I} \subseteq \{0,1\}^*$ *is a language. We call* $\mathcal{I}$ *the set of instances of* Π*.*
- *For each instance* $I \in \mathcal{I}$*, there is some set* $\mathcal{U}(I)$ *with* $|\mathcal{U}(I)| = |I|^{O(1)}$ *which we call the universe associated to the instance* I*.*
- *For each instance* $I \in \mathcal{I}$*, there is some (potentially empty) set* $\mathcal{S}(I) \subseteq 2^{\mathcal{U}(I)}$ *which we call the solution set associated to the instance* I*.*

An instance of an SSP problem is referred to as a yes-instance if $\mathcal{S}(I) \neq \emptyset$. Any LOP problem is transformable into an SSP problem by defining $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : d^{(I)}(S) \leq t^{(I)}\}$. We refer to this as the *SSP problem derived from an LOP problem*. There are problems that are more naturally represented as SSP problems, rather than as LOP problems. For instance, the problem 3Satisfiability can be derived as an SSP problem as follows.

3Satisfiability
Instances: Literal set $L = \{\ell_1, \ldots, \ell_n\} \cup \{\overline{\ell}_1, \ldots, \overline{\ell}_n\}$, clause set $C = \{C_1, \ldots, C_m\}$ such that $C_j \in L^3$ for all $j \in \{1, \ldots, m\}$.
Universe: $L =: \mathcal{U}$.
Solution set: The set of all subsets $L' \subseteq \mathcal{U}$ of the literals such that for all $i \in \{1, \ldots, n\}$ we have $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$, and such that $|L' \cap C_j| \geq 1$ for all clauses $C_j \in C$.

Grüne and Wulf introduce a novel type of reduction, termed SSP reduction. In essence, a typical polynomial-time reduction from a problem Π to another problem Π' possesses the SSP property if it includes an additional injective mapping $f : \mathcal{U} \to \mathcal{U}'$ that embeds the universe $\mathcal{U}$ elements of Π into the universe $\mathcal{U}'$ of Π'. This embedding allows Π to be viewed as a 'subinstance' of Π' while preserving the topology of solutions within the subset induced by the image of f. More precisely, we interpret W as the subinstance of Π within the instance of Π' and we require the following two conditions to be met:

1. For every solution S of Π, the set $f(S)$ is a partial solution of Π' and can be extended to a full solution using elements that are not in W.
2. For every solution S' of Π', the set $f^{-1}(S' \cap W)$ is a solution of Π.

These two conditions are encapsulated in the single equation (1). We indicate that such a reduction exists by $\Pi \leq_{\text{SSP}} \Pi'$. For a more intuitive explanation and an example that shows 3Sat $\leq_{\text{SSP}}$ Vertex Cover, we direct readers to [18].

▶ **Definition 3** (SSP Reduction, from [18]). *Let* $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ *and* $\Pi' = (\mathcal{I}', \mathcal{U}', \mathcal{S}')$ *be two SSP problems. We say that there is an SSP reduction from* Π *to* Π'*, and write* $\Pi \leq_{SSP} \Pi'$*, if*

- *There exists a function* $g : \{0,1\}^* \to \{0,1\}^*$ *computable in polynomial time in the input size* $|I|$*, such that* I *is a Yes-instance iff* $g(I)$ *is a Yes-instance (i.e.* $\mathcal{S}(I) \neq \emptyset$ *iff* $\mathcal{S}'(g(I)) \neq \emptyset$*).*

- *There exist functions $(f_I)_{I\in\mathcal{I}}$ computable in polynomial time in $|I|$ such that for all instances $I \in \mathcal{I}$, we have that $f_I : \mathcal{U}(I) \to \mathcal{U}'(g(I))$ is an injective function mapping from the universe of the instance I to the universe of the instance $g(I)$ such that*

$$\{f_I(S) : S \in \mathcal{S}(I)\} = \{S' \cap f_I(\mathcal{U}(I)) : S' \in \mathcal{S}'(g(I))\}. \tag{1}$$

We remark that the map f_I is required to be uniformly computable in polynomial time in $|I|$, i.e. there exists a fixed polynomial p such that $f_I(u)$ can be computed in time $p(|I|)$ for all $I \in \mathcal{I}$ and $u \in \mathcal{U}(I)$. In [18], it is demonstrated that SSP reductions are transitive, that is $\Pi_1 \leq_{\text{SSP}} \Pi_2$ and $\Pi_2 \leq_{\text{SSP}} \Pi_3$ imply $\Pi_1 \leq_{\text{SSP}} \Pi_3$. We denote the class of SSP-NP-complete problems by SSP-NPc. It is defined as the set of all SSP problems Π that are polynomial-time verifiable (containment) and there is an SSP reduction from Satisfiability, i.e. Satisfiability $\leq_{\text{SSP}} \Pi$ (hardness). The key insight in [18] is that many classical problems fall within the class SSP-NPc. Additionally, this insight can be leveraged to show that their corresponding min-max version are Σ_2^p-complete.

4 Recoverable Robust Problems

In this section, we consider recoverable robust optimization problems. We show that the recoverable robust optimization problem is Σ_3^p-complete for the following nominal problems: satisfiability, 3-satisfiability, vertex cover, dominating set, set cover, hitting set, feedback vertex set, feedback arc set, uncapacitated facility location, p-center, p-median, independent set, clique, subset sum, knapsack, partition, scheduling, (un)directed Hamiltonian path, (un)directed Hamiltonian cycle, traveling salesman, two directed disjoint path, k directed disjoint path, and Steiner tree.

Recoverable robust optimization problems are defined as follows: We are given some instance I of a linear optimization problem (like the shortest path problem, the traveling salesman problem, etc.), and are faced with an uncertain future. The goal is to find a feasible solution S_1 for the first stage (called here-and-now decision) such that after the reveal of the uncertainty we can find a feasible solution S_2 in the second stage (called wait-and-see decision) such that S_1 and S_2 are not far away from each other according to some distance measure. Formally we require $\text{dist}(S_1, S_2) \leq \kappa$, where $\text{dist}(S_1, S_2)$ is some abstract distance function (for example the Hamming distance $|S_1 \triangle S_2|$). The cost of the solution is given by $c_1(S_1) + c_2(S_2)$. Here, c_1 is a fixed cost function not affected by uncertainty, called the setup costs, and c_2 is affected by uncertainty. More specifically, we assume that $c_2 \in C_\Gamma$, that is, c_2 is affected by discrete budgeted uncertainty[2] C_Γ. Precisely, given $\Gamma \in \mathbb{N}$ and upper and lower bounds $\underline{c}(u) \leq \overline{c}(u)$ for all elements in the universe, the set C_Γ contains all cost functions such that at most Γ elements $u \in \mathcal{U}(I)$ have costs of $\overline{c}(u)$, while all other have $\underline{c}(u)$:

$$C_\Gamma := \{c_2 \mid \forall u \in \mathcal{U}(I) : c_2(u) = \underline{c}(u) + \delta_u(\overline{c}(u) - \underline{c}(u)),\ \delta_u \in \{0,1\},\ \sum_{u\in\mathcal{U}(I)} \delta_u \leq \Gamma\}$$

This leads to the following abstract definition of a recoverable robust problem. We remark that the formal definition in some sense "disregards" the cost functions d and the cost threshold t of the original LOP problem. This is intentional because in the new instance of the recoverable robust problem, it becomes necessary to replace the old function d and the old threshold t by new ones. However, these concepts are necessary to correctly understand the proofs.

