Recycling Valid Inequalities for Robust Combinatorial Optimization with Budget Uncertainty

Christina Büsing[1], Timo Gersing[1], and Arie M.C.A. Koster[2]

[1]Combinatorial Optimization, RWTH Aachen University,
{buesing,gersing}@combi.rwth-aachen.de
[2]Discrete Optimization, RWTH Aachen University,
koster@math2.rwth-aachen.de

Abstract

Robust combinatorial optimization with budget uncertainty is one of the most popular approaches for integrating uncertainty into optimization problems. The existence of a compact reformulation for (mixed-integer) linear programs and positive complexity results give the impression that these problems are relatively easy to solve. However, the practical performance of the reformulation is quite poor when solving robust integer problems, in particular due to its weak linear relaxation.

To overcome this issue, we propose procedures to derive new classes of valid inequalities for robust combinatorial optimization problems. For this, we recycle valid inequalities of the underlying deterministic problem such that the additional variables from the robust formulation are incorporated. The valid inequalities to be recycled may either be readily available model constraints or actual cutting planes, where we can benefit from decades of research on valid inequalities for classical optimization problems.

We first demonstrate the strength of the inequalities theoretically, by proving that recycling yields a facet-defining inequality in many cases, even if the original valid inequality was not facet-defining. Afterwards, we show in an extensive computational study that using recycled inequalities can lead to a significant improvement of the computation time when solving robust optimization problems.

Keywords: Robust Optimization, Combinatorial Optimization, Integer Programming, Polyhedral Combinatorics, Computation
Mathematics Subject Classification: 90C11, 90C17, 90C27, 90C57

1 Introduction

Uncertain parameters are a common issue in decision-making problems and should be treated with caution, as a seemingly optimal decision may reveal itself to be impractical once it is implemented in the real world [7]. A popular approach for dealing with uncertainties is robust optimization, in which one aims to optimize against the worst-case of a set of scenarios. Robust optimization was proposed first by Soyster [32] in the early 1970s. Kouvelis and Yu [27] considered it for combinatorial optimization problems and discrete uncertainty sets in the 1990s. The concept was then further analyzed by Ben-Tal and Nemirovski [5, 6, 7] as well as Bertsimas and Sim [11, 12] at the beginning of this century. An overview on the topic can be found in [4, 9, 17].

The concept of budgeted uncertainty by Bertsimas and Sim has received particular attention. This is illustrated by the fact that their paper [12] is the most cited document in the literature databases Scopus and Web of Science that contains "*robust optimization*" in its title, keywords or abstract. However, despite its popularity and the amount of research devoted to solving these kinds of robust optimization problems, instances of practical size often still pose a considerable challenge for MILP solvers [14]. To help overcoming this challenge, we propose new classes of valid inequalities for robust combinatorial optimization problems that are easy to compute and often lead to a significant reduction of the computation time.

We first define a standard, so called *nominal,* combinatorial problem without uncertainties

$$\text{(NOM)} \qquad \begin{aligned} \min \ & \sum_{i\in[n]} c_i x_i \\ \text{s.t. } & Ax \leq b, x \in \{0,1\}^n , \end{aligned}$$

where $c \in \mathbb{R}^n$ is an objective vector, $A \in \mathbb{R}^{m\times n}$ a constraint matrix with a right-hand side $b \in \mathbb{R}^m$ and $[n] = \{1, \ldots, n\}$. We now replace the objective coefficients c_i with uncertain coefficients c'_i from an interval $[c_i, c_i + \hat{c}_i]$ and say that c'_i can *deviate* from its *nominal value* c_i by up to the *deviation* $\hat{c}_i$. In the following, we call a vector of possible objective coefficients $\{c' \in \mathbb{R}^n | c'_i \in [c_i, c_i + \hat{c}_i]\}$ a *scenario*. Note that for any feasible solution x, the objective value is worst if all coefficients c'_i are equal to their maximum value $c_i + \hat{c}_i$. In practice, however, this extreme scenario is usually very unlikely such that it would be overly conservative to assume that it actually occurs. To adjust the level of conservatism, Bertsimas and Sim [12] propose a robust counterpart to NOM, in which they restrict the set of considered uncertain scenarios by defining a *uncertainty budget* $\Gamma \in [0, n]$. Given such a budget, we only consider these scenarios in which at most $\lfloor \Gamma \rfloor$ coefficients c'_i deviate to $c_i + \hat{c}_i$ and one coefficient

deviates to $c_i + (\Gamma - \lfloor\Gamma\rfloor)\,\hat{c}_i$. This robust counterpart can be stated as

$$\begin{aligned}\min\ & \sum_{i\in[n]} c_i x_i + \max_{\substack{S\cup\{t\}\subseteq[n]:\\ |S|\le\lfloor\Gamma\rfloor, t\notin S}} \left((\Gamma - \lfloor\Gamma\rfloor)\,\hat{c}_t x_t + \sum_{i\in S} \hat{c}_i x_i \right) \\ \text{s.t. } & Ax \le b, x \in \{0,1\}^n .\end{aligned}$$

Bertsimas and Sim [12] also show that we can resolve the non-linearity of the above problem by dualizing the inner maximization problem, which results in the compact robust problem

$$\text{(ROB)} \qquad \begin{aligned}\min\ & \Gamma z + \sum_{i\in[n]} (c_i x_i + p_i) \\ \text{s.t. } & (x,p,z) \in \mathcal{P}^{\text{ROB}}, x \in \{0,1\}^n\end{aligned}$$

with

$$\mathcal{P}^{\text{ROB}} = \left\{ (x,p,z) \middle| \begin{array}{ll} Ax \le b & \\ p_i + z \ge \hat{c}_i x_i & \forall i \in [n] \\ x \in [0,1]^n, p \in \mathbb{R}^n_{\ge 0}, z \in \mathbb{R}_{\ge 0} & \end{array} \right\}.$$

Unfortunately, the formulation $\mathcal{P}^{\text{ROB}}$ is quite weak, potentially leading to much higher computation times for solving ROB compared to NOM. In fact, the integrality gap of the formulation $\mathcal{P}^{\text{ROB}}$, that is the relative difference between the values of an optimal fractional solution and an optimal integer-feasible solution, may be arbitrarily large, even if the integrality gap of the corresponding nominal problem is zero. This is shown in the following example from [14].

Example 1. *Consider the easy problem of selecting the cheapest of n elements*

$$\begin{aligned}\min\ & \sum_{i\in[n]} c_i x_i \\ s.t.\ & \sum_{i\in[n]} x_i = 1, x \in \{0,1\}^n .\end{aligned}$$

The integrality gap of the above problem is zero for all $c \in \mathbb{R}^n$. However, if we consider an instance of the uncertain counterpart ROB with $c \equiv 0$, $\hat{c} \equiv 1$, and $\Gamma = 1$

$$\begin{aligned}\min\ & z + \sum_{i\in[n]} p_i & \\ s.t.\ & \sum_{i\in[n]} x_i = 1 & \\ & p_i + z \ge x_i & \forall i \in [n] \\ & x \in \{0,1\}^n, p \in \mathbb{R}^n_{\ge 0}, z \in \mathbb{R}_{\ge 0}, &\end{aligned}$$

then $(x,p,z) = \left(\frac{1}{n}, \ldots, \frac{1}{n}, 0, \ldots, 0, \frac{1}{n}\right)$ is the unique optimal fractional solution of value $\frac{1}{n}$, while the objective value of an optimal integer solution is 1. Hence, the integrality gap is $\frac{1-1/n}{1/n} = n-1$, and thus unbounded if $n \to \infty$.

The above example shows that optimal continuous solutions for ROB tend to be highly fractional, as small values of x_i allow for covering all right-hand sides $\hat{c}_i x_i$ in the constraints $p_i + z \geq \hat{c}_i x_i$ with a small value of z, while choosing $p_i = 0$. On the one hand, such solutions are exactly what we aim for when striving for robustness, as we distribute the risk as much as possible. On the other hand, highly fractional optimal solutions for the linear relaxation imply the need for much branching, and thus a high computational effort when solving ROB.

In the past, several approaches to solve ROB have been developed and evaluated. Bertsimas et al. [10] as well as Fischetti and Monaci [16] tested the practical performance of the compact reformulation $\mathcal{P}^{\text{ROB}}$ compared to a separation approach using an alternative formulation with exponentially many inequalities, each one modeling a scenario from the uncertainty set. Unfortunately, the alternative formulation is, despite its size, as weak as $\mathcal{P}^{\text{ROB}}$ and performs worse for robust integer problems (but better for continuous problems). Joung and Park [26] propose cuts that dominate the classic scenario inequalities and can be separated by considering the robustness term as a submodular function and greedily solving a maximization problem over the corresponding polymatroid. Atamtürk [3] addresses the issue by proposing four different problem-independent strong formulations. The strongest of these is theoretically as strong as possible, as it preserves the integrality gap of the nominal problem. However, the four formulations are very large and are thus computationally outperformed by the standard formulation $\mathcal{P}^{\text{ROB}}$, as shown in [14].

A famous approach to completely avoid the issues arising from the weak formulation $\mathcal{P}^{\text{ROB}}$ is to solve ROB via resorting to its nominal counterpart. Bertsimas and Sim [11] show that there always exists an optimal solution (x, p, z) to ROB such that $z \in \{0, \hat{c}_1, \ldots, \hat{c}_n\}$. Furthermore, solving ROB for a fixed z is equivalent to solving its nominal counterpart with different objective values. Hence, ROB can be solved by solving $|\{0, \hat{c}_1, \ldots, \hat{c}_n\}| \leq n + 1$ nominal problems. Álvarez-Miranda et al. [2] as well as Park and Lee [30] showed independently that it is sufficient to solve only $n + 2 - \Gamma$, or $n + 1 - \Gamma$ nominal problems respectively. Lee and Kwon [28] even improved these results later, showing that solving $\left\lceil \frac{n-\Gamma}{2} \right\rceil + 1$ nominal problems already suffices. Still, the computational effort of this approach is too high if n is large. To address this, Hansknecht et al. [24] propose a divide and conquer approach, in which one also solves nominal problems, but reduces the computational effort by pruning many non-optimal values for z using bounds on the respective optimal objective value. In [14], we proposed a branch and bound approach, in which non-optimal values for z are pruned even more efficiently. The branch and bound makes use of structural insights and strong linearizations derived from the following bilinear formulation

$$\mathcal{P}^{\text{BIL}} = \left\{ (x, p, z) \middle| \begin{array}{ll} Ax \leq b & \\ p_i + x_i z \geq \hat{c}_i x_i & \forall i \in [n] \\ x \in [0,1]^n, p \in \mathbb{R}^n_{\geq 0}, z \in \mathbb{R}_{\geq 0} & \end{array} \right\}.$$

This bilinear formulation strengthens the original robustness constraints $p_i + z \geq \hat{c}_i x_i$ by multiplying z with x_i. This is valid, as the bilinear inequality is equivalent to $p_i + z \geq \hat{c}_i x_i$ for $x_i = 1$ and $p_i \geq 0$ for $x_i = 0$. While the bilinearity is rather hindering for practical purposes, $\mathcal{P}^{\text{BIL}}$ is theoretically very strong. In fact, there exists no polyhedral formulation $\mathcal{P}$ for ROB with $\mathcal{P} \subsetneq \mathcal{P}^{\text{BIL}}$ [14].

Contribution In this paper, we use the bilinear formulation $\mathcal{P}^{\text{BIL}}$ as a foundation for the new class of *recycled inequalities*. To obtain these, we combine the strength of the bilinear inequalities with the structural properties provided by valid inequalities for the nominal problem NOM. By doing so, we can use valid inequalities for NOM a second time to improve the formulation $\mathcal{P}^{\text{ROB}}$.

In its simplest and most effective version, our recycling relies on the underlying valid inequality to be a knapsack inequality. We will show that in this case the corresponding recycled inequality often defines a facet of the convex hull of integer-feasible solutions

$$\mathcal{C}^{\text{ROB}} = \text{conv}\left(\left\{(x,p,z) \in \mathcal{P}^{\text{ROB}} \middle| x \in \{0,1\}^n\right\}\right).$$

Additionally, we show how to efficiently separate such recycled inequalities in a branch and cut algorithm. We also discuss how to recycle inequalities that are not of the knapsack type, to make use of a wider range of valid inequalities.

In an extensive computational study on robust versions of both classical combinatorial problems and real-world instances from MIPLIB 2017 [20], we verify that recycled inequalities can substantially strengthen the formulation $\mathcal{P}^{\text{ROB}}$, which is expressed by drastic reductions of the integrality gap. Together with the efficient separation of recycled inequalities, this leads to a significant improvement of solving times.

All implemented algorithms and generated test instances are published, together with a package of algorithms for solving robust combinatorial optimization problems [18] and benchmark instances [19].

Note that this is an extended version of a paper that appeared in the proceedings of IPCO 2023 [13].

Outline In Section 2, we show how to derive the new class of recycled inequalities from valid knapsack inequalities. In Section 3, we characterize inequalities for which the corresponding recycled inequality is facet-defining. In Section 4, we discuss different efficient approaches of separating recycled inequalities. In Section 5, we show how to recycle non-knapsack inequalities. Finally, in Section 6, we conduct our computational study.

2 Recycling Valid Inequalities

As already mentioned, the bilinear inequalities $p_i + x_i z \geq \hat{c}_i x_i$ play a crucial role for our recycled inequalities. To understand their strength intuitively, we recall our observations from Example 1. There, we noticed that choosing fractional

values for x_i is tempting, as we are then able to meet the inequalities $p_i + z \geq \hat{c}_i x_i$ with a small value of z and $p_i = 0$. However, this advantage vanishes for the bilinear inequalities $p_i + x_i z \geq \hat{c}_i x_i$, as we always have $z \geq \hat{c}_i$ for $x_i \neq 0$ and $p_i = 0$. To make use of this in practice, it would be beneficial to carry over the strength of the bilinear inequalities to a linear formulation.

Multiplying linear inequalities with variables as an intermediate step in order to achieve a stronger linear formulation is not a new approach. For the Reformulation-Linearization-Technique by Sherali and Adams [31], one multiplies constraints with variables and linearizes the resulting products afterwards via substitution with auxiliary variables. When taken to the extreme, where all constraints are multiplied with all possible combinations of variables, one obtains a formulation with exponentially many variables and constraints, whose projection on the original variables equals $\mathcal{C}^{\text{ROB}}$. Our approach is different in the sense that we don't directly linearize the bilinear inequalities, and thus don't create auxiliary variables. Instead, we combine several of the bilinear inequalities in order to estimate the non-linear terms against a linear term, using a valid inequality for the corresponding nominal problem. From now on, let

$$\mathcal{C}^{\text{NOM}} = \text{conv}\left(\{x \in \{0,1\}^n | Ax \leq b\}\right)$$

be the convex hull of all integer nominal solutions. Then we combine the bilinear inequalities and valid inequalities for $\mathcal{C}^{\text{NOM}}$ as follows.

Theorem 1. *Let $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ be a valid inequality for $\mathcal{C}^{NOM}$ with $\pi \in \mathbb{R}^{n+1}_{\geq 0}$. Then the inequality*

$$\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i \tag{1}$$

is valid for $\mathcal{C}^{ROB}$.

