



JAKOB WITZIG      TIMO BERTHOLD      STEFAN HEINZ

# **Experiments with Conflict Analysis in Mixed Integer Programming**

Zuse Institute Berlin  
Takustr. 7  
D-14195 Berlin

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

e-mail: [bibliothek@zib.de](mailto:bibliothek@zib.de)  
URL: <http://www.zib.de>

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

# Experiments with Conflict Analysis in Mixed Integer Programming

Jakob Witzig<sup>1</sup>, Timo Berthold<sup>2</sup> and Stefan Heinz<sup>2</sup>

<sup>1</sup>Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany  
witzig@zib.de

<sup>2</sup>Fair Isaac Germany GmbH, Takustr. 7, 14195 Berlin, Germany  
{timoberthold,stefanheinz}@fico.com

November 23, 2016

## Abstract

The analysis of infeasible subproblems plays an important role in solving mixed integer programs (MIPs) and is implemented in most major MIP solvers. There are two fundamentally different concepts to generate valid global constraints from infeasible subproblems. The first is to analyze the sequence of implications obtained by domain propagation that led to infeasibility. The result of the analysis is one or more sets of contradicting variable bounds from which so-called conflict constraints can be generated. This concept has its origin in solving satisfiability problems and is similarly used in constraint programming. The second concept is to analyze infeasible linear programming (LP) relaxations. The dual LP solution provides a set of multipliers that can be used to generate a single new globally valid linear constraint. The main contribution of this short paper is an empirical evaluation of two ways to combine both approaches. Experiments are carried out on general MIP instances from standard public test sets such as MIPLIB2010; the presented algorithms have been implemented within the non-commercial MIP solver SCIP. Moreover, we present a pool-based approach to manage conflicts which addresses the way a MIP solver traverses the search tree better than aging strategies known from SAT solving.

## 1 Introduction: MIP and Conflict Analysis

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

$$c^* = \min\{c^t x \mid Ax \geq b, \ell \leq x \leq u, x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}\}, \quad (1)$$

with objective function  $c \in \mathbb{R}^n$ , constraint matrix  $A \in \mathbb{R}^{m \times n}$ , constraint left-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  $\mathcal{N} = \{1, \dots, n\}$  be the index set of all variables. Let  $\mathcal{I} \subseteq \mathcal{N}$  such that  $x_i \in \mathbb{Z}$  for all  $i \in \mathcal{I}$ , i.e., the set of variables that need to be integral in every feasible solution.

When omitting the integrality requirements, we obtain the *linear program (LP)*

$$c_{LP}^* = \min\{c^t x \mid Ax \geq b, \ell \leq x \leq u, x \in \mathbb{R}^n\}. \quad (2)$$

The linear program (2) is called *LP relaxation* of (1). The LP relaxation provides a lower bound on the optimal solution value of the MIP (1), i.e.,  $c_{LP}^* \leq c^*$ . In LP-based branch-and-bound [11, 18], 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 (hopefully) easier to solve. During this procedure we may encounter infeasible subproblems. Infeasibility can be detected by contradicting implications, e.g., derived by domain propagation, or by an infeasible LP relaxation. Modern MIP solvers try to learn from infeasible subproblems, e.g., by *conflict analysis*. Conflict analysis for MIP has its origin in solving satisfiability problems (SAT) and goes back to [21]. Similar ideas are used in constraint programming, e.g., see [14, 15, 25]. First integrations of these techniques into MIP were independently suggested by [12], [24] and [2]. Further publications suggested to use conflict information for variable selection in branching, to tentatively generate conflicts before branching [3, 16], and to analyze infeasibility detected in primal heuristics [6, 7].

Today, conflict analysis is widely established in solving MIPs. The principal idea of conflict analysis, in MIP terminology, can be sketched as follows.

Given an infeasible node of the branch-and-bound tree defined by the subproblem

$$\min\{c^t x \mid Ax \geq b, \ell' \leq x \leq u', x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}\} \quad (3)$$

with local bounds  $\ell \leq \ell' \leq u' \leq u$ . In LP-based branch-and-bound, the infeasibility of a subproblem is typically detected by an infeasible LP relaxation (see next section) or by contradicting implications.

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. Note that in general, this is not a linear constraint and that by using different cuts in the graph, several different conflict constraints might be derived from a single infeasibility. A variant of conflict analysis close to the one described above is implemented in SCIP, the solver in which we will conduct our computational experiments. Also, a similar implementation is available in the FICO Xpress-Optimizer.

This short paper consists of two parts which are independent but complement each other in practice. The first part of this paper (Section 2) focuses on a MIP technique to analyze infeasibility based on LP theory. We discuss the interaction, differences, and commonalities between conflict analysis and the so-called *dual ray analysis*. Although both techniques have been known before, e.g., [2, 22], this will be, to the best of our knowledge, the first published direct comparison of the two. In the second part (Section 3), we present a new approach to drop conflicts that do not lead to variable bound reductions frequently. This new concept is an alternative to the aging scheme known from SAT. Finally, we present computational experiments comparing the techniques described in Section 2 and 3.

## 2 Analyzing Dual Unbounded Solutions

The idea of conflict analysis is tightly linked to domain propagation: conflict analysis studies a sequence of variable bound implications made by domain propagation routines. Besides domain propagation, there is another important subroutine in MIP solving which might prove infeasibility of a subproblem: the LP relaxation. The proof of LP infeasibility comes in form of a so-called “dual ray”, that is a list of multipliers on the model constraints and the variable bounds. Those give rise to a globally valid constraint that can be used similarly to a conflict constraint. In this section, we discuss the analysis of the LP infeasibility proof in more detail.

### 2.1 Analysis of Infeasible LPs: Theoretical Background

Consider a node of the branch-and-bound tree and the corresponding subproblem

$$\min\{c^t x \mid Ax \geq b, \ell' \leq x \leq u', x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}\} \quad (4)$$

defined by local bounds  $\ell \leq \ell' \leq u' \leq u$ . The *dual LP* of the corresponding LP relaxation of (4) is given by

$$\max\{y^t b + \underline{r}^t \ell' + \bar{r}^t u' \mid A^t y + \underline{r} + \bar{r} \leq c, y, \underline{r} \in \mathbb{R}_{\geq 0}^n, \bar{r} \in \mathbb{R}_{\leq 0}^n\}, \quad (5)$$

where  $A_{.i}$  is the  $i$ -th column of  $A$ ,  $\underline{r}_i = \max\{0, c_i - y^t A_{.i}\}$ , and  $\bar{r}_i = \min\{0, c_i - y^t A_{.i}\}$ . By LP theory each unbounded ray  $(\gamma, \underline{r}, \bar{r})$  of (5) proves infeasibility

of (4). A ray is called unbounded if multiplying the ray with an arbitrary scalar  $\alpha > 0$  will not change the feasibility. Note, in this case it holds

$$\underline{r}_i = \max\{0, -y^t A_{.i}\} \quad \text{and} \quad \bar{r}_i = \min\{0, -y^t A_{.i}\}.$$

Moreover, the Lemma of Farkas states that exactly one of the following two systems is satisfiable

$$(F_1) \quad \left. \begin{array}{l} Ax \geq b \\ \ell' \leq x \leq u' \end{array} \right\} \dot{\vee} \left\{ \begin{array}{l} \gamma^t A + \underline{r} + \bar{r} \leq 0 \\ \gamma^t b + \underline{r}^t \ell' + \bar{r}^t u' > 0 \end{array} \right. \quad (F_2)$$

It follows immediately, that if  $F_1$  is infeasible, there exists an unbounded ray  $(\gamma, \underline{r}, \bar{r})$  of (5) satisfying  $F_2$ . An infeasibility proof of (4) is given by a single constraint

$$\gamma^t Ax \geq \gamma^t b, \quad (6)$$

which is an aggregation of all rows  $A_j$  for  $j = 1, \dots, m$  with weight  $\gamma_j > 0$ . Constraint (6) is globally valid but violated in the local bounds  $[\ell', u']$  of subproblem (4). In the following, this constraint will be called *proof-constraint*.

## 2.2 Conflict Analysis of Infeasible LPs

The analysis of an infeasible LP relaxation, as it is implemented in SCIP, is a hybrid of the theoretical considerations made in Section 2.1 and the analysis of the conflict graph known from SAT. To use the concept of a conflict graph, all variables with a non-zero coefficient in the proof-constraint are converted to vertices of the conflict graph representing bound changes; global bound changes are omitted. Those vertices, called the *initial reason*, are then connected to the artificial sink representing the infeasibility. This neat idea was introduced in [1]. From thereon, conflict analysis can be applied as described in Section 1.

In practice, the proof-constraint is often quite dense, and therefore, it might be worthwhile to search for a sparser infeasibility proof. This can be done by a heuristic that relaxes some of the local bounds  $[\ell', u']$  that appear in the proof-constraint. Of course, the relaxed local bounds  $[\ell'', u'']$  with  $\ell < \ell'' \leq \ell' \leq u' \leq u'' < u$  still need to fulfill

$$\gamma^t b + \underline{r}^t \ell'' + \bar{r}^t u'' > 0.$$

The more bounds can be relaxed that way, the smaller gets the initial reason and consequently the stronger are the derived conflict constraints. Note again that these constraints do not need to be linear, if general integer or continuous variables are present.

## 2.3 Dual Ray Analysis of Infeasible LPs

The proof-constraint is globally valid but infeasible within the local bounds. It follows immediately by the Lemma of Farkas that the *maximal activity*

$$\Delta_{\max}(\gamma^t A, \ell', u') := \sum_{i \in \mathcal{N}: \gamma^t A_i > 0} (\gamma^t A_i) u'_i + \sum_{i \in \mathcal{N}: \gamma^t A_i < 0} (\gamma^t A_i) \ell'_i$$

of  $\gamma^t Ax$  w.r.t. variable bounds  $[\ell', u']$  is strictly less than the corresponding left-hand side  $\gamma^t b$ .

Instead of creating an “artificial” initial reason, the proof-constraint might also be used directly for domain propagation in the remainder of the search. It is a conic combination of global constraints, i.e., it is itself a valid (but redundant) global constraint. In contrast to the method described in Section 2.2, using a dual unbounded ray as a set of weights to aggregate model constraints yields exactly one linear constraint.

The proof-constraint along with an activity argument can be used to deduce local lower and upper variable bounds [2]. Therefore, consider a subproblem with local bounds  $[\ell', u']$ . For any  $i \in \mathcal{N}$  with a non-zero coefficient in the proof-constraint the *maximal activity residual* is given by

$$\Delta_{\max}^i(\gamma^t A, \ell', u') := \sum_{j \in \mathcal{N} \setminus i: \gamma^t A_j > 0} (\gamma^t A_j) u'_j + \sum_{j \in \mathcal{N} \setminus i: \gamma^t A_j < 0} (\gamma^t A_j) \ell'_j,$$

i.e., the maximal activity over all variables but  $x_i$ . Hence, valid local bounds are given by

$$\frac{\gamma^t b - \Delta_{\max}^i(\gamma^t A, \ell', u')}{a_i} \begin{cases} \leq \\ \geq \end{cases} x_i \begin{cases} \text{if } a_i > 0 \\ \text{if } a_i < 0 \end{cases} .$$

This is the so-called bound tightening procedure [9] which is widely used in all major MIP solvers, for all kinds of linear constraints.

Just like the dual ray might be heuristically shrunk to get a short initial reason for conflict analysis, it might be worthwhile to alter the proof-constraint itself before using it for propagation. This can include the application of pre-solving steps such as coefficient tightening to the constraint, projecting out continuous variables or applying mixed-integer rounding to get an alternative globally valid constraint which might be more powerful to propagate.

Finally, instead of generating a valid constraint from the dual ray, one could equivalently use the ray itself to simply check for infeasibility [22, 23] or to estimate the objective change during branch-and-bound and to derive branching decisions therefrom. While in Section 2.2, we described a way to reduce LP infeasibility analysis to conflict analysis based on domain propagation, one could as well try to generate a dual ray by solving the LP relaxation after having detected infeasibility by propagation.

### 3 Managing of Conflicts in a MIP Solver

Maintaining and propagating large numbers of conflict constraints might slow down a solver and create a big burden memory-wise. For instances with a high throughput of branch-and-bound nodes, a solver like SCIP might easily create hundreds of thousands of conflicts within an hour of running time. In order to avoid a slowdown or memory short-coming, an aging mechanism is used within SCIP. Once again, aging is a concept inspired by SAT and CP solving. Every

time a conflict constraint is considered for domain propagation an age counter (individually for each constraint) is increased if no deduction was found. If a deduction is found, the age will be reset to 0. If the age reaches a predefined threshold the conflict constraint is permanently deleted.

In SAT and CP, this mechanism is a well-established method to drop conflict constraints that do not frequently propagate. In the case of MIP solving, there are two main differences concerning the branch-and-bound search. First, domain propagation is most often not the most expensive part of node processing. Second, SAT and CP solvers often use a pure depth-first-search (DFS) node selection, while state-of-the-art MIP solvers use some hybrid between DFS and best-estimate-search or best-first-search (see, e.g., [2, 5, 20]). Therefore, it frequently happens that the node processed next is picked from a different part of the tree.