[2] Usually in literature, this set is denoted by $\mathcal{U}_\Gamma$, however, we have already used the letter $\mathcal{U}$ in this paper.

▶ **Definition 4** (Recoverable Robust Problem). *Let an LOP problem* $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{F}, d, t)$ *and a distance measure* $\mathrm{dist} : 2^{\mathcal{U}} \times 2^{\mathcal{U}} \to \mathbb{R}_{\geq 0}$ *be given. The recoverable robust problem associated to* Π *and* dist *is denoted by RR-*Π *and defined as follows: The input is an instance* $I \in \mathcal{I}$ *together with three cost functions* $c_1 : \mathcal{U}(I) \to \mathbb{Z}$, $\underline{c} : \mathcal{U}(I) \to \mathbb{Z}$ *and* $\overline{c} : \mathcal{U}(I) \to \mathbb{Z}$ *and a cost threshold* $t_{RR} \in \mathbb{Z}$*, an uncertainty parameter* $\Gamma \in \mathbb{N}_0$ *and a recoverability parameter* $\kappa \in \mathbb{N}_0$*. The question is whether*

$$\min_{S_1 \in \mathcal{F}(I)} \max_{c_2 \in C_\Gamma} \min_{\substack{S_2 \in \mathcal{F}(I) \\ \mathrm{dist}(S_1, S_2) \leq \kappa}} c_1(S_1) + c_2(S_2) \leq t_{RR}. \tag{2}$$

Example. Let $\Pi = \mathrm{TSP}$. TSP is encoded as LOP problem the following way: An instance is given by $I = (G, d, t)$, where $G = (V, E)$ is a complete undirected graph, $d : E \to \mathbb{N}_0$ are the edge costs and t is the cost threshold. The decision problem of TSP asks if there is a tour $T \subseteq E$ with cost $d(T) \leq t$ visiting all vertices from V exactly once. The universe is $\mathcal{U}(I) = E$. The set $\mathcal{F}(I) \subseteq 2^E$ is the set of all feasible tours (including those of cost greater than t). The set $\mathcal{S}(I)$ is the set of all tours of cost at most t. To turn the TSP into a recoverable robust problem, we "forget" about the cost function d and the threshold t. Given $c_1, \underline{c}, \overline{c}, \Gamma, \kappa, t_{RR}$, the decision problem associated to the recoverable robust TSP (under some fixed distance function dist) is to decide whether Equation (2) holds.

4.1 Distance Measures

In the literature, multiple different distance measures are used. All these distance functions share the abstract properties specified in Definition 5.

- the κ-*addition* or simply κ-*distance measure* is used in [5, 21]: $\mathrm{dist}(A_1, A_2) = |A_2 \setminus A_1|$
- the κ-*deletion distance* is used in [7]: $\mathrm{dist}(A_1, A_2) = |A_1 \setminus A_2|$
- the *Hamming distance* measure is used in [11, 16, 17]: $\mathrm{dist}(A_1, A_2) = |A_1 \triangle A_2|$

▶ **Definition 5** (Distance Measure). *A distance measure is a family* (dist_U)*, where* U *ranges over all finite sets, such that for all finite sets* U, U'*, and all subsets* $A_1, A_2 \subseteq U$*, the map* $\mathrm{dist}_U : 2^U \times 2^U \to \mathbb{R}_{\geq 0}$ *has the following properties:*

- *computable in polynomial time in* $|U|$
- *invariant on injective mappings* $f : U \to U'$*, i.e.* $\mathrm{dist}_U(A_1, A_2) = \mathrm{dist}_{U'}(f(A_1), f(A_2))$*,*
- *invariant on union, i.e.* $\mathrm{dist}_U(A_1, A_2) = \mathrm{dist}_U(A_1 \cup \{x\}, A_2 \cup \{x\})$ *for* $x \in U \setminus (A_1 \cup A_2)$*.*
- $\mathrm{dist}_U(A_1, A_1) = 0$

If U *is clear from the context, we omit the subscript.*

One can easily verify that all of the distance measures from above fulfill these criteria.

4.2 Containment in Σ_3^p

Recoverable robust problems can also be understood as a three-stage two-player game of an $\exists$-player playing against an adversary (the $\forall$-player). The $\exists$-player controls both min operators and is able to choose the solutions S_1 and S_2. On the other hand, the adversary controls the max operator and is able to choose the uncertainty scenario, i.e. the cost function c_2 from the set C_Γ. Thus, it is possible to reformulate the question to

$$\exists S_1 \subseteq \mathcal{U}(I) : \forall c_2 \in C_\Gamma : \exists S_2 \subseteq \mathcal{U}(I) : S_1, S_2 \in \mathcal{F}(I), c_1(S_1) + c_2(S_2) \leq t \text{ and } \mathrm{dist}(S_1, S_2) \leq \kappa.$$

With the game-theoretical perspective, it is intuitive to see the containment in Σ_3^p for all problems that have a polynomial-time verifier.

▶ **Theorem 6** ($\star$). *If* $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ *is an LOP problem in NP, then RR-*Π *is in* Σ_3^P*.*

4.3 Combinatorial Recoverable Robust Problems

To show the hardness of recoverable robust problems, we introduce a new problem which we call the *combinatorial version* of recoverable robust problems. There are two differences to Definition 4: In this new problem, the cost function is replaced by a set B of so-called *blockable elements* (this can be interpreted as the case where cost coefficients $\underline{c}, \overline{c}$ are restricted to come from $\{0, \infty\}$). Furthermore, in this new definition we replace $\mathcal{F}(I)$ by $\mathcal{S}(I)$. As we show later, the hardness of the new combinatorial version also implies the hardness of the cost version. Finally, as explained above, Definition 7 applies to all SSP problems.

▶ **Definition 7** (Combinatorial Recoverable Robust Problem). *Let an SSP problem* $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ *be given. The combinatorial recoverable robust problem associated to* Π *is denoted by* Comb. RR-Π *and defined as follows: The input is an instance* $I \in \mathcal{I}$ *together with a set* $B \subseteq \mathcal{U}(I)$*, an uncertainty parameter* $\Gamma \in \mathbb{N}_0$*, and a recoverability parameter* $\kappa \in \mathbb{N}_0$*. The question is whether*

$$\exists S_1 \subseteq \mathcal{U}(I) : \forall B' \subseteq B \text{ with } |B'| \leq \Gamma : \exists S_2 \subseteq \mathcal{U}(I) : \\ S_1, S_2 \in \mathcal{S}(I), S_2 \cap B' = \emptyset \text{ and } \operatorname{dist}(S_1, S_2) \leq \kappa.$$

5 An SSP Framework for Recoverable Robust Problems

In this section, we prove our main result, i.e. we introduce and aprove a sufficient condition for nominal problems such that the corresponding recoverable robust problem is Σ_3^p-hard. We first explain the rough idea.

