Zuse Institute Berlin

Takustr. 7
14195 Berlin
Germany

Jakob Witzig[1] Timo Berthold

Conflict-Free Learning for Mixed Integer Programming

[1] 0000-0003-2698-0767

ZIB Report 19-59 (November 2019)

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30-84185-0
Telefax: +49 30-84185-125

E-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Conflict-Free Learning for Mixed Integer Programming

Jakob Witzig[1] and Timo Berthold[2]

[1]Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
witzig@zib.de

[2]Fair Isaac Germany GmbH, Takustr. 7, 14195 Berlin, Germany
timoberthold@fico.com

December 2, 2019

Abstract

Conflict learning plays an important role in solving mixed integer programs (MIPs) and is implemented in most major MIP solvers. A major step for MIP conflict learning is to aggregate the LP relaxation of an infeasible subproblem to a single globally valid constraint, the dual proof, that proves infeasibility within the local bounds. Among others, one way of learning is to add these constraints to the problem formulation for the remainder of the search.

We suggest to not restrict this procedure to infeasible subproblems, but to also use global proof constraints from subproblems that are not (yet) infeasible, but can be expected to be pruned soon. As a special case, we also consider learning from integer feasible LP solutions. First experiments of this *conflict-free* learning strategy show promising results on the MIPLIB2017 benchmark set.

1 Introduction

In this paper, we consider *mixed integer programs (MIPs)* of the form

$$\min\{c^{\mathsf{T}}x \mid Ax \geq b,\ \ell \leq x \leq u,\ x_j \in \mathbb{Z}\ \forall j \in \mathcal{I}\}, \tag{1}$$

with objective coefficient vector $c \in \mathbb{R}^n$, constraint coefficient matrix $A \in \mathbb{R}^{m\times n}$, constraint right-hand side $b \in \mathbb{R}^m$, and variable bounds $\ell, u \in \overline{\mathbb{R}}^n$, where $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. Furthermore, let $\mathcal{N} = \{1,\ldots,n\}$ be the index set of all variables and let $\mathcal{I} \subseteq \mathcal{N}$ be the set of variables that need to be integr al in every feasible solution. Moreover, we allow that the constraint right-hand side b can be tightened at any point in time to $\tilde{b} \in \mathbb{R}^m$ with $\bar{b} \geq \tilde{b} \geq b$ and $\bar{b} \in \overline{\mathbb{R}}^m$. Note, we say that $\tilde{b}$ is greater or equal to b, if $\tilde{b}_i \geq b_i$ for all $i = 1,\ldots,m$. An important special case of this general setting is the tightening of the so-called *cutoff bound* $\bar{c}$ during the MIP search. The cutoff bound is either defined by the objective value of current incumbent, i.e., the currently best known, solution $\overline{x}$ or $+\infty$ if no solution has been found yet. It gives rise to the *objective cutoff constraint*

$$-c^{\mathsf{T}}x \geq -\bar{c}. \tag{2}$$

The objective cutoff constraint (2) models an upper bound on all MIP solutions found in the remainder of the search. In the following, we assume that the objective cutoff constraint (2) is explicitly contained in $Ax \geq b$. The computational experiments of this paper will focus on the case that the objective cutoff constraint is the only constraint being tightened during the search, the theoretic background, however, will be given for the general case $\bar{b} \geq \tilde{b} \geq b$. A lower bound on the MIP solution is given by the *linear programming (LP) relaxation* which omits the integrality conditions of (1). The optimal objective value of the LP relaxation provides a lower bound on the optimal solution value of the MIP (1).

In LP-based branch-and-bound (Dakin, 1965; Land and Doig, 1960), the most commonly used method to solve MIPs, the LP relaxation is used for bounding. Branch-and-bound is a divide-and-conquer method which splits the search space sequentially into smaller subproblems that are expected to be easier to solve. During this procedure, we may encounter infeasible subproblems. Infeasibility can be detected by contradicting implications, e.g., derived by domain propagation, by an infeasible LP relaxation, or an LP relaxation that exceeds the objective value of the current incumbent solution. Following our assumption that the objective cutoff constraint is part of the constraint matrix, the latter is just a special case of an infeasible LP relaxation.