Proof. Summing the bilinear constraints $p_i + x_i z \geq \hat{c}_i x_i$, each with a weight of π_i, we obtain

$$\sum_{i\in[n]} \pi_i p_i + \sum_{i\in[n]} \pi_i x_i z \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i,$$

which is a valid inequality for $\mathcal{C}^{\text{ROB}}$ due to $\pi \geq 0$. Now, since $z \geq 0$ holds, we have $\sum_{i\in[n]} \pi_i x_i z \leq \pi_0 z$, and thus the validity of (1). □

As we reuse the valid inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ to strengthen the formulation $\mathcal{P}^{\text{ROB}}$, we call Inequality (1) the *recycled inequality* of $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$. In accordance with the requirements of Theorem 1, we call $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ *recyclable* if it is valid for $\mathcal{C}^{\text{NOM}}$ and $\pi \geq 0$. In the following, we will only consider nominal inequalities $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ consisting exclusively of variables with uncertain objective coefficients, i.e., with $\hat{c}_i > 0$ for all $i \in [n]$ with $\pi_i \neq 0$. We call inequalities and their corresponding coefficients π with this property *uncertainty-exclusive*. Note that uncertainty-exclusive inequalities are

the only interesting ones for recycling, because we can always drop variables x_i with $\hat{c}_i = 0$. The resulting nominal inequality is weaker, and thus valid for $\mathcal{C}^{\mathrm{NOM}}$, but the corresponding recycled inequality becomes stronger by removing $\pi_i p_i$ from the left-hand side, while the right-hand side does not change due to $\pi_i \hat{c}_i x_i = 0$. By focusing on uncertainty-exclusive inequalities, we obtain the following statement.

Proposition 1. *Let $\pi \in \mathbb{R}^{n+1}$ be uncertainty-exclusive. If $\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i$ is valid for $\mathcal{C}^{ROB}$, then $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ is a recyclable inequality.*

Proof. First, note that the validity of $\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i$ already implies $\pi \geq 0$, since p and z are unbounded, while the right-hand side $\sum_{i\in[n]} \pi_i \hat{c}_i x_i$ is not. Now, assume that $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ is invalid, i.e., there exists a vector $\tilde{x} \in \mathcal{C}^{\mathrm{NOM}}$ with $\sum_{i\in[n]} \pi_i \tilde{x}_i > \pi_0$. Then there exists an index $j \in [n]$ with $\hat{c}_j \tilde{x}_j > 0$ and we define $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{C}^{\mathrm{ROB}}$ with $\tilde{z} = \min\{\hat{c}_i | i \in [n], \hat{c}_i \tilde{x}_i > 0\}$ as well as $\tilde{p}_i = (\hat{c}_i - \tilde{z})^+ \tilde{x}_i$ for all $i \in [n]$. Note that we write $(y)^+ = \max\{y, 0\}$ for arbitrary $y \in \mathbb{R}$. Then we have $\tilde{z} > 0$ and $\pi_i \tilde{p}_i = \pi_i (\hat{c}_i - \tilde{z}) \tilde{x}_i$ for all $i \in [n]$, which implies

$$\pi_0 \tilde{z} + \sum_{i\in[n]} \pi_i \tilde{p}_i = \tilde{z} \left(\pi_0 - \sum_{i\in[n]} \pi_i \tilde{x}_i \right) + \sum_{i\in[n]} \pi_i \hat{c}_i \tilde{x}_i < \sum_{i\in[n]} \pi_i \hat{c}_i \tilde{x}_i,$$

and thus proves that $\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i$ cannot be valid. □

The above shows that we can actually obtain all relevant valid inequalities of the form $\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i$ by recycling a valid nominal inequality. We also get a first understanding of the strength of the recycling approach. If there exists an inequality of the same form as the Recycled Inequality (1) that is better than the recycled one, then this does not mean that the recycling procedure is weak, but that there exists a better nominal inequality to recycle.

In order to see in which cases recycled inequalities are useful, let us consider how they compare to the bilinear inequalities over the course of their construction. First, note that the sum of the bilinear inequalities is weaker than the bilinear inequalities themselves. Hence, when separating a recycled inequality to cut off a fractional solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\mathrm{NOM}}$, our inequality to be recycled should only support indices $i \in [n]$ with $\pi_i > 0$ for which the bilinear inequality $\tilde{p}_i + \tilde{x}_i \tilde{z} \geq \hat{c}_i \tilde{x}_i$ is violated or tight. A second potential weakening occurs when applying the estimation $\sum_{i\in[n]} \pi_i x_i z \leq \pi_0 z$. This implies that recycling $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ is especially interesting if it is binding for $(\tilde{x}, \tilde{p}, \tilde{z})$.

Reconsider Example 1, for which we can recycle the valid inequality $\sum_{i\in[n]} x_i \leq 1$ implied by $\sum_{i\in[n]} x_i = 1$. The recycled inequality $z + \sum_{i\in[n]} p_i \geq \sum_{i\in[n]} x_i$ yields $z + \sum_{i\in[n]} p_i \geq 1$, and thus the optimal objective value of the linear relaxation is now equal to the optimal integer objective value. This intuitively highlights the strength of the recycled inequalities in the case where both properties, a binding recyclable valid inequality and the violation of supported bilinear

inequalities, coincide. In the next section, we will investigate the strength of recycled inequalities from a polyhedral point of view.

3 Facet-Defining Recycled Inequalities

In this section, we show that recycled inequalities often define facets of the convex hull of the robust problem $\mathcal{C}^{\mathrm{ROB}}$. Formally, for a polytope $P \subseteq \mathbb{R}^n$, each valid inequality $\sum_{i\in[n]} \pi_i x \leq \pi_0$ defines a *face* $F(\pi) = \left\{ x \in P \middle| \sum_{i\in[n]} \pi_i x = \pi_0 \right\}$ of P. A face $F(\pi)$ is called a *facet* if $\dim(F(\pi)) = \dim(P) - 1$ holds, where the dimension of a polytope P is defined as the maximum number of affinely independent points within P minus one [15]. We say that an inequality is *facet-defining* if its face is a facet. Facet-defining inequalities are highly interesting, as they are the best inequalities to describe a polytope P in the sense that a minimal representation

$$P = \left\{ x \in \mathbb{R}^n \middle| \begin{matrix} A^{=}x = b^{=} \\ A^{<}x \leq b^{<} \end{matrix} \right\},$$

with as few constraints as possible, consists of $n - \dim(P)$ equations $A^{=}x = b^{=}$ and otherwise only proper inequalities $A^{<}x \leq b^{<}$, all of which are facet-defining [15].

In order to prove that recycled inequalities can be facet-defining, we first determine the dimension of $\mathcal{C}^{\mathrm{ROB}}$. For the sake of simplicity, we assume for the rest of this paper that the solution sets $\mathcal{C}^{\mathrm{NOM}}$ and $\mathcal{C}^{\mathrm{ROB}}$ are non-empty.

Lemma 1. *We have* $\dim\left(\mathcal{C}^{ROB}\right) = \dim\left(\mathcal{C}^{NOM}\right) + n + 1$.

Proof. For a polytope $P \subseteq \mathbb{R}^n$, the number $n - \dim(P)$ equals the maximum number of linearly independent equations that are met by all vectors in P. Let $\sum_{i\in[n]} (\omega_i x_i + \omega_{n+i} p_i) + \omega_{2n+1} z = \omega_0$ be satisfied by all $(x, p, z) \in \mathcal{C}^{\mathrm{ROB}}$. Since p and z can be raised arbitrarily and $\mathcal{C}^{\mathrm{ROB}} \neq \emptyset$, we have $\omega_{n+1} = \cdots = \omega_{2n+1} = 0$, and thus $\sum_{i\in[n]} \omega_i x_i = \omega_0$. Hence, the equations that are met by all $(x, p, z) \in \mathcal{C}^{\mathrm{ROB}}$ are exactly the equations that are met by all $x \in \mathcal{C}^{\mathrm{NOM}}$, which implies

$$\dim\left(\mathcal{C}^{\mathrm{ROB}}\right) = 2n + 1 - \left(n - \dim\left(\mathcal{C}^{\mathrm{NOM}}\right)\right) = \dim\left(\mathcal{C}^{\mathrm{NOM}}\right) + n + 1.$$

□

Knowing the dimension of $\mathcal{C}^{\mathrm{ROB}}$, we are now able to study facet-defining recycled inequalities. Remember that we only consider uncertainty-exclusive inequalities, as these are the only interesting ones. The following theorem states the dimension of a recycled inequality's face, based on the projection of the nominal inequality's face on the supported variables. Together with Lemma 1, this yields a characterization of facet-defining recycled inequalities.

Theorem 2. *Let $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ be a recyclable, uncertainty-exclusive inequality and $S = \{i \in [n] | \pi_i > 0\}$. Then the face of the Recycled Inequality (1) has dimension*

$$\dim\left(\mathrm{proj}_S\left(F\left(\pi\right)\right)\right) + \dim\left(\mathcal{C}^{NOM}\right) + 1 + n - |S|\,,$$

where proj_S *is the projection on the supported variables* $\{x_i | i \in S\}$.

Hence, the Recycled Inequality (1) is facet-defining for $\mathcal{C}^{ROB}$ *if and only if*

$$\dim\left(\mathrm{proj}_S\left(F\left(\pi\right)\right)\right) = |S| - 1$$

holds.

Proof. There always exist $\dim\left(\mathcal{C}^{\mathrm{NOM}}\right)+1+n-|S|$ affinely independent vectors $(x,p,z) \in \mathcal{C}^{\mathrm{ROB}}$ satisfying (1) with equality. To see this, let $\left\{x^0, \ldots, x^{\dim\left(\mathcal{C}^{\mathrm{NOM}}\right)}\right\} \subseteq \mathcal{C}^{\mathrm{NOM}}$ be affinely independent. For each $j \in \left\{0, \ldots, \dim\left(\mathcal{C}^{\mathrm{NOM}}\right)\right\}$, we choose $\left(x^j, \hat{c} \odot x^j, 0\right)$, where $\hat{c} \odot x^j$ refers to the component-wise multiplication, i.e., $\left(\hat{c} \odot x^j\right)_i = \hat{c}_i x_i^j$. By definition, $\left(x^j, \hat{c} \odot x^j, 0\right)$ is within $\mathcal{C}^{\mathrm{ROB}}$ and we have

$$\pi_0 0 + \sum_{i\in[n]} \pi_i \left(\hat{c} \odot x^j\right)_i = \sum_{i\in[n]} \pi_i \hat{c}_i x_i^j.$$

Additionally, we choose $\left(x^0, \hat{c} \odot x^0 + e^j, 0\right)$ for each $j \in [n] \setminus S$, with $e^j \in \mathbb{R}^n$ being the j-th unit vector. Again, this vector is within $\mathcal{C}^{\mathrm{ROB}}$ and due to $\pi_j = 0$ it follows

$$\pi_0 0 + \sum_{i\in[n]} \pi_i \left(\hat{c} \odot x^0 + e^j\right)_i = \pi_j + \sum_{i\in[n]} \pi_i \hat{c}_i x_i^0 = \sum_{i\in[n]} \pi_i \hat{c}_i x_i^0.$$

Let $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}^j\right)_{j\in[k]}$ be a *maximal extension*, consisting of a maximum number of additional vectors in $\mathcal{C}^{\mathrm{ROB}}$ that satisfy the recycled inequality with equality and are affinely independent of the vectors above. It remains to show that $k = \dim\left(\mathrm{proj}_S\left(F\left(\pi\right)\right)\right) + 1$ holds. For this, we show that an extension is valid if and only if the following four properties hold:

1. $\tilde{p}_i^j = \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j$ for all $j \in [k]$, $i \in S$,
2. $\tilde{z}^j > 0$ for all $j \in [k]$,
3. $\left\{\mathrm{proj}_S\left(\tilde{x}^j\right) \middle| j \in [k]\right\}$ are affinely independent,
4. $\left\{\tilde{x}^j \middle| j \in [k]\right\} \subseteq F\left(\pi\right)$.

Note that Properties 3 and 4 already imply $k \leq \dim\left(\mathrm{proj}_S\left(F\left(\pi\right)\right)\right)+1$. For the other direction, it will then be sufficient to show that there always exist vectors $\left\{\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}^j\right) \middle| j \in \{0, \ldots, \dim\left(\mathrm{proj}_S\left(F\left(\pi\right)\right)\right)\}\right\}$ satisfying the first two properties.

Since we can assume without loss of generality that $\tilde{x}^j$ is binary for all $j \in [k]$, we have $\tilde{p}_i^j \geq \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j$ for all $i \in [n]$. Now, if Property 1 would not

hold, then there would exist $j \in [k]$ and $i \in S$ with $\tilde{p}_i^j > \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j$, and thus $\pi_i \tilde{p}_i^j > \pi_i \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j$. This is a contradiction due to

$$\begin{aligned}
\pi_0 \tilde{z}^j + \sum_{i \in [n]} \pi_i \tilde{p}_i^j &> \pi_0 \tilde{z}^j + \sum_{i \in [n]} \pi_i \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j \\
&\geq \sum_{i \in [n]} \pi_i \tilde{x}_i^j \tilde{z}^j + \sum_{i \in [n]} \pi_i \left(\hat{c}_i - \tilde{z}^j\right) \tilde{x}_i^j \\
&= \sum_{i \in [n]} \pi_i \hat{c}_i \tilde{x}_i^j.
\end{aligned}$$

The affine independency of the extension is equivalent to the linear independency of the $\dim\left(\mathcal{C}^{\mathrm{NOM}}\right) + n$ vectors obtained from subtracting $\left(x^0, \hat{c} \odot x^0, 0\right)$ of all others. We write these into the following matrix, assuming that $S = \{1, \ldots, |S|\}$

$$\left(\begin{array}{ccc|ccc|ccc}
\ldots & x_1^j - x_1^0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{x}_1^j - x_1^0 & \ldots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
\ldots & x_n^j - x_n^0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{x}_n^j - x_n^0 & \ldots \\
\hline
\ldots & \hat{c}_1 \left(x_1^j - x_1^0\right) & \ldots & 0 & \ldots & 0 & \ldots & \left(\hat{c}_1 - \tilde{z}^j\right) \tilde{x}_1^j - \hat{c}_1 x_1^0 & \ldots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
 & \vdots & & 0 & \ldots & 0 & & \left(\hat{c}_{|S|} - \tilde{z}^j\right) \tilde{x}_{|S|}^j - \hat{c}_{|S|} x_{|S|}^0 & \\
 & \vdots & & 1 & & 0 & & \tilde{p}_{|S|+1}^j - \hat{c}_{|S|+1} x_{|S|+1}^0 & \\
 & \vdots & & & \ddots & & & \vdots & \\
\ldots & \hat{c}_n \left(x_n^j - x_n^0\right) & \ldots & 0 & & 1 & \ldots & \tilde{p}_n^j - \hat{c}_n x_n^0 & \ldots \\
\hline
\ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{z}^j & \ldots
\end{array}\right).$$

For each $i \in [n]$, we subtract row i from row $n + i$ with factor $\hat{c}_i$ and obtain

$$\left(\begin{array}{ccc|ccc|ccc}
\ldots & x_1^j - x_1^0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{x}_1^j - x_1^0 & \ldots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
\ldots & x_n^j - x_n^0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{x}_n^j - x_n^0 & \ldots \\
\hline
\ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & -\tilde{z}^j \tilde{x}_1^j & \ldots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
 & \vdots & & 0 & \ldots & 0 & & -\tilde{z}^j \tilde{x}_{|S|}^j & \\
 & \vdots & & 1 & & 0 & & \tilde{p}_{|S|+1}^j - \hat{c}_{|S|+1} \tilde{x}_{|S|+1}^j & \\
 & \vdots & & & \ddots & & & \vdots & \\
\ldots & 0 & \ldots & 0 & & 1 & \ldots & \tilde{p}_n^j - \hat{c}_n \tilde{x}_n^j & \ldots \\
\hline
\ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & \tilde{z}^j & \ldots
\end{array}\right).$$

We use the middle columns with the unit vectors to eliminate the corresponding rows. Furthermore, since $\tilde{x}^j - x^0$ is linearly dependent of $\left\{x^1 - x^0, \ldots, x^{\dim(\mathcal{C}^{\mathrm{NOM}})} - x^0\right\}$

due to the dimension of $\mathcal{C}^{\text{NOM}}$, we can eliminate the first n rows in the last k columns and obtain

$$\left(\begin{array}{ccc|ccc|ccc}
\dots & x_1^j - x_1^0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
\dots & x_n^j - x_n^0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots \\
\hline
\dots & 0 & \dots & 0 & \dots & 0 & \dots & -\tilde{z}^j \tilde{x}_1^j & \dots \\
 & \vdots & & \vdots & & \vdots & & \vdots & \\
 & \vdots & & 0 & \dots & 0 & & -\tilde{z}^j \tilde{x}_{|S|}^j & \\
 & \vdots & & 1 & & 0 & & 0 & \\
 & \vdots & & & \ddots & & & \vdots & \\
\dots & 0 & \dots & 0 & & 1 & \dots & 0 & \dots \\
\hline
\dots & 0 & \dots & 0 & \dots & 0 & \dots & \tilde{z}^j & \dots
\end{array}\right).$$

These columns are linearly independent if and only if $\tilde{z}^j \neq 0$ holds for all $j \in [k]$ and $\left\{\text{proj}_S\left(\tilde{x}^j\right) \middle| j \in [k]\right\}$ are affinely independent. Thus, given Property 1, the affine independency of the extension is equivalent to Properties 2 and 3. Furthermore, given Properties 1 and 2, we have

$$\pi_0 \tilde{z}^j + \sum_{i \in [n]} \pi_i \tilde{p}_i^j = \sum_{i \in [n]} \pi_i \hat{c}_i \tilde{x}_i^j \;\Leftrightarrow\; \pi_0 \tilde{z}^j = \sum_{i \in [n]} \pi_i \tilde{z}^j \tilde{x}_i^j \;\Leftrightarrow\; \pi_0 = \sum_{i \in [n]} \pi_i \tilde{x}_i^j,$$

and thus $\left\{\tilde{x}^j \middle| j \in [k]\right\} \subseteq F(\pi)$. Hence, Property 4 holds if and only if $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}^j\right)_{j \in [k]}$ fulfill the Recycled Inequality (1) with equality. Overall, this shows that the validity of the extension is equivalent to Properties 1 to 4.

