Zuse Institute Berlin

Takustrasse 7
D-14195 Berlin-Dahlem
Germany

AMBROS M. GLEIXNER

TIMO BERTHOLD

BENJAMIN MÜLLER

STEFAN WELTGE

Three Enhancements for Optimization-Based Bound Tightening

This report has been published in the Journal of Global Optimization. Please cite as:

A. Gleixner, T. Berthold, B. Müller, S. Weltge. Three Enhancements for Optimization-Based Bound Tightening. Journal of Global Optimization, 67(4):731–757, DOI:10.1007/s10898-016-0450-4

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

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

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

Three Enhancements for Optimization-Based Bound Tightening*

Ambros M. Gleixner † Timo Berthold ‡ Benjamin Müller § Stefan Weltge ¶

March 7, 2016

Abstract

Optimization-based bound tightening (OBBT) is one of the most effective procedures to reduce variable domains of nonconvex mixed-integer nonlinear programs (MINLPs). At the same time it is one of the most expensive bound tightening procedures, since it solves auxiliary linear programs (LPs)—up to twice the number of variables many. The main goal of this paper is to discuss algorithmic techniques for an efficient implementation of OBBT.

Most state-of-the-art MINLP solvers apply some restricted version of OBBT and it seems to be common belief that OBBT is beneficial if only one is able to keep its computational cost under control. To this end, we introduce three techniques to increase the efficiency of OBBT: filtering strategies to reduce the number of solved LPs, ordering heuristics to exploit simplex warm starts, and the generation of Lagrangian variable bounds (LVBs). The propagation of LVBs during tree search is a fast approximation to OBBT without the need to solve auxiliary LPs.

We conduct extensive computational experiments on MINLPLib2. Our results indicate that OBBT is most beneficial on hard instances, for which we observe a speedup of 17% to 19% on average. Most importantly, more instances can be solved when using OBBT.

Keywords. MINLP, optimization-based bound tightening, optimality-based bound tightening, OBBT, propagation, bound tightening

1 Introduction

The state-of-the-art algorithm for solving general factorable nonconvex mixed-integer nonlinear programs to ε-global optimality is spatial branch-and-bound, see, e.g., [29, 47, 48]. Its performance mainly depends on the tightness of the convex relaxation, which in turn depends on the bounds of the variables involved in nonconvex nonlinear terms. For this reason, bound tightening algorithms have become a crucial component of global MINLP solvers. This paper is concerned with *optimization-based bound tightening*, in short *OBBT*, which in its most basic form is a straightforward technique that minimizes and maximizes each variable over the relaxation at hand. We develop methods to more efficiently deduce information from OBBT and to exploit dual information to learn valid inequalities. This reduces the overall computational cost of OBBT and the whole solving process.

We consider *mixed-integer nonlinear programs (MINLPs)* of the form

$$\begin{aligned} \min\{c^\mathsf{T} x \,|\, & g_k(x) \leqslant 0,\ k = 1,\dots,m, \\ & x \in [\ell, u], \\ & x_i \in \mathbb{Z} \text{ for } i \in \mathscr{I}\} \end{aligned} \tag{1}$$

where $c \in \mathbb{R}^n$ is the objective function vector, ℓ and u are the vectors of lower bounds $\ell \in (\mathbb{R} \cup \{-\infty\})^n$ and upper bounds $(u_i \in \mathbb{R} \cup \{+\infty\})^n$, $\mathscr{I} \subseteq \{1,\dots,n\}$ is the index set of variables required to take integral values, and the constraint functions $g_k : [\ell, u] \to \mathbb{R}$ (and hence the feasible region) may be nonconvex. We do not make any assumptions on the smoothness of the nonlinearities, since there are interesting applications with nonsmooth components, e.g., piecewise-linear functions, and algorithms that are based on polyhedral relaxations often allow nondifferentiabilities, see, e.g., [56].

*This work was supported by the Research Campus Modal "Mathematical Optimization and Data Analysis Laboratories" funded by the German Ministry of Education and Research.

†Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, gleixner@zib.de

‡Fair Isaac Europe Ltd, c/o Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, timoberthold@fico.com

§Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, benjamin.mueller@zib.de

¶Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany, weltge@ovgu.de

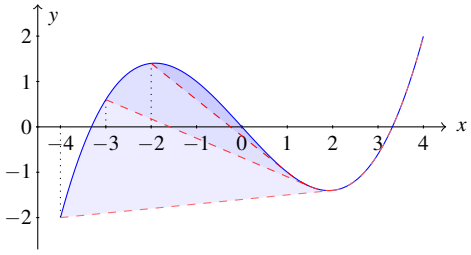


Figure 1: Convex envelope of $x \mapsto 0.1x^3 - 1.1x$ over domains $[-4,4]$, $[-3,4]$, and $[-2,4]$

Note that many MINLP solvers require the constraint functions to be *factorable* [36]. Furthermore, we may restrict ourselves to a linear objective function since we can rewrite any nonlinear objective function f as constraint $f(x) - x_0 \leqslant 0$ and minimize the additional auxiliary variable x_0.

There are many real-world applications that are inherently nonlinear and need to be tackled by MINLP, for an overview see, e.g., [27]. This induces a growing need for MINLP algorithms that are at the same time innovative from a theoretical perspective and efficient in practice, properly integrated into common, stable software implementations that target solving instances of relevant size from a broad field of applications.

In the following, we will discuss the particular importance of tight bounds for solving nonconvex MINLPs with spatial branch-and-bound and give an overview of existing bound tightening techniques from the literature.

1.1 Domain-Dependent Relaxations

The function of bound tightening algorithms for MINLPs is twofold: first to strengthen the associated convex relaxation and second to reduce the size of the domains of the variables over which enumerative search is performed. Though these objectives are different, they are closely related.

An important characteristic of branching on continuous variables is that the optimum x^* of the relaxation on which branching is performed typically stays inside the domains of both created subproblems. This stands in marked contrast to branching on integer variables with a fractional relaxation value, where x^* is immediately excluded from the domains of the subproblems by rounding. Besides generating locally valid outer approximation cuts, bound tightening is one of the most important features for an efficient implementation of a branch-and-bound algorithm for nonconvex MINLPs, see, e.g., [24].

Figure 1 illustrates how the convex envelope of the nonconvex function $x \mapsto 0.1x^3 - 1.1x$ depends on the size of the domain over which it is computed: As the lower bound on variable x increases, the convex envelope approaches the graph of the function. For $x \geqslant 0$, we can see that the function even becomes convex; in this case, i.e., if convexity is known, the feasibility of the constraint $x \mapsto 0.1x^3 - 1.1x \leqslant y$ could be enforced by separation instead of spatial branching, which is much more efficient.

This example demonstrates that *tighter bounds imply tighter relaxations* beyond the trivial fact that they reduce the size of the domains. In the following we will see that, vice versa, *tighter relaxations imply tighter bounds*.

1.2 Literature Review: Bound Tightening for MINLP

Presolving algorithms in mathematical programming aim at improving the performance and numerical robustness of solution algorithms by transforming the problem so that each optimal solution of the resulting problem can be transformed back to an optimal solution of the original formulation. Bound tightening algorithms may be considered to be a restricted form of presolving in which extensive reformulations of the problem are limited in a way that they can be applied easily during the branch-and-bound search. For mixed-integer linear programming, their study can be traced back to the seminal work of Dantzig et al. [20], who apply what is today known as *reduced cost fixing* to an instance of the traveling salesman problem with 49 cities. We refer to [50], [22], [1], and the recent study of [23] for an overview of presolving and propagation techniques for general MIPs. The following focuses on the nonlinear case.

Under *feasibility-based bound tightening* (FBBT), we subsume bound tightening algorithms that use purely primal feasibility arguments and remove parts of the domains in which no feasible solutions are contained. The most fundamental of these techniques is based on interval arithmetic, see, e.g., [9]. It amounts to computing the activity of nonlinear expressions over the domains of the variables (forward propagation) and conversely propagating the bounds on the constraint

activities back to the bounds of the variables (backward propagation). Implementations usually rely on the representation of the nonlinear terms as nodes of an expression graph, see [10] or [57] for details.

As Belotti et al. [8, 9] point out, this is an iterative process that may produce a series of smaller and smaller domains that converge only in the limit, even if all constraints are linear. As a consequence, MIP solvers typically restrict the maximum rounds of consecutive FBBT propagations. Belotti et al. show that the resulting vector of intervals can be interpreted as a fixed point of an operator on a suitable lattice and—for linear constraints—can be computed in finite time by solving a large linear program. Feasibility-based reductions can be strengthened by exploiting integer information as in MIP solvers. Examples are globally available information such as implications between variables and mutually exclusive binary assignments derived from constraints with knapsack-like structure, see, e.g., [1, Chap. 7, 10]. A more expensive form of FBBT is presented by Belotti [7], who proposes to apply FBBT to convex combinations of two linear inequalities of the LP relaxation.

Optimality-based reductions (not to be confused with OBBT) may additionally discard feasible and even optimal solutions as long as they guarantee that at least one optimal solution remains feasible for the reduced problem. They often exploit the knowledge of an *objective cut-off*, i.e., given that we are minimizing, an upper bound on the optimal objective function value of the MINLP, which can be obtained from the best known solution. A classical example is the technique of *marginals-based bound tightening* by [49], which extends reduced cost bound tightening for MIPs by using Lagrange multipliers from convex relaxations. Suppose $g(x) \leqslant g_0$ denotes a constraint that is active in an optimal solution to a convex NLP relaxation with associated dual multiplier $\lambda > 0$, and let U be the objective cut-off and L a lower bound on the optimal objective function value (typically from the relaxation), then

$$g(x) \geqslant g_0 - (U-L)/\lambda \tag{2}$$

is valid for all solutions with better (lower) objective function value than U.

If the constraint $g(x) \leqslant g_0$ is not active, then λ is zero and (2) cannot be applied. However, if we fix $g(x) = g_0$, solve the relaxation again, and obtain a tighter bound $\tilde{L} > U$, then this proves that $g(x) < g_0$ must hold for all optimal solutions. If the associated dual multiplier $\tilde{\lambda}$ in this restricted relaxation is positive, then

$$g(x) \leqslant g_0 - (U-\tilde{L})/\tilde{\lambda} \tag{3}$$

is valid. This is typically applied for the special case of bound constraints, i.e., when $g(x) = x_i$ for some variable x_i. In this case, the presented inequalities immediately yield tighter domains.

Note that there are optimality-based arguments that do not rely on the explicit availability of dual multipliers. Simple examples are the FBBT-like propagation of the objective function versus the objective cut-off value [1, Sec. 7.6] or the removal of dominated columns in linear programming [60].

The term *shaving* or *aggressive FBBT* (see, e.g., [43]) refers to what is called *probing* in mixed-integer linear programming [50]: tentatively tightening the lower or upper bound of a single variable and applying propagation in the hope of proving infeasibility and excluding this subdomain from further consideration. This is typically performed without solving relaxations, but it can include both feasibility- and optimality-based reductions. While in MIP solvers this is typically applied to binary variables only, see, e.g., [1], for MINLPs this has been found beneficial also for continuous variables [10, 41]. Iterating over all variables, it is considerably more expensive than applying FBBT to the global domains. At the same time it is potentially stronger because it exploits the appearance of variables across several constraints. Inequality (3) can be considered as an expensive form of shaving that solves the (restricted) relaxation.

In the related context of branching, [55] consider the situation when a node is pruned because the dual bound of the relaxation exceeds the primal objective limit. They discuss the possibility of extending the dual argument to a larger box and hence remove parts of the unpruned sibling node under specific assumptions.

In general, techniques based on dual relaxation solutions as reviewed above may only be considered cheap because the vector of dual multipliers is readily available after solving the relaxation. A technique that requires the solution of *additional* relaxations is optimization-based bound tightening, which is the central topic of this paper and will be explained in detail in the next section.

1.3 Optimization-Based Bound Tightening

Given an MINLP as in (1) with a convex relaxation $\mathscr{R}$ of its feasible region, classical OBBT computes the tightest bounds valid for all relaxation solutions by in turn minimizing and maximizing each variable,

$$\min/\max\{x_k \mid x \in \mathscr{R}\}. \tag{4}$$

Often a polyhedral relaxation is used. In this case, applying a full round of OBBT amounts to solving $2n$ linear programs. Even though this is possible efficiently, it is an expensive algorithm when compared to the average amount of work performed at a node of a spatial branch-and-bound tree.