In the following, we describe a pool-based approach to manage conflict constraints. Here, a pool refers to a fixed-size array that allows direct access to a particular element and which is independent of the model itself. The *conflict pool* is used to manage all conflict constraints, independently whether they were derived from domain propagation or an infeasible LP relaxation. The number of constraints that can be stored within the conflict pool at the same is limited. In our implementation the maximal size of the conflict pool depends on the number of variables and constraints of the presolved problem. However, the pool provides space for at least 1 000 and at most 50 000 conflict constraints at the same time. The conflict pool allows a central management of conflict constraints independently from the model constraints, i.e., they can be propagated, checked or deleted separately, without the need to traverse through all constraints.

To drop conflict constraints that don't lead to deductions frequently we implemented an update-routine that checks the conflict pool regularly, e.g., any time we create the first new conflict at a node. Moreover, we still use the concept of aging to determine the conflict constraints that are rarely used in propagation. Within this update procedure the oldest conflict constraints are removed.

Beside of the regular checks, the conflict pool is updated every time a new improving incumbent solution is found. Conflict constraints might depend on a (previous) best known solution, e.g., when the conflict was created from an LP whose infeasibility proof contained the objective cutoff. Such conflicts become weaker whenever a new incumbent is found and the chance that they lead to deductions becomes smaller the more the incumbent improves. Due to this, for each conflict constraint involving an incumbent solution we store the corresponding objective value. If this value is sufficiently worse than the new objective value, the conflict constraint will be permanently deleted. In our computational experiments (cf. Section 4) we use a threshold of 5%.



## 4 Computational Experiments

In our computational experiments, we compare combinations of the techniques presented in this paper: conflict analysis and dual ray analysis. To the best of our knowledge, most major MIP solvers either use conflict analysis of infeasible LPs and domain propagation (e.g., SCIP, FICO Xpress-Optimizer) or they employ dual ray analysis (e.g., Gurobi, SAS). We will refer to the former as the `conflict` setting and to the latter as the `dualray` setting. We compare those to a setting that uses conflict analysis and dual ray analysis simultaneously, the `combined` setting. Finally, we consider an extension of the `combined` setting that uses a pool for conflict management, the setting `combined+pool`.

All experiments were performed with the non-commercial MIP solver SCIP [13] (git hash 60f49ab, based on SCIP 3.2.1.2), using SoPlex 2.2.1.3 as LP solver. The experiments were run on a cluster of identical machines, each with an Intel Xeon Quad-Core with 3.2 GHz and 48 GB of RAM; a time limit of 3600 seconds was set.

We used two test sets: the MIPLIB2010 [17] benchmark test set and a selection of instances taken from the MIPLIB [8], MIPLIB2003 [4], MIPLIB2010, the COR@L [19] collection, the ALU<sup>1</sup>, and the MARKSHARE [10] test sets. From these we selected all instances for which (i) all of the above settings need at least 100 nodes, (ii) at least one setting finishes within the time limit of 3600 seconds, and (iii) at least one setting analyzes more than 100 infeasible subproblems successfully. We refer to this test set as the CONFLICT set, since it was designed to contain instances for which conflict or dual ray analysis is frequently used.

Aggregated results on the number of generated nodes and needed solving time can be found in Table 1. Detailed results can be found in Table 2 and 5 in the appendix.

We use the `conflict` setting as a base line (since it used to be the SCIP default), for which we give actual means of branch-and-bound nodes and the solving time. For all other settings, we instead give factors w.r.t. the base line. A number greater than one implies that the setting is inferior and a number less than one implies that the setting is superior to the `conflict` setting.

First of all, we observe that solely using dual ray analysis is inferior to using conflict analysis on both test sets and w.r.t. both performance measures. Note that we used a basic implementation of dual ray analysis; a solver that solely relies on it might implement further extensions that decrease this difference in performance, see also Section 5. However, the combination of conflict and dual ray analysis showed some significant performance improvements. We observed a speed-up of 3% and 18% on MIPLIB2010 and CONFLICT, respectively. Moreover, the number of generated nodes could be reduced by 5% and 25%, respectively. Finally, on the CONFLICT test set, the `combined` setting solved one instance more than the `conflict` setting and five more than the `dualray` setting. We take those results as an indicator that the two techniques complement each

---

<sup>1</sup>The instances are part of the contributed section of MIPLIB2003

Test set	conflict			dualray			combined			combined+pool		
	#	$n$	$t$	#	$n_Q$	$t_Q$	#	$n_Q$	$t_Q$	#	$n_Q$	$t_Q$
MIPLIB2010	60	14382	686	57	<i>1.365</i>	<i>1.167</i>	60	0.955	0.977	60	0.957	0.975
CONFLICT	105	16769	143	101	<i>1.616</i>	<i>1.256</i>	106	<b>0.755</b>	<b>0.827</b>	106	<b>0.759</b>	<b>0.829</b>

Table 1: Aggregated computational results. Columns marked with # show the number of solved instances. Columns 3 and 4 show the shifted geometric mean of absolute numbers of generated nodes ( $n$ , shift = 100) and needed solving time in seconds ( $t$ , shift = 10), respectively. All remaining columns show the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Column 3 and 4, respectively.

other nicely. In an additional experiment, we also tested to apply conflict analysis solely from domain propagation or solely from infeasible LPs. Both variants were inferior to the `conflict` setting and are therefore not discussed in detail (cf. Table 4 and 5 in the appendix).

To partially explain the different extent of the improvements on both tests set, we would like to point out that in the MIPLIB2010 benchmark set, there are only 31 instances which fulfill the filtering criteria mentioned above for the `CONFLICT` set. On those, the `combined` setting is 7.2% faster and needs 15.6% less nodes than the `conflict` setting.

Looking at individual instances, there are a few cases for which the `combined` setting is the clear winner, e.g., `neos-849702` `orbnatt350`. For `neos-849702` and `bnatt350`, the `dualray` setting has a timeout, while the `conflict` setting is a factor of 6.2 and 1.83 slower, respectively, than the `combined` setting. At the same time, `ns1766074` shows the largest deterioration from using a `combined` setting, being a factor of 1.63 slower than `conflict` and a factor of 1.06 slower than `dualray`.

As can be seen in Table 1, using a conflict pool in addition to an aging system makes hardly any difference w.r.t. the overall performance.

## 5 Conclusion and Outlook

In this short paper we discussed the similarities and differences of conflict analysis and dual ray analysis in solving MIPs. Our computational study indicates that a combination of both approaches can enhance the performance of a state-of-the-art MIP solver significantly. On instances where the analysis of infeasible subproblems succeeds frequently, the solving time improved by 17.3% and the number of branch-and-bound nodes by 24.5%. In contrast to that, using a pool-based approach in addition to an aging mechanism to manage conflict constraints showed hardly any impact.

There are several instances for which using either dual ray analysis or conflict analysis exclusively outperformed the combination of both. Thus, we will

plan to investigate a dynamic mechanism to switch between both techniques. Furthermore, applying dual ray analysis for infeasibility deduced by domain propagation as well as using more preprocessing (e.g., mixed integer rounding, projecting out continuous variables, etc.) techniques to modify constraints derived from dual ray analysis appear as promising directions for future research.

## 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

- [1] T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007.
- [2] T. Achterberg. Constraint integer programming, 2007.
- [3] T. Achterberg and T. Berthold. Hybrid branching. In W.-J. van Hoes and J. N. Hooker, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 6th International Conference, CPAIOR 2009*, volume 5547 of *Lecture Notes in Computer Science*, pages 309–311. Springer Berlin Heidelberg, May 2009.
- [4] T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):361–372, 2006.
- [5] M. Bénichou, J.-M. Gauthier, P. Girodet, G. Hentges, G. Ribière, and O. Vincent. Experiments in mixed-integer linear programming. *Mathematical Programming*, 1(1):76–94, 1971.
- [6] T. Berthold and A. M. Gleixner. Undercover: a primal MINLP heuristic exploring a largest sub-MIP. *Mathematical Programming*, 144(1–2):315–346, 2014.
- [7] T. Berthold and G. Hendel. Shift-and-propagate. *Journal of Heuristics*, 21(1):73–106, 2015.
- [8] R. E. Bixby, E. A. Boyd, and R. R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.
- [9] A. Brearley, G. Mitra, and H. Williams. Analysis of mathematical programming problems prior to applying the simplex algorithm. *Mathematical Programming*, 8:54–83, 1975.
- [10] G. Cornuéjols and M. Dawande. A class of hard small 0-1 programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 284–293. Springer, 1998.

- [11] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.
- [12] B. Davey, N. Boland, and P. J. Stuckey. Efficient intelligent backtracking using linear programming. *INFORMS Journal of Computing*, 14(4):373–386, 2002.
- [13] G. Gamrath, T. Fischer, T. Gally, A. M. Gleixner, G. Hendel, T. Koch, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, S. Schenker, R. Schwarz, F. Serrano, Y. Shinano, S. Vigerske, D. Weninger, M. Winkler, J. T. Witt, and J. Witzig. The scip optimization suite 3.2. Technical Report 15-60, ZIB, Takustr.7, 14195 Berlin, 2016.
- [14] M. L. Ginsberg. Dynamic backtracking. *Journal of Artificial Intelligence Research*, pages 25–46, 1993.
- [15] Y. Jiang, T. Richards, and B. Richards. No-good backmarking with min-con ict repair in constraint satisfaction and optimization. In *PPCP*, volume 94, pages 2–4. Citeseer, 1994.
- [16] F. Kilinç Karzan, G. L. Nemhauser, and M. W. P. Savelsbergh. Information-based branching schemes for binary linear mixed-integer programs. *Mathematical Programming Computation*, 1(4):249–293, 2009.
- [17] 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.
- [18] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.
- [19] J. T. Linderoth and T. K. Ralphs. Noncommercial software for mixed-integer linear programming. *Integer programming: theory and practice*, 3:253–303, 2005.
- [20] J. T. Linderoth and M. W. Savelsbergh. A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing*, 11(2):173–187, 1999.
- [21] J. P. Marques-Silva and K. Sakallah. Grasp: A search algorithm for propositional satisfiability. *Computers, IEEE Transactions on*, 48(5):506–521, 1999.
- [22] I. Pólik. (re)using dual information in milp. In *INFORMS Computing Society conference*, Richmond, VA, 2015.
- [23] I. Pólik. Some more ways to use dual information in milp. In *International Symposium on Mathematical Programming*, Pittsburgh, PA, 2015.

- [24] T. Sandholm and R. Shields. Nogood learning for mixed integer programming. In *Workshop on Hybrid Methods and Branching Rules in Combinatorial Optimization, Montréal*, 2006.
- [25] 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.

## A Appendix

A detailed overview of all computational results on MIPLIB2010 and CONFLICT test set can be found in Table 2 – 5. For each table we use the `conflict` setting as a base line, for which we give actual means of branch-and-bound nodes and the solving time. For all other settings, we instead give factors w.r.t. the base line. A number greater than one implies that the setting is inferior and a number less than one implies that the setting is superior to the `conflict` setting.

A comparison between the `conflict`, `dualray`, `combined`, and `combined+pool` setting can be found in Table 2 and 3.

In addition, results for applying conflict analysis solely from domain propagation or solely from infeasible LPs can be found in Table 4 and 5.