We want to show that recoverable problems are Σ_3^p-hard, so we have to choose some Σ_3^p-hard problem from which we start our reduction. The canonical Σ_3^p-complete problem is $\exists\forall\exists$-Satisfiability [33]. Intuitively, this satisfiability problem can be understood as a game between Alice and Bob. Alice first chooses a variable assignment on the variables of X, then Bob chooses an assignment of the variables Y, and then again Alice selects an assignment on the variables Z, where Alice wishes to satisfy formula $\varphi(X, Y, Z)$, and Bob wishes the opposite. On the other hand, in our target problem RR-Π, any solution consists of two solutions of the nominal problem Π, the first stage solution S_1 and the second stage solution S_2. The main challenge of our reduction is, that we have to model three variable sets X, Y, Z of the formula $\exists X \forall Y \exists Z \varphi(X, Y, Z)$ in the order of quantification into the problem RR-Π. Accordingly, both solutions S_1, S_2 need to include an assignment to all variables from X, Y, Z that satisfy $\varphi(X, Y, Z)$ in agreement with the nominal Sat problem. However, note that there is a difference in the solution structure of both problems, which we need to address. If we model Alice's decision on the X-variables in $\exists\forall\exists$-Sat in first stage solution S_1 of RR-Π, we need to make sure that Alice is not able to reassign the the X-variables in the second stage solution S_2 of RR-Π. Otherwise, Alice does not adhere to the order of quantification.

This problem can be circumvented by adding an additional property to the SSP reduction. We demand that for a given set of literals L_b the distance of two solutions S_1 and S_2 is small if and only if the partial solution of S_1 and S_2 restricted to L_b is the same. One can interpret this construction as a gadget that *blows up* the literals of L_b in comparison to all the other literals. Accordingly, we call the literals of L_b *blow-up literals* and the corresponding reduction *blow-up SSP reduction* with a *blow-up factor* β_I. Then, we can set $L_b = L_X$, where L_X is the set of literals corresponding to variables of X. Therefore, Alice is not able to change the assignment of the variable set X from the first stage solution S_1 to the second stage solution S_2 and thus has to adhere to the order of quantification. Then, it is also

possible to reuse the injective correspondence function f of the SSP reduction to set the blockable elements B and to show the correctness of the reduction by using the elementary correspondence between the SAT solution and the solution of Π. Formally, we define *blow-up SSP reductions* as follows.

▶ **Definition 8** (Blow-Up SSP Reduction). *Let* dist *be a distance measure. Let* $3\text{SAT} = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ *with* $\mathcal{U} = L = \{\ell_1, \dots, \ell_n\} \cup \{\overline{\ell}_1, \dots, \overline{\ell}_n\}$ *and* $\Pi' = (\mathcal{I}', \mathcal{U}', \mathcal{S}')$ *be two SSP problems. Then,* 3SAT *is SSP blow-up reducible to* Π' *with respect to* $\text{dist}(\cdot,\cdot)$ *if for all sets* $L_b \subseteq L$ *fulfilling* $\ell_i \in L_b \leftrightarrow \overline{\ell}_i \in L_b$ *there exists an SSP reduction* (g, f_I) *with the following property: There is a polynomial time computable blow-up factor* $\beta_I \in \mathbb{N}$ *corresponding to each instance* I *such that for all solutions* $S_1, S_2 \in \mathcal{S}'(g(I))$ *of* $g(I)$*:*

$$f(L_b) \cap S_1 = f(L_b) \cap S_2 \Leftrightarrow \text{dist}_{\mathcal{U}'(g(I))}(S_1, S_2) \leq \beta_I.$$

A concrete example of a blow-up reduction is given in Section 5.1.

Remark 1. Note that we define blow-up reductions to start at 3SAT. Because 3SAT is one of the 'first' NP-complete problems, many reductions start at 3SAT or have a short reduction chain from 3SAT. Thus, it is convenient to reuse these reductions or respectively reduction chains to construct blow-up reductions. As we will show later in Sections 5.1 and 6 this is not really a restriction.

Remark 2. Blow-up SSP reductions are not transitive. (This is because there is no restriction on newly introduced elements in Π' and how they behave corresponding to solutions in Π'. Therefore, the distance between solutions in Π' cannot be related to the distance between the corresponding solutions in Π.) We will tackle this problem in Section 6, in which we will show that we only need to find a blow-up reduction once at the beginning of a reduction chain and we can use so-called blow-up preserving reductions to append further problems to the reduction chain starting at 3SAT. This blow-up preserving reduction is typically much easier to find than a new blow-up reduction.

With everything in place, we show that blow-up SSP reductions enable us to show the Σ_3^p-hardness of recoverable robust problems as long as the nominal problem Π is SSP-NP-hard. In particular, our main theorem now states, that our newly introduced blow-up reductions indeed are a sufficient criterion for Σ_3^p-hardness.

▶ **Theorem 9.** *For all SSP-NP-hard problems* Π *that are blow-up SSP reducible from* 3SAT*, the combinatorial recoverable robust variant* COMB. *RR-*Π *is* Σ_3^p*-hard.*

Proof. For the proof of the main theorem, we require some definitions. Let $X = \{x_1, \dots, x_n\}$ be a set of binary variables and $L := \{x_1, \dots, x_n\} \cup \{\overline{x}_1, \dots, \overline{x}_n\} = X \cup \overline{X}$ be the set of corresponding literals. An *assignment* is a subset $A \subseteq X \cup \overline{X}$ such that $|A \cap \{x_i, \overline{x}_i\}| = 1$ for all $i = 1, \dots, n$. We say that the assignment A assigns the value 1 to variable x_i, if $x_i \in A$, and A assigns 0 to x_i, if $\overline{x}_i \in A$. We remark that this notation for an assignment is non-standard, but it turns out to be convenient in the context of our framework. A SAT-formula φ is in 3CNF, if it is a conjunction of clauses, and every clause has exactly three literals. We denote by $\varphi(A) \in \{0, 1\}$ the evaluation of the formula φ under the assignment A. In the following, we often consider the case, where the set of variables is partitioned into three disjoint sets X, Y, Z, and denote this case by writing $\varphi(X, Y, Z)$.

For the Σ_3^p-hardness proof, we require a known Σ_3^p-complete problem to reduce from. It turns out that instead of reducing from the classic problem $\exists\forall\exists$-SATISFIABILITY, it is more convenient to base our proof on ROBUST ADJUSTABLE SAT with budgeted uncertainty (in short, R-ADJ-SAT), introduced by Goerigk, Lendl and Wulf [16], which we reformulate to adhere to the notation of this paper:

Problem R-Adj-Sat
Instance: A Sat-formula $\varphi(X, Y, Z)$ in 3CNF. A partition of the set of variables into three disjoint parts $X \cup Y \cup Z$ with $|X| = |Y| = |Z|$. An integer $\Gamma \geq 0$.
Question: Is there an assignment $A_X \subseteq X \cup \overline{X}$ such that for all subsets $Y' \subseteq Y$ of size $|Y'| \leq \Gamma$, there exist assignments $A_Y \subseteq Y \cup \overline{Y}$ and $A_Z \subseteq Z \cup \overline{Z}$ such that $A_Y \cap Y' = \emptyset$, i.e. set all variables in Y' to 0, and $\varphi(A_X, A_Y, A_Z) = 1$?