1.1 Conflict Analysis in MIP

Modern MIP solvers try to 'learn' from infeasible subproblems, e.g., by applying *conflict graph analysis* or *dual proof analysis*. Conflict graph analysis for MIP has its origin in solving satisfiability problems (SAT) and goes back to (Marques-Silva and Sakallah, 1999). Similar ideas are used in constraint programming, e.g., see (Ginsberg, 1993; Jiang et al., 1994; Stallman and Sussman, 1977). First integration of these techniques into MIP were independently suggested by (Achterberg, 2007b; Davey et al., 2002; Sandholm and Shields, 2006). Dual proof analysis and its combination with conflict graph analysis has been recently studied for both MIPs (Pólik, 2015; Witzig et al., 2017) and mixed integer nonlinear programs (MINLPs) (Lubin et al., 2016; Witzig et al., 2019b). While conflict graph analysis is based on combinatorial arguments, dual proof analysis is a purely LP-based approach. We will briefly describe both concepts in the remainder of this section.

Assume we are given an infeasible node of the branch-and-bound tree defined by the subproblem

$$\min\{c^\mathsf{T} x \mid Ax \geq b,\ \ell' \leq x \leq u',\ x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}\} \tag{3}$$

with local bounds $\ell \leq \ell' \leq u' \leq u$. In LP-based branch-and-bound, the infeasibility of a node/subproblem is either detected by an infeasible LP relaxation or by contradicting implications in domain propagation.

In the latter case, a *conflict graph* gets constructed which represents the logic of how the set of branching decisions led to the detection of infeasibility. More precisely, the conflict graph is a directed acyclic graph in which the vertices represent bound changes of variables and the arcs (v, w) correspond to bound changes implied by propagation, i.e., the bound change corresponding to w is based (besides others) on the bound change represented by v. In addition to these inner vertices which represent the bound changes from domain propagation, the graph features source vertices for the bound changes that correspond to branching decisions and an artificial sink vertex representing the infeasibility. Then, each cut that separates the branching decisions from the artificial infeasibility vertex gives rise to a valid *conflict constraint*. A conflict constraint consists of a set of variables with associated bounds, requiring that in each feasible solution at least one of the variables has to take a value outside these bounds. This corresponds to no-good learning in CP. A variant of this procedure is implemented in `SCIP`, the solver in which we will conduct our computational experiments.

1.2 Deriving Dual Proofs for Infeasible LP Relaxations

If infeasibility is proven by the LP relaxation, however, the proof of infeasibility is given by a ray in the dual space. Consider a node of the branch-and-bound tree and the corresponding subproblem of type (3) with local bounds $\ell \leq \ell' \leq u' \leq u$. The *dual LP* of the corresponding LP relaxation of (3) is given by

$$\max\{y^\mathsf{T} b + r^\mathsf{T}\{\ell', u'\} \mid y^\mathsf{T} A + r^\mathsf{T} = c^\mathsf{T},\ y \in \mathbb{R}^n_{\geq 0},\ r \in \mathbb{R}^n\}, \tag{4}$$

$$r^\mathsf{T}\{\ell', u'\} := \sum_{j \in \mathcal{N}\colon\, r_j^\ell > 0} r_j^\ell \ell'_j - \sum_{j \in \mathcal{N}\colon\, -r_j^u < 0} r_j^u u'_j \tag{5}$$

with $r^\ell, r^u \in \mathbb{R}^n_+$ representing the dual variables on the finite bound constraints. Note, variable x_j can only be tight in at most one bound constraint, thus, r_j^ℓ and r_j^u cannot be non-zero at the same time. For every variable x_j it holds that $r_j = c_j - y^\mathsf{T} A_{\cdot j}$, where $A_{\cdot j}$ denotes the j-th column of A. By LP theory, each ray $(y, r) \in \mathbb{R}^{m+n}$ in the dual space that satisfies

$$\begin{aligned} y^\mathsf{T} A + r^\mathsf{T} &= 0 \\ y^\mathsf{T} b + r^\mathsf{T}\{\ell', u'\} &> 0 \end{aligned} \tag{6}$$

proves infeasibility of (the LP relaxation of) (3), which is a direct consequence of the Farkas Lemma (Farkas, 1902). Hence, there exists a solution (y, r) of (6) with