Table 2: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
30n20b8	51	268	933	<i>18.294</i>	1591.27	<i>5.933</i>	51	1.000	269.69	1.006	51	1.000	269.43	1.005
acc-tight5	920	140	357	<b>0.388</b>	76.88	<b>0.551</b>	1107	<i>1.203</i>	153.85	<i>1.102</i>	1107	<i>1.203</i>	153.74	<i>1.101</i>
afflow40b	163981	1333	160367	0.978	888.21	<b>0.666</b>	135529	<b>0.826</b>	634.29	<b>0.476</b>	135018	<b>0.823</b>	631.47	<b>0.474</b>
air04	218	74	180	<b>0.826</b>	73.71	0.997	218	1.000	69.62	<b>0.942</b>	218	1.000	70.40	0.952
app1-2	2	3600	71	<i>35.500</i>	2100.69	<b>0.584</b>	2	1.000	3600.00	1.000	2	1.000	3600.00	1.000
ash608gpia-3col	7	13	297	<i>42.429</i>	36.69	<i>2.720</i>	7	1.000	13.44	0.996	7	1.000	13.30	0.986
bab5	21993	3600	29410	<i>1.337</i>	3600.00	1.000	24407	<i>1.110</i>	3600.00	1.000	24417	<i>1.110</i>	3600.00	1.000
beasleyC3	832249	3600	1278442	<i>1.536</i>	3600.00	1.000	989653	<i>1.189</i>	3600.00	1.000	952794	<i>1.145</i>	3600.00	1.000
biella1	10755	1711	9961	<b>0.926</b>	1683.80	0.984	9134	<b>0.849</b>	1600.17	<b>0.935</b>	9134	<b>0.849</b>	1503.47	<b>0.879</b>
bienst2	242186	486	169723	<b>0.701</b>	326.13	<b>0.671</b>	148427	<b>0.613</b>	333.62	<b>0.686</b>	148427	<b>0.613</b>	332.34	<b>0.683</b>
binkar10_1	245091	380	245654	1.002	355.63	<b>0.936</b>	274286	<i>1.119</i>	426.07	<i>1.122</i>	274286	<i>1.119</i>	425.82	<i>1.121</i>
bley_xl1	1	238	1	1.000	236.30	0.991	1	1.000	238.00	0.998	1	1.000	225.80	<b>0.947</b>
bnatt350	4433	287	167031	<i>37.679</i>	3600.00	<i>12.540</i>	1913	<b>0.432</b>	156.83	<b>0.546</b>	1913	<b>0.432</b>	157.56	<b>0.549</b>
core2536-691	2335	1295	2051	<b>0.878</b>	1731.98	<i>1.337</i>	2335	1.000	1297.04	1.002	2335	1.000	1297.94	1.002
cov1075	1195289	3600	1193248	0.998	3600.00	1.000	1202635	1.006	3600.00	1.000	1113845	<b>0.932</b>	3600.00	1.000
csched010	398144	3600	386528	0.971	3600.00	1.000	366770	<b>0.921</b>	3600.00	1.000	357893	<b>0.899</b>	3600.00	1.000
danoint	1043564	3600	950242	<b>0.911</b>	3600.00	1.000	1015127	0.973	3600.00	1.000	1014726	0.972	3600.00	1.000
dfn-gwin-UUM	46722	96	46670	0.999	95.01	0.989	46456	0.994	95.95	0.999	46456	0.994	96.38	1.003
eil33-2	865	59	865	1.000	57.51	0.976	865	1.000	59.31	1.006	865	1.000	57.77	0.980
eilB101	12775	406	12775	1.000	408.54	1.006	12775	1.000	405.95	1.000	12775	1.000	409.21	1.008
enlight13	1	1	1	1.000	0.50	1.000	1	1.000	0.50	1.000	1	1.000	0.50	1.000
enlight14	1	1	1	1.000	0.50	1.000	1	1.000	0.50	1.000	1	1.000	0.50	1.000
ex9	1	36	1	1.000	36.33	1.004	1	1.000	35.89	0.992	1	1.000	36.82	1.018
glass4	5039212	3074	6252477	<i>1.241</i>	3600.00	<i>1.171</i>	4069131	<b>0.807</b>	3600.00	<i>1.171</i>	3839095	<b>0.762</b>	3600.00	<i>1.171</i>
gmu-35-40	5403540	3600	6607070	<i>1.223</i>	3600.00	1.000	5371633	0.994	3600.00	1.000	5425130	1.004	3600.00	1.000

cont. on next page

Table 2: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
iis-100-0-cov	88800	631	88800	1.000	630.10	0.999	88800	1.000	630.63	1.000	88800	1.000	630.81	1.000
iis-bupa-cov	180506	2425	180506	1.000	2426.71	1.001	180506	1.000	2635.21	<i>1.086</i>	180506	1.000	2425.22	1.000
iis-pima-cov	8090	241	8090	1.000	241.09	0.999	8090	1.000	242.11	1.003	8090	1.000	240.73	0.998
lectsched-4-obj	2399	269	29448	<i>12.275</i>	3600.00	<i>13.399</i>	9925	<i>4.137</i>	688.75	<i>2.563</i>	9925	<i>4.137</i>	685.39	<i>2.551</i>
m100n500k4r1	3292898	3600	3565574	<i>1.083</i>	3600.00	1.000	3592592	<i>1.091</i>	3600.00	1.000	3566478	<i>1.083</i>	3600.00	1.000
macrophage	878022	3600	865342	0.986	3600.00	1.000	903545	1.029	3600.00	1.000	903961	1.030	3600.00	1.000
map18	307	269	307	1.000	265.56	0.986	307	1.000	271.94	1.010	307	1.000	268.08	0.996
map20	385	256	385	1.000	255.38	0.998	385	1.000	255.75	1.000	385	1.000	256.51	1.003
mcsched	20358	206	18218	<b>0.895</b>	169.02	<b>0.821</b>	20358	1.000	193.54	<b>0.941</b>	20358	1.000	193.74	<b>0.942</b>
mik-250-1-100-1	633761	317	578416	<b>0.913</b>	287.90	<b>0.909</b>	633761	1.000	316.41	0.999	633761	1.000	317.71	1.003
mine-166-5	6651	37	4551	<b>0.684</b>	28.03	<b>0.763</b>	6651	1.000	36.60	0.997	6651	1.000	36.39	0.991
mine-90-10	45418	210	137594	<i>3.030</i>	254.67	<i>1.213</i>	37025	<b>0.815</b>	158.11	<b>0.753</b>	57388	<i>1.264</i>	219.77	1.047
msc98-ip	648	3600	1658	<i>2.559</i>	3600.00	1.000	602	<b>0.929</b>	3600.00	1.000	599	<b>0.924</b>	3600.00	1.000
mspp16	59	2142	121	<i>2.051</i>	2720.47	<i>1.270</i>	59	1.000	2158.66	1.008	59	1.000	2136.90	0.997
mzzv11	7274	1367	8588	<i>1.181</i>	1930.21	<i>1.412</i>	7259	0.998	1278.05	<b>0.935</b>	7271	1.000	1271.60	<b>0.930</b>
n3div36	123273	3600	130853	<i>1.061</i>	3600.00	1.000	122843	0.997	3600.00	1.000	122601	0.995	3600.00	1.000
n3seq24	6	3600	6	1.000	3600.00	1.000	6	1.000	3600.00	1.000	6	1.000	3600.00	1.000
n4-3	73370	926	88693	<i>1.209</i>	1172.36	<i>1.266</i>	90878	<i>1.239</i>	1210.23	<i>1.307</i>	90878	<i>1.239</i>	1200.98	<i>1.297</i>
neos-1109824	31218	162	42132	<i>1.350</i>	156.36	0.962	35139	<i>1.126</i>	183.91	<i>1.132</i>	35139	<i>1.126</i>	182.62	<i>1.124</i>
neos-1337307	224112	3600	241452	<i>1.077</i>	3600.00	1.000	226889	1.012	3600.00	1.000	225408	1.006	3600.00	1.000
neos-1396125	32266	707	159271	<i>4.936</i>	2391.59	<i>3.381</i>	44694	<i>1.385</i>	1037.98	<i>1.467</i>	44694	<i>1.385</i>	1029.89	<i>1.456</i>
neos-1601936	324	3600	2465	<i>7.608</i>	3600.00	1.000	335	1.034	3600.00	1.000	298	<b>0.920</b>	3600.00	1.000
neos-476283	3201	438	3201	1.000	437.63	1.000	3201	1.000	438.42	1.002	3201	1.000	435.71	0.996
neos-686190	149683	1077	61808	<b>0.413</b>	461.05	<b>0.428</b>	149683	1.000	1072.85	0.996	149683	1.000	1077.56	1.001
neos-849702	34345	648	287514	<i>8.371</i>	3600.00	<i>5.559</i>	1207	<b>0.035</b>	98.09	<b>0.151</b>	1207	<b>0.035</b>	98.27	<b>0.152</b>

cont. on next page



Table 2: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-916792	75359	248	75359	1.000	241.38	0.975	75359	1.000	239.41	0.967	75359	1.000	238.03	0.962
neos-934278	52	3600	52	1.000	3600.00	1.000	52	1.000	3600.00	1.000	52	1.000	3600.00	1.000
neos13	162158	3600	164002	1.011	3600.00	1.000	162805	1.004	3600.00	1.000	160925	0.992	3600.00	1.000
neos18	4919	27	76980	<i>15.650</i>	146.41	<i>5.488</i>	4919	1.000	26.93	1.009	4231	<b>0.860</b>	24.52	<b>0.919</b>
net12	2922	3600	5301	<i>1.814</i>	3600.00	1.000	2805	0.960	3114.80	<b>0.865</b>	3180	<i>1.088</i>	3241.24	<b>0.900</b>
netdiversion	60	3600	41	<b>0.683</b>	3600.00	1.000	84	<i>1.400</i>	3600.00	1.000	83	<i>1.383</i>	3600.00	1.000
newdano	3337954	3600	3301689	0.989	3600.00	1.000	3344550	1.002	3600.00	1.000	3334323	0.999	3600.00	1.000
noswot	584962	140	1810982	<i>3.096</i>	353.15	<i>2.517</i>	586950	1.003	145.09	1.034	574550	0.982	137.01	0.977
ns1208400	2373	3600	2296	0.968	2432.96	<b>0.676</b>	1074	<b>0.453</b>	3600.00	1.000	1022	<b>0.431</b>	3600.00	1.000
ns1688347	4308	231	32931	<i>7.644</i>	870.60	<i>3.777</i>	2781	<b>0.646</b>	269.30	<i>1.168</i>	2781	<b>0.646</b>	259.46	<i>1.126</i>
ns1758913	8	3600	2	<b>0.250</b>	3600.00	1.000	2	<b>0.250</b>	3600.00	1.000	1	<b>0.125</b>	3600.00	1.000
ns1766074	915997	971	1029467	<i>1.124</i>	1499.56	<i>1.544</i>	922327	1.007	1583.71	<i>1.631</i>	1005021	<i>1.097</i>	1317.95	<i>1.357</i>
ns1830653	39491	351	73960	<i>1.873</i>	615.22	<i>1.752</i>	28852	<b>0.731</b>	305.68	<b>0.870</b>	29502	<b>0.747</b>	287.45	<b>0.819</b>
opm2-z7-s2	8798	2208	5350	<b>0.608</b>	1327.74	<b>0.601</b>	8798	1.000	2211.72	1.002	8798	1.000	2209.04	1.000
pg5_34	300472	1383	298798	0.994	1318.28	0.953	300472	1.000	1334.09	0.965	300472	1.000	1333.86	0.965
pigeon-10	12882612	3600	12348998	0.959	3600.00	1.000	12571973	0.976	3600.00	1.000	12580260	0.977	3600.00	1.000
pw-myciel4	567066	3600	627543	<i>1.107</i>	2682.55	<b>0.745</b>	298632	<b>0.527</b>	3600.00	1.000	295734	<b>0.522</b>	3600.00	1.000
qiu	11256	49	11256	1.000	48.61	0.990	11256	1.000	44.65	<b>0.910</b>	11256	1.000	44.82	<b>0.913</b>
rail507	993	287	1183	<i>1.191</i>	369.52	<i>1.288</i>	993	1.000	284.98	0.993	993	1.000	285.31	0.994
ran16x16	340928	270	313667	<b>0.920</b>	245.13	<b>0.907</b>	303203	<b>0.889</b>	239.69	<b>0.887</b>	303203	<b>0.889</b>	238.93	<b>0.884</b>
reblock67	128157	235	74055	<b>0.578</b>	138.03	<b>0.587</b>	127595	0.996	215.33	<b>0.916</b>	130256	1.016	219.66	<b>0.934</b>
rmatr100-p10	768	162	768	1.000	163.23	1.005	768	1.000	162.70	1.002	768	1.000	163.12	1.005
rmatr100-p5	463	503	463	1.000	502.14	0.999	463	1.000	501.46	0.997	463	1.000	502.11	0.998
rmine6	562634	1146	807204	<i>1.435</i>	1549.12	<i>1.351</i>	562634	1.000	1152.77	1.006	562634	1.000	1141.78	0.996
rocII-4-11	15293	2126	35576	<i>2.326</i>	3600.00	<i>1.693</i>	11683	<b>0.764</b>	1608.66	<b>0.757</b>	12462	<b>0.815</b>	2009.13	<b>0.945</b>

cont. on next page

Table 2: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
rococoC10-001000	298459	2527	353552	<i>1.185</i>	2180.44	<b>0.863</b>	289711	0.971	2359.33	<b>0.934</b>	289711	0.971	2356.48	<b>0.933</b>
roll3000	1111977	3600	1177662	<i>1.059</i>	3600.00	1.000	1150992	1.035	3600.00	1.000	1151494	1.036	3600.00	1.000
satellites1-25	471	1787	1982	<i>4.208</i>	3600.00	<i>2.014</i>	721	<i>1.531</i>	1286.73	<b>0.720</b>	721	<i>1.531</i>	1291.66	<b>0.723</b>
sp98ic	15804	3600	17206	<i>1.089</i>	3600.00	1.000	17270	<i>1.093</i>	3600.00	1.000	17307	<i>1.095</i>	3600.00	1.000
sp98ir	5623	75	7980	<i>1.419</i>	88.89	<i>1.187</i>	5623	1.000	76.06	1.016	5623	1.000	75.24	1.005
tanglegram1	61	743	61	1.000	767.68	1.034	61	1.000	740.19	0.997	61	1.000	740.07	0.997
tanglegram2	3	6	3	1.000	6.43	0.998	3	1.000	6.41	0.995	3	1.000	6.80	<i>1.056</i>
timtab1	843361	435	872675	1.035	488.01	<i>1.123</i>	905328	<i>1.073</i>	544.35	<i>1.253</i>	905328	<i>1.073</i>	515.82	<i>1.187</i>
triptim1	1	499	1	1.000	501.62	1.005	1	1.000	500.00	1.002	1	1.000	509.57	1.021
unitcal_7	35212	1499	38141	<i>1.083</i>	1657.13	<i>1.106</i>	43582	<i>1.238</i>	2018.70	<i>1.347</i>	43582	<i>1.238</i>	1830.40	<i>1.221</i>
vpphard	1322	3600	3858	<i>2.918</i>	3600.00	1.000	2388	<i>1.806</i>	3600.00	1.000	2388	<i>1.806</i>	3600.00	1.000
zib54-UUE	487119	3600	505411	1.038	3600.00	1.000	481143	0.988	3600.00	1.000	480906	0.987	3600.00	1.000
geom.	8491.43	617.78	12356.545	1.455	724.631	1.173	8005.819	0.943	603.184	0.976	7958.263	0.937	602.033	0.975
sh. geom. [100, 10]	14381.73	685.80	19636.864	1.365	800.395	1.167	13737.172	0.955	670.337	0.977	13769.376	0.957	668.945	0.975