This straightforward idea must have been folklore knowledge in the mathematical programming community for a long time. It was first mentioned in the global optimization literature by Quesada and Grossmann [46] in the context of optimizing heat exchanger networks. Soon after, it became a regular component of generic global optimization algorithms, see [47], [34], or [52] for examples. In these references it is described as a bound tightening procedure applied at the root node of a tree search, i.e., to tighten the global bounds of the MINLP.

If an upper bound on the optimal objective function value is available, OBBT can be strengthened by adding an objective cut-off to the relaxation,

$$\min/\max\{x_k \mid x \in \mathscr{R}, c^\mathsf{T} x \leqslant U\}, \tag{5}$$

which renders it an optimality-based procedure. Zamora and Grossmann [62] have first used this idea in a "branch-and-contract" algorithm, which employs OBBT aggressively at every node of a spatial branch-and-bound tree, solving convex NLP relaxations. In order to control the effort that is spent on solving NLPs, they develop strategies to apply OBBT selectively to those variables which appear in nonconvex terms of the nonlinear functions and to those bounds which have large distance from the value of the variable in the current solution of the relaxation.

Examples of MINLP solvers implementing OBBT are αBB [3, 4], ANTIGONE [38, 40], Couenne [10, 17], LaGO [32, 44], and SCIP [1, 51, 57]. It is mostly applied at the root node and within the search tree only with limited frequency or based on its success rate. ANTIGONE, for instance, measures the success of OBBT by the reduction of the box volume and disables it for all children nodes once the rate of reduction drops below a given threshold. Even then it restarts OBBT within the search tree with a probability of $2^{\lambda - \text{tree depth}}$, $\lambda = 1$ by default. Couenne implements a similar strategy.

Because OBBT uses the convex relaxation directly, it is a particularly good example for the mutual interaction between a tight relaxation and tight bounds as discussed in Section 1.1. If the relaxation is refined after applying OBBT, another application of OBBT may result in further bound strengthenings. In this spirit, [14] present a theoretical study of an iterated version of OBBT.

The increased computational effort of OBBT is often justified because it can provide tighter bounds than the ones computed by many cheaper methods. Its strength lies in the consideration of (the relaxations of) all constraints at once and in combination with the objective cut-off. As such, it dominates any bound tightening technique that relies solely on the relaxation. This includes, e.g., FBBT on the linear relaxation and optimality-based reduced cost propagation.

Remark 1 (Optimization-Based and Optimality-Based Bound Tightening)**.** *Note that, while OBBT is sometimes labeled* optimality-*based bound tightening, this is slightly misleading: It is only one among many bound tightening procedures that exploit optimality and is applicable also when no objective cut-off limit is available. The defining feature of OBBT is the solution of* optimization *problems, hence we use it as an abbreviation for* optimization-based bound tightening*. In this paper we consider the solution of linear programs, but the optimization problems may be of more general form such as convex NLPs [62] or even partially solved MINLPs [30, 33].*

The rest of the paper is organized as follows. In Section 2, we show how dual solutions encountered during OBBT can be used to derive valid inequalities that approximate the effect of OBBT when propagated during the branch-and-bound search. In Section 3, we discuss strategies to reduce the computational cost of OBBT by saving unnecessary LP solves and exploiting the warm-starting capability of the simplex algorithm through a greedy ordering heuristic. We use an implementation based on the MINLP solver SCIP [51] to analyze the performance impact of these methods on a set of publicly available benchmark instances in Section 4. Section 5 gives concluding remarks.

2 One-Row Relaxations by OBBT

This section presents our main contribution: a new technique exploiting the potentially expensive solution of the OBBT LPs (5) beyond simply obtaining tighter bounds on variable x_k. To this end, we observe that the proof of optimality given by a dual solution of (5) can be used to generate globally valid inequalities that approximate OBBT locally.

2.1 Lagrangian Variable Bounds

Besides valid bounds for variable x_k, solving (5) yields dual multipliers that prove that no $x \in \mathscr{R}$ with $c^\mathsf{T} x \leqslant U$—and by that no MINLP solution with improved objective function value—can lie outside these bounds. The following theorem uses basic LP duality to derive this proof in form of a valid inequality.

Theorem 1 (Lagrangian Variable Bounds)**.** *Suppose we have a linear relaxation of an MINLP of form* (1)*, the feasibility set of this relaxation be*

$$\mathscr{R} = \{x \in \mathbb{R}^n \mid Ax \geqslant b, x \in [\ell, u]\} \tag{6}$$

with $A \in \mathbb{R}^{p\times n}$ and $b \in \mathbb{R}^p$, and a valid upper bound U on the optimal objective value of (1)*. Furthermore, suppose we have an optimal primal–dual solution $(\tilde{x}, \tilde{\lambda}, \tilde{\mu})$ to*

$$\min\{x_k \mid Ax \geqslant b, c^\mathsf{T}x \leqslant U, x \in [\ell, u]\} \tag{7}$$

where $\tilde{\lambda} \in \mathbb{R}^p_{\geq 0}$ is the vector of dual multipliers for $Ax \geqslant b$ and $\tilde{\mu} \leqslant 0$ is the dual multiplier associated with the objective cut-off constraint.

Let $\tilde{r} = e_k - A^\mathsf{T}\tilde{\lambda} - c\tilde{\mu}$ be the vector of reduced costs, where e_k is the k-th unit vector. Then

$$x_k \geqslant \sum_{j=1,\dots,n} \tilde{r}_j x_j + \tilde{\mu}U + \tilde{\lambda}^\mathsf{T} b \tag{8}$$

is a valid inequality for all $x \in \mathscr{R} \cap \{x \in \mathbb{R}^n \mid c^\mathsf{T}x \leqslant U\}$, which is tight at $\tilde{x}$. If $(\tilde{x}, \tilde{\lambda}, \tilde{\mu})$ with $\tilde{\lambda} \leqslant 0$ and $\tilde{\mu} \geqslant 0$ is optimal for

$$\max\{x_k \mid Ax \geqslant b, c^\mathsf{T}x \leqslant U, x \in [\ell, u]\} \tag{9}$$

then the same holds for

$$x_k \leqslant \sum_{j=1,\dots,n} \tilde{r}_j x_j + \tilde{\mu}U + \tilde{\lambda}^\mathsf{T} b. \tag{10}$$

Proof. Aggregating the rows of $Ax \geqslant b$ with $\tilde{\lambda}$ and adding $\tilde{\mu}$ times the cut-off constraint gives the valid inequality $(\tilde{\lambda}^\mathsf{T}A + \tilde{\mu}c^\mathsf{T})x \geqslant \tilde{\lambda}^\mathsf{T}b + \tilde{\mu}U$ in case of minimization and $(\tilde{\lambda}^\mathsf{T}A + \tilde{\mu}c^\mathsf{T})x \leqslant \tilde{\lambda}^\mathsf{T}b + \tilde{\mu}U$ for maximization. By definition of $\tilde{r}$, this is equivalent to (8) and (10), respectively. The tightness of (8) and (10) follows from complementary slackness. □

We will refer to valid inequalities of type (8) and (10) as *Lagrangian variable bounds* (LVBs). As can be seen, they are a consequence of basic duality theory. We want to note that their validity also follows from a special instantiation of the (PU-PR) range reduction framework presented in [55].

LVBs can be viewed as a one-row relaxation of the given MINLP. In mixed-integer linear programming one-row relaxations are a fundamental ingredient in the generation of general-purpose cutting planes such as Gomory mixed-integer cuts [25, 26], mixed-integer rounding cuts [42, 35], or Chvátal-Gomory cuts [15, 13]. We will use this concept for fast propagation.

Remark 2 (Nonexisting and Trivial LVBs)**.** *It is not guaranteed that useful LVBs exist:*

1. *If OBBT fails due to* (7) *or* (9) *being unbounded, then no dual feasible solution exists from which an LVB can be derived.*
2. *If the OBBT bound is implied only by problem constraints, i.e., $\tilde{r}$ and $\tilde{\mu}$ are zero in the dual solution, then the right-hand side of the LVB is the constant $\tilde{\lambda}^\mathsf{T}b$ and the LVB reads $x_k \geqslant \tilde{x}_k$ or $x_k \leqslant \tilde{x}_k$.*
3. *If OBBT fails to tighten bounds, i.e.,* (7) *equals ℓ_k or* (9) *equals u_k, then $\tilde{\lambda} = 0$, $\tilde{\mu} = 0$ are optimal dual multipliers. For these,* (8) *and* (10) *read $x_k \geqslant x_k$ and $x_k \leqslant x_k$, respectively.*

In the last two cases we call the LVBs trivial *since they do not contain any useful information. We call them* nontrivial *if a variable other than x_k appears with a nonzero multiplier on the right-hand side of* (8) *or* (10)*.*

Note that in the presence of dual degeneracy, the optimal dual multipliers $\tilde{\lambda}$ and $\tilde{\mu}$ may be nonzero even when OBBT fails to tighten the bound as in case three. We have encountered this in our experiments. In general, LVBs are frequently nontrivial as observed empirically in Section 4. This is illustrated by the following example.

Example 1 (Nontrivial LVBs)**.** *Consider the nonconvex, polynomial NLP*

$$\min\{y - x \mid y = 0.1x^3 - 1.1x, x \in [-4, 4], y \in [-2, 2]\}. \tag{11}$$

Figure 2a shows its feasible region—a one-dimensional curve—between the initial bounds -4 and 4. The shaded region over which OBBT is performed is defined by the linear relaxation

$$\begin{aligned}\mathscr{R} = \{(x,y) \in [-4,4] \times [-2,2] \mid\ & -0.1x + y \geqslant -1.6,\\ & -3.7x + y \geqslant -12.8,\\ & 0.1x - y \geqslant -1.6,\\ & 3.7x - y \geqslant -12.8\}\end{aligned} \tag{12}$$

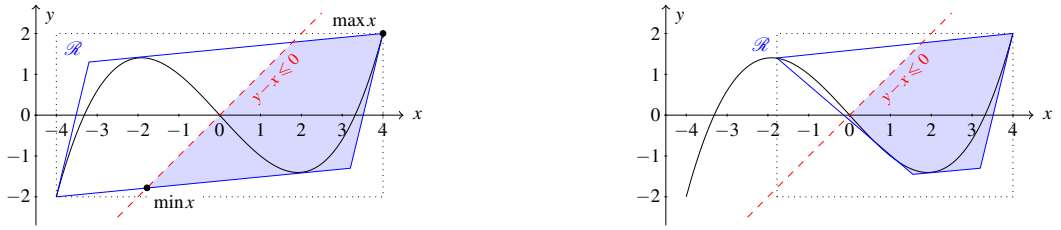
and the dashed objective cut-off resulting from the zero solution. Minimizing x over this region gives the dual solution $(\tilde{\lambda}, \tilde{\mu}) = (10/9, 0, 0, 0, -10/9)$, the lower bound $-16/9$, and the LVB

$$x \geqslant -\tfrac{10}{9}U - \tfrac{16}{9}. \tag{13}$$

As soon as a feasible solution with negative (better) objective function value is found, (13) *gives an even tighter lower bound. Maximizing x does not immediately tighten its upper bound, still the nontrivial LVB*

$$x \leqslant \tfrac{10}{37}y + \tfrac{128}{37} \tag{14}$$

can be generated from the dual solution $(\tilde{\lambda}, \tilde{\mu}) = (0, 0, 0, 10/37, 0)$. In this two-variable example, this is only the rightmost facet of $\mathscr{R}$. In general, this gives a valid, but redundant, inequality that might not be part of the relaxation. While the inequality is redundant for the relaxation, propagating it by using integrality information on the variables might lead to bound tightenings that could not be deduced from single rows of the relaxation. Figure 2b shows the new bounds and tightened relaxation.