The problem R-Adj-Sat is best understood as a game between Alice and Bob: Alice wants to satisfy the formula, Bob wants to achieve the opposite. Alice initially selects an assignment A_X of the variables in X. Afterwards, Bob is allowed to select a blocker $Y' \subseteq Y$ of size at most Γ and force all these variables to be "0". Finally, Alice is allowed to assign values to all variables in $Y \cup Z$ that have not yet been assigned. In [16] it is shown that R-Adj-Sat is Σ_3^p-complete. In order to provide additional insight for the reader, we quickly sketch the main argument in [16]: The idea is to reduce from the classic problem $\exists\forall\exists$-Satisfiability. Let $\exists A_X \forall A_Y \exists A_Z \varphi(A_X, A_Y, A_Z)$ be an instance of this problem. How do we transform it into a R-Adj-Sat problem? It is intuitive, that in such a reduction, Alice's rule of choosing assignments A_X and A_Z should stay roughly the same. However, how do we express the choice of A_Y in terms of a blocker Y' that is forced to 0? The idea is to split the variable $y_i \in Y$ into two new variables y_i^t, y_i^f for every $i = 1, \ldots, n$. We say that Bob plays honestly if for all $i = 1, \ldots, n$ we have $|\{y_i^t, y_i^f\} \cap Y'| = 1$. Such an honest choice of Y' naturally encodes an assignment A_Y. It turns out that, using the pigeonhole principle, and the budget constraint $|Y'| \leq \Gamma$, one can enrich the formula φ with a "cheat-detection-gadget". This gadget can be used by Alice to win the game trivially if and only if Bob cheats. Hence the modified game of R-Adj-Sat is equivalent to the original instance of $\exists\forall\exists$-Satisfiability.

We prove Σ_3^p-hardness by providing a reduction from R-Adj-Sat to Comb. RR-Π. This reduction will crucially rely on the blow-up SSP reduction from 3Sat to Π, which we assumed to exist. More formally, let $\Pi = (\mathcal{I}_\Pi, \mathcal{U}_\Pi, \mathcal{S}_\Pi)$ be an SSP problem, such that there is a blow-up reduction from 3Sat to Π defined by the mappings g, $(f_I)_{I \in \mathcal{I}_{\text{SAT}}}$, and blow-up factor $(\beta_I)_{I \in \mathcal{I}_{\text{SAT}}}$. Recall that by the definition of a blow-up reduction this means the following: Let the term $\mathcal{I}_{\text{SAT}}$ denote the set of all possible 3Sat instances, i.e. the set of all 3CNF formulas. For each fixed 3Sat instance $I_{\text{SAT}} \in \mathcal{I}_{\text{SAT}}$, its universe $\mathcal{U}(I_{\text{SAT}})$ is the set of positive and negative literals of variables appearing in that formula. By assumption, for every fixed 3Sat instance $I_{\text{SAT}} \in \mathcal{I}_{\text{SAT}}$ and every set $L_b \subseteq \mathcal{U}(I_{\text{SAT}})$ (with $\ell \in L_b$ iff $\overline{\ell} \in L_b$) there exists an equivalent instance $I_\Pi = g(I_{\text{SAT}})$ of Π. Furthermore, associated to each fixed 3Sat instance $I_{\text{SAT}} \in \mathcal{I}_{\text{SAT}}$ we have a function $f_{I_{\text{SAT}}}$, which describes in which way the universe of I_{SAT} can be injectively embedded into the universe of the equivalent instance I_Π such that the SSP property is true. Finally, there is a tight correspondence between solutions of I_Π having distance at most $\beta_{I_{\text{SAT}}}$, and (the pre-image of) the solutions agreeing on the set L_b. For brevity of notation, in the following we often write $\mathcal{U}(I_\Pi)$ instead of $\mathcal{U}_\Pi(I_\Pi)$ and $\mathcal{S}(I_\Pi)$ instead of $\mathcal{S}_\Pi(I_\Pi)$ to denote the solution set/universe associated to instance I_Π, if the correct subscript is clear from the context.

We now turn our attention to the Σ_3^p-hardness proof. As explained above, the problem R-Adj-Sat is Σ_3^p-complete. Let an R-Adj-Sat instance I_{RAS} be given. Our goal is to transform this instance into a new instance $I_{\text{RR}\Pi}$ of Comb. RR-Π such that

I_{RAS} is a Yes-instance of R-Adj-Sat $\Leftrightarrow$ $I_{\text{RR}\Pi}$ is a Yes-instance of Comb. RR-Π.

Clearly if we can achieve this goal we are done. By definition of the problem R-Adj-Sat, the instance I_{RAS} consists of the following parts:

$I_{\text{RAS}} = (\varphi(X, Y, Z), \Gamma)$,

where $\varphi \in \mathcal{I}_{\text{SAT}}$ is a 3CNF-formula, whose variables are partitioned into three parts X, Y, Z of equal size $n := |X| = |Y| = |Z|$, and $\Gamma \in \mathbb{N}_0$. Let the corresponding literal sets X, Y, Z be denoted by $L_X := X \cup \overline{X}$, $L_Y := Y \cup \overline{Y}$, and $L_Z := Z \cup \overline{Z}$. Note that the universe associated to the 3SAT-instance φ is $L_X \cup L_Y \cup L_Z$. Let us denote this fact by writing $\mathcal{U}(\varphi) = L_X \cup L_Y \cup L_Z$.

Given the instance I_{RAS} as input, we describe in the following how to construct the instance $I_{\text{RR}\Pi}$ of COMB. RR-Π, which consists of the following parts:

$$I_{\text{RR}\Pi} = (I_\Pi, B, \Gamma', \kappa).$$

Here, I_Π is an instance of the nominal problem Π, $B \subseteq \mathcal{U}(I_\Pi)$ is the set of blockable elements, $\Gamma' \in \mathbb{N}_0$ is the uncertainty budget, and $\kappa \in \mathbb{N}_0$ is the recoverability parameter.

Before we give a formal description of I_Π, B, Γ', and κ, we explain the rough idea: In particular, what is the right choice for the instance I_Π? The answer is provided by the blow-up reduction. Note that this reduction can take in any 3CNF formula and a subset L_b of the literals (with the property that $\ell \in L_b$ iff $\overline{\ell} \in L_b$ for all literals ℓ) and produce an instance I of Π. Note that the set L_b gets "blown up" by the reduction. We claim that $L_b = L_X$ is a good choice. Indeed, consider the following formal definition of $I_{\text{RR}\Pi}$.

Description of the Instance $I_{\text{RR}\Pi}$. Let g, $(f_\varphi)_{\varphi \in \mathcal{I}_{\text{SAT}}}$, and $(\beta_\varphi)_{\varphi \in \mathcal{I}_{\text{SAT}}}$ be the objects describing the blow-up reduction from 3SAT to Π for the fixed choice of $L_b := L_X$. We let $I_\Pi = g(\varphi)$ be the instance produced by the blow-up reduction applied to the formula φ and the literal set $L_b = L_X$. We remark that the following holds by the definition of the blow-up reduction: A blow-up reduction in particular is also an SSP reduction. Hence the function $f_\varphi : \mathcal{U}(\varphi) \to \mathcal{U}(I_\Pi)$ maps the literals of φ injectively to elements of the new universe $\mathcal{U}(I_\Pi)$. We can hence define the set of blockable elements

$$B := f_\varphi(Y),$$

where $Y \subseteq \mathcal{U}(\varphi)$ describes the positive literals in L_Y (recall that $L_Y = Y \cup \overline{Y}$). Furthermore, note that by definition of a blow-up reduction, there exists some $\beta_\varphi \in \mathbb{N}$, computable in poly-time, with the property that for all $S_1, S_2 \in \mathcal{S}(I_\Pi)$:

$$\text{dist}(S_1, S_2) \leq \beta_\varphi \Leftrightarrow S_1 \cap f_\varphi(L_X) = S_2 \cap f_\varphi(L_X).$$

In other words, the new instance I_Π has the property that two solutions of it have small distance if and only if they agree on $f_\varphi(L_X)$, i.e. they agree on those universe elements of $\mathcal{U}(I_\Pi)$ that correspond to L_X. We finally define

$$\kappa := \beta_\varphi \text{ and } \Gamma' := \Gamma.$$

This completes the description of the instance $I_{\text{RR}\Pi} = (I, B, \Gamma', \kappa)$.