Table 3: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (conflict). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
acc-tight5	920	139	357	<b>0.388</b>	80.27	<b>0.577</b>	1107	<i>1.203</i>	153.52	<i>1.104</i>	1107	<i>1.203</i>	154.34	<i>1.109</i>
afflow40b	163981	1338	160367	0.978	945.95	<b>0.707</b>	135529	<b>0.826</b>	649.99	<b>0.486</b>	135018	<b>0.823</b>	632.94	<b>0.473</b>
alu4_7	279	1	1643	<i>5.889</i>	1.88	<i>1.709</i>	250	<b>0.896</b>	1.12	1.018	250	<b>0.896</b>	1.17	<i>1.064</i>
alu4_8	649	1	28296	<i>43.599</i>	10.73	<i>8.007</i>	710	<i>1.094</i>	1.41	<i>1.052</i>	710	<i>1.094</i>	1.41	<i>1.052</i>
alu5_7	1704	2	6467	<i>3.795</i>	3.97	<i>2.005</i>	1368	<b>0.803</b>	1.67	<b>0.843</b>	1368	<b>0.803</b>	1.66	<b>0.838</b>
alu5_8	7492	4	212946	<i>28.423</i>	71.31	<i>17.266</i>	4473	<b>0.597</b>	3.09	<b>0.748</b>	4473	<b>0.597</b>	3.40	<b>0.823</b>
alu6_7	14365	11	277059	<i>19.287</i>	345.88	<i>30.394</i>	6024	<b>0.419</b>	6.07	<b>0.533</b>	6024	<b>0.419</b>	6.33	<b>0.556</b>
alu6_8	47135	33	739660	<i>15.692</i>	380.08	<i>11.459</i>	40659	<b>0.863</b>	28.92	<b>0.872</b>	40659	<b>0.863</b>	28.87	<b>0.870</b>
alu7_7	71481	76	739480	<i>10.345</i>	680.82	<i>8.966</i>	59906	<b>0.838</b>	69.68	<b>0.918</b>	59611	<b>0.834</b>	69.97	<b>0.922</b>
bell3a	23108	6	21050	<b>0.911</b>	4.12	<b>0.686</b>	21050	<b>0.911</b>	6.09	1.013	21050	<b>0.911</b>	6.23	1.037
bell5	1132	1	1346	<i>1.189</i>	0.61	<i>1.070</i>	1132	1.000	0.50	<b>0.877</b>	1132	1.000	0.60	<i>1.053</i>
biella1	10755	1710	9961	<b>0.926</b>	1685.98	0.986	9134	<b>0.849</b>	1503.26	<b>0.879</b>	9134	<b>0.849</b>	1500.85	<b>0.878</b>
bienst1	12591	44	15883	<i>1.261</i>	53.32	<i>1.220</i>	29295	<i>2.327</i>	87.96	<i>2.013</i>	29295	<i>2.327</i>	88.33	<i>2.022</i>
bienst2	242186	487	169723	<b>0.701</b>	326.33	<b>0.670</b>	148427	<b>0.613</b>	334.07	<b>0.686</b>	148427	<b>0.613</b>	334.26	<b>0.686</b>
binkar10_1	245091	379	245654	1.002	355.81	<b>0.938</b>	274286	<i>1.119</i>	426.45	<i>1.124</i>	274286	<i>1.119</i>	456.07	<i>1.202</i>
bnatt350	4433	288	166584	<i>37.578</i>	3600.00	<i>12.521</i>	1913	<b>0.432</b>	156.54	<b>0.544</b>	1913	<b>0.432</b>	157.05	<b>0.546</b>
enigma	1172	1	760	<b>0.648</b>	0.50	<b>0.862</b>	1172	1.000	0.59	1.017	1172	1.000	0.60	1.034
lectsched-4-obj	2399	268	29505	<i>12.299</i>	3600.00	<i>13.428</i>	9925	<i>4.137</i>	687.14	<i>2.563</i>	9925	<i>4.137</i>	685.40	<i>2.557</i>
lseu	601	1	530	<b>0.882</b>	0.88	0.978	576	0.958	0.79	<b>0.878</b>	576	0.958	0.68	<b>0.756</b>
markshare_3_0	4841	1	1843	<b>0.381</b>	0.50	<b>0.676</b>	1475	<b>0.305</b>	0.50	<b>0.676</b>	1475	<b>0.305</b>	0.50	<b>0.676</b>
markshare_3_2	5018	1	2861	<b>0.570</b>	0.50	<b>0.781</b>	3049	<b>0.608</b>	0.54	<b>0.844</b>	3049	<b>0.608</b>	0.50	<b>0.781</b>
markshare_3_3	4826	1	2645	<b>0.548</b>	0.50	<b>0.714</b>	2343	<b>0.485</b>	0.50	<b>0.714</b>	2343	<b>0.485</b>	0.50	<b>0.714</b>
markshare_3_4	3352	1	2349	<b>0.701</b>	0.50	1.000	1375	<b>0.410</b>	0.50	1.000	1375	<b>0.410</b>	0.50	1.000
markshare_3_5	2481	1	1421	<b>0.573</b>	0.50	1.000	1609	<b>0.649</b>	0.50	1.000	1609	<b>0.649</b>	0.50	1.000
markshare_4_0	643068	121	168883	<b>0.263</b>	16.77	<b>0.139</b>	129999	<b>0.202</b>	16.43	<b>0.136</b>	129999	<b>0.202</b>	17.52	<b>0.145</b>

cont. on next page

Table 3: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (conflict). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
markshare_4_1	531216	137	214185	<b>0.403</b>	21.93	<b>0.160</b>	108056	<b>0.203</b>	14.81	<b>0.108</b>	108056	<b>0.203</b>	14.74	<b>0.107</b>
markshare_4_2	369816	104	136123	<b>0.368</b>	13.52	<b>0.130</b>	101105	<b>0.273</b>	15.20	<b>0.146</b>	101105	<b>0.273</b>	15.34	<b>0.147</b>
markshare_4_4	420963	82	204375	<b>0.485</b>	20.46	<b>0.251</b>	89807	<b>0.213</b>	13.36	<b>0.164</b>	89807	<b>0.213</b>	13.40	<b>0.164</b>
markshare_4_5	504816	118	274293	<b>0.543</b>	27.82	<b>0.236</b>	145753	<b>0.289</b>	19.14	<b>0.162</b>	145753	<b>0.289</b>	18.59	<b>0.157</b>
mas74	3420026	685	3577586	1.046	717.66	1.047	3420026	1.000	689.28	1.006	3420282	1.000	686.09	1.001
mas76	519681	96	516536	0.994	86.56	<b>0.901</b>	519681	1.000	93.60	0.974	520189	1.001	89.72	<b>0.934</b>
mik-250-1-100-1	633761	315	578416	<b>0.913</b>	285.29	<b>0.905</b>	633761	1.000	316.66	1.005	633761	1.000	316.62	1.005
mine-166-5	6651	37	4551	<b>0.684</b>	27.87	<b>0.755</b>	6651	1.000	36.23	0.982	6651	1.000	37.03	1.004
mine-90-10	45418	214	137594	<i>3.030</i>	254.94	<i>1.191</i>	37025	<b>0.815</b>	156.94	<b>0.733</b>	57388	<i>1.264</i>	219.60	1.026
misc03	134	1	134	1.000	1.06	0.955	135	1.007	1.09	0.982	135	1.007	0.89	<b>0.802</b>
misc07	29168	21	38819	<i>1.331</i>	23.36	<i>1.125</i>	31784	<i>1.090</i>	21.97	<i>1.058</i>	31784	<i>1.090</i>	21.82	<i>1.051</i>
mzzv11	7274	1393	8588	<i>1.181</i>	1996.33	<i>1.433</i>	7259	0.998	1277.69	<b>0.917</b>	7271	1.000	1277.35	<b>0.917</b>
mzzv42z	3050	731	2320	<b>0.761</b>	529.74	<b>0.724</b>	3392	<i>1.112</i>	901.81	<i>1.233</i>	3392	<i>1.112</i>	897.06	<i>1.227</i>
neos-1061020	4040	1070	2704	<b>0.669</b>	701.54	<b>0.656</b>	10200	<i>2.525</i>	1600.51	<i>1.496</i>	10200	<i>2.525</i>	1600.38	<i>1.496</i>
neos-1109824	31218	163	42132	<i>1.350</i>	155.53	0.956	35139	<i>1.126</i>	182.47	<i>1.122</i>	35139	<i>1.126</i>	183.31	<i>1.127</i>
neos-1126860	5055	597	7015	<i>1.388</i>	589.86	0.989	5055	1.000	569.51	0.954	5055	1.000	571.85	0.958
neos-1173026	383075	1193	1305899	<i>3.409</i>	3291.14	<i>2.758</i>	103261	<b>0.270</b>	389.05	<b>0.326</b>	109510	<b>0.286</b>	412.64	<b>0.346</b>
neos-1208069	6041	457	161143	<i>26.675</i>	2583.86	<i>5.656</i>	6782	<i>1.123</i>	701.05	<i>1.535</i>	6782	<i>1.123</i>	754.67	<i>1.652</i>
neos-1208135	8457	1139	60146	<i>7.112</i>	2193.83	<i>1.926</i>	3651	<b>0.432</b>	492.98	<b>0.433</b>	3651	<b>0.432</b>	494.28	<b>0.434</b>
neos-1215259	2880	101	1645	<b>0.571</b>	56.00	<b>0.554</b>	7630	<i>2.649</i>	185.91	<i>1.840</i>	7630	<i>2.649</i>	195.43	<i>1.935</i>
neos-1215891	3890	1409	80007	<i>20.567</i>	3600.00	<i>2.554</i>	11676	<i>3.002</i>	2221.22	<i>1.576</i>	11676	<i>3.002</i>	2063.01	<i>1.464</i>
neos-1223462	1472	819	1878	<i>1.276</i>	1020.01	<i>1.245</i>	561	<b>0.381</b>	442.74	<b>0.541</b>	561	<b>0.381</b>	416.85	<b>0.509</b>
neos-1281048	1384	12	1221	<b>0.882</b>	10.75	<b>0.930</b>	1243	<b>0.898</b>	11.04	0.955	1243	<b>0.898</b>	11.29	0.977
neos-1396125	32266	708	159271	<i>4.936</i>	2393.14	<i>3.381</i>	44694	<i>1.385</i>	1029.00	<i>1.454</i>	44694	<i>1.385</i>	1030.02	<i>1.455</i>
neos-1420205	55079	18	26789	<b>0.486</b>	9.57	<b>0.545</b>	55079	1.000	17.83	1.016	55079	1.000	18.16	1.035