Now, let $\left\{\tilde{x}^j \middle| j \in \{0, \dots, \dim\left(\text{proj}_S\left(F(\pi)\right)\right)\}\right\} \subseteq F(\pi)$ be affinely independent in the components $\{x_i | i \in S\}$. As stated above, it suffices to construct vectors $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}^j\right)$ satisfying Properties 1 and 2. For each $j \in \{0, \dots, \dim\left(\text{proj}_S\left(F(\pi)\right)\right)\}$, we choose $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}\right)$ with $\tilde{z} = \min\left\{\hat{c}_i | i \in [n], \hat{c}_i > 0\right\}$ and $\tilde{p}_i^j = \left(\hat{c}_i - \tilde{z}\right)^+ \tilde{x}_i^j$ for all $i \in [n]$. Then $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}\right)$ is by definition within $\mathcal{C}^{\text{ROB}}$ and satisfies $\tilde{z} > 0$. Since $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ is uncertainty-exclusive, we have $\pi_i = 0$ for all $\hat{c}_i < \tilde{z}$, and thus $\tilde{p}_i^j = \left(\hat{c}_i - \tilde{z}\right) \tilde{x}_i^j$ for all $i \in S$. Therefore, $\left(\tilde{x}^j, \tilde{p}^j, \tilde{z}\right)$ satisfies both properties, which completes the proof. □

A powerful implication of Theorem 2 is that recycling an uncertainty-exclusive inequality yields always a facet-defining inequality if $\dim\left(F(\pi)\right) = n - 1$ holds. This is because there exist n affinely independent vectors satisfying $\sum_{i \in [n]} \pi_i x = \pi_0$, of which $|S|$ must be affinely independent when projected on the variables $\{x_i | i \in S\}$. Note that $\dim\left(F(\pi)\right) = n-1$ holds if $F(\pi)$ is either a facet of a full-dimensional polytope $\mathcal{C}^{\text{NOM}}$ or if $\sum_{i \in [n]} \pi_i x \leq \pi_0$ is actually an equation with $F(\pi) = \mathcal{C}^{\text{NOM}}$ and $\dim\left(\mathcal{C}^{\text{NOM}}\right) = n - 1$. This is summarized in the following corollary.

Corollary 1. *Let $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ be a recyclable, uncertainty-exclusive inequality. The Recycled Inequality (1) is facet-defining for $\mathcal{C}^{ROB}$ if one of the following holds:*

- *$\mathcal{C}^{NOM}$ is full-dimensional and $F(\pi)$ is a facet of $\mathcal{C}^{NOM}$,*
- *$\dim\left(\mathcal{C}^{NOM}\right) = n-1$ and $F(\pi) = \mathcal{C}^{NOM}$.*

Contrary to first intuition, it is also possible to obtain facet-defining inequalities by recycling weaker inequalities that are neither facet-defining nor equations. This is because the dimension of the face $F(\pi)$ can shrink by less than $n - |S|$ when projected on $\{x_i | i \in S\}$. Thus, inequalities defining low-dimensional faces can also yield facet-defining recycled inequalities if $\pi_i = 0$ holds for many $i \in [n]$. For example, consider an independent set problem on a graph with vertices $V = [n]$ and let $Q \subseteq V$ be a clique. Then the clique inequality $\sum_{i\in Q} x_i \leq 1$ dominates all inequalities $\sum_{i\in Q'} x_i \leq 1$ with $Q' \subsetneq Q$ and is facet-defining if and only if Q is a maximal clique with respect to inclusion [15]. However, the recycled inequality $z + \sum_{i\in Q'} p_i \geq \sum_{i\in Q'} \hat{c}_i x_i$ is facet-defining for all cliques $Q' \subseteq Q$, since the projection $\text{proj}_{Q'}(F(\pi))$, that is $S = Q'$, has dimension $|Q'| - 1$, with $\left\{e^j \middle| j \in Q'\right\}$ being the corresponding affinely independent vectors. Other examples include odd hole inequalities for the independent set problem [29] and minimal cover inequalities for the knapsack problem [15]. These are in general not facet-defining for their respective polytope, but yield facet-defining recycled inequalities for the robust counterpart.

One now might raise the question whether recycling dominated inequalities is actually of practical interest or whether these do not matter due to the special structure of the objective function, with all p_i having an objective coefficient of 1. The following example demonstrates that it can be beneficial to weaken an inequality before it is recycled.

Example 2. *Consider the robust problem*

$$\begin{aligned}
\min\ & 2z + \sum_{i\in[5]} -x_i + p_i \\
s.t.\ & 3x_5 + \sum_{i\in[4]} x_i \leq 3 \\
& z + p_i \geq x_i && \forall i \in [5] \\
& x \in \{0,1\}^5, p \in \mathbb{R}^5_{\geq 0}, z \in \mathbb{R}_{\geq 0}.
\end{aligned}$$

Choosing $x = \left(\frac{3}{4}, \ldots, \frac{3}{4}, 0\right)$, $p = 0$, and $z = \frac{3}{4}$ yields an optimal solution for the linear relaxation with objective value $-\frac{3}{2}$. Recycling constraint $3x_5 + \sum_{i\in[4]} x_i \leq 3$ yields $3z + 3p_5 + \sum_{i\in[4]} p_i \geq 3x_5 + \sum_{i\in[4]} x_i$. After adding the recycled inequality, an optimal solution is given by $x = \left(\frac{3}{4}, \ldots, \frac{3}{4}, 0\right)$, $p = \left(0, \ldots, 0, \frac{1}{4}\right)$, and $z = \frac{3}{4}$, with an objective value of $-\frac{5}{4}$. Note that we now choose $p_5 > 0$ even though $x_5 = 0$ holds. Since the bilinear inequality $p_5 + x_5 z \geq \hat{c}_5 x_5$ now has a slack of $\frac{1}{4}$, our observation from the last section suggests that it may be

beneficial to drop x_5 from the valid inequality for recycling. In fact, when recycling the dominated inequality $\sum_{i\in[4]} x_i \leq 3$ instead of the model constraint, we obtain $3z + \sum_{i\in[4]} p_i \geq \sum_{i\in[4]} x_i$ and an optimal solution is now given by $x = (1, 1, 1, 0, 0)$, $p = 0$, and $z = 1$, which yields an objective value of -1.

After discussing the strong implications of Theorem 2, let us now consider its limitations. The next example shows that it is indeed necessary that the inequality to be recycled is uncertainty-exclusive.

Example 3. *Consider the full-dimensional polytope*

$$\mathcal{C}^{ROB} = \operatorname{conv}\left(\left\{(x,p,z) \in \{0,1\}^3 \times \mathbb{R}^4_{\geq 0} \middle| \begin{array}{rl} x_1 + x_2 + x_3 \leq 2 & \\ z + p_i \geq \hat{c}_i x_i & \forall i \in [3] \end{array}\right\}\right),$$

with $\hat{c}_1 = \hat{c}_2 = 1$ and $\hat{c}_3 = 0$. The constraint $x_1 + x_2 + x_3 \leq 2$ is facet-defining for the corresponding $\mathcal{C}^{NOM}$, and thus meets all requirements of Corollary 1 with the exception that it is not uncertainty-exclusive. Indeed, the recycled inequality $2z + p_1 + p_2 + p_3 \geq x_1 + x_2$ is not facet-defining, as it is dominated by the sum of the constraints $z + p_1 \geq x_1$ and $z + p_2 \geq x_2$.

The observation in the example above is quite intuitive. While we can always transform an arbitrary recyclable inequality into an uncertainty-exclusive one by dropping all x_i with $\hat{c}_i = 0$, we loose information during this process and cannot expect to obtain a facet on the robust variables. Less obvious is the importance of the dimension of the nominal polytope $\mathcal{C}^{\mathrm{NOM}}$, which is a prerequisite for Corollary 1. In the following, we study lower-dimensional problems, for which we first consider another example.

Example 4. *Consider the four-dimensional polytope with five variables*

$$\mathcal{C}^{ROB} = \operatorname{conv}\left(\left\{(x,p,z) \in \{0,1\}^2 \times \mathbb{R}^3_{\geq 0} \middle| \begin{array}{rl} x_1 + x_2 = 1 & \\ z + p_i \geq x_i & \forall i \in [2] \end{array}\right\}\right).$$

The inequality $2x_1 + x_2 \leq 2$ defines a facet for the corresponding $\mathcal{C}^{NOM}$. However, the recycled inequality $2z + 2p_1 + p_2 \geq 2x_1 + x_2$ is not facet-defining for $\mathcal{C}^{ROB}$, as it is the sum of $z + p_1 \geq x_1$ and $z + p_1 + p_2 \geq x_1 + x_2$, where the latter is recycled from $x_1 + x_2 = 1$.

The example shows that in the lower dimensional case, inequalities recycled from facet-defining inequalities are not necessarily facet-defining. Note that the mapping from the set of recyclable inequalities to the corresponding recycled inequalities is a homomorphism in the sense that the recycled inequality of $\sum_{i\in[n]} \left(\pi^1_i + \pi^2_i\right) x_i \leq \pi^1_0 + \pi^2_0$ equals the sum of the recycled inequalities of $\sum_{i\in[n]} \pi^1_i x_i \leq \pi^1_0$ and $\sum_{i\in[n]} \pi^2_i x_i \leq \pi^2_0$. As recycled inequalities are proper inequalities, their sum is weaker than the separate inequalities and it is better to recycle each inequality individually.

This applies to the example, where $2x_1 + x_2 \leq 2$ is the sum of the recyclable inequality $x_1 \leq 1$ and the recyclable equation $x_1 + x_2 = 1$. Although both inequalities have the same face, recycling $x_1 \leq 1$ yields a facet-defining inequality, while recycling $2x_1 + x_2 \leq 2$ does not. Hence, for lower dimensional $\mathcal{C}^{\mathrm{NOM}}$, one cannot decide whether recycling yields a facet-defining inequality by solely relying on the recyclable inequality's face.

However, we can eliminate equations from an inequality and recycle the resulting one, which is equivalent for $\mathcal{C}^{\mathrm{NOM}}$. Formally, we call two inequalities $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$ and $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ *equivalent for a polytope* P if they define the same face [35]. This is implies that there exist equations $\sum_{i\in[n]} \omega_i^k x_i = \omega_0^k$, satisfied by all $x \in P$, such that $\left\{\pi', \omega^1, \ldots, \omega^\ell\right\}$ are linearly independent and $\lambda_0 \pi' + \sum_{k\in[\ell]} \lambda_k \omega^k = \pi$ holds for some $\lambda_0 > 0$ and $\lambda_k \in \mathbb{R}$ with $k \in [\ell]$.

Algorithm 1: Procedure for removing equations from π.

Input: A recyclable, uncertainty-exclusive inequality $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$ and equations $\sum_{i\in[n]} \omega_i^k x_i = \omega_0^k$ for $k \in [\ell]$ such that $\left\{\pi', \omega^1, \ldots, \omega^\ell\right\}$ are linearly independent

Output: An equivalent recyclable, uncertainty-exclusive inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ with $|\{i \in [n] | \pi_i = 0\}| \geq \ell$

1 Set $\pi = \pi'$
2 **for** $k \in [\ell]$ **do**
3 Choose $i^* \in \operatorname{argmin} \left\{ \left| \frac{\pi_i}{\omega_i^k} \right| \middle| i \in [n], \omega_i^k \neq 0, \right\}$
4 Update $\pi \leftarrow \pi - \frac{\pi_{i^*}}{\omega_{i^*}^k} \omega^k$
5 **for** $k' \in \{k+1, \ldots, \ell\}$ **do**
6 Update $\omega^{k'} \leftarrow \omega^{k'} - \frac{\omega_{i^*}^{k'}}{\omega_{i^*}^k} \omega^k$
7 **return** $\sum_{i\in[n]} \pi_i x \leq \pi_0$

We use Algorithm 1 to transform a recyclable inequality $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$ into an equivalent inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ that satisfies the conditions of Theorem 2. For given equations $\sum_{i\in[n]} \omega_i^k x_i = \omega_0^k$, the algorithm performs a specialized Gaussian elimination on $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$. We already noted above that having many zero coefficients is beneficial for obtaining facet-defining recycled inequalities. Therefore, for each equation given by ω^k, Algorithm 1 subtracts in line 4 a multiple $\frac{\pi_{i^*}}{\omega_{i^*}^k}$ of ω^k from π such that the coefficient π_{i^*} becomes zero. Note that the index i^* is chosen with respect to a bottleneck condition in line 3. This makes sure that $\frac{\pi_{i^*}}{\omega_{i^*}^k}$ is the multiple with the smallest absolute value such that one coefficient π_i with $\omega_i^k \neq 0$ becomes zero. This has two desirable implications. First, if π_i was positive before, then it will be non-negative after subtracting $\frac{\pi_{i^*}}{\omega_{i^*}^k} \omega_i^k$. Second, if π_i was already zero before, then it will still be zero afterwards. This implies that if $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$

is recyclable and uncertainty-exclusive, then this also holds for the resulting $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$. After eliminating π_{i^*}, we make sure in line 6 that $\omega^k_{i^*} = 0$ holds for all remaining equations such that i^* will be different in every iteration. By doing so, we guarantee that we have at least ℓ indices $i \in [n]$ with $\pi_i = 0$ at the end of Algorithm 1.

The following proposition uses Algorithm 1 to generalize Corollary 1 for lower-dimensional $\mathcal{C}^{\text{ROB}}$.