Correctness We start the correctness proof by showing

$$I_{\text{RAS}} \text{ is a Yes-instance} \Rightarrow I_{\text{RR}\Pi} = (I_\Pi, B, \Gamma, \kappa) \text{ is a Yes-instance.}$$

In this case, let $A_X \subseteq X \cup \overline{X}$ be an assignment of the variables X with the property that

$$\forall Y' \subseteq Y, |Y'| \leq \Gamma : \ \exists A_Y, A_Z : \ A_Y \cap Y' = \emptyset \text{ and } \varphi(A_X, A_Y, A_Z) = 1.$$

Such an A_X exists by assumption that I_{RAS} is a Yes-instance. Now, let $Y' \subseteq Y$ be an arbitrary subset of Y with $|Y'| \leq \Gamma$. Then, it is possible to choose assignments A_Y, A_Z

of Y, Z such that $A_Y \cap Y = \emptyset$ and $\varphi(A_X, A_Y, A_Z) = 1$. We consider such an assignment $A_1 := A_X \cup A_Y \cup A_Z$. Note that if we interpret φ as 3SAT instance, $A_1 \subseteq \mathcal{U}(\varphi)$ and even $A_1 \in \mathcal{S}(\varphi)$, due to $\varphi(A_1) = 1$. Since the blow-up reduction is in particular an SSP reduction, we can make use of the central property of SSP reductions. The property implies that the 3SAT solution A_1 can be "lifted" to a solution of Π. More precisely, there exists a solution $S_1 \in \mathcal{S}(I_\Pi)$, such that

$$S_1 \cap f_\varphi(\mathcal{U}(\varphi)) = f_\varphi(A_1).$$

(Intuitively, the solution $S_1 \in \mathcal{S}(I_\Pi)$ of the nominal problem instance I_Π corresponds to the solution $A_1 \in \mathcal{S}(\varphi)$ of the 3SAT instance φ when restricted to the injective embedded "sub-instance" of 3SAT inside Π.)

We claim that S_1 is a solution for $I_{\mathrm{RR}\Pi}$. To prove this claim we have to show that for all blockers $B' \subseteq B$ with $|B'| \leq \Gamma' = \Gamma$, there exists a solution $S_2 \in \mathcal{S}(I_\Pi)$ with $S_2 \cap B' = \emptyset$ and $\mathrm{dist}(S_1, S_2) \leq \kappa$. Indeed, such a solution S_2 exists for all choices of blockers B'. This can be seen by applying the following construction: Repeat exactly the same construction as for S_1, except that the set Y' is not chosen arbitrarily. Instead, choose Y' by letting $Y' := f_\varphi^{-1}(B')$. Note that by the definition of B, the set Y' is well-defined, i.e. $Y' \subseteq Y$, $|Y'| \leq \Gamma$, and $f_\varphi(Y') = B'$. Repeating the same construction, assignment A_X stays the same, but we obtain different assignments A'_Y, A'_Z with $A'_Y \cap Y' = \emptyset$ and $\varphi(A_X, A'_Y, A'_Z) = 1$. We let $A_2 := A_X \cup A'_Y \cup A'_Z$. Again, by the SSP property there exists a solution $S_2 \in \mathcal{S}(I_\Pi)$ such that $S_2 \cap f_\varphi(\mathcal{U}(\varphi)) = f_\varphi(A_2)$. Therefore we have

$$\emptyset = A_2 \cap Y' = f_\varphi(A_2) \cap f_\varphi(Y') = (S_2 \cap f_\varphi(\mathcal{U}(\varphi))) \cap B' = S_2 \cap B'.$$

Note that A_X is the same in both constructions. Therefore $S_1 \cap f_\varphi(L_X) = S_2 \cap f_\varphi(L_X)$, and hence by the definition of a blow-up reduction, we have $\mathrm{dist}(S_1, S_2) \leq \beta_\varphi = \kappa$. In total, we have shown that S_1 is a solution of Π such that for every blocker $B' \subseteq B$, with $|B'| \leq \Gamma$, there exists a good solution S_2. This shows that $I_{\mathrm{RR}\Pi}$ is yes-instance.
It remains to consider the reverse direction:

$$I_{\mathrm{RR}\Pi} = (I, B, \Gamma', \kappa) \text{ is a Yes-instance} \Rightarrow I_{\mathrm{RAS}} \text{ is a Yes-instance}$$

The strategy is very similar, with the difference that we use the SSP property and the blow-up property in the reverse direction. Consider some solution $S_1 \in \mathcal{S}(I_\Pi)$ which satisfies

$$\forall B' \subseteq B, |B'| \leq \Gamma' : \exists S_2 \in \mathcal{S}(I_\Pi), S_2 \cap B' = \emptyset : \mathrm{dist}(S_1, S_2) \leq \kappa.$$

We have to show that there exists an assignment $A_X \subseteq X \cup \overline{X}$ such that for all subsets $Y' \subseteq Y$ there are assignments A_Y, A_Z with $A_Y \cap Y = \emptyset$ and $\varphi(A_X, A_Y, A_Z) = 1$. Indeed, to define A_X, consider solution S_1. By the SSP property, since $S_1 \in \mathcal{S}(I_\Pi)$, the set $f_\varphi^{-1}(S_1) \subseteq \mathcal{U}(\varphi)$ is a set of literals which satisfies φ. We let A_X be the restriction of that satisfying assignment to the variables in X. More formally, we let $A_X := f_\varphi^{-1}(S_1) \cap L_X$.

We claim that this assignment A_X proves that I_{RAS} is a yes-instance. Indeed, let some set $Y' \subseteq Y$, with $|Y'| \leq \Gamma$, be given. We define the blocker to be $B' := f_\varphi(Y')$. Observe that $B' \subseteq f_\varphi(Y) = B$, and $|B'| \leq \Gamma$. Therefore there exists $S_2 \in \mathcal{S}(I)$ such that $S_2 \cap B' = \emptyset$ and $\mathrm{dist}(S_1, S_2) \leq \kappa$. We can again interpret the solution S_2 as an assignment $A_2 := f_\varphi^{-1}(S_2)$. By the SSP property, this assignment satisfies φ, that is $A_2 \in \mathcal{S}(\varphi)$. Since $\mathrm{dist}(S_1, S_2) \leq \kappa = \beta_\varphi$, we have $S_1 \cap f_\varphi(L_X) = S_2 \cap f_\varphi(L_X)$ by the property of a blow-up reduction. This in turn implies $A_2 \cap (X \cup \overline{X}) = A_X$. Finally, we have $\emptyset = S_2 \cap B' = f_\varphi^{-1}(S_2) \cap f_\varphi^{-1}(B') = A_2 \cap Y'$. This shows that I_{RAS} is a yes-instance, and concludes the proof.