cont. on next page

Table 3: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (conflict). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-1440225	1461	47	5584	<i>3.822</i>	152.10	<i>3.209</i>	12608	<i>8.630</i>	315.83	<i>6.663</i>	12608	<i>8.630</i>	317.68	<i>6.702</i>
neos-1460265	43673	66	26519	<b>0.607</b>	55.96	<b>0.850</b>	30942	<b>0.708</b>	59.61	<b>0.906</b>	30942	<b>0.708</b>	59.58	<b>0.905</b>
neos-1461051	3401	13	301005	<i>88.505</i>	222.35	<i>17.535</i>	3991	<i>1.173</i>	13.56	<i>1.069</i>	3380	0.994	12.11	0.955
neos-1480121	5608250	3600	2359174	<b>0.421</b>	324.99	<b>0.090</b>	565	<b>0.000</b>	0.94	<b>0.000</b>	565	<b>0.000</b>	1.10	<b>0.000</b>
neos-1582420	59182	621	31105	<b>0.526</b>	391.81	<b>0.630</b>	56913	0.962	596.55	0.960	56913	0.962	595.92	0.959
neos-1620807	1812678	1543	6291083	<i>3.471</i>	1844.94	<i>1.195</i>	1722467	<b>0.950</b>	1520.50	0.985	1729302	0.954	1222.67	<b>0.792</b>
neos-430149	25985	24	59691	<i>2.297</i>	48.23	<i>1.995</i>	27247	1.049	24.35	1.007	27247	1.049	24.23	1.002
neos-503737	9330	354	1743	<b>0.187</b>	131.56	<b>0.371</b>	2257	<b>0.242</b>	133.17	<b>0.376</b>	2257	<b>0.242</b>	133.63	<b>0.377</b>
neos-504674	36880	78	27888	<b>0.756</b>	51.73	<b>0.665</b>	32657	<b>0.885</b>	69.83	<b>0.898</b>	35401	0.960	74.54	0.958
neos-538867	46052	53	119585	<i>2.597</i>	88.74	<i>1.668</i>	42703	<b>0.927</b>	48.08	<b>0.904</b>	47156	1.024	52.99	0.996
neos-538916	29500	42	50111	<i>1.699</i>	44.14	1.047	29547	1.002	42.35	1.005	28916	0.980	41.11	0.975
neos-551991	5448	319	10487	<i>1.925</i>	1022.46	<i>3.205</i>	5448	1.000	318.02	0.997	5448	1.000	319.63	1.002
neos-555298	27669	200	37347	<i>1.350</i>	329.25	<i>1.642</i>	11145	<b>0.403</b>	178.05	<b>0.888</b>	11145	<b>0.403</b>	170.27	<b>0.849</b>
neos-584851	351	27	424	<i>1.208</i>	26.76	1.008	351	1.000	26.37	0.994	351	1.000	26.59	1.002
neos-585192	1577	35	1967	<i>1.247</i>	36.92	<i>1.068</i>	1478	<b>0.937</b>	32.89	0.952	1478	<b>0.937</b>	32.32	<b>0.935</b>
neos-595925	27416	59	14296	<b>0.521</b>	39.84	<b>0.672</b>	22257	<b>0.812</b>	50.22	<b>0.847</b>	22257	<b>0.812</b>	52.46	<b>0.884</b>
neos-686190	149683	1079	61808	<b>0.413</b>	479.98	<b>0.445</b>	149683	1.000	1085.06	1.006	149683	1.000	1074.97	0.996
neos-717614	265333	414	530703	<i>2.000</i>	749.59	<i>1.813</i>	15048	<b>0.057</b>	30.81	<b>0.075</b>	15048	<b>0.057</b>	30.92	<b>0.075</b>
neos-785912	267	107	797	<i>2.985</i>	185.41	<i>1.732</i>	319	<i>1.195</i>	115.74	<i>1.081</i>	319	<i>1.195</i>	115.31	<i>1.077</i>
neos-791021	636	1787	188	<b>0.296</b>	699.31	<b>0.391</b>	271	<b>0.426</b>	1034.35	<b>0.579</b>	271	<b>0.426</b>	1033.95	<b>0.579</b>
neos-803219	22155	35	24978	<i>1.127</i>	42.68	<i>1.223</i>	22478	1.015	37.22	<i>1.066</i>	22478	1.015	37.67	<i>1.079</i>
neos-803220	55512	89	48674	<b>0.877</b>	92.19	1.036	51137	<b>0.921</b>	103.62	<i>1.164</i>	51569	<b>0.929</b>	104.92	<i>1.179</i>
neos-806323	10041	29	14620	<i>1.456</i>	39.99	<i>1.378</i>	7717	<b>0.769</b>	22.50	<b>0.776</b>	7318	<b>0.729</b>	21.69	<b>0.748</b>
neos-807639	6123	16	3509	<b>0.573</b>	13.33	<b>0.819</b>	3575	<b>0.584</b>	13.34	<b>0.820</b>	3575	<b>0.584</b>	12.77	<b>0.785</b>
neos-807705	9210	24	5174	<b>0.562</b>	15.09	<b>0.623</b>	5240	<b>0.569</b>	15.41	<b>0.636</b>	5240	<b>0.569</b>	15.86	<b>0.655</b>

cont. on next page

Table 3: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (conflict). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-810286	945	502	381	<b>0.403</b>	122.82	<b>0.245</b>	1015	<i>1.074</i>	775.00	<i>1.543</i>	1015	<i>1.074</i>	789.36	<i>1.571</i>
neos-810326	2441	104	1188	<b>0.487</b>	61.46	<b>0.589</b>	3144	<i>1.288</i>	125.41	<i>1.201</i>	3144	<i>1.288</i>	125.06	<i>1.198</i>
neos-827015	187	1376	569	<i>3.043</i>	3268.56	<i>2.376</i>	187	1.000	1336.65	0.972	187	1.000	1331.51	0.968
neos-831188	4962	483	3861	<b>0.778</b>	372.08	<b>0.770</b>	5236	<i>1.055</i>	507.40	<i>1.051</i>	5236	<i>1.055</i>	506.56	1.049
neos-839859	18823	1130	20996	<i>1.115</i>	1214.60	<i>1.075</i>	13830	<b>0.735</b>	1009.15	<b>0.893</b>	13830	<b>0.735</b>	1008.80	<b>0.893</b>
neos-848845	36361	923	311122	<i>8.556</i>	3600.00	<i>3.900</i>	3316	<b>0.091</b>	189.45	<b>0.205</b>	3316	<b>0.091</b>	189.31	<b>0.205</b>
neos-849702	34345	647	299757	<i>8.728</i>	3600.00	<i>5.562</i>	1207	<b>0.035</b>	97.78	<b>0.151</b>	1207	<b>0.035</b>	98.48	<b>0.152</b>
neos-862348	1516	30	5222	<i>3.445</i>	38.85	<i>1.315</i>	2007	<i>1.324</i>	28.93	0.979	2007	<i>1.324</i>	29.02	0.982
neos-863472	262229	220	714473	<i>2.725</i>	453.29	<i>2.056</i>	328043	<i>1.251</i>	274.54	<i>1.245</i>	329331	<i>1.256</i>	278.13	<i>1.262</i>
neos-886822	66990	244	35286	<b>0.527</b>	218.88	<b>0.898</b>	66990	1.000	236.67	0.971	66990	1.000	242.12	0.993
neos-892255	1612	448	930	<b>0.577</b>	288.40	<b>0.643</b>	1612	1.000	448.19	1.000	1612	1.000	459.10	1.024
neos-905856	4334	55	33444	<i>7.717</i>	220.39	<i>3.979</i>	11715	<i>2.703</i>	93.84	<i>1.694</i>	11715	<i>2.703</i>	94.82	<i>1.712</i>
neos-906865	48822	122	49529	1.014	123.62	1.010	49279	1.009	123.11	1.006	49279	1.009	123.15	1.006
neos-912023	1827	21	53733	<i>29.411</i>	239.03	<i>11.291</i>	6968	<i>3.814</i>	47.97	<i>2.266</i>	6968	<i>3.814</i>	48.15	<i>2.274</i>
neos-914441	354	227	180	<b>0.508</b>	155.90	<b>0.686</b>	1350	<i>3.814</i>	322.93	<i>1.421</i>	1350	<i>3.814</i>	304.92	<i>1.342</i>
neos-942323	2567	16	4610	<i>1.796</i>	18.95	<i>1.220</i>	2382	<b>0.928</b>	14.60	<b>0.940</b>	2382	<b>0.928</b>	14.78	0.952
neos18	4919	27	76980	<i>15.650</i>	145.81	<i>5.498</i>	4919	1.000	26.55	1.001	4231	<b>0.860</b>	24.93	<b>0.940</b>
net12	1514	1690	3903	<i>2.578</i>	3600.00	<i>2.130</i>	1372	<b>0.906</b>	1675.85	0.992	1700	<i>1.123</i>	1772.64	1.049
noswot	584962	141	1810982	<i>3.096</i>	354.55	<i>2.507</i>	586950	1.003	143.77	1.016	574550	0.982	137.98	0.976
ns1208400	2495	3600	2296	<b>0.920</b>	2454.32	<b>0.682</b>	1031	<b>0.413</b>	3600.00	1.000	1074	<b>0.430</b>	3600.00	1.000
ns1688347	4308	228	32931	<i>7.644</i>	874.84	<i>3.831</i>	2781	<b>0.646</b>	260.66	<i>1.141</i>	2781	<b>0.646</b>	260.36	<i>1.140</i>
ns1766074	915997	978	1029467	<i>1.124</i>	1483.94	<i>1.517</i>	922327	1.007	1580.88	<i>1.616</i>	1005021	<i>1.097</i>	1297.55	<i>1.326</i>
ns1830653	39491	351	73960	<i>1.873</i>	619.02	<i>1.762</i>	28852	<b>0.731</b>	282.16	<b>0.803</b>	29502	<b>0.747</b>	288.85	<b>0.822</b>
pg5_34	300472	1332	298798	0.994	1317.25	0.989	300472	1.000	1335.26	1.002	300472	1.000	1333.51	1.001
prod1	29058	18	29822	1.026	15.83	<b>0.885</b>	29058	1.000	17.35	0.970	29058	1.000	17.99	1.006

cont. on next page

Table 3: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				combined				combined+pool			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
prod2	78750	63	150765	<i>1.914</i>	104.46	<i>1.649</i>	78750	1.000	65.36	1.032	78690	0.999	65.02	1.027
pw-myciel4	569026	3600	627543	<i>1.103</i>	2708.26	<b>0.752</b>	298057	<b>0.524</b>	3600.00	1.000	290010	<b>0.510</b>	3600.00	1.000
reblock67	128157	235	74055	<b>0.578</b>	130.51	<b>0.555</b>	127595	0.996	215.12	<b>0.914</b>	130256	1.016	219.11	<b>0.931</b>
rmine6	562634	1144	807204	<i>1.435</i>	1544.44	<i>1.350</i>	562634	1.000	1142.23	0.999	562634	1.000	1152.93	1.008
rococoC10-001000	298459	2527	353552	<i>1.185</i>	2185.99	<b>0.865</b>	289711	0.971	2357.66	<b>0.933</b>	289711	0.971	2360.75	<b>0.934</b>
rout	49976	58	137696	<i>2.755</i>	120.82	<i>2.099</i>	44264	<b>0.886</b>	50.01	<b>0.869</b>	44420	<b>0.889</b>	50.37	<b>0.875</b>
satellites1-25	471	1859	1982	<i>4.208</i>	3600.00	<i>1.937</i>	721	<i>1.531</i>	1289.84	<b>0.694</b>	721	<i>1.531</i>	1293.84	<b>0.696</b>
timtab1	843361	434	872675	1.035	489.78	<i>1.128</i>	905328	<i>1.073</i>	518.53	<i>1.194</i>	905328	<i>1.073</i>	515.51	<i>1.187</i>
geom.	16183.61	101.28	26196.478	1.619	128.768	1.271	12198.823	0.754	79.971	0.790	12270.609	0.758	80.171	0.792
sh. geom. [100, 10]	16769.05	143.22	27097.915	1.616	179.910	1.256	12661.862	0.755	118.510	0.827	12735.782	0.759	118.723	0.829