(a) Initial relaxation

(b) Tightened bounds and relaxation

Figure 2: Two-variable example for OBBT with objective cut-off constraint

Example 13 above also demonstrates that applying another round of OBBT would tighten the lower bound of x even further—from $-16/9$ to almost zero. As can be seen, continuing to iterate between OBBT and the refinement of the relaxation would result in an infinite series of lower bounds on x that converge towards zero. The difficulty in iterating OBBT is its high computational cost. The propagation of LVBs as explained in the next section, however, is cheap and may still be effective.

2.2 OBBT via FBBT

Lagrangian variable bounds can be seen as a local approximation of the effect of OBBT. Instead of minimizing a variable x_k over the complete linear relaxation and the objective cut-off constraint as in (7), we could have obtained the same bound tightening by minimizing x_k over the one-row relaxation given by the optimal dual solution $(\tilde{\lambda}, \tilde{\mu})$, which is equivalent to the LVB (8):

$$x_k \geqslant \min\left\{x_k \mid Ax \geqslant b, c^\mathsf{T}x \leqslant U, x \in [\ell, u]\right\} \tag{15}$$

$$\geqslant \min\left\{x_k \mid (\tilde{\lambda}^\mathsf{T}A + \tilde{\mu}c^\mathsf{T})x \leqslant \tilde{\lambda}^\mathsf{T}b + \tilde{\mu}U, x \in [\ell, u]\right\} \tag{16}$$

$$= \min\left\{x_k \mid x_k \geqslant \sum_{j=1,\dots,n} \tilde{r}_j x_j + \tilde{\mu}U + \tilde{\lambda}^\mathsf{T}b, x \in [\ell, u]\right\}. \tag{17}$$

The last minimization is now easily performed by replacing the right-hand side variables of the LVB with one of their bounds, giving the valid lower bound

$$\sum_{j:\tilde{r}_j>0} \tilde{r}_j \ell_j + \sum_{j:\tilde{r}_j<0} \tilde{r}_j u_j + \tilde{\mu} U + \tilde{\lambda}^\mathsf{T} b. \tag{18}$$

For maximization, we obtain the analogous upper bound

$$\sum_{j:r_j>0} \tilde{r}_j u_j + \sum_{j:r_j<0} \tilde{r}_j \ell_j + \tilde{\mu} U + \tilde{\lambda}^\mathsf{T} b. \tag{19}$$

This is essentially the same as applying FBBT to the LVB inequalities.

When initially created, this gives the same bound as OBBT because $(\tilde{\lambda}, \tilde{\mu})$ are optimal and hence (16) holds with equality. Later during the search, whenever the domain of some variable x_j appearing with $\tilde{r}_j \neq 0$ on the right-hand side is reduced, the LVB implies a tighter bound.[1] This holds both for tighter globally valid bounds as well as for bounds tightened through branching. Furthermore, if an improved primal solution is found, then the value of U may also be updated in the LVB inequalities. This is because Theorem 1 holds for *any* valid upper bound U. If $\tilde{\mu} < 0$, then this also leads to a tighter value in (18) and (19).

In these cases, the LVB bound may be weaker than the bound that would be obtained from reapplying OBBT for two reasons. First, the dual solution from which the LVB was constructed is still dual feasible, but may not be optimal anymore; second, the LP relaxation may have been tightened by additional rows that were not present when performing OBBT initially. However, in stark contrast to OBBT, LVBs can be propagated very efficiently. This motivates the application of LVBs within a spatial branch-and-bound algorithm in the following scheme:

1. Generate LVBs while performing OBBT during the root node.
2. Propagate them locally at the nodes of the search tree whenever bounds appearing on the right-hand side are reduced by branching and/or propagation.
3. Propagate them globally whenever an improved primal solution is found.

The high-level observation that dual feasible solutions encountered during the solution process may be used to construct a one-row relaxation of the LP at hand and that this inequality can be used to tighten the bounds of each variable involved has been made earlier by [55]. Applied unconditionally, however, this idea seems overly expensive and it is rather unclear how to put it to good use. Here, we suggest selecting the dual solutions from OBBT and propagating the resulting LVBs only towards the left-hand side variable.

Remark 3 (Redundancy of LVBs)**.** *The main purpose of the LVBs is to identify bounds that are already implied by the relaxation and—by making them explicit—allow improvement of the convex relaxations of nonconvex nonlinear constraints. Unlike MIP cutting planes, LVBs are not designed to cut off the LP optimum. Since they are redundant inequalities, it is not beneficial to add them to the LP relaxation.*

2.3 The LVB Graph

While propagating a single Lagrangian variable bound is a trivial task, a successful tightening may strengthen the effect of other LVBs. This raises the question in which order LVBs should be propagated and how often to repeat a round of propagation.

We address this issue by considering a directed graph D defined as follows: For each variable x_j, D contains a node v_j and a node w_j representing its lower bound and upper bound, respectively. Let

$$x_k \geqslant \sum_{j=1,\dots,n} \tilde{r}_j x_j + \tilde{\mu} U + \tilde{\lambda}^\mathsf{T} b.$$

be the latest stored LVB implying a lower bound on x_k. For each variable x_j with $j \neq k$, we draw a directed arc

- from v_j to v_k if $\tilde{r}_j > 0$, or
- from w_j to v_k if $\tilde{r}_j < 0$.

[1] Precisely, if ℓ_j increases for $\tilde{r}_j > 0$ in (18) or $\tilde{r}_j < 0$ in (19), and if u_j decreases for $\tilde{r}_j < 0$ in (18) or $\tilde{r}_j > 0$ in (19).

If otherwise

$$x_k \leqslant \sum_{j=1,\dots,n} \tilde{r}_j x_j + \tilde{\mu} U + \tilde{\lambda}^\mathsf{T} b,$$

is the latest stored LVB implying an upper bound on x_k, for each variable x_j with $j \neq k$, we draw a directed arc

- from w_j to w_k if $\tilde{r}_j > 0$, or
- from v_j to w_k if $\tilde{r}_j < 0$.

In other words, we draw an arc (a,b) if an improvement of bound a strengthens the bound propagated by the latest stored LVB affecting bound b.

2.3.1 Sorting the LVB Graph

Suppose that D does not contain any directed cycle and fix some topological order of its nodes. Clearly, given some (locally) valid lower and upper variable bounds, once we have performed LVB propagation for each single LVB with respect to the order of nodes of D, no additional propagation of a single LVB can further tighten some bound.

Unfortunately, in general the graph D contains directed cycles and hence does not admit a topological order of its nodes. In this case, it is easy to construct scenarios in which every order of generated LVBs allows further improvements after one round of LVB propagation (by again propagating single LVBs). Even worse, it can be seen that the propagation of LVBs may exhibit the same non-finite convergence behavior as described in [8, 9] for standard FBBT-like propagation of linear constraints.

In order to at least reduce this effect, we propose to compute a partial (topological) order over D. First, for each connected component of D, compute its strongly connected components, shrink each of them to a single node and topologically sort the resulting acyclic digraph. Finally, sort the LVBs according to that order by replacing each shrunk strongly connected component by some arbitrary order of its members. Note that such an order can be computed in time linear in the size of D, see, e.g., [54].

2.3.2 Efficient LVB Propagation

We are ready to propose a simple way to efficiently propagate generated LVBs. After every round of OBBT, store the latest computed LVB for each bound, update the LVB graph D and compute the partial order on D as described above. Whenever LVB propagation is called, for each (locally or globally) improved bound label its out-neighbors as *candidates*. If the global objective cut-off has been improved, add all bounds having an LVB with a nonzero coefficient $\tilde{\mu}$ on its right-hand side to the list of candidates. According to the partial order on D, for each candidate we propagate its associated LVB and, if successful, label all its out-neighbors as candidates and proceed with the next candidate with respect to the partial order.

In every LVB propagation call, this procedure is only performed once. As mentioned above, additional calls might result in further tightenings. However, this is a standard issue for all general propagation routines. For that reason, state-of-the-art solvers dynamically limit the effort spent on propagation and so we also propose to leave this decision to the solver framework itself.

3 Accelerating OBBT

Whereas Lagrangian variable bounds aim at leveraging the effect of OBBT, in this section we will discuss three ways to reduce the computational cost for applying OBBT in the first place. These are *selecting* promising types of variables for OBBT, *filtering* bounds a priori in which no tightening can occur, and *ordering* the LPs to be solved.

3.1 Selecting Variables

Because of its cost, it is a natural idea to apply OBBT not for all variables, but more selectively. Since the goal is the tightening of bounds that leads to tighter convex relaxations, it has been suggested by [3] and [62] to restrict OBBT to variables appearing in nonconvex terms, i.e., to candidates for spatial branching. We will refer to variables that appear in nonlinear and nonconvex nonlinear terms as “nonlinear” and “nonconvex” variables, respectively.

Binary variables should be excluded from OBBT because the effect of OBBT can be achieved more efficiently as follows: Tentatively fix the variable to zero and one and each time solve the resulting relaxation. The relaxation is infeasible on one side if and only if OBBT would fix the variable to the other side.

It has been observed, for instance, by [4], that this is computationally more efficient than minimizing and maximizing the binary variable. This is for two reasons: first, because it does not require the modification of the objective function and addition of the objective cut-off constraint; second, when using the simplex method to solve the LPs, then after fixing a variable the LP can be warm started by the dual simplex, while OBBT LPs are warm started by the primal simplex. Computationally, the primal simplex is often less efficient, see, e.g., [12]. Note that this tentative fixing to zero and one is equivalent to applying strong branching [5, 6], which has been found to be a crucial ingredient to state-of-the-art branching rules for mixed-integer linear programming. Strong branching is also used by most state-of-the-art MINLP solvers for branching on integer variables with non-integral value in the relaxation solution, see, e.g., [10, 39, 57].

Finally, for numerical reasons, convex nonlinearities can not always be enforced by separating outer approximation constraints; hence also variables that appear only in convex constraints can potentially be selected as branching candidates. Tighter bounds on branching candidates reduce the extent to which enumerative search must be performed. Hence, in the implementation presented in Section 4, we apply OBBT to all nonbinary nonlinear variables, also to those that appear only in convex nonlinear constraints.

3.2 Filtering Bounds

As shown above, LVBs are a dual certificate of the validity of tightened OBBT bounds. The primal counterpart is given by the simple observation that any feasible relaxation solution $x^* \in \mathscr{R} \cap \{x \mid c^\mathsf{T} x \leqslant U\}$, certifies that the bounds that are tight at this solution cannot be improved by OBBT. More generally, if $|x_i^* - \ell_i|$ or $|x_i^* - u_i| \leqslant \varepsilon$ for some $\varepsilon \geqslant 0$, then OBBT can strengthen the corresponding bound by at most ε.

This idea is exploited in the branch-and-contract algorithm [62], which inspects the optimal solution x^* of the convex relaxation at the current node of the branch-and-bound tree. OBBT is executed only for those lower and upper bounds for which $|x_i^* - \ell_i|$ and $|x_i^* - u_i|$, respectively, exceeds a specified fraction of the domain size of variable x_i. This reduces the number of expensive solutions of OBBT problems for which no or only a small improvement of the bound is possible.

We observe that it is not necessary to restrict this filtering of bounds to an optimal solution of the relaxation, but that any feasible solution of the relaxation can be used as long as it satisfies the objective cut-off. This includes, in particular, all primal solutions encountered during OBBT. These have the advantage that they are readily available at no additional cost. Beyond that, we can ask the question of how to generate solutions or a sequence of solutions that filter a large amount of bounds at a lower cost than applying OBBT to each of these bounds. To rephrase the task: compute points in $\mathscr{R} \cap \{x \in \mathbb{R}^n \mid c^\mathsf{T} x \leqslant U\}$ that are tight at or close to a large number of yet unfiltered bounds.

A natural method to achieve this is given by Algorithm 1, which we call *aggressive bound filtering*. It takes a general direction $v \in \mathbb{R}^n$ with positive objective coefficient for the variables that we want to become tight at their upper bound and negative ones for variables that we want to push towards their lower bound. The algorithm maximizes $v^\mathsf{T} x$ over the feasible region over which OBBT operates and filters all bounds at which the resulting solution is tight. Subsequently, positive objective coefficients are set to zero if the corresponding upper bound was filtered and negative objective coefficients are set to zero if the lower bound was filtered. Reoptimizing the filtering problem with reduced v may now give a point at which further bounds become tight and can be filtered. This procedure is iterated until all bounds are filtered, v is reduced to a unit direction—in which case the filtering problem would be identical to applying OBBT in this direction—, or until the success rate, i.e., the number of bounds filtered in the last iteration, drops below a specified threshold δ.