Proposition 2. *Let $\sum_{i\in[n]} \pi'_i x_i \leq \pi'_0$ be a recyclable, uncertainty-exclusive inequality such that $F(\pi)$ is either a facet of $\mathcal{C}^{NOM}$ or $F(\pi) = \mathcal{C}^{NOM}$. Let furthermore ℓ be the maximum number of equations $\sum_{i\in[n]} \omega^k_i x_i = \omega^k_0$ for $k \in [\ell]$ such that $\left\{\pi', \omega^1, \ldots, \omega^\ell\right\}$ are linearly independent. Then Algorithm 1 computes an equivalent recyclable inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ whose Recycled Inequality (1) is facet-defining for $\mathcal{C}^{ROB}$.*

Proof. Since the returned inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ from Algorithm 1 is recyclable and uncertainty-exclusive, it only remains to show by Theorem 2 that $\dim\left(\text{proj}_S\left(F(\pi)\right)\right) = |S|-1$ holds, with $S = \{i \in [n] | \pi_i > 0\}$. Due to the choice of ℓ, we have

$$\ell = \begin{cases} n - \dim\left(\mathcal{C}^{\text{NOM}}\right) - 1 & \text{, if } F(\pi) = \mathcal{C}^{\text{NOM}} \\ n - \dim\left(\mathcal{C}^{\text{NOM}}\right) & \text{, otherwise.} \end{cases}$$

Hence, there exists a set $\left\{x^1, \ldots, x^{n-\ell}\right\} \subseteq F(\pi)$ of affinely independent vectors. Let furthermore $T \subseteq [n] \setminus S$ consist of the ℓ indices i^* that were chosen in Algorithm 1. We show that the vectors $\left\{\text{proj}_{[n]\setminus T}\left(x^1\right), \ldots, \text{proj}_{[n]\setminus T}\left(x^{n-\ell}\right)\right\}$ are affinely independent, i.e., $\dim\left(\text{proj}_{[n]\setminus T}\left(F(\pi)\right)\right) \geq n - |T| - 1$. Since $S \subseteq [n] \setminus T$ holds, this implies $\dim\left(\text{proj}_S\left(F(\pi)\right)\right) \geq |S| - 1$. Furthermore, since the equation induced by π is only on the variables $\{x_i | i \in S\}$, we have $\dim\left(\text{proj}_S\left(F(\pi)\right)\right) < |S|$, which then proves the proposition.

Assume that the projections $\text{proj}_{[n]\setminus T}\left(x^j\right)$ are not affinely independent. Then there exist coefficients $\lambda \in \mathbb{R}^{n-\ell}$ with $\lambda \neq 0$, $\sum_{j\in[n-\ell]} \lambda_j = 0$, and $\sum_{j\in[n-\ell]} \lambda_j x^j_i = 0$ for all $i \in [n]\setminus T$. Consider a fixed but arbitrary index $i^* \in T$. Since i^* was chosen in Algorithm 1, there exists an equation $\sum_{i\in[n]} \omega^k_i x_i = \omega^k_0$ with $\omega^k_{i^*} \neq 0$. Without loss of generality, we can assume $\omega^k_{i^*} = 1$ and obtain

$$x^j_{i^*} = \omega^k_0 - \sum_{i\in[n]\setminus\{i^*\}} \omega^k_i x^j_i$$

for all $j \in [n-\ell]$, and thus

$$\begin{aligned}\sum_{j\in[n-\ell]} \lambda_j x^j_{i^*} &= \sum_{j\in[n-\ell]} \lambda_j \left(\omega^k_0 - \sum_{i\in[n]\setminus\{i^*\}} \omega^k_i x^j_i\right) \\ &= \omega^k_0 \underbrace{\sum_{j\in[n-\ell]} \lambda_j}_{=0} - \sum_{i\in[n]\setminus\{i^*\}} \omega^k_i \underbrace{\sum_{j\in[n-\ell]} \lambda_j x^j_i}_{=0} \qquad = 0.\end{aligned}$$

However, as this applies for all $i^* \in T$, we have $\sum_{j\in[n-\ell]} \lambda_j x^j_i = 0$ for all $i \in [n]$. This implies that the vectors $\left\{x^1, \ldots, x^{n-\ell}\right\}$ are affinely dependent, which contradicts their choice and completes the proof. □

Note that we do not always know the dimension of $\mathcal{C}^{\text{NOM}}$ in practice, let alone all equations $\sum_{i\in[n]} \omega^k_i x_i = \omega^k_0$. We tested Algorithm 1 using the already present equations in the constraint matrix $Ax \leq b$ of the robust instances generated from the MIPLIB 2017, which we use in our computational study in Section 6.6. Interestingly, we observed no improvement in the dual bound provided by the linear relaxation compared to the setting in which we didn't use Algorithm 1. Thus, Algorithm 1 is more of a theoretical tool for Proposition 2, as practitioners seem to usually model their problems such that we can immediately derive strong recycled inequalities.

Now that we have established a good theoretical understanding of the strength of recycled inequalities, we discuss in the next section how to use them in practice.

4 Separating Recycled Inequalities

In the previous section, we have seen that recycling can yield a vast number of facet-defining inequalities. For example, in the case of the independent set problem, every clique inequality can be recycled to a facet-defining inequality. Therefore, potentially exponentially many facet-defining recycled inequalities exist, which raises the need for an efficient separation.

4.1 Separation of Recycled Constraints

A straightforward separation approach is to simply recycle the constraints $Ax \leq b$ of the nominal problem. Given a row $\sum_{i\in[n]} a_{ji} x_i \leq b_j$ of the constraint matrix, we first remove all negative entries on the left-hand side. Since the variables x_i are binary,

$$\sum_{i\in[n]:a_{ji}\geq 0} a_{ji} x_i \leq b_j - \sum_{i\in[n]:a_{ji}<0} a_{ji}$$

is a recyclable inequality for $\mathcal{C}^{\text{ROB}}$. We may either add the corresponding recycled inequalities directly to the formulation $\mathcal{P}^{\text{ROB}}$ or precalculate and store

them for later separation during branch and cut. In both cases, we restrict ourselves to inequalities with

$$\sum_{i\in[n]:a_{ji}\geq 0} a_{ji} > b_j - \sum_{i\in[n]:a_{ji}<0} a_{ji},$$

as the corresponding recycled inequality is otherwise dominated by the constraints $p_i + z \geq \hat{c}_i x_i$. When using the precalculated recyclable inequalities for separation to cut off a fractional solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\mathrm{ROB}}$, we also make sure to only include variables x_i with $\hat{c}_i\tilde{x}_i - \tilde{p}_i \geq 0$, as this maximizes the violation $\sum_{i\in[n]} \pi_i (\hat{c}_i\tilde{x}_i - \tilde{p}_i) - \pi_0\tilde{z}$ of the recycled inequality. That is, we iterate over every row in the constraint matrix $Ax \leq b$ and recycle

$$\sum_{\substack{i\in[n]:a_{ji}\geq 0,\\ \hat{c}_i\tilde{x}_i-\tilde{p}_i\geq 0}} a_{ji}x_i \leq b_j - \sum_{i\in[n]:a_{ji}<0} a_{ji}$$

if the resulting recycled inequality is violated. We will see in our computational study that this simple approach already improves the solvers performance drastically in many cases.

4.2 Separation of Recycled Cuts

Another approach is to benefit from the research on the nominal problem and recycle well studied cutting planes, i.e., inequalities that are valid for the convex hull $\mathcal{C}^{\mathrm{NOM}}$, but not for the linear relaxation $\mathcal{P}^{\mathrm{NOM}}$. Let $\Pi \subseteq \mathbb{R}^{n+1}_{\geq 0}$ be such that $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ is a recyclable inequality for all $\pi \in \Pi$. When separating inequalities to cut off a fractional solution to the nominal problem $\tilde{x} \in \mathcal{P}^{\mathrm{NOM}}$, we would usually search for a $\pi \in \Pi$ with positive violation $\sum_{i\in[n]} \pi_i\tilde{x}_i - \pi_0 > 0$. When separating recycled inequalities for a given solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\mathrm{ROB}}$, we require $\sum_{i\in[n]} \pi_i (\hat{c}_i\tilde{x}_i - \tilde{p}_i) - \pi_0\tilde{z} > 0$ instead. Note that the coefficients $\hat{c}_i\tilde{x}_i - \tilde{p}_i$ and $\tilde{z}$ are fixed in this case. Therefore, the same algorithms for finding a violated nominal inequality can be applied for separating violated recycled inequalities, provided these do not rely on some special structure of the objective function of the separation problem. In our computational study, we will test a heuristic separation of recycled clique inequalities for the robust independent set problem. We will show that these facet-defining recycled inequalities improve the formulation significantly.

4.3 Exact Separation via Recycling Combined Constraints

Obviously, an exact separation of violated recycled inequalities is $\mathcal{NP}$-hard in general. For example, in the case of recycled clique inequalities, an exact separation would require solving a maximum weighted clique problem. However, we can separate recycled inequalities from valid inequalities for $\mathcal{P}^{\mathrm{NOM}}$ in polynomial time in the size of the constraint matrix $Ax \leq b$ via solving an LP. In particular, if we already know the convex hull of the nominal problem, i.e.,

$\mathcal{C}^{\mathrm{NOM}} = \mathcal{P}^{\mathrm{NOM}}$, then an exact separation of recycled inequalities can be done in polynomial time. To see this, note that an inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ is valid for $\mathcal{P}^{\mathrm{NOM}}$ if and only if it can be expressed as a conic combination of the rows in $Ax \leq b$ as well as the box constraints $x_i \leq 1$ and $-x_i \leq 0$. That is, we have $\nu_i - \mu_i + \sum_{j\in[m]} a_{ji}\lambda_j = \pi_i$ and $\sum_{i\in[n]} \nu_i + \sum_{j\in[m]} b_j\lambda_j = \pi_0$ for some $\lambda \in \mathbb{R}^m_{\geq 0},\ \mu, \nu \in \mathbb{R}^n_{\geq 0}$. Hence, there exists a recyclable inequality for $\mathcal{P}^{\mathrm{NOM}}$ whose recycled inequality is violated by $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\mathrm{ROB}}$ if and only if the following separation LP has a solution with positive objective value

$$
\begin{aligned}
\max\ & \sum_{i\in[n]} \pi_i \left(\hat{c}_i \tilde{x}_i - \tilde{p}_i\right) - \pi_0 \tilde{z} \\
\text{s.t. } & \nu_i - \mu_i + \sum_{j\in[m]} a_{ji}\lambda_j = \pi_i && \forall i \in [n] \\
& \sum_{i\in[n]} \nu_i + \sum_{j\in[m]} b_j \lambda_j = \pi_0 \\
& \pi \in \mathbb{R}^{n+1}_{\geq 0},\ \lambda \in \mathbb{R}^m_{\geq 0},\ \mu, \nu \in \mathbb{R}^n_{\geq 0}.
\end{aligned}
\tag{SLP}
$$

As the optimal objective value of the SLP is either zero or unbounded due to arbitrary scaling, we normalize the recyclable inequality $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ by fixing $\pi_0 = 1$. This imposes no restriction on finding violated recycled inequalities, as we always have $\pi_0 > 0$ for all relevant recyclable inequalities and can thus achieve $\pi_0 = 1$ via scaling. This is because the left-hand side $\sum_{i\in[n]} \pi_i x_i$ is non-negative, and thus $\pi_0 = 0$ would imply $x_i = 0$ for all $x \in \mathcal{P}^{\mathrm{NOM}}$ and $i \in [n]$ with $\pi_i > 0$. In both cases, the right-hand side of the recycled inequality $\sum_{i\in[n]} \pi_i \hat{c}_i \tilde{x}_i$ would always be zero, which renders the inequality void.

Remember that recycling is a homomorphism on the set of recyclable inequalities. Hence, recycling a conic combination of inequalities, as given by a solution to SLP, is only reasonable if some of the combined inequalities are not recyclable themselves. That is, if we have $\pi = \pi^1 + \pi^2$, with π^1, π^2 linearly independent and both define recyclable inequalities, then it would be better to recycle each of these inequalities separately. In practice, this is guaranteed by fixing $\pi_0 = 1$, as a combination of π^1 and π^2 is never a vertex of SLP and can only be an optimal solution if the violation of their respective recycled inequalities is equal.

We observe two issues when using SLP for separation in practice. First, solving SLP is relatively time consuming if the number of inequalities is large. Second, we obtain only one optimal solution when solving SLP, and can thus only separate one recycled inequality at a time, although MILP solvers usually perform better when several cuts are added at once. The following proposition helps in this regard, showing that we can partition the constraints into sets that can be considered independently for combination. Doing so, we can solve one smaller LP for each set of the partition, yielding multiple (possibly violated) recycled inequalities within the same separation round.

Proposition 3. *Let $A = (a_{ji})_{j\in[m],i\in[n]}$ be the left-hand side of the constraints*

$Ax \leq b$ (not including $0 \leq x \leq 1$) and let $G = (V, E)$ be the graph with nodes $V = [m]$ and edges $E = \left\{\{j, j'\} \in \binom{V}{2} \middle| \exists i \in [n] : a_{ji} < 0 < a_{j'i}\right\}$. Let $\{C_1, \ldots, C_k\} \subseteq 2^V$ be the connected components in G. Then every inequality that is recycled from a valid inequality for $\mathcal{P}^{NOM}$ is dominated by inequalities recycled from recyclable inequalities of the form

$$\sum_{i \in [n]} \left(\nu_i^\ell - \mu_i^\ell + \sum_{j \in C_\ell} a_{ji} \lambda_j \right) x_i \leq \sum_{i \in [n]} \nu_i^\ell + \sum_{j \in C_\ell} b_j \lambda_j \tag{2}$$

with $\ell \in [k]$, $\lambda \in \mathbb{R}^m_{\geq 0}$, and $\mu^\ell, \nu^\ell \in \mathbb{R}^n_{\geq 0}$.

Proof. We write $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ as

$$\sum_{i \in [n]} \left(\nu_i - \mu_i + \sum_{j \in [m]} a_{ji} \lambda_j \right) x_i \leq \sum_{i \in [n]} \nu_i + \sum_{j \in [m]} b_j \lambda_j,$$

which is a conic combination of $Ax \leq b$ and $-x \leq 0$ as well as $x \leq 1$ with coefficients $\lambda \in \mathbb{R}^m_{\geq 0}$, $\mu, \nu \in \mathbb{R}^n_{\geq 0}$. We show that if $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ is recyclable, then it is also a conic combination of the recyclable inequalities from the statement.

Note that we can assume $\nu_i = 0$ or $\mu_i = 0$ for all $i \in [n]$, since we would otherwise decrease both and then recycle $\sum_{i \in [n]} \pi_i x_i \leq \pi_0 - \sum_{i \in [n]} \min\{\nu_i, \mu_i\}$ instead. We can also assume $\nu_{i'} = \left(-\sum_{j \in [m]} a_{ji'} \lambda_j\right)^+$, as otherwise $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ would be a combination of $x_{i'} \leq 1$ and the recyclable inequality obtained by decreasing $\nu_{i'}$ to $\left(-\sum_{j \in [m]} a_{ji'} \lambda_j\right)^+$. Moreover, we can assume $\mu_{i'} \in \left\{0, \sum_{j \in [m]} a_{ji'} \lambda_j\right\}$, since both values result in recyclable inequalities and all other values would imply that $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ is a convex combination of these two. Accordingly, we have $\mu_{i'} = 0$ for $\pi_{i'} > 0$ and $\mu_{i'} = \left(\sum_{j \in [m]} a_{ji'} \lambda_j\right)^+$ for $\pi_{i'} = 0$. Thus, we can rewrite $\sum_{i \in [n]} \pi_i x_i \leq \pi_0$ as

$$\begin{aligned}
&\sum_{i \in [n]} \left(\left(-\sum_{j \in [m]} a_{ji} \lambda_j \right)^+ + \sum_{j \in [m]} a_{ji} \lambda_j \right) x_i - \sum_{i \in [n] : \pi_i = 0} \left(\sum_{j \in [m]} a_{ji} \lambda_j \right)^+ x_i \\
&\leq \sum_{i \in [n]} \left(-\sum_{j \in [m]} a_{ji} \lambda_j \right)^+ + \sum_{j \in [m]} \lambda_j b_j.
\end{aligned}$$

Note that $\sum_{j \in [m]} a_{ji} \lambda_j$ and $\sum_{j \in C_\ell} a_{ji} \lambda_j$ have the same sign for all $i \in [n]$ and $\ell \in [k]$. Otherwise, there exist $\ell, \ell' \in [k]$ with $j \in C_\ell$ and $j' \in C_{\ell'}$ such that $a_{ji} < 0 < a_{j'i}$. However, this implies that constraints j and j' are

adjacent in the graph G, and thus $\ell = \ell'$. It follows $\left(\pm\sum_{j\in[m]} a_{ji}\lambda_j\right)^+ = \sum_{\ell\in[k]}\left(\pm\sum_{j\in C_\ell} a_{ji}\lambda_j\right)^+$, and thus we can rewrite $\sum_{i\in[n]}\pi_i x_i \le \pi_0$ again as

$$\sum_{\ell\in[k]}\left(\sum_{i\in[n]}\left(\nu_i^\ell - \mu_i^\ell + \sum_{j\in C_\ell} a_{ji}\lambda_j\right)x_i\right) \le \sum_{\ell\in[k]}\left(\sum_{i\in[n]}\nu_i^\ell + \sum_{j\in C_\ell} b_j\lambda_j\right),$$

with $\mu_i^\ell = \left(\sum_{j\in C_\ell} a_{ji}\lambda_j\right)^+$ for $\pi_i = 0$ and $\mu_i^\ell = 0$ for $\pi_i > 0$ as well as $\nu_i^\ell = \left(-\sum_{j\in C_\ell} a_{ji}\lambda_j\right)^+$.