Table 4: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
30n20b8	51	268	933	<i>18.294</i>	1591.27	<i>5.933</i>	1207	<i>23.667</i>	1477.92	<i>5.511</i>	6754	<i>132.431</i>	3600.00	<i>13.423</i>
acc-tight5	920	140	357	<b>0.388</b>	76.88	<b>0.551</b>	207	<b>0.225</b>	69.40	<b>0.497</b>	1615	<i>1.755</i>	280.53	<i>2.009</i>
aflow40b	163981	1333	160367	0.978	888.21	<b>0.666</b>	158904	0.969	1259.54	<b>0.945</b>	199108	<i>1.214</i>	1486.33	<i>1.115</i>
air04	218	74	180	<b>0.826</b>	73.71	0.997	180	<b>0.826</b>	73.85	0.999	218	1.000	69.87	<b>0.945</b>
app1-2	2	3600	71	<i>35.500</i>	2100.69	<b>0.584</b>	2	1.000	3600.00	1.000	4	<i>2.000</i>	3600.00	1.000
ash608gpia-3col	7	13	297	<i>42.429</i>	36.69	<i>2.720</i>	297	<i>42.429</i>	36.65	<i>2.717</i>	7	1.000	13.44	0.996
bab5	21993	3600	29410	<i>1.337</i>	3600.00	1.000	23609	<i>1.073</i>	3600.00	1.000	28573	<i>1.299</i>	3600.00	1.000
beasleyC3	832249	3600	1278442	<i>1.536</i>	3600.00	1.000	542564	<b>0.652</b>	3600.00	1.000	1548537	<i>1.861</i>	3600.00	1.000
biella1	10755	1711	9961	<b>0.926</b>	1683.80	0.984	9356	<b>0.870</b>	1460.30	<b>0.853</b>	7155	<b>0.665</b>	950.74	<b>0.556</b>
bienst2	242186	486	169723	<b>0.701</b>	326.13	<b>0.671</b>	184318	<b>0.761</b>	333.64	<b>0.686</b>	136397	<b>0.563</b>	295.24	<b>0.607</b>
binkar10_1	245091	380	245654	1.002	355.63	<b>0.936</b>	247948	1.012	360.19	<b>0.948</b>	242089	0.988	376.38	0.991
bley_xl1	1	238	1	1.000	236.30	0.991	1	1.000	238.66	1.001	1	1.000	233.57	0.980
bnatt350	4433	287	167031	<i>37.679</i>	3600.00	<i>12.540</i>	148078	<i>33.404</i>	3600.00	<i>12.540</i>	4949	<i>1.116</i>	281.28	0.980
core2536-691	2335	1295	2051	<b>0.878</b>	1731.98	<i>1.337</i>	2051	<b>0.878</b>	1733.00	<i>1.338</i>	2335	1.000	1295.13	1.000
cov1075	1195289	3600	1193248	0.998	3600.00	1.000	1204446	1.008	3600.00	1.000	1195478	1.000	3600.00	1.000
csched010	398144	3600	386528	0.971	3600.00	1.000	386405	0.971	3600.00	1.000	410822	1.032	3600.00	1.000
danoint	1043564	3600	950242	<b>0.911</b>	3600.00	1.000	1030940	0.988	3600.00	1.000	982068	<b>0.941</b>	3256.07	<b>0.904</b>
dfn-gwin-UUM	46722	96	46670	0.999	95.01	0.989	46030	0.985	95.92	0.999	46722	1.000	96.39	1.003
eil33-2	865	59	865	1.000	57.51	0.976	865	1.000	58.41	0.991	865	1.000	57.28	0.972
eilB101	12775	406	12775	1.000	408.54	1.006	12775	1.000	409.03	1.008	12775	1.000	405.62	0.999
enlight13	1	1	1	1.000	0.50	1.000	1	1.000	0.50	1.000	1	1.000	0.50	1.000
enlight14	1	1	1	1.000	0.50	1.000	1	1.000	0.50	1.000	1	1.000	0.50	1.000
ex9	1	36	1	1.000	36.33	1.004	1	1.000	35.84	0.991	1	1.000	37.02	1.024
glass4	5039212	3074	6252477	<i>1.241</i>	3600.00	<i>1.171</i>	4871908	0.967	3600.00	<i>1.171</i>	2397374	<b>0.476</b>	1430.13	<b>0.465</b>
gmu-35-40	5403540	3600	6607070	<i>1.223</i>	3600.00	1.000	6882538	<i>1.274</i>	3600.00	1.000	5433076	1.005	3600.00	1.000

cont. on next page



Table 4: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
iis-100-0-cov	88800	631	88800	1.000	630.10	0.999	88800	1.000	630.28	0.999	88800	1.000	630.51	1.000
iis-bupa-cov	180506	2425	180506	1.000	2426.71	1.001	180506	1.000	2425.33	1.000	180506	1.000	2428.11	1.001
iis-pima-cov	8090	241	8090	1.000	241.09	0.999	8090	1.000	240.66	0.997	8090	1.000	259.62	<i>1.076</i>
lectsched-4-obj	2399	269	29448	<i>12.275</i>	3600.00	<i>13.399</i>	20488	<i>8.540</i>	1649.51	<i>6.139</i>	5744	<i>2.394</i>	725.57	<i>2.700</i>
m100n500k4r1	3292898	3600	3565574	<i>1.083</i>	3600.00	1.000	3569939	<i>1.084</i>	3600.00	1.000	3604469	<i>1.095</i>	3600.00	1.000
macrophage	878022	3600	865342	0.986	3600.00	1.000	863221	0.983	3600.00	1.000	887139	1.010	3600.00	1.000
map18	307	269	307	1.000	265.56	0.986	307	1.000	266.44	0.989	307	1.000	265.94	0.988
map20	385	256	385	1.000	255.38	0.998	385	1.000	254.63	0.995	385	1.000	254.33	0.994
mcsched	20358	206	18218	<b>0.895</b>	169.02	<b>0.821</b>	18218	<b>0.895</b>	179.88	<b>0.874</b>	20358	1.000	193.24	<b>0.939</b>
mik-250-1-100-1	633761	317	578416	<b>0.913</b>	287.90	<b>0.909</b>	578416	<b>0.913</b>	284.89	<b>0.900</b>	633761	1.000	315.92	0.998
mine-166-5	6651	37	4551	<b>0.684</b>	28.03	<b>0.763</b>	5176	<b>0.778</b>	26.50	<b>0.722</b>	4860	<b>0.731</b>	36.68	0.999
mine-90-10	45418	210	137594	<i>3.030</i>	254.67	<i>1.213</i>	53696	<i>1.182</i>	142.27	<b>0.678</b>	1253859	<i>27.607</i>	3600.00	<i>17.144</i>
msc98-ip	648	3600	1658	<i>2.559</i>	3600.00	1.000	1068	<i>1.648</i>	3600.00	1.000	774	<i>1.194</i>	3600.00	1.000
mspp16	59	2142	121	<i>2.051</i>	2720.47	<i>1.270</i>	123	<i>2.085</i>	2645.91	<i>1.235</i>	63	<i>1.068</i>	2132.71	0.996
mzzv11	7274	1367	8588	<i>1.181</i>	1930.21	<i>1.412</i>	10482	<i>1.441</i>	1929.12	<i>1.411</i>	6352	<b>0.873</b>	1214.28	<b>0.888</b>
n3div36	123273	3600	130853	<i>1.061</i>	3600.00	1.000	130806	<i>1.061</i>	3600.00	1.000	118014	0.957	3600.00	1.000
n3seq24	6	3600	6	1.000	3600.00	1.000	6	1.000	3600.00	1.000	6	1.000	3600.00	1.000
n4-3	73370	926	88693	<i>1.209</i>	1172.36	<i>1.266</i>	73370	1.000	934.17	1.009	78500	<i>1.070</i>	986.20	<i>1.065</i>
neos-1109824	31218	162	42132	<i>1.350</i>	156.36	0.962	52693	<i>1.688</i>	201.42	<i>1.240</i>	48097	<i>1.541</i>	240.14	<i>1.478</i>
neos-1337307	224112	3600	241452	<i>1.077</i>	3600.00	1.000	248820	<i>1.110</i>	3600.00	1.000	242427	<i>1.082</i>	3600.00	1.000
neos-1396125	32266	707	159271	<i>4.936</i>	2391.59	<i>3.381</i>	151173	<i>4.685</i>	2277.34	<i>3.219</i>	36926	<i>1.144</i>	921.15	<i>1.302</i>
neos-1601936	324	3600	2465	<i>7.608</i>	3600.00	1.000	1842	<i>5.685</i>	3600.00	1.000	333	1.028	3600.00	1.000
neos-476283	3201	438	3201	1.000	437.63	1.000	3201	1.000	438.50	1.002	3201	1.000	435.39	0.995
neos-686190	149683	1077	61808	<b>0.413</b>	461.05	<b>0.428</b>	74321	<b>0.497</b>	630.72	<b>0.586</b>	30924	<b>0.207</b>	249.65	<b>0.232</b>
neos-849702	34345	648	287514	<i>8.371</i>	3600.00	<i>5.559</i>	242167	<i>7.051</i>	3600.00	<i>5.559</i>	24076	<b>0.701</b>	484.31	<b>0.748</b>

cont. on next page

Table 4: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-916792	75359	248	75359	1.000	241.38	0.975	75359	1.000	238.15	0.962	75359	1.000	244.54	0.988
neos-934278	52	3600	52	1.000	3600.00	1.000	52	1.000	3600.00	1.000	52	1.000	3600.00	1.000
neos13	162158	3600	164002	1.011	3600.00	1.000	164177	1.012	3553.56	0.987	161319	0.995	3600.00	1.000
neos18	4919	27	76980	<i>15.650</i>	146.41	<i>5.488</i>	352971	<i>71.757</i>	447.81	<i>16.784</i>	4919	1.000	26.56	0.996
net12	2922	3600	5301	<i>1.814</i>	3600.00	1.000	6393	<i>2.188</i>	3600.00	1.000	3084	<i>1.055</i>	3600.00	1.000
netdiversion	60	3600	41	<b>0.683</b>	3600.00	1.000	48	<b>0.800</b>	3600.00	1.000	114	<i>1.900</i>	3600.00	1.000
newdano	3337954	3600	3301689	0.989	3600.00	1.000	2908845	<b>0.871</b>	3250.59	<b>0.903</b>	2799973	<b>0.839</b>	3600.00	1.000
noswot	584962	140	1810982	<i>3.096</i>	353.15	<i>2.517</i>	1139205	<i>1.947</i>	226.58	<i>1.615</i>	880169	<i>1.505</i>	210.01	<i>1.497</i>
ns1208400	2373	3600	2296	0.968	2432.96	<b>0.676</b>	519	<b>0.219</b>	1550.57	<b>0.431</b>	5630	<i>2.373</i>	3600.00	1.000
ns1688347	4308	231	32931	<i>7.644</i>	870.60	<i>3.777</i>	19314	<i>4.483</i>	378.82	<i>1.643</i>	4175	0.969	335.15	<i>1.454</i>
ns1758913	8	3600	2	<b>0.250</b>	3600.00	1.000	9	<i>1.125</i>	3600.00	1.000	4	<b>0.500</b>	3600.00	1.000
ns1766074	915997	971	1029467	<i>1.124</i>	1499.56	<i>1.544</i>	936139	1.022	945.65	0.974	1373951	<i>1.500</i>	572.80	<b>0.590</b>
ns1830653	39491	351	73960	<i>1.873</i>	615.22	<i>1.752</i>	64211	<i>1.626</i>	583.96	<i>1.663</i>	27436	<b>0.695</b>	341.45	0.972
opm2-z7-s2	8798	2208	5350	<b>0.608</b>	1327.74	<b>0.601</b>	5350	<b>0.608</b>	1327.74	<b>0.601</b>	8798	1.000	2206.64	0.999
pg5_34	300472	1383	298798	0.994	1318.28	0.953	300472	1.000	1332.62	0.964	298798	0.994	1314.34	<b>0.950</b>
pigeon-10	12882612	3600	12348998	0.959	3600.00	1.000	11449875	<b>0.889</b>	3600.00	1.000	13299845	1.032	3600.00	1.000
pw-myciel4	567066	3600	627543	<i>1.107</i>	2682.55	<b>0.745</b>	616372	<i>1.087</i>	3600.00	1.000	693585	<i>1.223</i>	3600.00	1.000
qiu	11256	49	11256	1.000	48.61	0.990	11256	1.000	44.58	<b>0.908</b>	11256	1.000	47.92	0.976
rail507	993	287	1183	<i>1.191</i>	369.52	<i>1.288</i>	1183	<i>1.191</i>	368.83	<i>1.286</i>	993	1.000	285.79	0.996
ran16x16	340928	270	313667	<b>0.920</b>	245.13	<b>0.907</b>	310218	<b>0.910</b>	242.85	<b>0.899</b>	337130	0.989	265.36	0.982
reblock67	128157	235	74055	<b>0.578</b>	138.03	<b>0.587</b>	101852	<b>0.795</b>	176.10	<b>0.749</b>	119391	<b>0.932</b>	205.31	<b>0.873</b>
rmatr100-p10	768	162	768	1.000	163.23	1.005	768	1.000	163.35	1.006	768	1.000	162.35	1.000
rmatr100-p5	463	503	463	1.000	502.14	0.999	463	1.000	532.16	<i>1.058</i>	463	1.000	502.88	1.000
rmine6	562634	1146	807204	<i>1.435</i>	1549.12	<i>1.351</i>	1010123	<i>1.795</i>	1949.63	<i>1.701</i>	484579	<b>0.861</b>	1010.17	<b>0.881</b>
rocII-4-11	15293	2126	35576	<i>2.326</i>	3600.00	<i>1.693</i>	29759	<i>1.946</i>	3600.00	<i>1.693</i>	24742	<i>1.618</i>	2806.38	<i>1.320</i>

cont. on next page

Table 4: Detailed computational results on MIPLIB2010 test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
rococoC10-001000	298459	2527	353552	<i>1.185</i>	2180.44	<b>0.863</b>	217769	<b>0.730</b>	1595.62	<b>0.631</b>	361057	<i>1.210</i>	2740.30	<i>1.084</i>
roll3000	1111977	3600	1177662	<i>1.059</i>	3600.00	1.000	932197	<b>0.838</b>	3600.00	1.000	1055375	<b>0.949</b>	3600.00	1.000
satellites1-25	471	1787	1982	<i>4.208</i>	3600.00	<i>2.014</i>	6212	<i>13.189</i>	1930.49	<i>1.080</i>	1082	<i>2.297</i>	3600.00	<i>2.014</i>
sp98ic	15804	3600	17206	<i>1.089</i>	3600.00	1.000	17299	<i>1.095</i>	3600.00	1.000	17294	<i>1.094</i>	3600.00	1.000
sp98ir	5623	75	7980	<i>1.419</i>	88.89	<i>1.187</i>	8102	<i>1.441</i>	89.17	<i>1.191</i>	5623	1.000	75.03	1.002
tanglegram1	61	743	61	1.000	767.68	1.034	61	1.000	744.40	1.003	61	1.000	747.47	1.007
tanglegram2	3	6	3	1.000	6.43	0.998	3	1.000	6.60	1.025	3	1.000	6.41	0.995
timtab1	843361	435	872675	1.035	488.01	<i>1.123</i>	863707	1.024	441.25	1.015	3612719	<i>4.284</i>	1665.76	<i>3.833</i>
triptim1	1	499	1	1.000	501.62	1.005	1	1.000	507.33	1.016	1	1.000	500.72	1.003
unitcal_7	35212	1499	38141	<i>1.083</i>	1657.13	<i>1.106</i>	42530	<i>1.208</i>	1851.53	<i>1.235</i>	35212	1.000	1481.98	0.989
vpphard	1322	3600	3858	<i>2.918</i>	3600.00	1.000	2824	<i>2.136</i>	3600.00	1.000	924	<b>0.699</b>	3600.00	1.000
zib54-UUE	487119	3600	505411	1.038	3600.00	1.000	475796	0.977	3600.00	1.000	502361	1.031	3600.00	1.000
geom.	8491.43	617.78	12356.545	1.455	724.631	1.173	11550.090	1.360	708.962	1.148	9768.339	1.150	663.745	1.074
sh. geom. [100, 10]	14381.73	685.80	19636.864	1.365	800.395	1.167	18728.848	1.302	783.014	1.142	16243.066	1.129	736.524	1.074