In order to ensure that the filtering problem is bounded, we force objective coefficients to be nonpositive for variables without upper bound and nonnegative for variables without lower bound. We assume that the relaxation is not empty because this procedure is applied after solving this relaxation for computing the dual bound. The resulting optimal solution can be used to compute the initial sets of unfiltered bounds that are required as input. The repeated solution of the filtering problem is particularly efficient if the relaxation is polyhedral and can be solved using the dual simplex algorithm with warm starts.

Remark 4 (Filtering Directions)**.** *Natural candidates for initial directions v are $v_i = 1$ for all $i \in \mathscr{U}_0$, zero otherwise, in order to filter upper bounds; and $v_i = -1$ for all $i \in \mathscr{L}_0$, zero otherwise, in order to filter lower bounds. Note that even if $v_i > 0$, the resulting filtering solution x^k may be tight at—and hence filter—the lower bound ℓ_i. One frequently encounters MINLP models that feature nonnegative variables only and allow the zero solution as feasible, at least for the relaxation. In this case, if the objective cut-off bound U is positive, Algorithm 1 started with the "-1" direction requires only one iteration to detect that no lower bounds can be strengthened.*

Algorithm 1: Aggressive Bound Filtering

in : variable bounds $[\ell, u]$, convex relaxation $\mathscr{R} \subseteq [\ell, u]$, objective function vector c, objective limit U, unfiltered bound indices $\mathscr{L}_0, \mathscr{U}_0 \subseteq \{1, \ldots, n\}$, initial direction $v \in \mathbb{R}^n$, termination threshold $\delta \in \mathbb{N}$

out : reduced index sets $\tilde{\mathscr{L}} \subseteq \mathscr{L}_0$, $\tilde{\mathscr{U}} \subseteq \mathscr{U}_0$

1 **begin**

2 $\quad \mathscr{L}_1 \leftarrow \{i \in \mathscr{L}_0 \mid \ell_i \neq -\infty\}$ `/* 0. exclude unbounded directions */`

3 $\quad \mathscr{U}_1 \leftarrow \{i \in \mathscr{U}_0 \mid u_i \neq \infty\}$

4 $\quad$ **for** $k \leftarrow 1, 2, \ldots$ **do** `/* filtering loop */`

$\quad\quad$ `/* 1. reduce v to unfilt. directions */`

5 $\quad\quad$ **foreach** $i \in \mathscr{L}_{k-1} \setminus \mathscr{L}_k$, $v_i < 0$ **do** $v_i \leftarrow 0$

6 $\quad\quad$ **foreach** $i \in \mathscr{U}_{k-1} \setminus \mathscr{U}_k$, $v_i > 0$ **do** $v_i \leftarrow 0$

7 $\quad\quad$ **if** $\mathscr{L}_k \cup \mathscr{U}_k = \emptyset$ or $v = 0$ **then** `/* 2. check termination */`

8 $\quad\quad\quad$ break

9 $\quad\quad$ **else if** $k \geqslant 2$ and $|\mathscr{L}_{k-1} \setminus \mathscr{L}_k| + |\mathscr{U}_{k-1} \setminus \mathscr{U}_k| \leqslant \delta$ **then**

10 $\quad\quad\quad$ break

11 $\quad\quad$ $x^k = \arg\max\{v^\mathsf{T} x \mid x \in \mathscr{R}, c^\mathsf{T} x \leqslant U\}$ `/* 3. minimize v */`

12 $\quad\quad$ $\mathscr{L}_k \leftarrow \{i \in \mathscr{L}_{k-1} \mid x^k_i \neq \ell_i\}$ `/* 4a. filter lower bounds */`

13 $\quad\quad$ $\mathscr{U}_k \leftarrow \{i \in \mathscr{U}_{k-1} \mid x^k_i \neq u_i\}$ `/* 4a. filter upper bounds */`

14 $\quad$ **return** $\tilde{\mathscr{L}} \leftarrow \mathscr{L}_k$, $\tilde{\mathscr{U}} \leftarrow \mathscr{U}_k$

To summarize, we have identified three different types of solutions that can be used to filter bounds for which applying OBBT is fruitless. In order of increasing computational cost, these are

- optimal solutions of the convex relaxation computed for bounding,
- the solutions encountered during OBBT, and
- the solutions produced by aggressive bound filtering.

We will refer to the first two as *trivial* filtering. If tightness is tested exactly as in lines 12 and 13 of Algorithm 1, then the filtering is exact in the sense that only bounds are excluded for which OBBT is guaranteed to find no tightening. This is easily replaced by a test using an absolute or relative threshold. A practical implementation based on floating-point arithmetic needs to use a small nonzero tolerance, even when testing for exact equality.

Remark 5 (Filtering vs. LVB Generation)**.** *As noted in Section 2.1, nontrivial LVBs can be learned also from dual OBBT solutions that do not tighten the current bound. In this sense, the filtering of such bounds might reduce the number of generated LVBs. In order to counteract this effect, our implementation uses the following technique that is specific to the simplex method.*

Suppose x^* is a solution that we want to use for filtering tight bounds, which might be the initial relaxation solution, a solution of an OBBT LP, or the solution computed during aggressive filter in line 11 of Algorithm 1. Suppose further that $x^*_i = \ell_i$, but due to primal degeneracy variable x_i is basic in the simplex solution. Then we know that its lower bound is not part of the proof of optimality in the current simplex basis. In this case, there is the chance of generating a nontrivial LVB by using the primal simplex to minimize x_i starting from the current basis. We perform this step because it should take only few simplex iterations; afterwards we can mark the bound as processed. Note that the LP solution may change in the course of these calls to the simplex solver if pivots occur due to floating-point numerics.

By contrast, if x^*_i is nonbasic in the simplex solution then we know that the simplex solver will (in all likelihood) return the trivial dual solution that only involves this bound, leading to no useful LVB. Hence we filter nonbasic variables immediately without solving the above LP.

3.3 Ordering LP Solves

By construction, the optimal solutions computed during OBBT are often widely scattered across the feasible region. When using a polyhedral relaxation, this fact challenges the warm-starting capabilities of the simplex algorithm, which is the algorithm of choice for solving relaxations in most spatial branch-and-bound implementations. In the following, we

devise schemes for the order in which OBBT explores the bounds such as to heuristically minimize the time and iterations spent by a simplex-based LP solver.

Since a practical implementation will employ working limits on the number of LP iterations spent by OBBT, ideally, a good ordering strategy does not only minimize the total number of LP iterations, but also keeps low the number of LP iterations that are spent between any two successive candidates. An intuitive strategy is the following greedy heuristic: select the next bound candidate by considering the distance between the value of a variable in the current LP solution and the bound of the variable that we want to tighten. This is formalized in Algorithm 2, which executes a complete round of OBBT in this greedy order, interleaved with filtering steps. Note that if OBBT tightens the bound of an integer variable to a non-integral value, the bound may be tightened further by rounding it, see Lines 13 and 18 of Algorithm 2. As argued in Remark 5, LVBs are not only generated in Line 19, but also in Lines 4 and 20.

Algorithm 2: OBBT with Filtering and Greedy Ordering

in : variable bounds $[\ell, u]$, convex relaxation $\mathscr{R} \subseteq [\ell, u]$, objective function vector c, objective limit U, bound indices $\mathscr{L}, \mathscr{U} \subseteq \{1, \ldots, n\}$, index set of integer variables $\mathscr{I}$, initial relaxation solution x^*

out : tightened variable bounds $[\ell', u'] \subseteq [\ell, u]$

1 **begin**
2 $[\ell', u'] \leftarrow [\ell, u]$
3 reduce $(\mathscr{L}, \mathscr{U})$ by trivial filtering on x^*
4 reduce $(\mathscr{L}, \mathscr{U})$ by aggressive bound filtering (optional)
5 $\tilde{x} \leftarrow$ last LP solution
6 **while** $\mathscr{L} \cup \mathscr{U} \neq \emptyset$ **do** /* 1. select bound */
7 $val_{\mathscr{L}} \leftarrow \min\{\tilde{x}_k - \ell_k \mid k \in \mathscr{L}\}$ /* ∞ if $\mathscr{L} = \emptyset$ */
8 $val_{\mathscr{U}} \leftarrow \min\{u_k - \tilde{x}_k \mid k \in \mathscr{U}\}$ /* ∞ if $\mathscr{U} = \emptyset$ */
9 **if** $val_{\mathscr{L}} \leq val_{\mathscr{U}}$ **then**
10 $i \leftarrow \arg\min\{\tilde{x}_k - \ell_k \mid k \in \mathscr{L}\}$
11 $\mathscr{L} \leftarrow \mathscr{L} \setminus \{i\}$
12 $\tilde{x} \leftarrow \arg\min\{\tilde{x}_i \mid \tilde{x} \in \mathscr{R}, c^\mathsf{T} x \leq U\}$ /* 2a. minimize */
13 **if** $i \in \mathscr{I}$ **then** $\ell'_i \leftarrow \lceil \tilde{x}_i \rceil$ **else** $\ell'_i \leftarrow \tilde{x}_i$
14 **else**
15 $i \leftarrow \arg\min\{u_k - \tilde{x}_k \mid k \in \mathscr{U}\}$
16 $\mathscr{U} \leftarrow \mathscr{U} \setminus \{i\}$
17 $\tilde{x} \leftarrow \arg\max\{\tilde{x}_i \mid \tilde{x} \in \mathscr{R}, c^\mathsf{T} x \leq U\}$ /* 2b. maximize */
18 **if** $i \in \mathscr{I}$ **then** $u'_i \leftarrow \lfloor \tilde{x}_i \rfloor$ **else** $u'_i \leftarrow \tilde{x}_i$
19 generate LVB, store if nontrivial
20 reduce $(\mathscr{L}, \mathscr{U})$ by trivial filtering on $\tilde{x}$ /* 3. filter */
21 **return** $[\ell', u']$

When OBBT is applied with working limits, e.g., on the number of simplex iterations, the order in which OBBT LPs are processed becomes important. There are two criteria for an efficient ordering: finding as many reductions as possible within the given limits and finding "important" reductions, which are often those with a large ratio between the original and the reduced domain. These criteria might be mutually exclusive.

In Lines 7 and 8, we use a greedy strategy for ordering the OBBT LPs. This is motivated by the expectation that we need few simplex iterations when a variable's relaxation value is close to the bound that should be tightened. One disadvantage may be, though, that it explores those variables first which are closest to their bound and therefore show the smallest potential for a tightening. Therefore just the opposite strategy could be equally beneficial: a *reverse greedy* ordering that always chooses variables with largest distance between their value in the current LP solution and the bound of interest. This might lead to more bound tightenings among the early LPs solved; at the same time, these LPs may be more expensive to solve. A different heuristic, which we called *min–max*, first solves all minimization LPs, then all maximization LPs. This is motivated by the conjecture that the solutions to the minimization LPs are close to each other and that processing them sequentially may (indirectly) exploit the simplex algorithm's warm starting capabilities; likewise for the solutions to the maximization LPs. Finally, as a control setting, the experiments in Section 4 also used a trivial *random* choice strategy as a baseline that any more sophisticated strategy should outperform.