$$\Delta_{\max}(y^\mathsf{T} A, \ell', u') < y^\mathsf{T} b,$$

where $\Delta_{\max}(y^\mathsf{T} A, \ell', u') := \sum_{y^\mathsf{T} A<0}(y^\mathsf{T} A)\ell' + \sum_{y^\mathsf{T} A>0}(y^\mathsf{T} A)u'$ is called the *maximal activity* of $y^\mathsf{T} A$ w.r.t. the local bounds ℓ' and u'. Consequently, the inequality

$$y^\mathsf{T} Ax \geq y^\mathsf{T} b \tag{7}$$

has to be fulfilled by every feasible solution of the MIP. In the following, this type of constraint will be called *dual proof constraint*. If locally valid constraints are present in subproblem (3), e.g., due to the separation of local cutting planes, the corresponding dual multipliers are assumed to be zero, thereby leaving those constraints out of aggregation (7). Otherwise, the resulting dual proof constraint might not be globally valid anymore.

Observation 1 *Let $b \in \mathbb{R}^m$ be the right-hand side vector and $y^\mathsf{T} Ax \geq y^\mathsf{T} b$ be a dual proof constraint that was derived from an infeasible subproblem. If tightening the (global) right-hand side to $\tilde{b} \in \mathbb{R}^m$ with $\bar{b} \geq \tilde{b} \geq b$, the following holds.*

(i) $y^\mathsf{T} Ax \geq y^\mathsf{T} b$ is still globally valid.

(ii) The dual proof can be strengthened to $y^\mathsf{T} Ax \geq y^\mathsf{T} \tilde{b}$, while preserving global validity.

2 LP-Based Solution Learning

Conflict-driven learning or no-good learning (Prosser, 1993; Zhang et al., 2001), is a fundamental concept in SAT and CP. Besides learning from infeasibility, the methodology of solution-driven learning or good-learning (Chu and Stuckey, 2019; Giunchiglia et al., 2003), i.e., learning from feasibility, has been applied in SAT and CP. Recently, good learning has been successfully applied to nested constraint programming (Chu and Stuckey, 2014). Generally, algorithms for infeasibility learning can be extended to solution learning by pretending that the corresponding cutoff constraint with the updated incumbent was already present for the current subproblem and would prove it to be infeasible (after the incumbent update).

To the best of our knowledge, solution learning has not yet been studied for MIP. Every LP that yields an optimal solution that is MIP-feasible, i.e., feasible for (1), can be used to apply *LP-based solution learning*.

Consider a subproblem (3) with local bounds ℓ' and u'. Moreover, let $x^\star_{\mathrm{LP}}$ be an optimal solution of its LP relaxation that is feasible for MIP (1). If $x^\star_{\mathrm{LP}}$ is an improving solution, i.e., $c^\mathsf{T} x^\star_{\mathrm{LP}} < \bar{c}$, $x^\star_{\mathrm{LP}}$ defines the new incumbent solution. Consequently, the cutoff bound can be updated to

$$\bar{c} = c^\mathsf{T} x^\star_{\mathrm{LP}} - \epsilon$$

with $\epsilon > 0$. Note that MIP solvers using floating point arithmetic typically subtract a small epsilon in the order or magnitude of the used tolerances, e.g., SCIP uses $\epsilon = 10^{-6}$, to enforce strict improvement during the search. If all variables with a non-zero coefficient in the objective function are integral, the minimal improvement in the objective value can be computed by a GCD-like algorithm and used as an epsilon. For example, the objective value always improves by a multiple of 1 if $c_j \in \{-1, 0, 1\}$ with $j \in \mathcal{I}$ and $c_j = 0$ with $j \in \mathcal{N} \setminus \mathcal{I}$. Thus, $\epsilon = 1$ could be used in this case.

Since the objective cutoff constraint (2) is part of the constraint matrix A, the right-hand side vector b changes when updating the cutoff bound. Assume that row index k represents the matrix row associated to the objective cutoff constraint, i.e., $A_{k\cdot}x \geq b_k$ with $-c^\mathsf{T} = A_{k\cdot}$ and $b_k = -\overline{c}$. After an incumbent update, the right-hand side vector changes to $\tilde{b}$ with $\tilde{b}_k = b_k + \overline{c} - c^\mathsf{T} x^\star_{\text{LP}}$ and $\tilde{b}_i = b_i$ for all $i = 1, \ldots, m$ with $i \neq k$. Thus, the feasible LP relaxation defined by the local bounds ℓ' and u' turns infeasible after the update. Henceforth, we can apply both conflict graph analysis and dual proof analysis to learn from LP relaxations that yield integer feasible solutions.