Table 5: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
acc-tight5	920	139	357	<b>0.388</b>	80.27	<b>0.577</b>	207	<b>0.225</b>	66.28	<b>0.476</b>	1615	<i>1.755</i>	280.88	<i>2.019</i>
afflow40b	163981	1338	160367	0.978	945.95	<b>0.707</b>	158904	0.969	1262.21	<b>0.943</b>	199108	<i>1.214</i>	1486.37	<i>1.111</i>
alu4_7	279	1	1643	<i>5.889</i>	1.88	<i>1.709</i>	343	<i>1.229</i>	1.16	<i>1.055</i>	301	<i>1.079</i>	1.04	<b>0.945</b>
alu4_8	649	1	28296	<i>43.599</i>	10.73	<i>8.007</i>	1802	<i>2.777</i>	1.72	<i>1.284</i>	1141	<i>1.758</i>	1.77	<i>1.321</i>
alu5_7	1704	2	6467	<i>3.795</i>	3.97	<i>2.005</i>	4051	<i>2.377</i>	2.68	<i>1.354</i>	2147	<i>1.260</i>	2.12	<i>1.071</i>
alu5_8	7492	4	212946	<i>28.423</i>	71.31	<i>17.266</i>	132692	<i>17.711</i>	45.63	<i>11.048</i>	4580	<b>0.611</b>	3.22	<b>0.780</b>
alu6_7	14365	11	277059	<i>19.287</i>	345.88	<i>30.394</i>	13099	<b>0.912</b>	10.36	<b>0.910</b>	15776	<i>1.098</i>	10.04	<b>0.882</b>
alu6_8	47135	33	739660	<i>15.692</i>	380.08	<i>11.459</i>	495193	<i>10.506</i>	161.01	<i>4.854</i>	69219	<i>1.469</i>	41.93	<i>1.264</i>
alu7_7	71481	76	739480	<i>10.345</i>	680.82	<i>8.966</i>	119479	<i>1.671</i>	83.15	<i>1.095</i>	91263	<i>1.277</i>	91.06	<i>1.199</i>
bell3a	23108	6	21050	<b>0.911</b>	4.12	<b>0.686</b>	23108	1.000	4.67	<b>0.777</b>	23728	1.027	6.13	1.020
bell5	1132	1	1346	<i>1.189</i>	0.61	<i>1.070</i>	1132	1.000	0.55	0.965	1343	<i>1.186</i>	0.84	<i>1.474</i>
biella1	10755	1710	9961	<b>0.926</b>	1685.98	0.986	9356	<b>0.870</b>	1457.89	<b>0.852</b>	7155	<b>0.665</b>	950.67	<b>0.556</b>
bienst1	12591	44	15883	<i>1.261</i>	53.32	<i>1.220</i>	13962	<i>1.109</i>	46.97	<i>1.075</i>	16133	<i>1.281</i>	54.61	<i>1.250</i>
bienst2	242186	487	169723	<b>0.701</b>	326.33	<b>0.670</b>	184318	<b>0.761</b>	333.07	<b>0.684</b>	136397	<b>0.563</b>	296.19	<b>0.608</b>
binkar10_1	245091	379	245654	1.002	355.81	<b>0.938</b>	247948	1.012	358.96	<b>0.946</b>	242089	0.988	374.55	0.987
bnatt350	4433	288	166584	<i>37.578</i>	3600.00	<i>12.521</i>	148395	<i>33.475</i>	3600.00	<i>12.521</i>	4949	<i>1.116</i>	281.06	0.978
enigma	1172	1	760	<b>0.648</b>	0.50	<b>0.862</b>	1413	<i>1.206</i>	0.83	<i>1.431</i>	987	<b>0.842</b>	0.67	<i>1.155</i>
lectsched-4-obj	2399	268	29505	<i>12.299</i>	3600.00	<i>13.428</i>	20488	<i>8.540</i>	1654.63	<i>6.172</i>	5744	<i>2.394</i>	728.26	<i>2.716</i>
lseu	601	1	530	<b>0.882</b>	0.88	0.978	491	<b>0.817</b>	0.77	<b>0.856</b>	631	<i>1.050</i>	0.62	<b>0.689</b>
markshare_3_0	4841	1	1843	<b>0.381</b>	0.50	<b>0.676</b>	4436	<b>0.916</b>	0.77	1.041	6810	<i>1.407</i>	0.54	<b>0.730</b>
markshare_3_2	5018	1	2861	<b>0.570</b>	0.50	<b>0.781</b>	3565	<b>0.710</b>	0.50	<b>0.781</b>	4951	0.987	0.50	<b>0.781</b>
markshare_3_3	4826	1	2645	<b>0.548</b>	0.50	<b>0.714</b>	4780	0.990	0.63	<b>0.900</b>	5607	<i>1.162</i>	0.60	<b>0.857</b>
markshare_3_4	3352	1	2349	<b>0.701</b>	0.50	1.000	3160	<b>0.943</b>	0.50	1.000	3938	<i>1.175</i>	0.50	1.000
markshare_3_5	2481	1	1421	<b>0.573</b>	0.50	1.000	4145	<i>1.671</i>	0.56	<i>1.120</i>	4699	<i>1.894</i>	0.57	<i>1.140</i>
markshare_4_0	643068	121	168883	<b>0.263</b>	16.77	<b>0.139</b>	512822	<b>0.797</b>	102.12	<b>0.846</b>	1131625	<i>1.760</i>	101.99	<b>0.845</b>

cont. on next page

Table 5: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (conflict). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (improvement) and italic and red (deterioration).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
markshare_4_1	531216	137	214185	<b>0.403</b>	21.93	<b>0.160</b>	491671	<b>0.926</b>	135.45	0.987	1093586	<i>2.059</i>	88.70	<b>0.646</b>
markshare_4_2	369816	104	136123	<b>0.368</b>	13.52	<b>0.130</b>	359908	0.973	102.52	0.983	703463	<i>1.902</i>	56.86	<b>0.545</b>
markshare_4_4	420963	82	204375	<b>0.485</b>	20.46	<b>0.251</b>	398662	<b>0.947</b>	102.63	<i>1.257</i>	826934	<i>1.964</i>	68.57	<b>0.840</b>
markshare_4_5	504816	118	274293	<b>0.543</b>	27.82	<b>0.236</b>	1106341	<i>2.192</i>	339.34	<i>2.873</i>	1172893	<i>2.323</i>	105.06	<b>0.890</b>
mas74	3420026	685	3577586	1.046	717.66	1.047	3431529	1.003	677.46	0.989	3705205	<i>1.083</i>	724.45	<i>1.057</i>
mas76	519681	96	516536	0.994	86.56	<b>0.901</b>	519458	1.000	96.44	1.004	516019	0.993	87.27	<b>0.908</b>
mik-250-1-100-1	633761	315	578416	<b>0.913</b>	285.29	<b>0.905</b>	578416	<b>0.913</b>	287.72	<b>0.913</b>	633761	1.000	318.41	1.010
mine-166-5	6651	37	4551	<b>0.684</b>	27.87	<b>0.755</b>	5176	<b>0.778</b>	26.66	<b>0.723</b>	4860	<b>0.731</b>	36.66	0.994
mine-90-10	45418	214	137594	<i>3.030</i>	254.94	<i>1.191</i>	53696	<i>1.182</i>	141.84	<b>0.662</b>	1250935	<i>27.543</i>	3600.00	<i>16.812</i>
misc03	134	1	134	1.000	1.06	0.955	128	0.955	1.32	<i>1.189</i>	124	<b>0.925</b>	1.11	1.000
misc07	29168	21	38819	<i>1.331</i>	23.36	<i>1.125</i>	33828	<i>1.160</i>	21.58	1.039	31818	<i>1.091</i>	21.70	1.045
mzzv11	7274	1393	8588	<i>1.181</i>	1996.33	<i>1.433</i>	10482	<i>1.441</i>	1943.27	<i>1.395</i>	6352	<b>0.873</b>	1211.37	<b>0.870</b>
mzzv42z	3050	731	2320	<b>0.761</b>	529.74	<b>0.724</b>	1470	<b>0.482</b>	417.16	<b>0.570</b>	3332	<i>1.092</i>	939.63	<i>1.285</i>
neos-1061020	4040	1070	2704	<b>0.669</b>	701.54	<b>0.656</b>	4530	<i>1.121</i>	1128.83	<i>1.055</i>	22397	<i>5.544</i>	3600.00	<i>3.365</i>
neos-1109824	31218	163	42132	<i>1.350</i>	155.53	0.956	52693	<i>1.688</i>	210.15	<i>1.292</i>	48097	<i>1.541</i>	243.93	<i>1.500</i>
neos-1126860	5055	597	7015	<i>1.388</i>	589.86	0.989	8011	<i>1.585</i>	574.66	0.963	4877	0.965	552.22	<b>0.926</b>
neos-1173026	383075	1193	1305899	<i>3.409</i>	3291.14	<i>2.758</i>	1091276	<i>2.849</i>	3600.00	<i>3.017</i>	1359963	<i>3.550</i>	3600.00	<i>3.017</i>
neos-1208069	6041	457	161143	<i>26.675</i>	2583.86	<i>5.656</i>	12846	<i>2.126</i>	599.19	<i>1.312</i>	126495	<i>20.939</i>	3600.00	<i>7.881</i>
neos-1208135	8457	1139	60146	<i>7.112</i>	2193.83	<i>1.926</i>	33349	<i>3.943</i>	1893.66	<i>1.662</i>	188506	<i>22.290</i>	3600.00	<i>3.160</i>
neos-1215259	2880	101	1645	<b>0.571</b>	56.00	<b>0.554</b>	4310	<i>1.497</i>	118.86	<i>1.177</i>	2557	<b>0.888</b>	89.30	<b>0.884</b>
neos-1215891	3890	1409	80007	<i>20.567</i>	3600.00	<i>2.554</i>	5344	<i>1.374</i>	1412.94	1.003	52912	<i>13.602</i>	3600.00	<i>2.554</i>
neos-1223462	1472	819	1878	<i>1.276</i>	1020.01	<i>1.245</i>	586	<b>0.398</b>	794.53	0.970	1350	<b>0.917</b>	613.92	<b>0.750</b>
neos-1281048	1384	12	1221	<b>0.882</b>	10.75	<b>0.930</b>	1120	<b>0.809</b>	10.64	<b>0.920</b>	1753	<i>1.267</i>	12.86	<i>1.112</i>
neos-1396125	32266	708	159271	<i>4.936</i>	2393.14	<i>3.381</i>	151173	<i>4.685</i>	2272.84	<i>3.211</i>	36926	<i>1.144</i>	920.86	<i>1.301</i>
neos-1420205	55079	18	26789	<b>0.486</b>	9.57	<b>0.545</b>	20946	<b>0.380</b>	7.37	<b>0.420</b>	25828	<b>0.469</b>	9.82	<b>0.560</b>