Remark 6 (Minimum Spanning Tree Orders)**.** *During our experiments we obtained the following "negative" result. If the simplex basis at an optimal OBBT solution can be stored and restored, this allows for the design of more sophisticated ordering algorithms since one is not bound to warm start the LP solves from the last solution. We have experimented with a heuristic similar to Prim's algorithm [45] for computing a minimum spanning tree in a graph. To this end, consider the bounds of the variables as the nodes of a complete graph. The number of LP iterations spent between the LP solves associated with two bounds gives weights on the edges; these weights are, in general, not well-defined and a priori unknown, but the distances used in the above greedy ordering can serve as a simple proxy. If we assume symmetric weights then, at least in theory, a minimum-weight spanning tree of the undirected graph provides an order of LP solves that minimizes the total number of LP iterations. In more detail, if during the algorithm of Prim we add an edge* $\{v,w\}$ *to the current tree* T*, where* $v \in T$ *and* $w \notin T$*, we load the LP basis stored at* v *and solve the corresponding OBBT LP for* w*. In our experiments, however, this was not efficient due to the large overhead for storing and restoring the basis information, which additionally invalidates the underlying factorization of the simplex solver.*

4 Computational Experiments

In this section, we investigate the computational impact of OBBT with the enhancements described in Section 3 and the effectiveness of the LVB propagation introduced in Section 2. We implemented these techniques as part of the MINLP solver SCIP [51]. For an overview of SCIP's general solving algorithm and MINLP features we refer to [1, 57].

4.1 Implementation

We extended SCIP by two separate propagator plug-ins for OBBT and LVB propagation. Since the validity of OBBT is based on dual feasibility, we chose a dual feasibility tolerance of 10^{-9} for the OBBT LP solves (the SCIP default is 10^{-7}).

Because of its potentially high computational cost, we gave OBBT the lowest priority and executed it only when no other propagators found problem reductions. By default, it is only called at the root node and the number of LP iterations spent by OBBT during one call is limited to ten times the LP iterations spent so far at the root node. Note that SCIP may call OBBT repeatedly at one node. In this case we continue with the variable directions not yet processed at this node; each variable direction is tested at most once per node. The LP optimum is used for the initial filtering, followed by aggressive bound filtering, and the filtering via OBBT solutions.

From each successfully solved OBBT LP at the root node, globally valid Lagrangian variable bounds are transferred to the LVB propagator and propagated at each node as described in Section 2.3. In order to avoid frequent resorting triggered by the addition of new LVBs, we sort only after the root node processing. Before that, we propagate without making use of the graph structure.

4.2 Experimental Setup

We conducted two main experiments. For the first experiment, we processed only the root node of SCIP. For the second one, we analyzed the overall performance impact on the complete tree search.

The goal of the root node experiment was to answer the following questions:

How many nontrivial LVBs can be generated during OBBT? As explained in Remark 2, LVBs may be trivial and not contain information useful for bound tightening. Furthermore, what is the effect of filtering? And what is the effect of different ordering strategies on the performance of OBBT?

In our root node experiment, we chose a "pure" OBBT setting: perform OBBT once at the root node, solving one OBBT LP for the bounds of each unfixed, nonbinary, nonlinear variable as explained in Section 3.1. In order to ensure comparability, we did not apply any working limits on the number of LPs or LP iterations spent in the root node experiments. We compared the ordering strategies described in Section 3.3 (greedy, reverse greedy, min–max, and random) without aggressive bound filtering. We analyzed the impact of filtering on the number of bound tightenings and LVBs found by OBBT using the following settings:

- no filtering,
- trivial filtering after each solved OBBT LP, and
- trivial plus aggressive filtering.

The goal of the tree experiment was to analyze the impact of OBBT alone and of OBBT enhanced by LVB propagation on the overall performance of SCIP. To this end, we measured the running time and the number of branch-and-bound nodes taken by SCIP when solving to ε-global optimality[2] for the gap limit $\varepsilon = 10^{-4}$. In this second experiment, we compared

- SCIP without OBBT,
- SCIP with OBBT at the root node, with working limits as described in Section 4.1, using greedy ordering and aggressive filtering, and
- SCIP with OBBT as above and LVB propagation during the tree search.

We imposed a time limit of one hour per instance for both the root node and the tree experiment.

We also conducted preliminary experiments applying OBBT within the tree, as opposed to calling it only once at the root and exploiting its deductions during tree search. While this can be beneficial on some instances, a simple execution of OBBT at each node of the tree easily incurs a large overhead in running time on average. The main research question that we address in this paper is not how to tune OBBT for application at local nodes of the tree, but how to efficiently use it for the deduction of global bounds and how to generate and harness LVBs as an approximation of iterated OBBT during the entire solving process. It turns out that this already requires a careful balancing of the effort that OBBT spends a priori versus the savings gained later through smaller search trees. Achieving similar results for local OBBT constitutes a far more difficult challenge. This being said, the ordering and filtering strategies found superior in our root node experiments may be a good starting point for investigating how to perform OBBT locally in the tree.

Test Sets. We used the publicly available instances of the MINLPLib2 [37], which at time of the experiments contained 1,357 instances. This includes amongst others instances from the first MINLPLib, the nonlinear programming library GLOBALLib, and the recent CMU-IBM initiative minlp.org [16]. We selected the instances that were available in OSiL format and consisted of nonlinear expressions that could be handled by SCIP: 1,275 instances. Of these, 36 instances were solved by SCIP's presolving or were linear after presolving; we removed them because they are not relevant for our experiments. For 287 instances OBBT was never called because SCIP solved them to proven optimality or hit the time limit before; we excluded those instances because they are not relevant to our experiments. For a more detailed analysis, we subdivided the remaining 952 instances into the 605 MINLPs which contained integer variables after presolving (test set INT) and the 347 instances that were NLPs after presolving (test set GO, for *g*lobal *o*ptimization).

Note that OBBT terminates if the LP solver fails to solve an OBBT LP, e.g., due to numerical difficulties. Such an early termination for one of the settings would affect the observed success of this setting and compromise the comparability of the results. Hence, only for the root node experiments, we removed instances for which an LP error occurred for at least one of the settings. This was the case for 28 of the INT instances and for 5 of the GO instances. For the tree experiments we did not exclude instances because of LP errors in order to measure the performance of the actual OBBT implementation used by the solver SCIP as accurately as possible.

Hardware and Software. The experiments were conducted on a cluster of 64bit Intel Xeon X5672 CPUs at 3.2 GHz with 12 MB cache and 48 GB main memory. In order to safeguard against a potential mutual slowdown of parallel processes, we ran only one job per node at a time. We used SCIP developer version 3.1.0.1 with SoPlex 2.0.1 as LP solver [61, 53], CppAD 20140000.1 [18], and Ipopt 3.12.4 as NLP solver [58, 19].

Averages and Statistical Tests. In order to evaluate algorithmic performance over a large set of benchmark instances, we compare geometric means, which provide a measure for relative differences. This avoids results being dominated by outliers with large absolute values as is the case for the arithmetic mean. In order to also avoid an over-representation of differences among very small values, we use the shifted geometric mean. The *shifted geometric mean* of values $v_1, \ldots, v_N \geqslant 0$ with shift $s \geqslant 0$ is defined as

$$\left(\prod_{i=1}^{N} (v_i + s) \right)^{1/N} - s. \tag{20}$$

[2]Let U and L be the global upper and lower bounds, respectively. Then SCIP terminates if either $U - L \leqslant \varepsilon$ (absolute gap limit reached) or U and L have same sign, $|U|, |L| > 10^{-9}$, and $|U - L| / \min\{|U|, |L|\} \leqslant \varepsilon$ (relative gap limit reached).

See also the discussion in [1, 2, 28]. As shift values we used 10 for averaging over the number of LP solves, 100 for averaging over the number LP iterations, 10 seconds for averaging over the running time, and 100 nodes for averaging over the size of search trees.

However, comparing performance of different algorithms only based on the mean of certain performance measures may be misleading: it does not allow us to draw conclusions on how a change in the mean value is distributed over the test set. This can be quantified by using statistical hypothesis testing. We use the nonparametric *Wilcoxon signed rank test* [59] in the fashion of [28, Sec. 3.2] and [11, Sec. 11.1] in order to test whether changes we observe in the shifted geometric mean are significant in this sense. We use a one-sided test with a p-value of 5%.

4.3 Computational Results

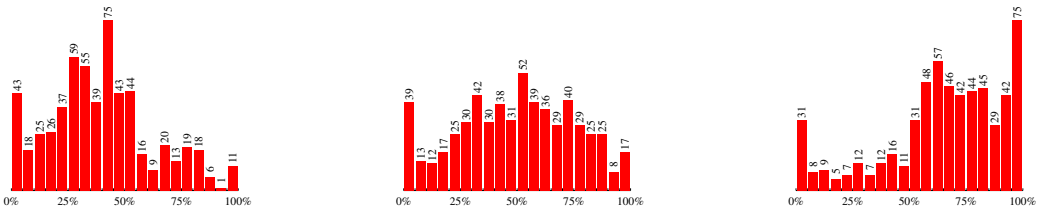
4.3.1 Root Node Experiments

For each of the root node experiments, detailed instance-wise results can be found in Tables 3, 4, 5, and 6 in the appendix. In the following we discuss these results in an aggregated fashion. First, we analyze how frequently LVBs are generated and how LVB generation interacts with different filtering strategies. Second, we compare various ordering heuristics w.r.t. the number of bound reductions and LVBs they find when given working limits on the overall number of LP iterations.

Generated LVBs. We counted the number of nontrivial LVBs that were generated per number of OBBT LPs solved. This gives a *success rate* for the generation of nonlinear LVBs; the success rate lies between 0% and 100% for each instance. Figure 3 plots a histogram of these success rates on test set INT when using greedy ordering and the three different filtering strategies; Figure 4 plots the same for the NLP test set GO. In Figure 3a, for example, we can see that for 11 instances it was possible to generate an LVB from 95–100% of the LPs solved.

Our first observation is that both trivial and aggressive filtering each increase the success rates significantly: visually, the "mass" of the histogram moves towards higher success rates as we go from Figure 3a through 3b to 3c; likewise for Figure 4. This shows that we filter primarily those OBBT LPs from which no LVB could have been generated, so filtering seems to affect the generation of LVBs only slightly. This is a positive result: the increase in efficiency of OBBT does not compromise the effective generation of LVBs. To highlight specific numbers, with aggressive filtering, for more than 75% of the instances the LVB success rate was 50% or higher, i.e., on these instances, more than half of the OBBT LPs led to a nontrivial LVB.

For the GO instances, there is a much larger group on which almost no LVBs could be generated. Even with aggressive filtering, 59 out of 342 instances are in the 0–5% bin; for 48 of these, no LVB could be generated at all. Except for this difference, the distribution of success rates and the effect of filtering is very similar.

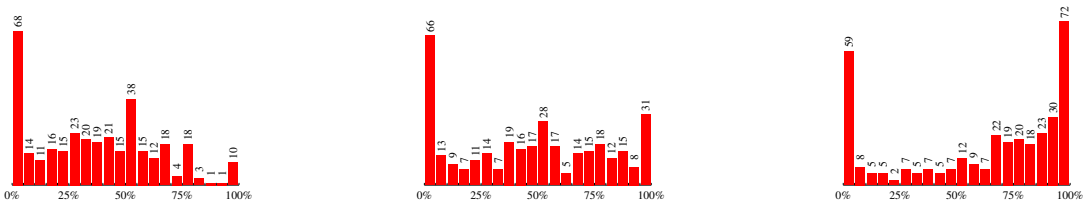


(a) no filtering (b) trivial filtering (c) trivial + aggressive filtering

Figure 3: Histograms of LVB success rates for test set INT

Overall Performance of Ordering and Filtering Strategies. Table 1 shows average performance measures on six different OBBT strategies. The base setting is the greedy ordering with trivial filtering. In the first three strategies, we changed the ordering heuristic to one of greedy reverse, min–max, and random. Additionally, we tested one setting with no filtering and one setting with aggressive bound filtering activated. We compare the number of LPs solved, the number of LP iterations spent, the number of bounds tightened, and the number of LVBs generated.

As a first observation, the average number of bounds that could be tightened (column "b_{obbt}") is almost identical for all strategies. This is expected since we processed the same variables for all strategies. However, the limited precision of floating-point operations can potentially lead to different outcomes, as can be seen, e.g., at the average value 13.2 for "greedy (base)" versus the value 13.1 for the other strategies on the INT test set. The number of LVBs generated is similar



(a) no filtering (b) trivial filtering (c) trivial + aggressive filtering

Figure 4: Histograms of LVB success rates for test set GO

over different orders. However, in column "lvb" it can be noted that turning filtering off causes 3.1% (39.4 vs. 38.2) and 4.0% (18.1 vs. 17.4) more LVBs to be generated on the INT and GO test set, respectively. This can be explained by the fact that we filter bounds for which, at the current simplex basis, no nontrivial LVB can be generated because the corresponding variable is nonbasic, see also Remark 5. Without filtering, these bounds may become basic at a later stage and admit a nontrivial LVB.