Polynomial Time. The instance $I_{\mathrm{RR}\Pi} = (I_\Pi, B, \Gamma', \kappa)$ can indeed be constructed in polynomial time. The nominal instance I_Π can be computed polynomially, since the map g in the SSP reduction can be computed polynomially. The set B can be computed polynomially, since the map f_φ in the SSP reduction can be computed polynomially. The number $\kappa = \beta_\varphi$ can be computed polynomially by definition of a blow-up SSP reduction. The number Γ remains the same. ◀

We have shown that the combinatorial version is indeed Σ_3^p-hard as long as the nominal SSP problem Π is SSP-NP-hard with a blow-up SSP reduction from Satisifiability. This however does not directly imply that the linear optimization version RR-Π is also Σ_3^p-hard. We remedy this problem with a short reduction that simulates the set of blockable elements in Comb. RR-Π with the cost functions in RR-Π that results in the following theorem.

▶ **Theorem 10** ($\star$). *For all LOP problems $\Pi \in NP$ with the property that the SSP problem derived from it is blow-up SSP reducible from 3Sat, the recoverable robust version RR-Π is Σ_3^p-complete.*

5.1 Blow-Up SSP Reductions for Various Problems

We want to apply this framework to several problems. In order to convey the intuition how a blow-up SSP reduction works, we only present the reduction from satisfiability to vertex cover and defer the other reductions to the full version.

Recall the definitions of vertex cover and 3-satisfiability as given in Section 3. Now, we can use a modification of the reduction by Garey and Johnson [13] from 3Sat to Vertex Cover as blow-up SSP reduction. We first describe the original reduction and then argue why it is SSP. At last, we show how to modify the reduction to a blow-up SSP reduction with a blow-up factor β_I.

Let (L, C) be the 3Sat instance of literals L and clauses C. We construct the corresponding Vertex Cover instance (G, k) with graph $G = (V, E)$ and integer k as follows. The reduction maps each literal ℓ to a vertex v_ℓ. Then the vertices v_ℓ and $v_{\overline{\ell}}$ of a literal ℓ and its negation $\overline{\ell}$ are connected by the edge $\{v_\ell, v_{\overline{\ell}}\}$. Next, for all clauses $c \in C$ there is a 3-clique on the three vertices $\{v_\ell^c : \ell \in C\}$. All the edges connecting the vertices in the 3-clique to their corresponding literal vertex are added, i.e. the edge set $\{\{v_\ell^c, v_\ell\} : c \in C, \ell \in c\}$. The threshold of the vertex cover instance is set to $k = |L|/2 + 2|C|$.

We now describe the blow-up gadget for every possible 3Sat instance I, blow-up literal set L_b, and number $\beta_I \in \mathbb{N}$. The blow-up gadget for $\ell, \overline{\ell} \in L_b$ is a duplication of the literal vertices v_ℓ and $v_{\overline{\ell}}$ forming a complete bipartite graph K_{β_I+1,β_I+1} between the vertex sets $V_\ell^{\beta_I} = \{v_\ell, v_{\ell^1}, \dots, v_{\ell^{\beta_I}}\}$ and $V_{\overline{\ell}}^{\beta_I} = \{v_{\overline{\ell}}, v_{\overline{\ell}^1}, \dots, v_{\overline{\ell}^{\beta_I}}\}$ as depicted in Figure 1. Then, we have the equivalence classes $Q_\ell = \{v_{\ell^1}, \dots, v_{\ell^{\beta_I}}\}$ and $Q_{\overline{\ell}} = \{v_{\overline{\ell}^1}, \dots, v_{\overline{\ell}^{\beta_I}}\}$. Now, consider the graph G as described above and modify it the following way: For each pair $\ell, \overline{\ell} \in L_b$, we identify the two vertices $v_\ell, v_{\overline{\ell}}$ in G and in the blow-up gadget for $\ell, \overline{\ell} \in L_b$ and merge them together. This results in the graph

$$G'(L_b, \beta_I) = (V(G) \cup \bigcup_{\ell \in L_b} V_\ell^{\beta_I},\ E(G) \cup \bigcup_{\ell \in L_b} \{\{a, b\} \mid a \in V_\ell^{\beta_I}, b \in V_{\overline{\ell}}^{\beta_I}\})$$

An example of such a graph $G'(L_b, \beta_I)$ can be found in Figure 1.

The threshold for the vertex cover is then increased such that $\beta_I + 1$ vertices (instead of only one) for every blow-up literal pair can be taken into the solution, i.e. $k'(\beta_I) = (\beta_I + 1) \cdot |L_b|/2 + |L \setminus L_b|/2 + 2|C|$.

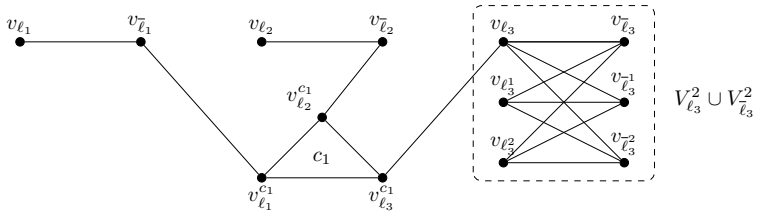


Figure 1 The graph $G'(L_b, \beta_I)$ with $L_b = \{\ell_3, \overline{\ell}_3\}$ and $\beta_I = 2$ of the instance $\varphi = (\overline{\ell}_1 \vee \overline{\ell}_2 \vee \ell_3)$. The blow-up gadget of the literals ℓ_3 and $\overline{\ell}_3$ consiting of vetex sets $V_{\ell_3}^2 \cup V_{\overline{\ell}_3}^2$ is outlined with dashed lines.

▷ Claim 11. For the three choices of dist from Section 4.1 and $\beta_I := |V(G)|$, $G'(L_b, \beta_I)$ and $k'(\beta_I)$ describe a blow-up SSP reduction from 3Sat to Vertex Cover.

Proof. We show that g, $(f_I)_{I \in \mathcal{I}}$, and $(\beta_I)_{I \in \mathcal{I}}$ defined by

$$g(I) = (G'(L_b, \beta_I), k'(\beta_I)),\ f(\ell) = v_\ell,\ \text{and}\ \beta_I = |V(G)|$$

describe a blow-up SSP reduction from 3Sat to Π. We start by showing that the reduction is still SSP.

For this, we first make the observation that for a bipartite graph K_{β_I+1,β_I+1} between the vertex sets $V_\ell^{\beta_I}$ and $V_{\overline{\ell}}^{\beta_I}$, there are exactly two vertex covers of size $\beta_I + 1$: either one takes all v_{ℓ^i} together with v_ℓ or all $v_{\overline{\ell}^i}$ together with $v_{\overline{\ell}}$ into the vertex cover. Otherwise, there is at least one edge $\{v, w\}$ with $v \in V_\ell^{\beta_I}$ and $w \in V_{\overline{\ell}}^{\beta_I}$ not covered.

With this observation, we can follow that every vertex cover solution still consists of

1. exactly one of the original vertices v_ℓ and $v_{\overline{\ell}}$ for $\ell, \overline{\ell} \in L$, and
2. the β_I additional vertices $\{v_{\ell^i} : 1 \leq i \leq \beta_I\}$ of the blow-up gadget if and only if $v_\ell \in L_b$ is in the vertex cover solution, and
3. exactly two of the three vertices $v_{\ell_1}^{c_j}, v_{\ell_2}^{c_j}, v_{\ell_3}^{c_j}$ for $c_j \in C$.

Thus, the original injective SSP mapping $f(\ell) = v_\ell$ is still a valid SSP mapping for this reduction.