Thus, the above inequality decomposes into the k Inequalities (2), all of which are recyclable since $\nu_i^\ell - \mu_i^\ell + \sum_{j\in C_\ell} a_{ji}\lambda_j \ge 0$ holds by the definition of ν^ℓ, μ^ℓ. □

In the special case where all constraints are recyclable, the graph G from the proposition above contains no edges. Thus, we don't have to consider any combinations of constraints, but can solely rely on recycling constraints as in Section 4.1.

Corollary 2. *Let all constraints in $Ax \le b$ be recyclable. Then every inequality that is recycled from a valid inequality for $\mathcal{P}^{NOM}$ is dominated by the recycled inequalities from $\sum_{i\in I} a_{ji}x_i \le b_j$ for $j \in [m]$ and $I \subseteq [n]$.*

In the case where we have $\mathcal{P}^{\text{NOM}} = \mathcal{C}^{\text{NOM}}$ and all constraints are recyclable, the above corollary shows together with Proposition 1 that we can separate inequalities of the form $\pi_0 z + \sum_{i\in[n]}\pi_i p_i \ge \sum_{i\in[n]}\pi_i \hat{c}_i x_i$ exactly in linear time. In our computational study, we show the strength of such an exact separation for the example of the robust weighted matching problem on bipartite graphs, for which we have $\mathcal{P}^{\text{NOM}} = \mathcal{C}^{\text{NOM}}$ and all constraints are indeed recyclable.

Even if not all constraints are recyclable, in practice an optimal solution to SLP often corresponds to an already recyclable constraint in $Ax \le b$. Hence, we observe that it is beneficial to first check whether we can separate violated recycled inequalities from constraints, as described in Section 4.1. Only if none of these are violated, we solve SLP to check whether there exists a violated recycled inequality from a combined inequality. We will see in our tests on robust instances generated from the MIPLIB 2017 that solving SLP sometimes yields very strong recycled inequalities, even if recycling the constraints in $Ax \le b$ has no effect at all (cf. Section (6.6)).

5 Partially Recycling of Non-Recyclable Inequalities

Let $\sum_{i\in[n]}\pi_i x_i \le \pi_0$ be a non-recyclable valid inequality. In the last section, we transformed such inequalities into $\sum_{i\in[n]:\pi_i>0}\pi_i x_i \le \pi_0 - \sum_{i\in[n]:\pi_i<0}\pi_i$ for

recycling, by estimating $\pi_i x_i \geq \pi_i$. Intuitively, the resulting recycled inequality seems to be unnecessarily weak if the estimated terms $\pi_i \tilde{x}_i$ are actually (near to) zero for a fractional solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\mathrm{ROB}}$ to be cut off. To resolve this, we propose another procedure, using the recyclable part of generally non-recyclable inequalities.

Note that $\sum_{i\in[n]:\pi_i\geq 0} \pi_i x_i \leq \pi_0$ is a recyclable inequality for the restricted nominal solution space $\left\{x \in \mathcal{C}^{\mathrm{NOM}} \middle| x_i = 0 \ \forall i \in [n] : \pi_i < 0\right\}$, and can thus be recycled to a valid inequality for $\left\{(x,p,z) \in \mathcal{C}^{\mathrm{ROB}} \middle| x_i = 0 \ \forall i \in [n] : \pi_i < 0\right\}$. In order to obtain a valid inequality for $\mathcal{C}^{\mathrm{ROB}}$, we can then lift the fixed variables into the recycled inequality. For this, we compute *lifting coefficients* $\alpha_i \in \mathbb{R}$ for $i \in [n]$ with $\pi_i < 0$ such that

$$\pi_0 z + \sum_{i\in[n]:\pi_i>0} \pi_i p_i \geq \sum_{i\in[n]:\pi_i>0} \pi_i \hat{c}_i x_i + \sum_{i\in[n]:\pi_i<0} \alpha_i x_i$$

is a valid inequality.

In general, one wants to choose maximal lifting coefficients α, such that the *lifted inequality* is as strong as possible. Whether one obtains a facet-defining inequality is not trivial to say, as this not only depends on the inequality to be lifted and the maximality of the lifting coefficients, but also on the considered polyhedron. However, roughly speaking, lifting is more likely to yield a facet-defining inequality if the original inequality is facet-defining for the restricted solution space, where the variables to be lifted are fixed to zero [21, 34, 36]. Using Theorem 2, we can state in which case this applies for our recycled inequalities.

Corollary 3. *Let $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ be valid and $S^+ = \{i \in [n] | \pi_i > 0\}$ as well as $S^- = \{i \in [n] | \pi_i < 0\}$. The recycled inequality $\pi_0 z + \sum_{i\in S^+} \pi_i p_i \geq \sum_{i\in S^+} \pi_i \hat{c}_i x_i$ is facet defining for the restricted solution space $\left\{(x,p,z) \in \mathcal{C}^{ROB} \middle| x_i = 0 \ \forall i \in S^-\right\}$ if $\hat{c}_i > 0$ holds for all $i \in S^+$, i.e., it is uncertainty-exclusive on $\{x_i | i \in S^+\}$, and*

$$\dim\left(\mathrm{proj}_{S^+}\left(\left\{x \in F(\pi) \middle| x_i = 0 \ \forall i \in S^-\right\}\right)\right) = \left|S^+\right| - 1.$$

Hence, the approach of fixing, recycling, and lifting is promising if the original inequality is strong on the variables $\{x_i | i \in S^+\}$ and if we are able to compute good lifting coefficients α. Computing maximal lifting coefficients often involves solving multiple $\mathcal{NP}$-hard optimization problems. For example, when only lifting the variable x_ℓ into the recycled inequality $\pi_0 z + \sum_{i\in S^+} \pi_i p_i \geq \sum_{i\in S^+} \pi_i \hat{c}_i x_i$, we need to solve

$$\begin{aligned} \min\ & \pi_0 z + \sum_{i\in S^+} \pi_i p_i - \sum_{i\in S^+} \pi_i \hat{c}_i x_i \\ \text{s.t.}\ & (x,p,z) \in \mathcal{C}^{\mathrm{ROB}}, x_\ell = 1. \end{aligned}$$

That is, we minimize the slack of the inequality to be lifted while fixing $x_\ell = 1$. This (in our case non-positive) slack is then the maximal lifting coefficient of x_ℓ. This theoretical complexity implies the need for an efficient heuristic approach. The following proposition shows how to compute lifting coefficients by solving a sequence of easy fractional knapsack problems.

Proposition 4. *Let $\sum_{i\in[n]} \pi_i x_i \leq \pi_0$ be a valid inequality for $\mathcal{C}^{NOM}$ with $\pi_0 \geq 0$ and $S^+ = \{i \in [n] | \pi_i > 0\}$ as well as $S^- = \{i \in [n] | \pi_i < 0\}$. Consider the fractional knapsack problem*

$$f(y) = \max \left\{ \sum_{i\in S^+} \pi_i \hat{c}_i x_i \middle| \sum_{i\in S^+} \pi_i x_i \leq y, x \in [0,1]^n \right\}$$

and let $\alpha_i = f(\pi_0) - f(\pi_0 - \pi_i)$ for $i \in S^-$. Then

$$\pi_0 z + \sum_{i\in S^+} \pi_i p_i \geq \sum_{i\in S^+} \pi_i \hat{c}_i x_i + \sum_{i\in S^-} \alpha_i x_i$$

is a valid inequality for $\mathcal{C}^{ROB}$.

Proof. We sequentially lift the variables $\{x_{i_1}, \ldots, x_{i_k}\} = \{x_i | i \in S^-\}$. After lifting variable x_{i_ℓ}, this yields a valid inequality for the restricted solution space $\left\{(x,p,z) \in \mathcal{C}^{\text{ROB}} \middle| x_{i_{\ell+1}} = \cdots = x_{i_k} = 0\right\}$. Assume that we already lifted variables $x_{i_1}, \ldots, x_{i_{\ell-1}}$ with coefficients $\alpha_1, \ldots, \alpha_{\ell-1}$ and consider the problem of lifting variable x_{i_ℓ}

$$\begin{aligned}
\min\ & \pi_0 z + \sum_{i\in S^+} \pi_i p_i - \sum_{i\in S^+} \pi_i \hat{c}_i x_i - \sum_{j\in[\ell-1]} \alpha_j x_{i_j} \\
\text{s.t. } & x_{i_\ell} = 1 \\
& x_{i_{\ell+1}}, \ldots, x_{i_k} = 0 \\
& (x,p,z) \in \mathcal{C}^{\text{ROB}},
\end{aligned}$$

We relax this problem by only considering the bilinear constraints $p_i + x_i z \geq \hat{c}_i x_i$ as well as $\sum_{i\in S^+} \pi_i x_i \leq \pi_0 - \pi_{i_\ell} - \sum_{j\in[\ell-1]} \alpha_j x_{i_j}$ and allowing all variables but $x_{i_1}, \ldots, x_{i_\ell}$ to be fractional. By assuming that $S \subseteq [\ell - 1]$ defines an optimal choice for the already lifted variables $x_{i_1}, \ldots, x_{i_{\ell-1}}$, with $x_{i_j} = 1$ iff $j \in S$, we obtain the following relaxed lifting problem

$$\begin{aligned}
\min\ & \pi_0 z + \sum_{i\in S^+} \pi_i p_i - \sum_{i\in S^+} \pi_i \hat{c}_i x_i - \sum_{j\in S} \alpha_j \\
\text{s.t. } & \sum_{i\in S^+} \pi_i x_i \leq y \\
& p_i + x_i z \geq \hat{c}_i x_i && \forall i \in [n] \\
& x \in [0,1]^n, p \in \mathbb{R}^n_{\geq 0}, z \in \mathbb{R}_{\geq 0}
\end{aligned} \tag{RLP}$$

with $y = \pi_0 - \pi_{i_\ell} - \sum_{j\in S} \pi_{i_j}$. We will first show that the optimal solution value of the RLP equals $f(\pi_0) - f(y) - \sum_{j\in S} \alpha_j$ for all $y \geq \pi_0$. Afterwards, we show that $S = \emptyset$ is an optimal choice, which proves that $f(\pi_0) - f\left(\pi_0 - \pi_{i_j}\right) = \alpha_j$ is a feasible lifting coefficient.

Since $f(y)$ is a fractional knapsack problem with capacity y, values $\pi_i \hat{c}_i$ and weights π_i, we can sort the variables with respect to non-decreasing $\frac{\pi_i \hat{c}_i}{\pi_i} = \hat{c}_i$

and greedily fill the knapsack until the capacity is reached. Let x^* be such an optimal greedy solution to $f(y)$. We show that x^*, together with appropriate p^*, z^*, is also an optimal solution to RLP. For this, let (x, p, z) be an optimal solution to RLP. We can assume $p_i = (\hat{c}_i - z)^+ x_i$, and thus obtain

$$\pi_0 z + \sum_{i \in S^+} \pi_i p_i - \sum_{i \in S^+} \pi_i \hat{c}_i x_i - \sum_{j \in S} \alpha_j = \pi_0 z - \sum_{i \in S^+} \pi_i \min\{\hat{c}_i, z\} x_i - \sum_{j \in S} \alpha_j.$$

Hence, for fixed z, we have a fractional knapsack problem with values $\pi_i \min\{\hat{c}_i, z\}$ and weights π_i. Since sorting with respect to $\hat{c}_i$ also yields a sorting with respect to $\frac{\pi_i \min\{\hat{c}_i, z\}}{\pi_i} = \min\{\hat{c}_i, z\}$, the above greedy solution x^* is again optimal.

Now, we choose

$$z^* = \min \left\{ z \in \{0, \hat{c}_1, \ldots, \hat{c}_n\} \middle| \sum_{i \in S^+, \hat{c}_i > z} \pi_i x_i^* \leq \pi_0 \right\}$$

together with $p_i^* = (\hat{c}_i - z^*)^+ x_i^*$. We first show that the value of this solution equals $f(\pi_0) - f(y) - \sum_{j \in S} \alpha_j$ and show afterwards that it is optimal.

If $\sum_{i \in S^+} \pi_i \leq \pi_0$ holds then we have $z^* = 0$, and thus

$$\begin{aligned} &\pi_0 z^* + \sum_{i \in S^+} \pi_i p_i^* - \sum_{i \in S^+} \pi_i \hat{c}_i x_i^* - \sum_{j \in S} \alpha_j \\ =& \sum_{i \in S^+} \pi_i (\hat{c}_i - 0)^+ x_i^* - \sum_{i \in S^+} \pi_i \hat{c}_i x_i^* - \sum_{j \in S} \alpha_j = -\sum_{j \in S} \alpha_j. \end{aligned}$$

Since the increased capacity $y \geq \pi_0$ has no effect on the objective value of the relaxed knapsack problem, we also have $f(\pi_0) - f(y) - \sum_{j \in S} \alpha_j = -\sum_{j \in S} \alpha_j$.

Otherwise, if $\sum_{i \in S^+} \pi_i > \pi_0$ holds, we assume $0 =: \hat{c}_0 \leq \hat{c}_1 \leq \cdots \leq \hat{c}_n$ and let $j^* \in \{0, \ldots, n\}$ be the smallest index such that $\sum_{i \in S^+, \hat{c}_i > \hat{c}_{j^*}} \pi_i \leq \pi_0$. It follows $z^* = \hat{c}_{j^*}$ and $x_i^* = 1$ for all $i > j^*$ with $\pi_i > 0$, which implies that (x^*, p^*, z^*) is a solution to RLP of value

$$\begin{aligned} &\pi_0 z^* + \sum_{i \in S^+} \pi_i p_i^* - \sum_{i \in S^+} \pi_i \hat{c}_i x_i^* - \sum_{j \in S} \alpha_j \\ =& \pi_0 \hat{c}_{j^*} + \sum_{i \in \{j^*+1, \ldots, n\} : \pi_i > 0} \pi_i (\hat{c}_i - \hat{c}_{j^*}) - f(y) - \sum_{j \in S} \alpha_j \\ =& \left(\pi_0 - \sum_{i \in \{j^*+1, \ldots, n\} : \pi_i > 0} \pi_i \right) \hat{c}_{j^*} + \sum_{i \in \{j^*+1, \ldots, n\} : \pi_i > 0} \pi_i \hat{c}_i - f(y) - \sum_{j \in S} \alpha_j \\ =& f(\pi_0) - f(y) - \sum_{j \in S} \alpha_j. \end{aligned}$$

Here, the last equation holds since $x_i^* = 1$ for all $i > j^*$ with $\pi_i > 0$ and

$$x_{j^*}^* = \frac{\pi_0 - \sum_{i \in \{j^*+1, \ldots, n\} : \pi_i > 0} \pi_i}{\pi_{j^*}}$$

is an optimal solution to $f(\pi_0)$.