In general, the number of bound tightenings found is rather low: only about 15.6% on the INT (13.2 from 84.5) and 13.0% on the GO test set (7.6 from 58.4) of the OBBT LPs solved by the base setting led to a tighter bound. However, we can see that many more LVBs could be generated: 2.9 times as many LVBs as tightened bounds on the INT test set, and 2.3 times as many on the GO test set. This shows that in many cases valuable information in the form of nontrivial LVBs can be learned even if OBBT does not yield tighter bounds.

To evaluate the efficiency of the different strategies, we first look at the number of LP iterations. Although the greedy strategy had to solve more OBBT LPs than the other ordering heuristics, it did so in significantly fewer iterations: between 14.7% and 26.6% fewer than for reverse greedy, random, and min–max. Not surprisingly, reverse greedy even took more iterations than the random order. The min–max order was better than the random order, but could not outperform the greedy order. Also w.r.t. time, the greedy strategy is superior to the other ordering strategies.

Aggressive filtering increases the number of LP iterations in total, but decreases the number of LPs solved. Because each LP solve incurs a fixed setup time, OBBT with aggressive filtering is overall faster than OBBT with trivial filtering only: in combination with the greedy strategy it leads to a 16% speedup of OBBT on the GO test set; for the INT test it is performance-neutral. For single instances we observe tremendous improvements. For `elec200` of the GO test set, aggressive filtering reduces the time for OBBT from 714.3s to 62.6s and the LP iterations from 143,948 to 38,811. For `space960` from the INT test set, it reduces the time spent for OBBT from 502.1s to 89.7s and the LP iterations from 804,391 to 120,398. Therefore, we applied aggressive filtering in the tree experiments of Section 4.3.2.

Moreover, we compared the relative domain reductions found or implied by the OBBT or LVB propagator in the root node between the different ordering strategies. The relative reduction for all different strategies are almost the same. For the INT test set all ordering strategies reduced the volume of the domain by 19% compared to 15% reduction without using OBBT. Also, for the GO test set there are almost no differences in the volume reduction between the different ordering strategies. All strategies reduce the volume by 25% compared to 13% if we are not using OBBT.

We conclude that the greedy strategy is most promising when OBBT shall be performed for all variables. The next paragraph considers the case that we conduct only a partial execution of OBBT.

Bound Tightening Profiles. The following analysis is relevant to the tree experiments presented later. An efficient implementation as part of an MINLP solver will impose working limits, e.g., on the number of LP iterations, in order to control the effort spent on OBBT. Thus, the above comparison based on a full execution of OBBT does not show the full picture: we do not so much want to find the strategy that is fastest at processing all bounds, but rather the strategy that succeeds most often with limited computational effort. This of course depends on the chosen limits; ideally, a solver uses dynamic limits. Therefore, we performed a slightly involved graphical analysis that is inspired by performance profiles as used for benchmarking optimization software [21]. The result, comparing the ordering strategies from Table 1, can be seen in Figure 5.

These profiles were generated as follows: For one run of OBBT on one instance, we recorded the number of bound tightenings found within the first n iterations vs. n, resulting in an increasing step function. This gives one profile per instance per strategy. For each instance, we normalized its profiles on both axes, dividing the number of iterations by the maximum number of iterations over all strategies compared, and likewise dividing the number of bound tightenings by the maximum number of bound tightenings over all strategies compared. The scaled profiles are between zero and

Table 1: Aggregated results comparing six OBBT strategies at the root node

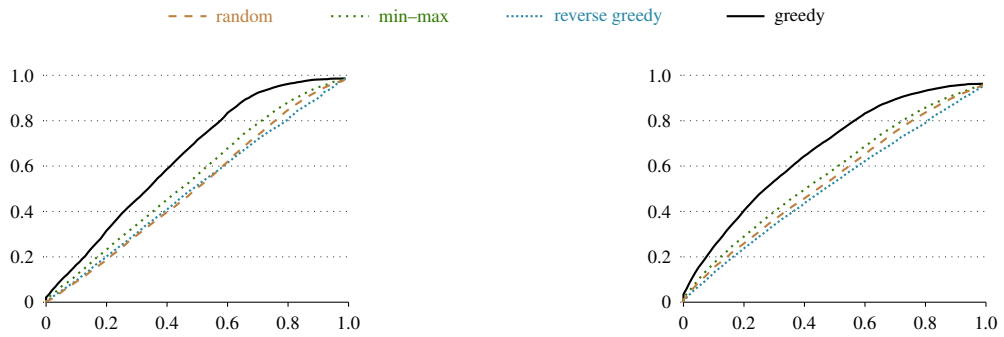
lp — number of LPs solved during OBBT (shifted geo. mean)
lp_{filt} — number of LPs solved during filtering (not incl. in lp, shifted geo. mean)
iter — number of LP iterations during OBBT (shifted geo. mean)
$\text{iter}_{\text{filt}}$ — number of LP iterations during filtering (not incl. in iter, shifted geo. mean)
b_{obbt} — number of bounds tightened by OBBT (shifted geo. mean)
t_{obbt} — time used by OBBT (shifted geo. mean)
lvb — number of LVBs found by OBBT (shifted geo. mean)

Test set	strategy	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	t_{obbt}	lvb
INT	random	77.3	15.6	1830.0	45.1	13.1	0.93	37.9
	min–max	77.7	15.5	1765.2	44.6	13.1	0.89	37.8
	reverse greedy	76.0	16.8	1847.5	47.6	13.1	0.92	37.8
	greedy (base)	84.5	10.6	1478.1	32.5	13.2	0.79	38.2
	greedy+filt. off	109.9	0.0	1505.1	0.0	13.1	0.79	39.4
	greedy+filt. aggr.	63.1	27.5	982.9	701.8	13.1	0.79	38.7
GO	random	55.1	9.5	429.6	22.5	7.6	0.42	17.7
	min–max	55.5	9.7	401.6	24.2	7.6	0.39	17.7
	reverse greedy	54.3	10.2	443.7	22.6	7.6	0.42	17.7
	greedy (base)	58.4	6.8	351.7	17.8	7.6	0.37	17.4
	greedy+filt. off	74.8	0.0	358.7	0.0	7.6	0.40	18.1
	greedy+filt. aggr.	40.2	18.2	212.4	204.6	7.6	0.31	18.3

one on both axes and can be averaged pointwise over all instances. This gives one profile per strategy as displayed in Figure 5. Loosely speaking, a point (x,y) in this profile means that after $x\%$ of the total LP iterations, $y\%$ of the total bound tightenings have been found (on average). Profiles that are higher and further left indicate better performance: more bound tightenings and more LVBs are found earlier. If the profile of strategy A is consistently above the profile of strategy B, we know that A outperforms B *independently* of how we choose working limits on the number of LP iterations: at whichever iteration we stop OBBT, we know that strategy A will (on average) have found more bound tightenings and LVBs than strategy B.

Only for plotting these profiles, we excluded instances where all strategies took less than 1,000 LP iterations or achieved less than ten bound tightenings. The former criterion handles instances for which OBBT is trivial and different orders have hardly any effect. The latter criterion excludes instances for which OBBT, and hence different strategies for OBBT, have limited impact on the overall performance. We generated profiles both for the number of bounds tightened and the number of LVBs generated; for the latter, the same procedure can be applied. We did not distinguish between the two test sets INT and GO, because their profiles look almost identical.

As can be seen in Figure 5, all ordering heuristics reach the same level of success eventually. The greedy order, however, significantly outperforms the others. The min–max strategy performs slightly better than the random order.



(a) Tightened bounds vs. LP iterations

(b) Generated LVBs vs. LP iterations

Figure 5: Performance curves for different ordering strategies

To summarize the results from the root node experiments, we saw that the greedy ordering heuristic outperformed the other orderings in our experiments. Furthermore, the overhead of aggressive bound filtering in terms of LP iterations was compensated by the fewer numbers of LPs being solved, leading to faster overall running times of OBBT. Finally, a greedy ordering strategy is expected to have a higher success rate when operated with limits on LP iterations. Thus, we made greedy ordering and aggressive filtering the default strategies for our tree experiment. We close this section with a remark on the relationship between greedy ordering and trivial filtering.

Remark 7 (Greedy Filtering)**.** *If some of the variables in the current LP solution are tight at a yet unprocessed bound (and filtering is turned off), then the greedy strategy would select one of these tight bounds for the next OBBT LP. Hence, it "filters" this bound automatically by explicitly solving the LP, usually without a single pivot when warm starting the primal simplex method. As explained in Remark 5, we already partially solve these LPs even when filtering is turned on in order to generate LVBs. In this sense, a plain greedy ordering makes trivial filtering almost superfluous, with the exception of variables with nonbasic status that should be skipped.*

4.3.2 Tree Experiments

Having arrived at efficient default settings for OBBT, we now turn to the results for comparing the overall performance of SCIP without OBBT, SCIP with OBBT only, and SCIP with OBBT and LVB propagation. Tables 7 and 8 in the appendix give detailed results for each instance of the INT and GO test set, respectively. We do not compare instances that could not be solved by any of the settings within the time limit (226 on INT and 160 on GO). Furthermore, we exclude 38 INT instances and 21 GO instances where one of the settings aborted due to insufficient memory, returned a solution that violated feasibility by more than 10^{-6}, or returned primal or dual bounds that were inconsistent with the values stated on the MINLPLib2 webpage. None of the settings ran into these cases significantly more often than the others.

This leaves 341 instances of the INT test set and 166 instances of the GO test set. For each of these test sets, Table 2 gives aggregated results over several subgroups. The group "all solved" contains the instances that could be solved by all settings within the time limit. Additionally, we analyze the instances which could be solved by all, but for which at least one setting took at least 1, 10, and 1000 seconds, respectively ("$\geqslant 1, 10, 100s$"). These groups are contained in "all solved" and form a hierarchy of harder and harder instances which are defined in an unbiased way by excluding instances that are easy to *all* settings; see, e.g., [2] for this benchmarking methodology. To be able to condense the presentation into one table, we state running time and number of branch-and-bound nodes on top of each other in one column. The key observations are:

- On all subgroups, SCIP+OBBT and SCIP+OBBT+LVB reduce the average number of branch-and bound nodes: by at least 6.2% (SCIP+OBBT on test set GO, group "all solved"), up to 41.9% (SCIP+OBBT+LVB on test set INT, group "$\geqslant 100s$").
- For the two easier INT subgroups "$\geqslant 1s$" and "$\geqslant 10s$" SCIP+OBBT alone gives a small slowdown, but the combination SCIP+OBBT+LVB is slightly faster than plain SCIP.
- On the harder INT subgroup "$\geqslant 100s$", SCIP+OBBT and SCIP+OBBT+LVB reduce the average running time by 5.7% and 17.1%, respectively.
- On the GO test set, SCIP+OBBT did not change the average running time significantly (at most 1.4% slower or faster than plain SCIP).
- For the GO subgroups $\geqslant 1, 10s$ and $\geqslant 100s$, SCIP+OBBT+LVB results in a speedup of 9.8%, 14.1%, and 18.8%: the harder the instances are, the larger are the reductions in running time achieved.
- Since LVBs are applied during the entire solving process, LVBs perform significantly more (one to three orders of magnitude more) bound tightenings than OBBT at the root node.
- On all subgroups LVBs lead to an additional reduction of running time and tree size as compared to SCIP+OBBT.

Note that the above groups do not contain instances that could be solved with one of the settings, but not with all. There were 11 such "timeout" instances in the INT test set and 9 in the GO test set. Out of the 11 "timeout" INT instances, plain SCIP solved only 1, while SCIP+OBBT solved 6, and SCIP+OBBT+LVB solved even 10 instances. Out of the 9 "timeout" GO instances, plain SCIP solved only 3, while SCIP+OBBT and SCIP+OBBT+LVB each solved 7 instances. Comparing plain SCIP with SCIP+OBBT+LVB on both test sets, SCIP+OBBT+LVB solved all but 3 instances which were solved by plain SCIP, and most importantly solved 16 additional instances for which plain SCIP timed out.