2.1 Implementation

In our implementation, LP-based solution learning is applied whenever the LP relaxation yields a feasible solution, i.e., all integrality conditions are satisfied, that improved the incumbent solution. Note, in principle LP-based solution learning could also be applied for all improving solutions, e.g., found within a heuristic, with an objective value equal to the objective value of the LP relaxation. However, this is not considered in this publication and subject of future research.

Since `sollearning` can immediately be applied when the feasible LP relaxation turns into a bound exceeding LP both *conflict graph analysis* and *dual proof analysis* (Witzig et al., 2019a) are applied in our implementation without introducing much computational overhead.

3 Conflict-Free Dual Proofs

State-of-the-art MIP solvers like SCIP and FICO Xpress do not actively steer the tree seach towards the exploration of infeasible subproblems. Thus, learning from infeasibility information can be considered to be a "byproduct".

Here, we will discuss how the concept of conflict analysis can be extended to learn from subproblems that are not (yet) infeasible. Therefore, we consider dual proofs of form (7) that are *conflict-free*.

Definition 2 (Conflict-Free Dual Proof) *Let $\ell \leq \ell' \leq u' \leq u$ be a set of local bounds and $y^\mathsf{T} Ax \geq y^\mathsf{T} b$ be an aggregation of globally valid constraint weighted by $y \in \mathbb{R}^m_{\geq 0}$. The inequality $y^\mathsf{T} Ax \geq y^\mathsf{T} b$ is called* conflict-free dual proof *with respect to ℓ' and u' if* *i)* $\Delta_{\max}(y^\mathsf{T} A, \ell', u') \geq y^\mathsf{T} b$ *and*

ii) $\exists\ \tilde{b} \in \mathbb{R}$ *with* $\overline{b} \geq \tilde{b} \geq b$ *such that* $\Delta_{\max}(y^\mathsf{T} A, \ell', u') < y^\mathsf{T} \tilde{b}$.

Within a black-box MIP solver (e.g., `SCIP`, `FICO Xpress`, `Gurobi`, and `CPLEX`) that considers the cutoff bound for pruning subproblems, the concept of conflict-free dual proofs simplifies as follows. W.l.o.g. let $A_{m\cdot} x \geq b_m$ be the row assigned to the cutoff bound. Moreover, let $\hat{A} \in \mathbb{R}^{n\times(m-1)}$ be the coefficient matrix without the row assigned to the objective cutoff constraint (2) and $\hat{b} \in \mathbb{R}^{m-1}$ the corresponding constraint right-hand side, i.e.,

$$A := \begin{bmatrix} \hat{A} \\ -c \end{bmatrix} \quad \text{and} \quad b := \begin{bmatrix} \hat{b} \\ -\overline{c} \end{bmatrix}.$$

Let (y, r) be a dual feasible solution for (4) with respect to the local bounds ℓ' and u'. From complementary slackness, it follows that $y_m = 0$. Thus, it holds that

$$\begin{aligned}
& & c^\mathsf{T}\overline{x} &\geq y^\mathsf{T} b + r^\mathsf{T}\{\ell', u'\} \\
&\Leftrightarrow & \overline{c} &\geq y^\mathsf{T} b + (c - y^\mathsf{T} A)\{\ell', u'\} \\
&\Leftrightarrow & \overline{c} &\geq \hat{y}^\mathsf{T}\hat{b} + y_m\overline{c} + (c - (\hat{y}^\mathsf{T}\hat{A} + y_m c))\{\ell', u'\} \\
&\Leftrightarrow & \overline{c} &\geq \hat{y}^\mathsf{T}\hat{b} + (c - \hat{y}^\mathsf{T}\hat{A})\{\ell', u'\} \\
&\Leftrightarrow & (\hat{y}^\mathsf{T}\hat{A} - c)\{\ell', u'\} &\geq \hat{y}^\mathsf{T}\hat{b} - \overline{c} \\
&\Leftrightarrow & y^\mathsf{T} A\{\ell', u'\} &\geq y^\mathsf{T} b \quad \text{with } y_m = 1.
\end{aligned}$$