To see that the choice of p^*, z^* is actually optimal, first consider $z' > z^*$ and p' with $p'_i \geq (\hat{c}_i - z')\, x^*_i$. By definition of z^*, we have $\sum_{i \in S^+, \hat{c}_i > z^*} \pi_i x^*_i \leq \pi_0$, and thus

$$\begin{aligned}
\pi_0 z^* + \sum_{i \in S^+} \pi_i p^*_i &= \pi_0 z^* + \sum_{i \in S^+, \hat{c}_i > z^*} \pi_i \left(z' - z^* + \hat{c}_i - z'\right) x^*_i \\
&\leq \pi_0 z^* + \pi_0 \left(z' - z^*\right) + \sum_{i \in S^+, \hat{c}_i > z^*} \pi_i \left(\hat{c}_i - z'\right) x^*_i \\
&\leq \pi_0 z' + \sum_{i \in S^+, \hat{c}_i > z'} \pi_i \left(\hat{c}_i - z'\right) x^*_i \\
&\leq \pi_0 z' + \sum_{i \in S^+} \pi_i p'_i.
\end{aligned}$$

Second, consider $z' < z^*$ with an appropriate p'. Due to the minimality of z^*, we have $\sum_{i \in S^+, \hat{c}_i \geq z^*} \pi_i x^*_i > \pi_0$, and thus

$$\begin{aligned}
\pi_0 z^* + \sum_{i \in S^+} \pi_i p^*_i &= \pi_0 \left(z' + z^* - z'\right) + \sum_{i \in S^+, \hat{c}_i \geq z^*} \pi_i \left(\hat{c}_i - z^*\right) x^*_i \\
&< \pi_0 z' + \sum_{i \in S^+, \hat{c}_i \geq z^*} \pi_i \left(z^* - z' + \hat{c}_i - z^*\right) \\
&\leq \pi_0 z' + \sum_{i \in S^+, \hat{c}_i > z'} \pi_i \left(\hat{c}_i - z'\right) x^*_i \\
&\leq \pi_0 z' + \sum_{i \in S^+} \pi_i p'_i,
\end{aligned}$$

which shows the optimality of z^* and p^*.

We have shown that $f(\pi_0) - f\left(\pi_0 - \pi_{i_\ell} - \sum_{j \in S} \pi_{i_j}\right) - \sum_{j \in S} \alpha_j$ is the optimal value of RLP for some $S \subseteq [\ell - 1]$. Thus, it only remains to show that $S = \emptyset$ is optimal. To see this, note that f is submodular due to the diminishing utility of additional capacity. That is, we have $f(y' + \varepsilon) - f(y') \geq f(y + \varepsilon) - f(y)$ for $y' \leq y$ and $\varepsilon \geq 0$. Since all π_{i_j} are negative, this implies

$$\sum_{j \in S} \left(f\left(\pi_0 - \pi_{i_j}\right) - f\left(\pi_0\right)\right) \geq f\left(\pi_0 - \sum_{j \in S} \pi_{i_j}\right) - f\left(\pi_0\right),$$

and thus we have

$$\begin{aligned}
&f(\pi_0) - f\left(\pi_0 - \pi_{i_\ell} - \sum_{j\in S}\pi_{i_j}\right) - \sum_{j\in S}\alpha_j \\
=&f(\pi_0) - f\left(\pi_0 - \pi_{i_\ell} - \sum_{j\in S}\pi_{i_j}\right) + \sum_{j\in S}\left(f\left(\pi_0 - \pi_{i_j}\right) - f(\pi_0)\right) \\
\geq&f\left(\pi_0 - \sum_{j\in S}\pi_{i_j}\right) - f\left(\pi_0 - \pi_{i_\ell} - \sum_{j\in S}\pi_{i_j}\right) \\
\geq&f(\pi_0) - f(\pi_0 - \pi_{i_\ell}),
\end{aligned}$$

which proves the statement. □

In practice, when cutting off a fractional solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\text{ROB}}$ with a lifted recycled inequality, we again drop all variables x_i from the inequality with $\pi_i > 0$ and $\hat{c}_i\tilde{x}_i > \tilde{p}_i$, as these negatively impact the violation of the recycled inequality. We do this before lifting the variables x_i with $\pi_i < 0$, as doing so restricts the lifting problem RLP, and thus yields potentially better lifting coefficients.

Note that we require $\pi_0 \geq 0$ in the above proposition, as a negative coefficient of the unbounded variable z would imply infinite lifting coefficients α for obtaining a valid inequality. Hence, if the original inequality has $\pi_0 < 0$, we first have to estimate some $\pi_i x_i \geq \pi_i$ with $\pi_i < 0$ to obtain a non-negative right-hand side. This raises the question of which variables should be estimated and which should be lifted. Moreover, even if $\pi_0 \geq 0$ holds, it is reasonable to check whether lifting or estimating a variable yields a higher violation. For example, when cutting off a fractional solution $(\tilde{x}, \tilde{p}, \tilde{z})$ with $\tilde{z} = 0$, we obtain a higher violation by estimating $x_i\pi_i \geq \pi_i$ and recycling $\sum_{i\in S^+}\pi_i x_i \leq \pi_0 - \sum_{i\in S^+}\pi_i$, as the higher coefficient of z in the recycled inequality is irrelevant in this case. Contrary to that, if $\tilde{z} > 0$ and $\tilde{x}_i = 0$ hold, then it is preferable to lift x_i.

Since we add $\alpha_i\tilde{x}_i$ to the violation when lifting and $\pi_i\tilde{z}$ when estimating, we want to lift those variables with $\alpha_i\tilde{x}_i > \pi_i\tilde{z}$. However, when we decide to estimate $\pi_i x_i \geq \pi_i$ for $i \in S \subseteq [n]$, we obtain a new inequality with a greater right-hand side $\pi_0 - \sum_{i\in S}\pi_i$, which influences the lifting coefficients $\alpha_j(S) = f\left(\pi_0 - \sum_{i\in S}\pi_i\right) - f\left(\pi_0 - \sum_{i\in S}\pi_i - \pi_j\right)$ of the other variables x_j, and thus our lifting decision. Let $\{i_1, \ldots, i_k\} = \{i \in [n] | \pi_i < 0\}$ such that $\tilde{x}_{i_1} \geq \cdots \geq \tilde{x}_{i_k}$. Since variables with higher solution values $\tilde{x}_i$ are less preferable for lifting, it is reasonable to assume that a good decision for S consists of variables $x_{i_1}, \ldots, x_{i_j}$ for some $j \in [k]$. Therefore, we first set $S = \emptyset$ and assume that all variables will be lifted. Afterwards, we greedily decide for $i \in \{i_1, \ldots, i_k\}$ whether x_i should better not be lifted and instead added to S. For this, we check whether

$$\pi_i\tilde{z} + \sum_{j\in\{i_1,\ldots,i_k\}\setminus(S\cup\{i\})}\alpha_j(S\cup\{i\})\,\tilde{x}_j \geq \sum_{j\in\{i_1,\ldots,i_k\}\setminus S}\alpha_j(S)\,\tilde{x}_j$$

holds, i.e., whether the change of the violation is positive when not lifting x_i. Note that the values $\alpha_j(S \cup \{i\})$ can be updated efficiently from $\alpha_j(S)$ by greedily extending the solutions of the corresponding fractional knapsack problems.

We use this approach in the following computational study, which shows the practical relevance of recycling in general and also indicates the potential of partially recycling.

6 Computational Study

In this section, we assess the performance of recycled inequalities computationally. We first discuss numerical pitfalls that can occur in practice when using recycled inequalities and present a remedy for these. Afterwards, we lay out our methodology for measuring an algorithm's performance. Using this methodology, we test different aspects of recycling inequalities for different classes of robust combinatorial optimization problems.

With the study of the robust independent set problem in Section 6.3, we examine the contribution of recycling problem specific cuts. For this, we heuristically separate recycled clique inequalities, which are always facet-defining for the robust problem (cf. Section 3). In Section 6.4, we test the recycling of model constraints for the robust bipartite matching problem. Since the standard formulation of the nominal version consists exclusively of recyclable inequalities and also describes the convex hull $\mathcal{C}^{\text{NOM}}$, every non-dominated recycled inequality corresponds to a model constraint (cf. Corollary 2). In Section 6.5, we consider the robust bipartite matching problem with penalties, in which we allow the violation of matching constraints at the cost of a penalty. Using the new model constraints, which are no longer recyclable, we test the partially recycling of non-recyclable inequalities.

After considering the combinatorial problems above, we evaluate the practical relevance of recycling on a broad set of robustified real world instances from the MIPLIB 2017 [20], which have been used for benchmarking in [14]. For these instances, we also test the generic approach of separating recycled inequalities via solving SLP.

All algorithms have been implemented in Java 11 and are performed on a single core of a Linux machine with an Intel® Core™ i7-5930K CPU @ 3.50GHz, with 4 GB RAM reserved for each calculation. All LPs and MILPs are solved using Gurobi version 9.5.0 [23] in single thread mode and all other settings at default. Furthermore, we use a time limit of 3600 seconds for each algorithm and instance.

All implemented algorithms [18] and generated test instances [19] are freely available online.

6.1 Dealing with Numerical Issues

MILP solvers relying on numeric arithmetic have to face the threat of numerical instability, leading to inconsistent results. One source of numerical instability is a constraint matrix $Ax \leq b$ with a high range in the order of magnitude of coefficients a_{ij}, e.g., with $a_{11} = 10^{-4}$ and $a_{12} = 10^{10}$. E.g., Gurobi recommends that the range of coefficients in the constraint matrix should be within six orders of magnitude [22]. In the case of recycled inequalities $\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \hat{c}_i x_i$, the coefficients $\pi_i \hat{c}_i$ on the right-hand side might violate this desirable property if $\hat{c}_i$ and π_i are simultaneously very large or very small. As a consequence, we observed for three instances in our computational study on the MIPLIB that sub-optimal solutions were reported as optimal. To tackle this problem, we scale the deviations $\hat{c}_i$ as well as the variables p, z in an attempt to reduce the range of coefficients in the recycled inequalities. Let $\hat{c}_{\max} = \max\{\hat{c}_1, \ldots, \hat{c}_n\}$ and $\hat{c}_{\min} = \min\{\hat{c}_i | \hat{c}_i > 0\}$ be the maximum and minimum (proper) deviations. If $\hat{c}_{\max}$ is very large and $\hat{c}_{\min}$ simultaneously very small, then our problem is predisposed to be numerically unstable anyway. However, if both are either very large or very small, then we can scale the deviations such that $\hat{c}_{\max}$ and $\hat{c}_{\min}$ are closer to one. For this, we divide all deviations $\hat{c}_i$ by $\lambda = \sqrt{\hat{c}_{\max}\hat{c}_{\min}}$. This implies $\frac{\hat{c}_{\max}}{\lambda}\frac{\hat{c}_{\min}}{\lambda} = 1$, i.e., the scaled maximum and minimum deviation have the same distance to one in orders of magnitudes. To compensate this change, we multiply z and p in the objective function with λ. Thus, our new problem, which is equivalent to ROB, reads

$$
\begin{aligned}
\min\ & \lambda\Gamma z + \sum_{i\in[n]} (c_i x_i + \lambda p_i) \\
\text{s.t. } & Ax \leq b \\
& p_i + z \geq \frac{\hat{c}_i}{\lambda} x_i && \forall i \in [n] \\
& x \in \{0,1\}^n, p \in \mathbb{R}^n_{\geq 0}, z \in \mathbb{R}_{\geq 0}
\end{aligned}
$$

and the recycled inequalities are

$$\pi_0 z + \sum_{i\in[n]} \pi_i p_i \geq \sum_{i\in[n]} \pi_i \frac{\hat{c}_i}{\lambda} x_i.$$

This small change resolves the observed issues for the MIPLIB instances. For comparability, we will always use the scaled problem when solving ROB. However, for the sake of simplicity, we will only write down the non-scaled problem in the following sections.

6.2 Performance Indicators

Rating the performance of generic algorithms is not trivial, as different use cases imply different requirements for an algorithm. While we aim to find an

optimal solution as fast as possible for some practical problems, it is important to find a good solution within seconds for other problems. Therefore, we need performance indicators that appropriately reflect the spectrum of use cases.

To reflect the aim of solving problems as fast as possible to optimality, we consider the elapsed time required to solve an instance. As it is standard in the literature, we set the elapsed time to the time limit in case an algorithm is not able to solve an instance within this limit. Note that this favors algorithms that often hit the time limit.

In addition, we use the *primal-dual integral*, which was proposed by Berthold [8] with the aim that this metric "reflects the development of the solution quality over the complete optimization process". The primal-dual integral is defined to be the integral of the gap between the current primal and dual bound for each point in time. Since the optimality gap, as reported by, e.g., Gurobi, is in general not bounded and at the start even infinite, the primal-dual integral is defined over an adapted gap. Let $\overline{v}(t)$ be the primal bound and $\underline{v}(t)$ be the dual bound at time step t, with $\overline{v}(t) = \infty$ or $\underline{v}(t) = -\infty$ respectively if no bound is known. We define the step function

$$g(t) = \begin{cases} 1, & \text{if } \overline{v}(t) = \infty \text{ or } \underline{v}(t) = -\infty \text{ or } \underline{v}(t) \cdot \overline{v}(t) < 0, \\ 0, & \text{if } \underline{v}(t) = \overline{v}(t), \\ \frac{\overline{v}(t) - \underline{v}(t)}{\max\{|\overline{v}(t)|, |\underline{v}(t)|\}}, & \text{else,} \end{cases}$$

with respect to the piecewise constant bounds $\overline{v}(t), \underline{v}(t)$, for which we can easily compute the primal-dual integral

$$G(T) = \int_{t=0}^{T'} g(t) \, \mathrm{d}\, t,$$

with T' being the time at which the algorithm is terminated or finishes. The primal-dual integral reflects improvements of the gap over the whole computation process, and is thus a reasonable additional performance indicator alongside the computation time.

Since displaying our performance indicators for all algorithms and instances is impractical, we give aggregate values using the *shifted geometric mean*, as proposed by Achterberg [1]. This is defined as $\left(\Pi_{i=1}^{k} (v_i + s)^{1/k}\right) - s$ for values $v_1, \ldots, v_k \in \mathbb{R}_{\geq 0}$ and a *shifting parameter* $s \in \mathbb{R}_{\geq 0}$. In the following, we always use $s = 1$ second for computation times and $s = 100\%$ for primal-dual integrals, which corresponds to the integral of one second at maximum gap. Besides computation times and primal-dual integrals, we will also report integrality gaps to compare the strength of the linear relaxation with and without recycled inequalities. For aggregating these, we use the shifted geometric mean with $s = 1\%$.