Table 2: Aggregated results comparing SCIP without OBBT, SCIP with OBBT only, and SCIP with OBBT and LVB propagation

N — number of instances
$\underline{t}\sqrt{n}$ — running time of SCIP in seconds (first line) and number of branch-and-bound nodes (second line, both shifted geo. mean)
% — relative time/nodes w.r.t. SCIP plain (in %)
b_{obbt} — number of bounds tightened by OBBT (shifted geo. mean)
b_{lvb} — number of bounds tightened by LVB propagation (shifted geo. mean)

			SCIP plain	SCIP+OBBT			SCIP+OBBT+LVB		
Test set	subgroup	N	$\underline{t}\sqrt{n}$	$\underline{t}\sqrt{n}$	%	b_{obbt}	$\underline{t}\sqrt{n}$	%	b_{lvb}
INT	all solved	330	20.7	21.1	101.6	12.0	20.3	97.7	130.4
			2084.4	1894.3	90.9		1742.3	83.6	
	$\geqslant 1s$	228	40.1	40.9	101.9	14.3	39.0	97.2	292.4
			7196.2	6226.8	86.5		5605.0	77.9	
	$\geqslant 10s$	142	102.1	104.8	102.6	17.1	98.4	96.4	659.4
			24245.4	20140.6	83.1		18363.7	75.7	
	$\geqslant 100s$	69	353.5	333.5	94.3	11.9	293.0	82.9	811.0
			119914.6	86908.9	72.5		69714.2	58.1	
GO	all solved	157	4.6	4.6	99.1	5.2	4.3	93.5	54.6
			451.4	423.5	93.8		412.0	91.3	
	$\geqslant 1s$	43	27.7	27.3	98.5	7.2	25.0	90.2	2087.5
			16819.0	14147.3	84.1		13560.1	80.6	
	$\geqslant 10s$	20	124.8	123.0	98.6	8.2	107.2	85.9	9012.2
			140227.9	123725.6	88.2		116555.9	83.1	
	$\geqslant 100s$	11	362.7	367.7	101.4	7.8	294.6	81.2	11004.5
			357429.9	306267.6	85.7		294799.8	82.5	

On the INT subgroups "all solved", "$\geqslant 1s$", and "$\geqslant 10s$" the reduction in the number of branch-and-bound nodes is consumed by the overhead in time spent on the solution of the auxiliary LPs. On the harder instances in the subgroup "$\geqslant 100s$", OBBT reduces running time by 5.7%. We believe that the slowdown on easier instances is an acceptable price to pay given the drastic improvements on harder instances and, even more importantly, the additional instances that can be solved with OBBT, but not with plain SCIP.

The number of LVB propagations grows with tree size on the subgroups of increasingly harder instances. This shows that LVB propagation is not only effective early, but throughout the solving process. However, it does not increase as drastically as tree size; this indicates that for most instances LVBs are most effective at upper and medium levels of the tree. LVBs are an approximation of OBBT generated from the LP relaxation at the root node; hence, it is intuitive that their effectiveness would decrease as the LP relaxation at local nodes of the search tree differs more and more from the root node relaxation.

Finally, the Wilcoxon signed rank test allows us to analyze how consistently the improvements seen in the shifted geometric means are distributed across the test set. First, we analyze the changes due to OBBT alone, i.e., the change in average running time between plain SCIP and SCIP+OBBT. The small slowdowns on the INT test set are all judged significant, while the 5.7% speedup on the INT subgroup "$\geqslant 100s$" is not. Though this may be surprising at first, it shows the importance of significance testing: the statistical test here reveals the fact that this average speedup mainly comes from 5 instances with high speedup factors of 90, 163, 310, 334, and 357.[3] None of the small changes on the GO test set are significant. Second, the improvements due to LVBs, i.e., the reductions in average running time between SCIP+OBBT and SCIP+OBBT+LVB, are all confirmed as significant except for the two GO subgroups "$\geqslant 10s$" and "$\geqslant 100s$". The failing of the Wilcoxon test on these two subgroups is most likely due to their small sample size of 20 and 11 instances, respectively.

Finally, we discuss some distinguished results for individual instances. To ensure that the observed effects are indeed due to OBBT and LVBs and not just random noise, we exploit the idea of performance variability (see, e.g., [31]). To this end, we have solved each of the instances ten times with different permutations of their constraints. In the following, we will present average running times, taken over the ten permuted runs.

First, consider instance `fin2bb`. OBBT causes a slowdown from 224.3 to 291.7 seconds in shifted geometric mean. However, by using LVBs we could reduce the run time to 132.9 seconds. This is an example where the immediate bound changes of OBBT at the root do not weigh out its computational overhead, but the implied bound changes by LVBs during tree search lead to a significant improvement. For the instance `rsyn0805h`, OBBT leads to a mean running time of only 0.8 seconds, while for all permutations SCIP times out with an average gap of 36.3% when not using OBBT. Applying LVB in addition does not have an impact on the running time. This is one of the instances for which OBBT makes the difference between extremely hard (for SCIP) and almost trivial to solve. Finally, instance `smallinvDAXr3b150-165` is one of the biggest success stories for both, plain OBBT and LVBs. By applying OBBT we could solve it in 106.4s instead of 1367.3s. This is a speedup factor of 12.8. By using LVBs we can increase the speedup factor to 213.6 and solve the instance in 6.4s in shifted geomatric mean.

Our computational results show that the impact of OBBT on the INT and GO instances lies not in delivering a consistent speedup over all instances, but in being a "game changer": using OBBT, significantly more instances can be solved and some hard instances are solved much faster.

5 Conclusion

In this paper, we have presented three enhancements to optimization-based bound tightening for MINLP and an extensive computational study of their performance impact over a large and heterogeneous test set. In reverse order of presentation these are: ordering strategies to reduce the number of LP iterations for a simplex-based implementation, filtering techniques to reduce the number of LPs solved, and an approximation of OBBT in form of a one-row relaxation, which can be propagated repeatedly and efficiently via FBBT, which we termed LVB propagation.

Our experiments on benchmark instances from MINLPLib2, using an implementation that is part of the MINLP solver SCIP, show that both OBBT and LVB propagation each improve the computational performance. We analyzed the results separately for instances with and without integer variables after SCIP's presolving. On those instances that could be solved with and without our OBBT algorithms within a time limit of one hour, our implementation reduced the average running time by 6.5% on the continuous test set and by 2.3% on the integer test set. On the subset of harder instances that took at least 100 seconds with one of the settings, the speedup was more pronounced: 17.1% and 18.8% on the continuous

[3] `smallinvDAXr2b100-110`, `smallinvDAXr4b150-165`, `smallinvDAXr1b100-110`, `rsyn0805h`, `nvs09`

and integer test set, respectively. We noticed that OBBT alone does not give a significant speedup on average, because the average reduction in the number of branch-and-bound nodes is mostly compensated by the overhead in time for solving LPs at the root node. LVB propagation, however, leverages the effect of OBBT and leads to a statistically significant reduction in average running time.

Most importantly, OBBT and LVB propagation lead to significantly more instances being solved on each test set. With OBBT and LVB propagation SCIP could solve 16 additional instances, while it exceeded the time limit only on three instances which plain SCIP could solve without OBBT.

Acknowledgements

We want to thank Pietro Belotti and Domenico Salvagnin for fruitful discussions and their valuable comments on earlier versions of this paper, as well as all SCIP developers, especially Tobias Achterberg for creating SCIP and Stefan Vigerske for providing the MINLP core of SCIP. Furthermore, we would like to acknowledge the support of Gregor Hendel in the evaluation of the computational experiments, for which we used his Ipet tool [28]. This work was supported by the Research Campus Modal "Mathematical Optimization and Data Analysis Laboratories" funded by the German Ministry of Education and Research.

References

[1] Achterberg, T.: Constraint integer programming. Ph.D. thesis, Technische Universität Berlin (2007). URN:nbn:de:kobv:83-opus-16117

[2] Achterberg, T., Wunderling, R.: Mixed integer programming: Analyzing 12 years of progress. In: M. Jünger, G. Reinelt (eds.) Facets of Combinatorial Optimization, pp. 449–481. Springer Berlin Heidelberg (2013). DOI:10.1007/978-3-642-38189-8_18

[3] Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs—II. Implementation and computational results. Computers & Chemical Engineering **22**(9), 1159–1179 (1998). DOI:10.1016/S0098-1354(98)00218-X

[4] Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of mixed-integer nonlinear problems. AIChE Journal **46**(9), 1769–1797 (2000). DOI:10.1002/aic.690460908

[5] Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: Finding cuts in the TSP (A preliminary report). Tech. Rep. 95-05, Center for Discrete Mathematics & Theoretical Computer Science (DIMACS) (1995)

[6] Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, USA (2007)

[7] Belotti, P.: Bound reduction using pairs of linear inequalities. Journal of Global Optimization **56**(3), 787–819 (2013). DOI:10.1007/s10898-012-9848-9

[8] Belotti, P., Cafieri, S., Lee, J., Liberti, L.: Feasibility-based bounds tightening via fixed points. In: W. Wu, O. Daescu (eds.) Combinatorial Optimization and Applications, *Lecture Notes in Computer Science*, vol. 6508, pp. 65–76. Springer Berlin Heidelberg (2010). DOI:10.1007/978-3-642-17458-2_7

[9] Belotti, P., Cafieri, S., Lee, J., Liberti, L.: On feasibility based bounds tightening. Tech. Rep. 3325, Optimization Online (2012). http://www.optimization-online.org/DB_HTML/2012/01/3325.html

[10] Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods & Software **24**, 597–634 (2009). DOI:10.1080/10556780903087124

[11] Berthold, T.: Heuristic algorithms in global MINLP solvers. Ph.D. thesis, Technische Universität Berlin (2014)

[12] Bixby, R.E.: Solving real-world linear programs: A decade and more of progress. Operations Research **50**(1), 3–15 (2002). DOI:10.1287/opre.50.1.3.17780

[13] Caprara, A., Fischetti, M.: $\{0, \frac{1}{2}\}$-Chvátal-Gomory cuts. Mathematical Programming **74**(3), 221–235 (1996). DOI:10.1007/BF02592196

[14] Caprara, A., Locatelli, M.: Global optimization problems and domain reduction strategies. Mathematical Programming **125**, 123–137 (2010). DOI:10.1007/s10107-008-0263-4

[15] Chvátal, V.: Edmonds polytopes and a hierarchy of combinatorial problems. Discrete Mathematics **4**(4), 305–337 (1973). DOI:10.1016/0012-365X(73)90167-2

[16] CMU-IBM Cyber-Infrastructure for MINLP. http://www.minlp.org/

[17] COIN-OR: Couenne, an exact solver for nonconvex MINLPs. http://www.coin-or.org/Couenne

[18] COIN-OR: CppAD, a package for differentiation of C++ algorithms. http://www.coin-or.org/CppAD

[19] COIN-OR: Ipopt, Interior point optimizer. http://www.coin-or.org/Ipopt

[20] Dantzig, G.B., Fulkerson, D.R., Johnson, S.M.: Solution of a large-scale traveling-salesman problem. Operations Research **2**, 393–410 (1954). DOI:10.1287/opre.2.4.393

[21] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**(2), 201–213 (2002). DOI:10.1007/s101070100263

[22] Fügenschuh, A., Martin, A.: Computational integer programming and cutting planes. In: K. Aardal, G.L. Nemhauser, R. Weismantel (eds.) Discrete Optimization, *Handbooks in Operations Research and Management Science*, vol. 12, pp. 69–121. Elsevier (2005). DOI:10.1016/S0927-0507(05)12002-7

[23] Gamrath, G., Koch, T., Martin, A., Miltenberger, M., Weninger, D.: Progress in presolving for mixed integer programming. Mathematical Programming Computation **7**(4), 367–398 (2015). DOI:10.1007/s12532-015-0083-5

[24] Gleixner, A., Vigerske, S.: Analyzing the computational impact of individual MINLP solver components. Talk at MINLP 2014, Carnegie Mellon University, Pittsburgh, PA, USA (2014). http://minlp.cheme.cmu.edu/2014/papers/gleixner.pdf