Consequently, from every dual feasible solution $(y, r) \in \mathbb{R}^{m+n}$ a globally valid constraint

$$y^\mathsf{T} A x \geq y^\mathsf{T} b \tag{8}$$

can be derived. This constraint is generally not violated with respect to ℓ, u and will not be violated with respect to ℓ' and u' either, when the local LP relaxation is feasible. Moreover, let $\underline{c} \in \overline{\mathbb{R}}$ be the current dual bound, i.e., the global lower bound on the MIP solution value. If there exists a $\tilde{c} \in \mathbb{R}$ with $\underline{c} < \tilde{c} < \overline{c}$ such that

$$(\hat{y}^\mathsf{T}\hat{A} - c)\{\ell', u'\} < \hat{y}^\mathsf{T}\hat{b} - \tilde{c},$$

then (8) is a *conflict-free dual proof*. The new global right-hand side is defined by $\tilde{b} := \begin{bmatrix} b - \tilde{c} \end{bmatrix}^\mathsf{T}$.

3.1 Implementation

In our implementation, we maintain a storage of conflict-free dual proofs which is restricted to at most 200 entries. For every conflict-free dual proof we calculate the *primal target bound* $\tilde{c} := \overline{c} + (\Delta_{\max}(y^{\mathsf{T}}A, \ell', u') - \hat{y}^{\mathsf{T}}b)$. The decision whether a conflict-free dual proof is added to the storage for later considerations only depends on the primal target bound. If the storage maintains less than 200 entries, a conflict-free dual proof is accepted if its primal target bound is at least the current global dual bound. In case of a completely filled storage, the newly derived conflict-free dual proof is immediately rejected if its primal target bound is smaller (i.e., worse) than the smallest target bound among all maintained conflict-free dual proofs. Otherwise, the conflict-free dual proof is accepted if it has a larger (i.e., better) target bound are less nonzero entries. With this strategy we aim to prefer short conflict-free dual proofs that tend to propagate earlier with respect to the cutoff bound, i.e., the improvements on the primal side. Whenever a new conflict-free dual proof is derived, all maintained conflict-free dual proofs whose primal target bound become worse than the global dual bound are immediately removed from the storage.

If a new incumbent solution $\overline{x}$ is found, we add at most 10 conflict-free dual proofs for which $\tilde{c} \geq c^{\mathsf{T}}\overline{x}$ holds to the actual solving process, i.e., these (conflict-free) dual proofs become "active" and are considered during the remainder of the search for, e.g., variable bound propagation. Moreover, we allow for slight relaxed primal target bounds. Thus, every conflict-free dual proof for which $\tilde{c} \geq (1+\alpha)c^{\mathsf{T}}\overline{x}$, with $\alpha \geq 0$, holds is considered to become active. In our computational experiments we used $\alpha = 0.1$.

4 Computational Experiments

This section presents a first computational study of solution learning and conflict-free learning for MIP. Our preliminary implementation covers the main features, but is still missing some fine-tuning, as we will see in the following.

We implemented the techniques presented in this paper within the academic MIP solver `SCIP` 6.0.2, using `SoPlex` 4.0.2 as LP solver (Gleixner et al., 2017). In the following, we will refer to `SCIP` with default settings as `default` and to `SCIP` with enabled LP-based solution learning and enabled conflict-free learning as `sollearning` and `conffree`, respectively. To `SCIP` using both techniques simultaneously, we will refer to as `combined`. Our experiments were run on a cluster of identical machines equipped with Intel Xeon E5-2690 CPUs with 2.6 GHz and 128 GB of RAM. A time limit of 7200 seconds was set.

As test set we used the benchmark set of Miplib 2017 (Gleixner et al., 2019) which consists of 240 MIP problems. To account for the effect of performance variability (Koch et al., 2011; Lodi and Tramontani, 2013) all experiments were performed with three different global random seeds. Every pair of MIP problem and seed is treated as an individual observation, effectively resulting in a test set of 720 *instances*. Instances where at least one setting finished with numerical violations are not considered in the following.

Aggregated results on Miplib 2017 comparing all three configurations to `SCIP` with default settings as baseline are shown in Table 1. For every set of instances the group of affected and hard instances is shown. We denote an

Table 1: Aggregated computational results on MIPLIB 2017 benchmark over three random seeds. Improvements by at least 5% are highlighted in bold and blue.