6.3 Robust Independent Set

To show the effect of recycling a class of well-known valid inequalities in a separation procedure, we consider the robust maximum weighted independent

set problem on a graph G with nodes V and edges E. The robust counterpart of the standard formulation with edge constraints $x_v + x_w \leq 1$ for $\{v, w\} \in E$ reads

$$\begin{aligned} \max \ & \sum_{v \in V} c_v x_v - \left(\Gamma z + \sum_{v \in V} p_v \right) \\ \text{s.t. } & x_v + x_w \leq 1 && \forall \{v, w\} \in E \\ & p_v + z \geq \hat{c}_v x_v && \forall v \in V \\ & x \in \{0,1\}^V, p \in \mathbb{R}^V_{\geq 0}, z \in \mathbb{R}_{\geq 0}. \end{aligned}$$

As seen in Section 3, recycling a clique inequality $\sum_{v \in Q} x_v \leq 1$ yields a facet-defining inequality for all cliques $Q \subseteq V$. We compare the separation of recycled clique inequalities in the root node of the branching tree against the robust default formulation $\mathcal{P}^{\text{ROB}}$, which solely uses the constraints $p_i + z \geq \hat{c}_i x_i$. For this, we use Gurobi's callback to add the recycled inequalities as user cuts [23]. Every time Gurobi invokes the callback in the root node and reports a current optimal fractional solution $(\tilde{x}, \tilde{p}, \tilde{z}) \in \mathcal{P}^{\text{ROB}}$, we heuristically separate cliques $Q \subseteq V$ for which the recycled inequality $z + \sum_{v \in Q} p_v \geq \sum_{v \in Q} \hat{c}_v x_v$ is violated. We do so as in Section 4.2, that is, we heuristically solve maximum weighted clique problems on the graph $G = (V, E)$ with weights $\hat{c}_v \tilde{x}_v - \tilde{p}_v$. To separate many recycled inequalities at once, we extend each node $v \in V$ with $\hat{c}_v \tilde{x}_v - \tilde{p}_v > 0$ greedily to a clique $Q_v \subseteq V$ with $v \in Q_v$. For this, we start with $Q_v = \{v\}$ and then iteratively add $v' \in N(Q_v)$ such that $\hat{c}_{v'} \tilde{x}_{v'} - \tilde{p}_{v'}$ is maximal and non-negative. Finally, we return the corresponding recycled inequality to Gurobi as a user cut if its violation is positive.

As a basis for our instances, we use the graphs of the second DIMACS implementation challenge on the clique problem [25]. Of the 66 DIMACS graphs, we choose the 46 graphs that have at most 500 nodes, as otherwise the nominal problem is already very hard. For each node $v \in V$, we generate independent and uniformly distributed values $c_v \in \{900, \dots, 1000\}$ and correlated deviations $\hat{c}_v = \lceil \xi_v c_v \rceil$, with $\xi_v \in [0.45, 0.55]$ being an independent and uniformly distributed random variable. Since robust problems tend to be hard for Γ being somewhere around half the number of variables with $x_i = 1$ [14], we greedily compute a maximal independent set $S \subseteq V$ and define $\Gamma = \left\lfloor \frac{|S|}{2} \right\rfloor$. Using this procedure, we randomly generate five robust independent set problems for each of the 46 DIMACS graphs, leaving us with 230 robust instances.

We show computational results for the default formulation and the recycling of clique inequalities in Table 1. Here, we see that the shifted geometric mean of the integrality gaps is reduced absolutely by roughly 220% from 1427.09% to 1206.91% when using recycled clique inequalities. For computing these gaps, we use the dual bound obtained by heuristically separating recycled clique inequalities for subsequent linear relaxations until no violated inequalities are found. While the absolute reduction of the integrality gap is quite impressive, the relative reduction does not adequately reflect the strength of the recycled

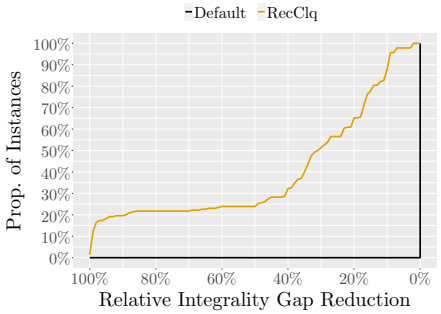


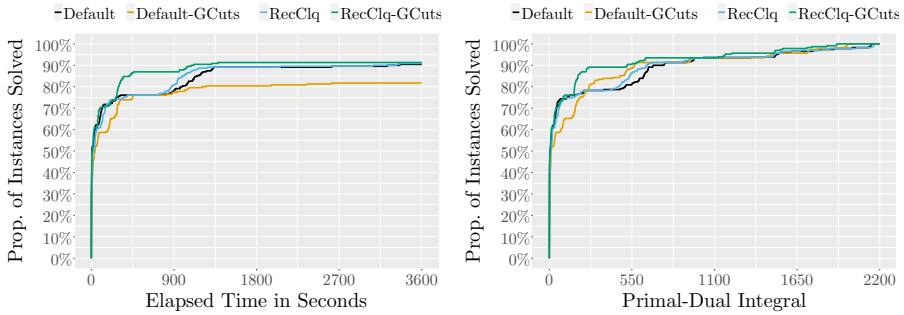
Figure 1: Proportion of instances whose integrality gap has been reduced by at least a specific percentage relative to the clique formulation without recycling.

inequalities. This is due to the large integrality gap of the nominal problem, which constitutes a major part of the total gap. Therefore, we also test an additional formulation for the nominal problem, in which we replace every constraint $x_v + x_w \leq 1$ for an edge $\{v, w\} \in E$ with a constraint $\sum_{v \in Q} x_v \leq 1$ for a clique $Q \subseteq V$ with $\{v, w\} \subseteq Q$. This *clique formulation* has a much tighter linear relaxation compared to the previous *edge formulation*, and thus reduces the contribution of the nominal problem to the integrality gap. Indeed, Table 1 shows that separating recycled clique inequalities reduces the integrality gap by more than half when using the clique formulation. Figure 1 gives a more detailed view of the improvement, by showing for how many instances the integrality gap is reduced by at least a specific percentage. Here, we see that recycling cliques reduces the integrality gap by at least 30% for more than 50% of all instances. Moreover, we have a reduction of 90% for almost 20% of the instances.

Apart from the analysis of the integrality gap, the clique formulation is not of practical interest, as Gurobi seems to be better trained on the edge formulation. Table 1 shows for the edge formulation that we solve one more instance when recycling clique inequalities, but have an increase in the computation time and the primal-dual integral. This seems to be due to some interference with Gurobi's own cutting planes. When disabling Gurobi's cutting planes, recycling is much better than using the default formulation, as it approximately halves

		Default Formulation				Separate Recycled Clique Inequalities			
formulation	GCuts	timeout	time	integral	int Gap	timeout	time	integral	int Gap
edge	enable	22	26.13	14.12	1427.09%	21	33.22	16.99	1206.91%
	disable	42	51.23	22.91		20	20.93	11.72	
clique	enable	61	116.13	67.66	135.50%	64	135.15	77.06	56.49%
	disable	74	168.47	83.30		53	82.07	49.97	

Table 1: Computational results for 230 instances of the robust maximum weighted independent set problem. We use different nominal formulations and test with Gurobi's own cuts enabled or disabled.



(a) Proportion of instances solved within a specific number of seconds.

(b) Proportion of instances for which the primal-dual integral is at most a specific value.

Figure 2: Performance indicators when using the edge formulation.

the computation time and the primal-dual integral. In fact, disabling Gurobi's cuts and using recycled clique inequalities is the overall best performing approach, solving the most instances in the least amount of computation time. This is supported by Figures 2a and 2b. While both recycling approaches and the default solve 75% of all instances within a similar time and have a similar primal-dual integral, especially the recycling approach without Gurobi's cuts performs better on the harder instances.

6.4 Robust Bipartite Matching

To study the recycling of model constraints, we now consider the robust maximum weighted matching problem

$$\begin{aligned} \max \ & \sum_{e \in E} c_e x_e - \left(\Gamma z + \sum_{e \in E} p_e \right) \\ \text{s.t.} \ & \sum_{e \in \delta(v)} x_e \leq 1 && \forall v \in V \\ & p_e + z \geq \hat{c}_e x_e && \forall e \in E \\ & x \in \{0,1\}^E, p \in \mathbb{R}^E_{\geq 0}, z \in \mathbb{R}_{\geq 0} \end{aligned}$$

on a bipartite graph with nodes V and edges E. As mentioned above, the bipartite matching problem has the interesting property that the constraints $\sum_{e \in \delta(v)} x_e \leq 1$ for $v \in V$ and $0 \leq x_e$ for $e \in E$ already define the convex hull of the nominal problem [15]. Moreover, since all constraints are recyclable, the properties from Corollary 2 are fulfilled, which allows for an exact separation of recycled inequalities in linear time, and thus enables us to test their strength to the limit.

We randomly generate instances by first dividing a given set of nodes $V = [n]$ into two partitions $U = \left[\left\lceil \frac{n}{2} \right\rceil\right]$ and $W = \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n\right\}$. Afterwards, we sam-

		Default Formulation				Recycle Constraints				Recycle Constraints + Separate Exactly			
nodes	GCuts	timeout	time	integral	int Gap	timeout	time	integral	int Gap	timeout	time	integral	int Gap
50	enable	0	1.73	0.04	19.532%	0	0.48	0.04	0.326%	0	0.77	0.04	0.319%
	disable	10	3600.00	192.48		0	0.25	0.05		0	0.35	0.05	
100	enable	9	2269.14	3.49	22.820%	0	4.50	0.16	0.319%	0	6.28	0.17	0.316%
	disable	10	3600.00	550.96		0	15.16	0.21		0	16.26	0.20	
150	enable	7	2223.68	2.56	23.660%	0	150.40	0.59	0.269%	0	168.12	0.61	0.265%
	disable	10	3600.00	635.91		6	1887.56	2.51		8	1960.65	2.65	

Table 2: Computational results for the robust maximum weighted bipartite matching problem. We generate ten instances per number of nodes and test with Gurobi's own cuts enabled or disabled.

ple for each node $u \in U$ a random number $\phi_u \in [0, 1]$, modeling the probability with which an edge incident to u exists. Then for every $w \in W$, we add the edge $\{u, w\}$ with probability ϕ_u. Given the constructed graph, we generate weights c_e and deviations $\hat{c}_e$ analogously to the independent set problem. Every weight is a random number $c_e \in \{900, \dots, 1000\}$ and the correlated deviations are $\hat{c}_e = \lceil \xi_e c_e \rceil$ with $\xi_e \in [0.45, 0.55]$. Finally, as the number of edges in a solution will most likely be near to $\frac{n}{2}$, we set $\Gamma = \left\lfloor \frac{n}{4} \right\rfloor$. We use this procedure to generate ten instances for different numbers of nodes $n \in \{50, 100, 150\}$.

Table 2 shows computational results for the robust default formulation and two different approaches for using recycled inequalities. The first approach recycles all constraints $\sum_{e \in \delta(v)} x_e \leq 1$ for $v \in V$. The second approach additionally separates violated inequalities $\sum_{e \in E'} x_e \leq 1$ with $E' \subseteq \delta(v)$ for $v \in V$ in the root node of the branch and bound tree.

It is evident that recycling inequalities is significantly better than solely using the default formulation. We observe a significant strengthening of the formulation, leading to a reduction of the integrality gap to nearly one-hundredth for $n = 150$ nodes. This strength also translates to a higher number of instances solved and much lower computation times. For $n = 150$ with Gurobi's cuts enabled, recycling constraints leads to a reduction of the computation time by 93%. Still, the primal-dual integral is quite low for the default formulation, suggesting that the solver is very close to optimality from the beginning. This changes once we disable Gurobi's cuts. In this case, we are not even able to solve any instance with the default formulation. Furthermore, the primal-dual integral for the default formulation is 253-times as large as the one for the recycling of constraints.

The recycling of dominated inequalities $\sum_{e \in E'} x_e \leq 1$ compared to the sole recycling of constraints $\sum_{e \in \delta(v)} x_e \leq 1$ yields an improvement of the integrality gap. However, as the recycled constraints already perform very well for these instances, the improvement in the linear relaxation is very small. In fact, the minor strengthening of the linear relaxation cannot compensate for the computational load imposed by the additional inequalities, which leads to higher computation times. Our study on the MIPLIB instances will reveal that recycling dominated inequalities can have a much greater effect on the integrality gap, and thus lead to lower computation times.

6.5 Robust Bipartite Matching with Penalties

Until now, we only considered problems for which all valid inequalities are recyclable. In order to test our approach of partially recycling from Section 5, we alter the bipartite matching problems from above such that none of the constraints is recyclable. To this end, we allow a solution to match each node $v \in V$ up to two times. However, when matching v more than once, we have to pay a penalty $c_v > 0$. This yields the following robust problem

$$\begin{aligned}
\max \ & \sum_{e \in E} c_e x_e - \sum_{v \in V} c_v y_v - \left(\Gamma z + \sum_{e \in E} p_e \right) \\
\text{s.t.} \ & \sum_{e \in \delta(v)} x_e - y_v \leq 1 && \forall v \in V \\
& p_e + z \geq \hat{c}_e x_e && \forall e \in E \\
& x \in \{0,1\}^E, y \in \{0,1\}^V, p \in \mathbb{R}^E_{\geq 0}, z \in \mathbb{R}_{\geq 0},
\end{aligned}$$

where $y_v = 1$ is chosen if node v is matched twice. We use the same graphs and parameters as in the previous section, together with random penalties $c_v \in \{450, \dots, 500\}$, which is on average half the value $c_e \in \{900, \dots, 1000\}$ of the edge $e \in E$. Note that we do not consider uncertainties on the penalty coefficients.

Table 3 shows computational results for the robust default formulation as well as the separation of recycled inequalities via estimating $\sum_{e \in \delta(v)} x_e \leq 2$, as in Section 4.1, and the separation of partially recycled inequalities, as in Section 5. Again, we only separate within the root node of the branch and bound tree.

The separation of recycled inequalities via estimation still significantly improves the formulation, cutting the integrality gap in half. However, the effect is clearly weaker compared to the improvement for the original matching problem. The partially recycling approach is considerably stronger, reducing the integrality gap to one-tenth. This is despite the fact that approximately three-fourths of the variables to be fixed and lifted y_v are actually equal to 1 in the computed solutions. Considering that the partially recycled inequalities are especially strong when the lifted variables are zero, this shows that the approach can be very effective.

		Default Formulation				Separate Estimated				Separate Partially Recycled			
nodes	GCuts	timeout	time	integral	int Gap	timeout	time	integral	int Gap	timeout	time	integral	int Gap
50	enable	0	65.67	0.35	18.14%	0	67.20	0.37	10.58%	0	27.07	0.17	1.67%
	disable	10	3600.00	289.70		9	2907.91	72.20		0	127.79	0.48	
100	enable	10	3600.00	17.79	20.89%	10	3600.00	17.73	11.02%	10	3600.00	13.84	2.02%
	disable	10	3600.00	524.65		10	3600.00	252.57		10	3600.00	35.51	
150	enable	10	3600.00	28.15	20.80%	10	3600.00	27.00	9.81%	10	3600.00	21.27	2.03%
	disable	10	3600.00	575.09		10	3600.00	267.58		10	3600.00	45.15	

Table 3: Computational results for the robust maximum weighted bipartite matching with penalties problem. We generate ten instances per number of nodes and test with Gurobi's own cuts enabled or disabled.

Even with the partially recycling procedure, the matching with penalties is apparently much harder to solve. Since all approaches always hit the time limit for $n \in \{100, 150\}$, we cannot compare computation times. However, we still see that the partially recycling results in significantly smaller primal-dual integrals compared to the other approaches, especially when Gurobi's cuts are disabled.

6.6 MIPLIB

So far, we have seen that, given the right setting, recycling inequalities can have a significant impact on the strength of the formulation, and thus on the computational performance. To evaluate the use of recycled inequalities for practical instances, we also perform tests on a broad set of robust instances that have been generated in [14]. The test set contains 804 robust instances based on 67 nominal benchmark instances from MIPLIB 2017 [20]. For each nominal instance, 12 robust instances were generated by combining three different ranges of deviations and four different values for Γ.

We consider four different approaches for integrating recycled inequalities in the optimization process. Note that we don't have any insight into the structure of the nominal problems of our test instances, and thus we only recycle inequalities in a generic fashion based on the constraints in $Ax \leq b$. Our first two approaches are as described in Section 4.1. We call the approach in which we directly add recycled constraints to the default formulation *Recycle Constraints* (short: RecCons). The second approach, in which we separate recycled constraints in the root node of the branch and bound tree, is called *Separate Recycled Constraints* (short: SepCons). For the third approach, we first separate recycled constraints as for SepCons. If this separation is unsuccessful, we solve SLP for a more refined separation of violated recycled inequalities. We call this approach *Separate LP* (short: SepLP). The fourth approach is as SepCons, but we also consider non-recyclable constraints as in Section 5. We call this approach *Separate Partially Recycled* (short: SepPart).