To show that the blow-up property holds, we have to show that for $\beta_I := |V(G)|$ and for all solutions of $g(I)$, $S_1, S_2 \in \mathcal{S}'(g(I))$,

$$f(L_b) \cap S_1 = f(L_b) \cap S_2 \Leftrightarrow \text{dist}(S_1, S_2) \leq \beta_I. \tag{3}$$

We begin with analyzing the maximum distances of two solutions $S_1, S_2 \in \mathcal{S}'(g(I))$ for the three distance measures if S_1 and S_2 agree on each blow-up gadget. To reach the maximum distance between two solutions S_1 and S_2 for the three distance measures, S_1 includes all the v_ℓ for all $\ell \in L \setminus L_b$, while S_2 includes all the opposite $v_{\overline{\ell}}$. Additionally, S_1 includes a different pair $v_{\ell_1}^{c_j}, v_{\ell_2}^{c_j}$ for all $c_j \in C$ in comparison to S_2. Thus, all $v_{\overline{\ell}}$ of $\overline{\ell} \in L \setminus L_b$ and $v_{\ell_3}^{c_j}$ for all $c_j \in C$ have to be added (while deleting v_ℓ and one of $v_{\ell_1}^{c_j}, v_{\ell_2}^{c_j}$). In conclusion, $\beta_I = |V(G)|$ is an upper bound for the maximal possible *κ-addition* and *κ-deletion* distance $(|C| + |L \setminus L_b|/2)$ as well as for the maximal possible *Hamming distance* (here one has to count for all deletions *and* additions such that $\beta_I = |V(G)| \geq 2|C| + |L \setminus L_b| \geq |S_1 \triangle S_2|$ is also sufficient). It follows that $f(L_b) \cap S_1 = f(L_b) \cap S_2 \Rightarrow \text{dist}(S_1, S_2) \leq \beta_I$.

We conclude with the situation that S_1 and S_2 do not agree on each blow-up gadget. Then corresponding to the analysis above for *κ-addition* and *κ-deletion*, we reach at least a distance of $|V(G)| + 1 > \beta_I$ and for the Hamming distance we reach at least $2|V(G)| + 2 > \beta_I$. It follows that $f(L_b) \cap S_1 \neq f(L_b) \cap S_2 \Rightarrow \text{dist}(S_1, S_2) > \beta_I$.

Clearly β_I and thus $G'(L_b, \beta_I)$ and $k'(\beta_I)$ are polynomial time computable. This completes the correctness proof of the reduction. ◁

6 The Issue of Transitivity: Preserving the Blow-up

Since blow-up SSP reductions require more structure, we unfortunately lose transitivity in comparison to normal SSP reductions. However, we would like to reuse an existing blow-up SSP reduction such that we do not need to start every reduction at 3Sat and additionally construct blow-up gadgets. For this, we introduce *blow-up preserving SSP reductions*, and show that they are transitive and preserve the structure of the blow-up gadgets. The idea is that we only need to show that there is a blow-up SSP reduction for the first reduction in the reduction chain and then the reduction chain can be prolonged by adding further problems, between which there are blow-up preserving SSP reductions.

The idea of a blow-up preserving reduction between problem Π and problem Π' is that we partition the universe into three sets, the set of elements which originate in the problem Π and a set of elements U_{off}, which are never of a solution of the instance of Π', and a set of elements U_{on}, which are part of every solution of the instance of Π'. Therefore, every solution S in Π has a correspondent solution $S' = f(S) \cup U_{on}$ by applying the injective function f. Since the distance measures that we consider are invariant on injective functions and union, the distance between the solutions in problem Π is the same as in problem Π'. Correspondingly, the blow-up SSP reduction from 3Sat to Π can be extended to Π'.

From a different point of view, a blow-up preserving SSP reduction can be understood as an SSP reduction, which "adds" only two kind of new elements to the universe: Those that are trivially contained in every (optimal) solution, and those that are never contained in an (optimal) solution. It is intuitively easy to understand that compared to the old instance, the newly "added" elements cannot influence the term $\text{dist}(S_1, S_2)$. Hence Σ_3^p-hardness of recoverable problems is maintained.

▶ **Definition 12** (Blow-Up Preserving SSP Reduction). *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ and $\Pi' = (\mathcal{I}', \mathcal{U}', \mathcal{S}')$ be two SSP problems. Then, there is a blow-up preserving SSP reduction from Π to Π' if there exists an SSP reduction $(g, (f_I)_{I \in \mathcal{I}})$ such that for all instances $I \in \mathcal{I}$ the following holds: For all elements $u' \in \mathcal{U}'(g(I))$, either $u' \in f(\mathcal{U}(I))$, or for all $S' \in \mathcal{S}' : u' \notin S'$, i.e. $u' \in U_{off}$, or for all $S' \in \mathcal{S}' : u' \in S'$, i.e. $u' \in U_{on}$.*

With this definition of a blow-up preserving SSP reduction, we are able to derive the following important properties.

▶ **Theorem 13** (⋆). *Let Π be an SSP-NP-complete problem for which a blow-up SSP reduction from 3Sat exists. Then every problem Π', which is blow-up preserving SSP reducible from Π, is blow-up SSP reducible from 3Sat.*

▶ **Lemma 14** (⋆). *Blow-up preserving SSP reductions are transitive.*

6.1 Blow-up Gadgets for various Problems

In the full version of the paper, we show that for many classical problems, blow-up or blow-up-preserving reductions exists. An overview of all the reductions is provided in Figure 2. As a corollary, we obtain Theorem 15.

▶ **Theorem 15** (⋆). *The recoverable robust version of the following nominal problems is Σ_3^p-complete: Sat, 3Sat, Vertex Cover, Dominating Set, Set Cover, Hitting Set, Feedback Vertex/Arc Set, Uncap. Facility Location, p-Center, p-Median, Independent Set, Clique, Subset Sum, Knapsack, Partition, Scheduling, (Un)Directed Hamiltonian Path/Cycle, TSP, k-Vertex Directed Disjoint Path, Steiner Tree*

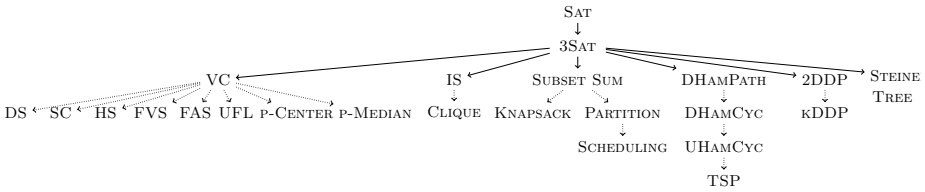


Figure 2 The tree of SSP reductions for all considered problems. Solid edges indicate blow-up SSP reductions and dotted edges indicate blow-up preserving SSP reductions.

7 Conclusion

We have shown that for a large class of NP-complete problems their corresponding recoverable robust problem is Σ_3^p-complete for several different distance measures. For this, we have introduced two new types of reductions: blow-up SSP reductions and blow-up preserving SSP reductions. Blow-up SSP reductions are the basis to show Σ_3^p-completeness of a recoverable robust problem by a reduction from 3Sat and further problems can be appended to the reduction chain by using the transitive blow-up preserving SSP reductions. In particular, we are able to show that 24 recoverable robust problems are Σ_3^p-complete with the ability to apply the framework to further problems.