	#	default			conffree			sollearning			combined		
		S	**T**	**N**	**S**	$\mathbf{T_Q}$	$\mathbf{N_Q}$	**S**	$\mathbf{T_Q}$	$\mathbf{N_Q}$	**S**	$\mathbf{T_Q}$	$\mathbf{N_Q}$
MIPLIB 2017													
all	716	369	1124	6069	370	1.001	0.976	368	0.995	0.991	368	1.000	0.973
affected	127	122	567	30265	123	0.993	**0.927**	121	0.975	0.952	121	0.987	**0.902**
≥100s	107	102	969	45972	103	0.987	**0.919**	101	0.970	**0.944**	101	0.979	**0.887**
MIXED BINARY													
all	523	272	1153	5639	273	1.000	0.976	272	0.991	0.983	273	0.996	0.967
affected	83	81	517	26406	82	0.983	**0.913**	81	**0.943**	**0.901**	82	0.958	**0.857**
≥100s	72	70	817	42289	71	0.973	**0.899**	70	**0.934**	**0.886**	71	**0.942**	**0.834**

instance to be hard, when at least one setting takes more than 100 seconds and as affected, if it could be solved by at least one setting and the number of nodes differs among settings. The columns of Table 1 show the number of instances in every groups (#) and the number of solved instances (**S**). For the baseline (`default`) the shifted geometric mean (Achterberg, 2007b) of solving times in seconds (**T**, shift = 1) and explored search tree nodes (**N**, shift = 100) is shown. For `conffree`, `sollearning`, and `combined` relative solving times ($\mathbf{T_Q}$) and nodes ($\mathbf{N_Q}$) compared to `default` are shown. Relative numbers less than 1 indicate improvements.

Our computational experiments indicate that both individual techniques and the combination of them are superior compared to `default` on affected instances. There, we observe an overall speed-up of up to 2.5 % (`sollearning`). At the same time, the tree size reduced by up to 10 % (`combined`). Regarding solving time and tree size, `sollearning` alone is superior to `conffree`. `combined` is superior to both individual settings regarding nodes and almost identical to `sollearning` regarding solving time. For the set of all instances, the number of nodes reduces for all settings, while the impact on running time is almost neutral.

A reason why `conffree` seems to be less powerful than `sollearning` might be the fact that dual proof constraints are known to work better in the neighborhood of the subproblem where they were derived from, which is usually controlled by maintaining a small pool of around 100 dual proof constraints (Achterberg et al., 2016; Witzig et al., 2017, 2019a). In our implementation, the origin of conflict-free dual proofs is not yet considered; this is a direction of future research. Also, conflict-free learning is applied much more frequently than solution learning (every feasible LP relaxation versus every integral LP relaxation), leading to a larger overhead. While `sollearning` only increases the time `SCIP` spends during conflict analysis by marginal 2.4 %, `conffree` learning increases it by a *factor* of 3.4. This shows the need to better choose at which nodes to run conflict-free learning in future implementations.

In our computational study we observed that both techniques perform poorly on instances with general integer variables. One reason for the deterioration might be that for such instances, conflict graph analysis will generate bound disjunction constraints (Achterberg, 2007a) which are generally weaker than conflict constraints on binary variables. Table 1 also presents results when applying the techniques only to (mixed) binary problems. In this case, improvements of over 5 % (`sollearning`) with respect to running time and 14 % (`combined`) with respect to the number of nodes can be observed on affected individual.

5 Conclusion and Outlook

In this paper, we discussed how conflict analysis techniques can be applied to learn from subproblems that are not (yet) proven to be infeasible. For our computational study, we implemented two conflict-free learning techniques, namely conflict-free dual proofs and LP-based solution learning, within the academic MIP solver `SCIP`. The results of our study indicate promising results on the benchmark set of Miplib 2017 when applying conflict-free learning techniques within `SCIP`. In particular, our experiments indicate that solution learning seems to work best on mixed binary instances.

For future research, we plan to consider the locality of derived proofs to increase the efficiency and we plan to predict, e.g., by ML techniques, from which subproblems conflict-free dual proofs should be derived to reduce the overhead.

Acknowledgments

The work for this article has been conducted within the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund number 05M14ZAM).

References

T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007a.

T. Achterberg. Constraint integer programming, 2007b.

T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, and D. Weninger. Presolve reductions in mixed integer programming. Technical Report 16-44, ZIB, Takustr. 7, 14195 Berlin, 2016.