		Default Formulation				Recycle Constraints				Separate Recycled Constraints			
instances	GCuts	timeout	time	integral	int Gap	timeout	time	integral	int Gap	timeout	time	integral	int Gap
all	enable	348	226.52	35.35	15.90%	333	198.96	35.95	8.61%	309	193.73	31.97	7.95%
	disable	497	670.46	99.48		394	368.22	51.14		409	395.64	51.69	
affected	enable	273	303.21	47.59	24.64%	254	252.93	45.17	11.15%	235	246.43	41.63	10.06%
	disable	398	842.92	147.33		295	377.53	60.50		311	416.77	61.85	

		Separate LP				Separate Partially Recycled			
instances	GCuts	timeout	time	integral	int Gap	timeout	time	integral	int Gap
all	enable	308	199.98	33.16	7.07%	316	195.52	32.33	7.94%
	disable	403	405.32	52.97		408	400.17	52.53	
affected	enable	234	251.90	42.65	8.65%	242	247.79	41.99	10.05%
	disable	305	425.30	62.88		310	423.35	63.24	

Table 4: Computational results for robustified MIPLIB instances. We test with Gurobi's own cuts enabled or disabled and show results aggregated for all 804 instances as well as only the 608 on which at least one recycling approach had an effect.

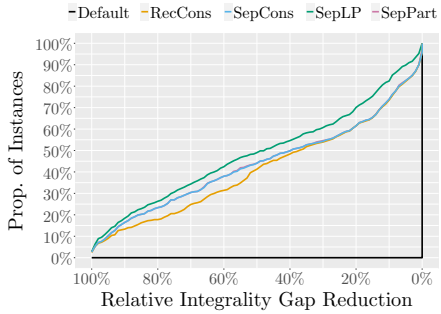


Figure 3: Proportion of the affected instances whose integrality gap has been reduced by at least a specific percentage relative to the default formulation.

Table 4 shows computational results for the four recycling approaches and the default formulation. As for the combinatorial problems in the last sections, recycling inequalities is very effective when Gurobi's cuts are disabled. In this setting, the recycling approaches require between 39% and 45% less time in the shifted geometric mean of the computation time over all instances. When only considering the affected instances, that are the instances for which at least one of the recycling approaches provides a better integrality gap compared to the default formulation, the speed-up is even higher. Out of the 804 instances in our test set, 608 were affected by recycling. For these, the recycling approaches require between 49% and 55% less time, which clearly highlights the practical potential of recycling inequalities.

This performance boost is due to the substantially strengthened linear relaxations. RecCons already cuts the integrality gap nearly in half, from 15.90% to 8.61%. SepCons yields an even better integrality gap, since we also recycle dominated inequalities in this approach. Note that the relative improvement of the integrality gap using SepCons is much larger compared to our observations for the robust bipartite matching problem. We deduce from this that considering dominated inequalities is more important when the coefficients in the constraints are not all the same, as in Example 2 in contrast to the matching problem.

SepPart yields nearly no improvement of the integrality gap compared to SepCons. While we were able to prove the great potential of this approach for the matching with penalties, the considered instances of the MIPLIB apparently don't contain many constraints of the necessary structure with both positive and negative coefficients on the left-hand side.

In contrast, SepLP yields another substantial improvement down to 7.07%. In comparison with the integrality gap of the default formulation, this is a relative reduction of 55% in the shifted geometric mean over all instances. When only considering the affected instances, we even see an improvement from 24.64% to 8.65%, which is a relative reduction of 65%.

To get a better picture of the improvement on the affected instances, we show

in Figure 3 for how many of these the integrality gap is reduced by at least a specific percentage. Note that SepCons and SepPart have nearly identical lines, as they mostly compute the same cuts for these instances.

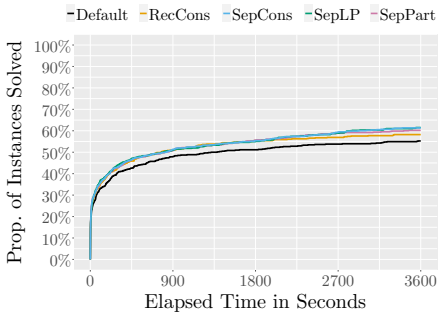
Of the 608 affected instances, RecCons, SepCons, and SepPart close the integrality gap completely for 19 and SepLP even for 20 instances. Interestingly, this includes not only instances with a low default integrality gap, but 10 instances with a default gap of more than 10%, of which one is even $59,756\%$. Moreover, these 10 instances are based on 6 different nominal instances from the MIPLIB 2017. That is, for nearly every eleventh nominal instance, there is at least one corresponding robust instance for which we close the integrality gap completely.

In addition to these extreme cases, we see that SepLP is able to halve the integrality gap for 49% of the instances. Furthermore, SepLP achieves a reduction for some problems on which RecCons, SepCons, and SepPart have no effect. This gives hope that practitioners with a good understanding of their problem might be able to benefit from problem specific fast separations of recycled inequalities that don't correspond directly to the constraints $Ax \leq b$.

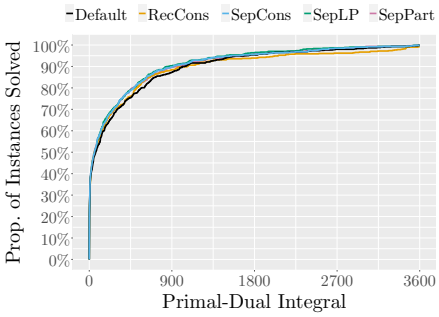
The strong linear relaxations also translate to an improved performance when Gurobi's cuts are enabled, with all recycling approaches solving more instances in shorter time. Table 4 shows that SepCons has the lowest shifted geometric mean for the computation time and primal-dual integral. Compared to the default formulation, the computation times are 14.5% lower for all instances and 18.7% for the affected ones. Moreover, the lower primal-dual integral implies that separating recycled constraints usually improves the performance across the whole optimization process. SepLP solves one instance more, but is on average slightly slower than SepCons. This is because the overhead of handling and solving SLP only pays off for specific instances. SepPart performs worse than SepCons, as both compute almost the same cuts, with SepPart requiring more time doing so. RecCons is on average clearly slower compared to the other recycling approaches because many of the added recycled constraints are actually uninteresting for strengthening the linear relaxation, and thus impose unnecessary computational load due to the bigger constraint matrix. Nevertheless, we will see in the following that RecCons can actually be very useful in practice.

Just like for the integrality gap, Figure 4 shows the distribution of performance indicators. Figures 4a and 4b display for each approach the proportion of affected instances that were solved within a given number of seconds and whose primal-dual integral is below a specific value. We see that the line for SepCons is almost always above the line for Default. Although the difference in the primal-dual integral appears to be small, a paired Wilcoxon signed-rank test [33] reveals for both, the computation time and the primal-dual integral, that the improvement of SepCons is significant with a confidence level above 99%.

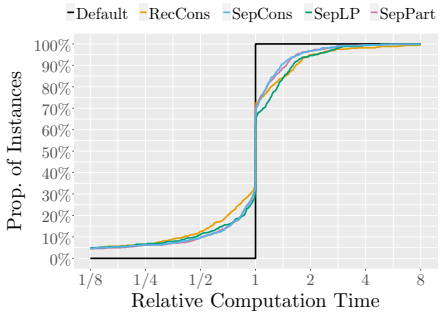
Figures 4c and 4d display the performance indicators of our recycling approaches relative to the default formulation. In Figure 4c, we see that all recycling approaches are 8-times as fast for roughly 5%, while requiring 8-times as



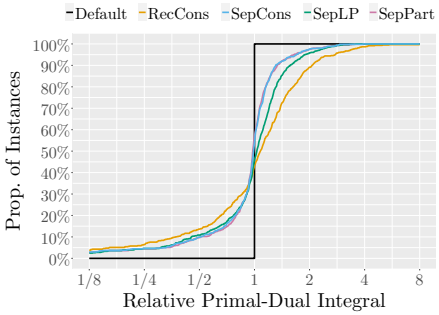
(a) Proportion of instances solved within a specific number of seconds.



(b) Proportion of instances for which the primal-dual integral is at most a specific value.



(c) Proportion of instances whose computation time relative to the default formulation is at most a certain ratio.



(d) Proportion of instances whose primal-dual integral relative to the default formulation is at most a certain ratio.

Figure 4: Performance indicators for the set of affected instances.

much time for only 0.5% of the instances. The most balanced of all approaches is SepCons, which is 2-times as fast for 9.5% and halve as fast for 3.3% of the instances. Similar observations can be made for the primal-dual integral in Figure 4d. However, the most interesting observation about Figure 4c and 4d is that RecCons' and SepLP's performance is quite extreme. Regarding computation time and primal-dual integral, both approaches perform badly for more instances than SepCons, but the number of instances on which they perform very well is also higher. This is no surprise for SepLP, as we already observed above that the higher effort in separating cuts pays off for some specific instances. For RecCons, we see that, given the proper problem structure, recycling constraints directly is not only the most easy approach, but also very efficient. This is good news for the practical use of recycled inequalities, as practitioners will often know whether their optimization problem contains promising recyclable constraints, like clique constraints or almost binding capacity restrictions, that are worth recycling. Recycling precisely these constraints, and not all as we do here for RecCons, might result in a good speed-up for the respective problem. In comparison to SepCons, this yields the advantage that the added recycled inequalities are present from the beginning of the optimization process, which is beneficial because the solver can use the additional information for preprocessing. When integrated into a general robust optimization solver, some further engineering might enable us to combine the stable performance of SepCons with the performance peaks of RecCons. In any case, we have seen that recycling inequalities can help to

7 Conclusions

In this paper, we proposed and analyzed recycled inequalities for robust combinatorial optimization problems with budget uncertainty. Given a valid knapsack inequality for the nominal problem, the corresponding recycled inequality can be derived in linear time, which gives the possibility to reuse model-constraints and well known classes of valid inequalities in order to strengthen the linear relaxation of the robust problem. We highlighted the theoretical strength of such recycled inequalities by proving that they often define facets of the convex hull of the robust problem, even when the underlying valid inequality is dominated.

To make recycled inequalities usable in practice, we discussed different separation procedures that either depend on separation algorithms for classical cutting planes or simply work on the constraint matrix in a generic fashion. One of these separation procedures even implies that recycled inequalities can be separated exactly in polynomial time if the convex hull of the nominal problem is known. Furthermore, we showed that inequalities that are not of the knapsack type can be partially recycled on a restricted solution space and lifted afterwards to obtain a valid inequality for the robust problem.

To test the strength of recycled inequalities and the practicability of their separation, we conducted an extensive computational study on robust versions of three classes of combinatorial problems and a set of nominal benchmark in-

stances. Our experiments show that recycled inequalities are not only interesting from a theoretical point of view, but can also yield a substantial speed-up in the optimization process.

For future research, it would be interesting to further analyze the recycling of non-knapsack inequalities and evaluate whether one can obtain facet-defining robust inequalities from specific classes of nominal inequalities. Furthermore, the effect of recycling should be tested for robust problems with uncertain constraints.

Statements and Declarations

Funding This work was partially supported by the German Federal Ministry of Education and Research (grants no. 05M16PAA) within the project "HealthFaCT - Health: Facility Location, Covering and Transport", the Freigeist-Fellowship of the Volkswagen Stiftung, and the German research council (DFG) Research Training Group 2236 UnRAVeL.

Competing Interests The authors declare that they have no competing interests.

Author Contributions Christina Büsing: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition
Timo Gersing: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization
Arie Koster: Conceptualization, Writing - Review & Editing, Supervision, Project administration, Funding acquisition

Availability of data and material All test instances used in our computational study are published and available for download, sharing, and reuse, see [19].

Code availability All tested algorithms have been implemented in Java and are available on GitHub, see [18].

References

[1] Tobias Achterberg. Constraint integer programming. *Ph. D. Thesis, Technische Universitat Berlin*, 2007.

[2] Eduardo Álvarez-Miranda, Ivana Ljubić, and Paolo Toth. A note on the bertsimas & sim algorithm for robust combinatorial optimization problems. *4OR*, 11(4):349–360, 2013.

[3] Alper Atamtürk. Strong formulations of robust mixed 0–1 programming. *Mathematical Programming*, 108(2-3):235–250, 2006.

[4] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton Series in Applied Mathematics. Princeton university press, USA, 2009.

[5] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[6] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[7] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

[8] Timo Berthold. Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6):611–614, 2013.

[9] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[10] Dimitris Bertsimas, Iain Dunning, and Miles Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.

[11] Dimitris Bertsimas and Melvyn Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.

[12] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[13] Christina Büsing, Timo Gersing, and Arie Koster. Recycling inequalities for robust combinatorial optimization with budget uncertainty. In Alberto Del Pia and Volker Kaibel, editors, *Integer Programming and Combinatorial Optimization. IPCO 2023*, volume 13904 of *Lecture Notes in Computer Science*, pages 58–71, Cham, May 2023. Springer.

[14] Christina Büsing, Timo Gersing, and Arie MCA Koster. A branch and bound algorithm for robust binary optimization with budget uncertainty. *Mathematical Programming Computation*, pages 269–326, 2023.

[15] Michele Conforti, Gérard Cornuéjols, Giacomo Zambelli, et al. *Integer programming*, volume 271. Springer, Cham, 2014.

[16] Matteo Fischetti and Michele Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273, 2012.

[17] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[18] Timo Gersing. Algorithms for robust binary optimization, December 2022.

[19] Timo Gersing, Christina Büsing, and Arie Koster. Benchmark instances for robust combinatorial optimization with budgeted uncertainty, December 2022.

[20] Ambros Gleixner, Gregor Hendel, Gerald Gamrath, Tobias Achterberg, Michael Bastubbe, Timo Berthold, Philipp M. Christophel, Kati Jarck, Thorsten Koch, Jeff Linderoth, Marco Lübbecke, Hans D. Mittelmann, Derya Ozyurt, Ted K. Ralphs, Domenico Salvagnin, and Yuji Shinano. MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library. *Mathematical Programming Computation*, 2021.

[21] Zonghao Gu, George L Nemhauser, and Martin WP Savelsbergh. Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4:109–129, 2000.

[22] Gurobi Optimization, LLC. Advanced user scaling. `https://www.gurobi.com/documentation/9.5/refman/advanced_user_scaling.html`. Accessed: 2022-09-27.

[23] Gurobi Optimization, LLC. Gurobi optimizer reference manual, version 9.5, 2022.

[24] Christoph Hansknecht, Alexander Richter, and Sebastian Stiller. Fast robust shortest path computations. In *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASIcs)*, pages 5:1–5:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[25] David S Johnson and Michael A Trick. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge, October 11-13, 1993*, volume 26. American Mathematical Soc., USA, 1996.

[26] Seulgi Joung and Sungsoo Park. Robust mixed 0-1 programming and submodularity. *INFORMS Journal on Optimization*, 3(2):183–199, 2021.

[27] P Kouvelis and G Yu. *Robust discrete optimization and its applications*. Springer, USA, 1997.

[28] Taehan Lee and Changhyun Kwon. A short note on the robust combinatorial optimization problems with cardinality constrained uncertainty. *4OR*, 12(4):373–378, 2014.

[29] Manfred W Padberg. On the facial structure of set packing polyhedra. *Mathematical programming*, 5(1):199–215, 1973.

[30] Kyungchul Park and Kyungsik Lee. A note on robust combinatorial optimization problem. *Management Science and Financial Engineering*, 13(1):115–119, 2007.

[31] Hanif D Sherali and Warren P Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, volume 31. Springer, New York, 2013.

[32] Allen L Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

[33] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1(6):80–83, 1945.

[34] Laurence A Wolsey. Facets and strong valid inequalities for integer programs. *Operations research*, 24(2):367–372, 1976.

[35] Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, 1999.

[36] Eitan Zemel. Lifting the facets of zero–one polytopes. *Mathematical Programming*, 15:268–277, 1978.