cont. on next page

Table 5: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-1440225	1461	47	5584	<i>3.822</i>	152.10	<i>3.209</i>	127980	<i>87.598</i>	2620.81	<i>55.291</i>	1756	<i>1.202</i>	56.22	<i>1.186</i>
neos-1460265	43673	66	26519	<b>0.607</b>	55.96	<b>0.850</b>	23940	<b>0.548</b>	52.86	<b>0.803</b>	10826	<b>0.248</b>	40.66	<b>0.618</b>
neos-1461051	3401	13	301005	<i>88.505</i>	222.35	<i>17.535</i>	546261	<i>160.618</i>	408.46	<i>32.213</i>	3608	<i>1.061</i>	13.10	1.033
neos-1480121	5608250	3600	2359174	<b>0.421</b>	324.99	<b>0.090</b>	17799189	<i>3.174</i>	3600.00	1.000	328071	<b>0.058</b>	58.65	<b>0.016</b>
neos-1582420	59182	621	31105	<b>0.526</b>	391.81	<b>0.630</b>	17976	<b>0.304</b>	323.46	<b>0.520</b>	27241	<b>0.460</b>	396.75	<b>0.638</b>
neos-1620807	1812678	1543	6291083	<i>3.471</i>	1844.94	<i>1.195</i>	4881156	<i>2.693</i>	1483.78	0.961	1654801	<b>0.913</b>	1400.46	<b>0.907</b>
neos-430149	25985	24	59691	<i>2.297</i>	48.23	<i>1.995</i>	30031	<i>1.156</i>	25.36	1.049	49726	<i>1.914</i>	39.88	<i>1.649</i>
neos-503737	9330	354	1743	<b>0.187</b>	131.56	<b>0.371</b>	7813	<b>0.837</b>	260.84	<b>0.736</b>	5797	<b>0.621</b>	232.06	<b>0.655</b>
neos-504674	36880	78	27888	<b>0.756</b>	51.73	<b>0.665</b>	32446	<b>0.880</b>	66.97	<b>0.861</b>	36880	1.000	77.56	0.997
neos-538867	46052	53	119585	<i>2.597</i>	88.74	<i>1.668</i>	105752	<i>2.296</i>	68.48	<i>1.287</i>	40285	<b>0.875</b>	47.81	<b>0.899</b>
neos-538916	29500	42	50111	<i>1.699</i>	44.14	1.047	49107	<i>1.665</i>	41.35	0.981	30631	1.038	43.95	1.043
neos-551991	5448	319	10487	<i>1.925</i>	1022.46	<i>3.205</i>	10487	<i>1.925</i>	1019.42	<i>3.195</i>	5448	1.000	321.11	1.006
neos-555298	27669	200	37347	<i>1.350</i>	329.25	<i>1.642</i>	39681	<i>1.434</i>	333.98	<i>1.666</i>	7968	<b>0.288</b>	98.32	<b>0.490</b>
neos-584851	351	27	424	<i>1.208</i>	26.76	1.008	424	<i>1.208</i>	26.37	0.994	351	1.000	26.50	0.998
neos-585192	1577	35	1967	<i>1.247</i>	36.92	<i>1.068</i>	2300	<i>1.458</i>	39.45	<i>1.141</i>	1683	<i>1.067</i>	34.59	1.001
neos-595925	27416	59	14296	<b>0.521</b>	39.84	<b>0.672</b>	15337	<b>0.559</b>	37.96	<b>0.640</b>	9774	<b>0.357</b>	27.71	<b>0.467</b>
neos-686190	149683	1079	61808	<b>0.413</b>	479.98	<b>0.445</b>	74321	<b>0.497</b>	629.68	<b>0.584</b>	30924	<b>0.207</b>	245.01	<b>0.227</b>
neos-717614	265333	414	530703	<i>2.000</i>	749.59	<i>1.813</i>	2388414	<i>9.002</i>	3600.00	<i>8.705</i>	575150	<i>2.168</i>	989.23	<i>2.392</i>
neos-785912	267	107	797	<i>2.985</i>	185.41	<i>1.732</i>	346	<i>1.296</i>	99.28	<b>0.928</b>	6577	<i>24.633</i>	585.12	<i>5.467</i>
neos-791021	636	1787	188	<b>0.296</b>	699.31	<b>0.391</b>	2738	<i>4.305</i>	2466.50	<i>1.380</i>	423	<b>0.665</b>	792.06	<b>0.443</b>
neos-803219	22155	35	24978	<i>1.127</i>	42.68	<i>1.223</i>	25400	<i>1.146</i>	38.17	<i>1.094</i>	32205	<i>1.454</i>	45.11	<i>1.293</i>
neos-803220	55512	89	48674	<b>0.877</b>	92.19	1.036	55071	0.992	89.53	1.006	51782	<b>0.933</b>	80.60	<b>0.905</b>
neos-806323	10041	29	14620	<i>1.456</i>	39.99	<i>1.378</i>	10165	1.012	29.62	1.021	10117	1.008	28.52	0.983
neos-807639	6123	16	3509	<b>0.573</b>	13.33	<b>0.819</b>	6265	1.023	16.42	1.009	6245	1.020	16.46	1.012
neos-807705	9210	24	5174	<b>0.562</b>	15.09	<b>0.623</b>	4870	<b>0.529</b>	14.70	<b>0.607</b>	5270	<b>0.572</b>	15.47	<b>0.639</b>

cont. on next page

Table 5: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray				inf. LP				dom. prop.			
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
neos-810286	945	502	381	<b>0.403</b>	122.82	<b>0.245</b>	100	<b>0.106</b>	82.65	<b>0.165</b>	454	<b>0.480</b>	430.23	<b>0.856</b>
neos-810326	2441	104	1188	<b>0.487</b>	61.46	<b>0.589</b>	1188	<b>0.487</b>	62.08	<b>0.594</b>	2312	<b>0.947</b>	96.79	<b>0.927</b>
neos-827015	187	1376	569	<i>3.043</i>	3268.56	<i>2.376</i>	517	<i>2.765</i>	3144.85	<i>2.286</i>	158	<b>0.845</b>	1826.65	<i>1.328</i>
neos-831188	4962	483	3861	<b>0.778</b>	372.08	<b>0.770</b>	4011	<b>0.808</b>	421.50	<b>0.873</b>	13608	<i>2.742</i>	1260.03	<i>2.609</i>
neos-839859	18823	1130	20996	<i>1.115</i>	1214.60	<i>1.075</i>	23734	<i>1.261</i>	1325.53	<i>1.173</i>	23683	<i>1.258</i>	1680.63	<i>1.487</i>
neos-848845	36361	923	311122	<i>8.556</i>	3600.00	<i>3.900</i>	141118	<i>3.881</i>	2737.30	<i>2.966</i>	50669	<i>1.393</i>	1030.13	<i>1.116</i>
neos-849702	34345	647	299757	<i>8.728</i>	3600.00	<i>5.562</i>	258988	<i>7.541</i>	3600.00	<i>5.562</i>	24076	<b>0.701</b>	483.54	<b>0.747</b>
neos-862348	1516	30	5222	<i>3.445</i>	38.85	<i>1.315</i>	3581	<i>2.362</i>	37.16	<i>1.258</i>	4024	<i>2.654</i>	40.14	<i>1.359</i>
neos-863472	262229	220	714473	<i>2.725</i>	453.29	<i>2.056</i>	432913	<i>1.651</i>	323.06	<i>1.465</i>	296521	<i>1.131</i>	245.66	<i>1.114</i>
neos-886822	66990	244	35286	<b>0.527</b>	218.88	<b>0.898</b>	84519	<i>1.262</i>	236.22	0.969	60753	<b>0.907</b>	239.36	0.982
neos-892255	1612	448	930	<b>0.577</b>	288.40	<b>0.643</b>	1516	<b>0.940</b>	401.37	<b>0.895</b>	1852	<i>1.149</i>	520.46	<i>1.161</i>
neos-905856	4334	55	33444	<i>7.717</i>	220.39	<i>3.979</i>	8228	<i>1.898</i>	90.27	<i>1.630</i>	60237	<i>13.899</i>	378.37	<i>6.831</i>
neos-906865	48822	122	49529	1.014	123.62	1.010	49183	1.007	123.64	1.010	48822	1.000	122.45	1.001
neos-912023	1827	21	53733	<i>29.411</i>	239.03	<i>11.291</i>	221277	<i>121.115</i>	881.14	<i>41.622</i>	1111	<b>0.608</b>	16.68	<b>0.788</b>
neos-914441	354	227	180	<b>0.508</b>	155.90	<b>0.686</b>	166	<b>0.469</b>	153.96	<b>0.678</b>	182	<b>0.514</b>	154.60	<b>0.680</b>
neos-942323	2567	16	4610	<i>1.796</i>	18.95	<i>1.220</i>	22291	<i>8.684</i>	44.25	<i>2.849</i>	674	<b>0.263</b>	10.41	<b>0.670</b>
neos18	4919	27	76980	<i>15.650</i>	145.81	<i>5.498</i>	352971	<i>71.757</i>	447.33	<i>16.868</i>	4919	1.000	26.58	1.002
net12	1514	1690	3903	<i>2.578</i>	3600.00	<i>2.130</i>	5342	<i>3.528</i>	3600.00	<i>2.130</i>	2621	<i>1.731</i>	2359.03	<i>1.396</i>
noswot	584962	141	1810982	<i>3.096</i>	354.55	<i>2.507</i>	1139205	<i>1.947</i>	229.39	<i>1.622</i>	880169	<i>1.505</i>	212.14	<i>1.500</i>
ns1208400	2495	3600	2296	<b>0.920</b>	2454.32	<b>0.682</b>	519	<b>0.208</b>	1550.37	<b>0.431</b>	5623	<i>2.254</i>	3600.00	1.000
ns1688347	4308	228	32931	<i>7.644</i>	874.84	<i>3.831</i>	19314	<i>4.483</i>	380.48	<i>1.666</i>	4175	0.969	335.34	<i>1.468</i>
ns1766074	915997	978	1029467	<i>1.124</i>	1483.94	<i>1.517</i>	936139	1.022	940.16	0.961	1373951	<i>1.500</i>	570.14	<b>0.583</b>
ns1830653	39491	351	73960	<i>1.873</i>	619.02	<i>1.762</i>	64211	<i>1.626</i>	584.50	<i>1.664</i>	27436	<b>0.695</b>	313.80	<b>0.893</b>
pg5_34	300472	1332	298798	0.994	1317.25	0.989	300472	1.000	1332.08	1.000	298798	0.994	1313.72	0.986
prod1	29058	18	29822	1.026	15.83	<b>0.885</b>	25958	<b>0.893</b>	16.12	<b>0.901</b>	26142	<b>0.900</b>	14.58	<b>0.815</b>

cont. on next page

Table 5: Detailed computational results on CONFLICT test set. The table shows the absolute number of generated nodes ( $n$ ) and needed solving time in seconds ( $t$ ), as well as the relative number of generated nodes ( $n_Q$ ) and needed solving time ( $t_Q$ ) w.r.t. Columns 2 and 3 (**conflict**). All changes in the number of nodes or solving time of at least 5% are highlighted in bold and blue (**improvement**) and italic and red (*deterioration*).

Instance	conflict		dualray			inf. LP				dom. prop.				
	$n$	$t$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$	$n$	$n_Q$	$t$	$t_Q$
prod2	78750	63	150765	<i>1.914</i>	104.46	<i>1.649</i>	113606	<i>1.443</i>	91.29	<i>1.441</i>	118860	<i>1.509</i>	89.94	<i>1.420</i>
pw-myciel4	569026	3600	627543	<i>1.103</i>	2708.26	<b>0.752</b>	619421	<i>1.089</i>	3600.00	1.000	697295	<i>1.225</i>	3600.00	1.000
reblock67	128157	235	74055	<b>0.578</b>	130.51	<b>0.555</b>	101852	<b>0.795</b>	175.47	<b>0.746</b>	119391	<b>0.932</b>	201.96	<b>0.859</b>
rmine6	562634	1144	807204	<i>1.435</i>	1544.44	<i>1.350</i>	1010123	<i>1.795</i>	2175.24	<i>1.902</i>	484579	<b>0.861</b>	1010.23	<b>0.883</b>
rococoC10-001000	298459	2527	353552	<i>1.185</i>	2185.99	<b>0.865</b>	217769	<b>0.730</b>	1594.96	<b>0.631</b>	361057	<i>1.210</i>	2738.66	<i>1.084</i>
rout	49976	58	137696	<i>2.755</i>	120.82	<i>2.099</i>	168051	<i>3.363</i>	135.10	<i>2.347</i>	94508	<i>1.891</i>	103.05	<i>1.790</i>
satellites1-25	471	1859	1982	<i>4.208</i>	3600.00	<i>1.937</i>	6212	<i>13.189</i>	1930.08	1.038	1082	<i>2.297</i>	3600.00	<i>1.937</i>
timtab1	843361	434	872675	1.035	489.78	<i>1.128</i>	863707	1.024	441.72	1.017	3612719	<i>4.284</i>	1782.76	<i>4.106</i>
geom.	16183.61	101.28	26196.478	1.619	128.768	1.271	26064.206	1.611	135.274	1.336	20016.548	1.237	108.826	1.075
sh. geom. [100, 10]	16769.05	143.22	27097.915	1.616	179.910	1.256	27119.469	1.617	189.288	1.322	20728.280	1.236	155.943	1.089