G. Chu and P. J. Stuckey. Nested constraint programs. In B. O'Sullivan, editor, *Principles and Practice of Constraint Programming*, pages 240–255, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10428-7.

G. Chu and P. J. Stuckey. Goods and Nogoods for Nested Constraint Programming. 2019.

R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965. doi: 10.1093/comjnl/8.3.250.

B. Davey, N. Boland, and P. J. Stuckey. Efficient Intelligent Backtracking Using Linear Programming. *INFORMS Journal of Computing*, 14(4):373–386, 2002.

J. Farkas. Theorie der einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik*, 124:1–27, 1902. URL `http://eudml.org/doc/149129`.

M. L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, 1:25–46, 1993.

E. Giunchiglia, M. Narizzano, and A. Tacchella. Backjumping for quantified boolean logic satisfiability. *Artificial Intelligence*, 145(1):99 – 120, 2003. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(02)00373-9. URL `http://www.sciencedirect.com/science/article/pii/S0004370202003739`.

A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, F. Serrano, Y. Shinano, J. M. Viernickel, S. Vigerske, D. Weninger, J. T. Witt, and J. Witzig. The scip optimization suite 5.0. Technical Report 17-61, ZIB, Takustr. 7, 14195 Berlin, 2017.

A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. Christophel, K. Jarck, T. Koch, J. Linderoth, et al. MIPLIB 2017: Data-driven compilation of the 6th mixed-integer programming library. Technical report, Technical report, Optimization Online, 2019.

Y. Jiang, T. Richards, and B. Richards. No-good backmarking with min-conflict repair in constraint satisfaction and optimization. In *PPCP*, volume 94, pages 2–4. Citeseer, 1994.

T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011. doi: 10.1007/s12532-011-0025-9. URL `http://mpc.zib.de/index.php/MPC/article/view/56/28`.

A. H. Land and A. G. Doig. An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28(3):497–520, 1960.

A. Lodi and A. Tramontani. Performance variability in mixed-integer programming. In *Theory Driven by Influential Applications*, pages 1–12. INFORMS, 2013.

M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma. *Extended Formulations in Mixed-Integer Convex Programming*, pages 102–113. Springer International Publishing, Cham, 2016. ISBN 978-3-319-33461-5.

doi: 10.1007/978-3-319-33461-5_9. URL http://dx.doi.org/10.1007/978-3-319-33461-5_9.

J. P. Marques-Silva and K. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions of Computers*, 48(5):506–521, 1999.

I. Pólik. Some more ways to use dual information in MILP. In *International Symposium on Mathematical Programming*, Pittsburgh, PA, 2015.

P. Prosser. HYBRID ALGORITHMS FOR THE CONSTRAINT SATISFACTION PROBLEM. *Computational Intelligence*, 9(3):268–299, aug 1993. doi: 10.1111/j.1467-8640.1993.tb00310.x. URL https://doi.org/10.1111%2Fj.1467-8640.1993.tb00310.x.

T. Sandholm and R. Shields. Nogood learning for mixed integer programming. In *Workshop on Hybrid Methods and Branching Rules in Combinatorial Optimization, Montréal*, 2006.

R. M. Stallman and G. J. Sussman. Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial intelligence*, 9(2):135–196, 1977. doi: http://dx.doi.org/10.1016/0004-3702(77)90029-7.

J. Witzig, T. Berthold, and S. Heinz. Experiments with Conflict Analysis in Mixed Integer Programming. In D. Salvagnin and M. Lombardi, editors, *Integration of AI and OR Techniques in Constraint Programming*, volume 10335 of *LNCS*, pages 211–220, Cham, May 2017. Springer, Springer International Publishing. ISBN 978-3-319-59776-8.

J. Witzig, T. Berthold, and S. Heinz. Computational Aspects of Infeasibility Analysis in Mixed Integer Programming. Technical Report 19-54, ZIB, Takustr. 7, 14195 Berlin, 2019a.

J. Witzig, T. Berthold, and S. Heinz. A status report on conflict analysis in mixed integer nonlinear programming. In L.-M. Rousseau and K. Stergiou, editors, *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 84–94, Cham, 2019b. Springer, Springer International Publishing. ISBN 978-3-030-19212-9.

L. Zhang, C. F. Madigan, M. H. Moskewicz, and S. Malik. Efficient conflict driven learning in a Boolean satisfiability solver. In *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 279–285. IEEE Press, 2001.