These results and the framework might be extendible in several different dimensions. First the natural question arises, whether these results can also be applied to most problems that are just in NP but not NP-complete, because we are only able to use this result for NP-complete problems. The results by Jackiewicz, Kasperski and Zieliński [23] indicate that this might be indeed the case. Secondly, we have only shown that these results hold for subset search problems. Partition problems or permutation problems are not directly covered by this result. Thus, it is of interest to classify the complexity of these problems as well. Furthermore, it is of interest to apply the framework to more subset search problems, i.e. to show that more problems are blow-up (preserving) SSP reducible from the examined problems.

References

1 Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009. doi:10.1515/9781400831050.

2 Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Math. Program.*, 98(1-3):49–71, 2003. doi:10.1007/S10107-003-0396-4.

3 Matthew Bold and Marc Goerigk. Investigating the recoverable robust single machine scheduling problem under interval uncertainty. *Discret. Appl. Math.*, 313:99–114, 2022. doi:10.1016/j.dam.2022.02.005.

4 Christina Büsing. *Recoverable robustness in combinatorial optimization*. Cuvillier Verlag, 2011.

5 Christina Büsing. Recoverable robust shortest path problems. *Networks*, 59(1):181–189, 2012. doi:10.1002/net.20487.

6 Christina Büsing, Sebastian Goderbauer, Arie M. C. A. Koster, and Manuel Kutschka. Formulations and algorithms for the recoverable Γ-robust knapsack problem. *EURO J. Comput. Optim.*, 7(1):15–45, 2019. doi:10.1007/s13675-018-0107-9.

7 Christina Büsing, Arie M. C. A. Koster, and Manuel Kutschka. Recoverable robust knapsacks: Γ-scenarios. In Julia Pahl, Torsten Reiners, and Stefan Voß, editors, *Network Optimization - 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*,

volume 6701 of *Lecture Notes in Computer Science*, pages 583–588. Springer, 2011. doi:10.1007/978-3-642-21527-8_65.

8 Christina Büsing, Arie M. C. A. Koster, and Manuel Kutschka. Recoverable robust knapsacks: the discrete scenario case. *Optim. Lett.*, 5(3):379–392, 2011. doi:10.1007/s11590-011-0307-1.

9 André B. Chassein and Marc Goerigk. On the recoverable robust traveling salesman problem. *Optim. Lett.*, 10(7):1479–1492, 2016. doi:10.1007/s11590-015-0949-5.

10 André B. Chassein, Marc Goerigk, Adam Kasperski, and Pawel Zielinski. On recoverable and two-stage robust selection problems with budgeted uncertainty. *Eur. J. Oper. Res.*, 265(2):423–436, 2018. doi:10.1016/j.ejor.2017.08.013.

11 Mitre Costa Dourado, Dirk Meierling, Lucia Draque Penso, Dieter Rautenbach, Fábio Protti, and Aline Ribeiro de Almeida. Robust recoverable perfect matchings. *Networks*, 66(3):210–213, 2015. doi:10.1002/net.21624.

12 Dennis Fischer, Tim A. Hartmann, Stefan Lendl, and Gerhard J. Woeginger. An investigation of the recoverable robust assignment problem. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.IPEC.2021.19.

13 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

14 Marc Goerigk and Michael Hartisch. *An Introduction to Robust Combinatorial Optimization*, volume 361 of *International Series in Operations Research & Management Science*. Springer, 2024.

15 Marc Goerigk, Stefan Lendl, and Lasse Wulf. Recoverable robust representatives selection problems with discrete budgeted uncertainty. *Eur. J. Oper. Res.*, 303(2):567–580, 2022. doi:10.1016/j.ejor.2022.03.001.

16 Marc Goerigk, Stefan Lendl, and Lasse Wulf. On the complexity of robust multi-stage problems with discrete recourse. *Discret. Appl. Math.*, 343:355–370, 2024. doi:10.1016/J.DAM.2023.10.018.

17 Christoph Grüne. The complexity classes of hamming distance recoverable robust problems. In José A. Soto and Andreas Wiese, editors, *LATIN 2024: Theoretical Informatics - 16th Latin American Symposium, Puerto Varas, Chile, March 18-22, 2024, Proceedings, Part I*, volume 14578 of *Lecture Notes in Computer Science*, pages 321–335. Springer, 2024. doi:10.1007/978-3-031-55598-5_21.

18 Christoph Grüne and Lasse Wulf. Completeness in the polynomial hierarchy for many natural problems in bilevel and robust optimization. *CoRR*, abs/2311.10540, 2023. doi:10.48550/arXiv.2311.10540.

19 Christoph Grüne and Lasse Wulf. On the complexity of recoverable robust optimization in the polynomial hierarchy. *arXiv preprint arXiv:2411.18590*, 2024. doi:10.48550/arXiv.2411.18590.

20 Felix Hommelsheim, Nicole Megow, Komal Muluk, and Britta Peis. Recoverable robust optimization with commitment. *CoRR*, abs/2306.08546, 2023. doi:10.48550/arXiv.2306.08546.

21 Mikita Hradovich, Adam Kasperski, and Pawel Zielinski. Recoverable robust spanning tree problem under interval uncertainty representations. *J. Comb. Optim.*, 34(2):554–573, 2017. doi:10.1007/s10878-016-0089-6.

22 Mikita Hradovich, Adam Kasperski, and Pawel Zielinski. The recoverable robust spanning tree problem with interval costs is polynomially solvable. *Optim. Lett.*, 11(1):17–30, 2017. doi:10.1007/S11590-016-1057-X.

23 Marcel Jackiewicz, Adam Kasperski, and Pawel Zielinski. Computational complexity of the recoverable robust shortest path problem with discrete recourse. *CoRR*, abs/2403.20000, 2024. doi:10.48550/arXiv.2403.20000.

24 Marcel Jackiewicz, Adam Kasperski, and Pawel Zielinski. Recoverable robust shortest path problem under interval uncertainty representations. *CoRR*, abs/2401.05715, 2024. doi:10.48550/arXiv.2401.05715.

25 Robert G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Math. Program.*, 32(2):146–164, 1985. doi:10.1007/BF01586088.

26 Berit Johannes. New classes of complete problems for the second level of the polynomial hierarchy, 2011.

27 Adam Kasperski and Pawel Zielinski. Robust recoverable and two-stage selection problems. *Discret. Appl. Math.*, 233:52–64, 2017. doi:10.1016/j.dam.2017.08.014.

28 Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

29 Thomas Lachmann, Stefan Lendl, and Gerhard J. Woeginger. A linear time algorithm for the robust recoverable selection problem. *Discret. Appl. Math.*, 303:94–107, 2021. doi:10.1016/j.dam.2020.08.012.

30 Stefan Lendl, Britta Peis, and Veerle Timmermans. Matroid bases with cardinality constraints on the intersection. *Math. Program.*, 194(1):661–684, 2022. doi:10.1007/S10107-021-01642-1.

31 Christian Liebchen, Marco E. Lübbecke, Rolf H. Möhring, and Sebastian Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In Ravindra K. Ahuja, Rolf H. Möhring, and Christos D. Zaroliagis, editors, *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, volume 5868 of *Lecture Notes in Computer Science*, pages 1–27. Springer, 2009. doi:10.1007/978-3-642-05465-5_1.

32 Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

33 Larry J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.

34 Gerhard J. Woeginger. The trouble with the second quantifier. *4OR*, 19(2):157–181, 2021. doi:10.1007/s10288-021-00477-y.