[25] Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Society **64**, 275–278 (1958)

[26] Gomory, R.E.: An algorithm for the mixed integer problem. Tech. Rep. P-1885, The RAND Corporation (1960)

[27] Grossmann, I.E., Sahinidis, N.V.: Special issue on mixed integer programming and its application to engineering, part I. Optimization and Engineering **3**(4) (2002)

[28] Hendel, G.: Empirical analysis of solving phases in mixed integer programming. Master's thesis, Technische Universität Berlin (2014). URN:nbn:de:0297-zib-54270

[29] Horst, R., Tuy, H.: Global optimization: Deterministic approaches. Springer (1996)

[30] Huang, W.: Operative planning of water supply networks by mixed integer nonlinear programming. Master's thesis, Freie Universität Berlin (2011)

[31] Koch, T., Achterberg, T., Andersen, E., Bastert, O., Berthold, T., Bixby, R.E., Danna, E., Gamrath, G., Gleixner, A.M., Heinz, S., Lodi, A., Mittelmann, H., Ralphs, T., Salvagnin, D., Steffy, D.E., Wolter, K.: MIPLIB 2010. Mathematical Programming Computation **3**(2), 103–163 (2011). DOI:10.1007/s12532-011-0025-9

[32] LaGO – Lagrangian Global Optimizer. https://projects.coin-or.org/LaGO

[33] Lodi, A., Nogales-Gómez, A., Belotti, P., Fischetti, M., Monaci, M., Salvagnin, D., Bonami, P.: Indicator constraints in mixed-integer programming. Talk at SCIP Workshop 2014, Zuse Institute Berlin, Germany (2014). http://scip.zib.de/workshop/scip_lodi.pdf

[34] Maranas, C.D., Floudas, C.A.: Global optimization in generalized geometric programming. Computers & Chemical Engineering **21**(4), 351–369 (1997). DOI:10.1016/S0098-1354(96)00282-7

[35] Marchand, H., Wolsey, L.A.: Aggregation and mixed integer rounding to solve MIPs. Operations Research **49**(3), 363–371 (2001). DOI:10.1287/opre.49.3.363.11211

[36] McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. Mathematical Programming B **10**(1), 147–175 (1976). DOI:10.1007/BF01580665

[37] MINLP library 2. http://gamsworld.org/minlp/minlplib2

[38] Misener, R., Floudas, C.A.: Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. Mathematical Programming **136**(1), 155–182 (2012). DOI:10.1007/s10107-012-0555-6

[39] Misener, R., Floudas, C.A.: GloMIQO: Global mixed-integer quadratic optimizer. Journal of Global Optimization **57**, 3–50 (2013). DOI:10.1007/s10898-012-9874-7

[40] Misener, R., Floudas, C.A.: ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. Journal of Global Optimization **59**(2-3), 503–526 (2014). DOI:10.1007/s10898-014-0166-2

[41] Nannicini, G., Belotti, P., Lee, J., Linderoth, J., Margot, F., Wächter, A.: A probing algorithm for MINLP with failure prediction by SVM. In: T. Achterberg, J.C. Beck (eds.) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, *Lecture Notes in Computer Science*, vol. 6697, pp. 154–169. Springer Berlin Heidelberg (2011). DOI:10.1007/978-3-642-21311-3_15

[42] Nemhauser, G.L., Wolsey, L.A.: A recursive procedure to generate all cuts for 0–1 mixed integer programs. Mathematical Programming **46**(1-3), 379–390 (1990). DOI:10.1007/BF01585752

[43] Neumaier, A.: Complete search in continuous global optimization and constraint satisfaction. Acta Numerica **13**, 271–369 (2004). DOI:10.1017/S0962492904000194

[44] Nowak, I., Vigerske, S.: LaGO: a (heuristic) branch and cut algorithm for nonconvex MINLPs. Central European Journal of Operations Research **16**(2), 127–138 (2008). DOI:10.1007/s10100-007-0051-x

[45] Prim, R.C.: Shortest connection networks and some generalizations. Bell System Technical Journal **36**(6), 1389–1401 (1957). DOI:10.1002/j.1538-7305.1957.tb01515.x

[46] Quesada, I., Grossmann, I.E.: Global optimization algorithm for heat exchanger networks. Industrial & engineering chemistry research **32**(3), 487–499 (1993). DOI:10.1021/ie00015a012

[47] Quesada, I., Grossmann, I.E.: A global optimization algorithm for linear fractional and bilinear programs. Journal of Global Optimization **6**, 39–76 (1995). DOI:10.1007/BF01106605

[48] Ryoo, H., Sahinidis, N.: Global optimization of nonconvex NLPs and MINLPs with applications in process design. Computers & Chemical Engineering **19**(5), 551–566 (1995). DOI:10.1016/0098-1354(94)00097-2

[49] Ryoo, H.S., Sahinidis, N.V.: A branch-and-reduce approach to global optimization. Journal of Global Optimization **8**(2), 107–138 (1996). DOI:10.1007/BF00138689

[50] Savelsbergh, M.W.: Preprocessing and probing techniques for mixed integer programming problems. ORSA Journal on Computing **6**(4), 445–454 (1994). DOI:10.1287/ijoc.6.4.445

[51] SCIP – Solving Constraint Integer Programs. http://scip.zib.de

[52] Smith, E.M., Pantelides, C.C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Computers & Chemical Engineering **23**, 457–478 (1999). DOI:10.1016/S0098-1354(98)00286-5

[53] SoPlex—the Sequential object-oriented simPlex. http://soplex.zib.de/

[54] Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing **1**(2), 146–160 (1972). DOI:10.1137/0201010

[55] Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming **99**, 563–591 (2004). DOI:10.1007/s10107-003-0467-6

[56] Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Mathematical Programming **103**(2), 225–249 (2005). DOI:10.1007/s10107-005-0581-8

[57] Vigerske, S.: Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming. Ph.D. thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II (2013). URN:nbn:de:kobv:11-100208240

[58] Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming **106**(1), 25–57 (2006). DOI:10.1007/s10107-004-0559-y

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

[60] Williams, H.P.: A reduction procedure for linear and integer programming models. In: Redundancy in Mathematical Programming, *Lecture Notes in Economics and Mathematical Systems*, vol. 206, pp. 87–107. Springer Berlin Heidelberg (1983). DOI:10.1007/978-3-642-45535-3_9

[61] Wunderling, R.: Paralleler und objektorientierter Simplex-Algorithmus. Ph.D. thesis, Technische Universität Berlin (1996). URN:nbn:de:0297-zib-5386

[62] Zamora, J.M., Grossmann, I.E.: A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. Journal of Global Optimization **14**, 217–249 (1999). DOI:10.1023/A:1008312714792

A Experimental Data and Results

A.1 Root Node Experiments

Table 3: Detailed results comparing filtering strategies at the root node on test set INT. See Table 1 for aggregated results.

lp — number of LPs solved during OBBT
lp_{filt} — number of LPs solved during OBBT's filtering
iter — number of LP iterations during OBBT
$\text{iter}_{\text{filt}}$ — number of LP iterations of OBBT's filtering LPs
b_{obbt} — number of bounds tightened by OBBT
lvb — number of LVBs found by OBBT

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
4stufen	54	9	649	3	8	31	84	0	771	0	8	31	43	21	1087	338	8	30
alan	6	0	33	0	5	5	6	0	33	0	5	5	5	2	43	14	5	5
batch0812_nc	76	9	1723	1	40	54	104	0	1716	0	40	60	65	25	2045	352	40	55
batch0812	51	2	672	0	34	44	68	0	679	0	34	52	46	17	720	177	34	45
batchdes	15	0	104	0	9	11	18	0	104	0	9	12	16	8	150	46	9	14
batch_nc	56	10	1947	3	20	36	66	0	1395	0	20	40	43	18	2051	528	20	40
batch	36	0	638	0	16	27	44	0	638	0	16	30	26	10	680	236	16	24
batchs101006m	86	1	1710	1	19	32	98	0	1662	0	19	33	45	28	1816	582	19	25
batchs121208m	105	0	7776	0	23	39	118	0	8006	0	23	40	88	31	10115	2558	23	48
batchs151208m	108	0	3504	0	23	48	124	0	3335	0	23	49	54	43	3514	1795	23	45
batchs201210m	119	2	4635	0	23	47	134	0	4774	0	23	53	64	44	6004	3075	23	52
bchoco05	84	6	1848	12	37	80	118	0	1884	0	37	82	85	27	1972	332	37	78
bchoco06	142	42	6622	90	6	115	192	0	6374	0	6	121	137	50	6798	569	6	112
bchoco08	747	172	117650	2584	60	590	1228	0	72343	0	12	341	755	161	63019	4448	13	254
blend029	36	5	469	4	2	22	52	0	437	0	2	22	24	13	436	289	2	18
blend146	108	13	3729	53	0	54	152	0	4199	0	0	48	69	38	2512	1427	0	43
blend480	142	26	5335	78	0	88	216	0	4954	0	0	88	111	53	4025	1578	0	86
blend531	118	15	2311	48	0	57	186	0	2848	0	0	65	80	42	2270	1115	0	65
blend718	90	6	2087	20	1	43	148	0	2091	0	1	48	66	39	2309	974	1	55
blend721	104	17	2903	95	0	58	152	0	2734	0	0	60	75	47	3178	1698	0	61
blend852	145	14	3981	55	0	72	216	0	4259	0	0	76	94	42	3042	1432	0	70
carton7	39	1	1644	2	4	16	46	0	1881	0	4	16	31	31	2616	2015	4	11
carton9	60	7	1945	19	11	21	62	0	2343	0	11	19	55	28	3739	2005	11	21
casctanks	452	53	22032	22	185	352	692	0	23294	0	185	360	413	118	26492	4349	185	359
cecil_13	108	2	3864	2	43	79	112	0	3864	0	43	79	78	29	6698	3381	43	78
chp_partload	1561	297	146966	137	399	995	1926	0	148278	0	399	1001	1475	378	137553	12581	399	1003
clay0203h	264	140	3895	100	0	235	288	0	4530	0	0	227	247	137	6162	3528	0	233
clay0203m	12	0	135	0	0	0	12	0	135	0	0	0	3	6	160	125	0	0
clay0204h	357	176	5427	121	0	321	384	0	5303	0	0	317	345	172	7806	3355	0	328
clay0204m	15	0	158	0	0	1	16	0	158	0	0	1	1	6	196	175	0	1
clay0205h	489	267	16457	162	0	430	624	0	7621	0	0	441	456	384	12136	7768	0	381
clay0205m	19	0	394	0	0	3	20	0	394	0	0	3	7	7	391	249	0	1
clay0303h	454	249	4629	16	0	332	656	0	4498	0	0	326	322	331	4191	4161	0	315
clay0303m	12	0	174	0	0	1	12	0	174	0	0	1	1	8	133	133	0	1
clay0304h	589	323	9242	82	0	425	880	0	6392	0	0	433	444	448	16901	16052	0	414
clay0304m	16	0	236	0	0	1	16	0	236	0	0	1	5	11	195	195	0	4
clay0305h	757	413	15207	269	0	558	1096	0	8800	0	0	562	550	529	26717	22846	0	532
clay0305m	20	1	288	1	0	0	20	0	288	0	0	0	2	9	280	280	0	1
crudeoil_lee1_05	91	34	3787	70	32	50	128	0	3830	0	33	52	86	49	4794	1581	34	52
crudeoil_lee1_06	117	48	8147	150	21	59	160	0	7376	0	22	60	104	61	8782	2291	21	64
crudeoil_lee1_07	135	50	10008	116	25	72	192	0	9692	0	25	74	116	66	11580	3242	25	78
crudeoil_lee1_08	176	69	12957	140	28	95	224	0	12791	0	30	94	150	83	15563	3383	32	96
crudeoil_lee1_09	192	83	18964	201	36	104	256	0	16841	0	36	107	173	97	18111	4484	35	100
crudeoil_lee1_10	195	64	18502	161	42	115	280	0	20539	0	44	119	187	105	26933	7243	43	117
crudeoil_lee2_05	158	48	20051	86	39	86	222	0	20192	0	39	92	139	74	26227	5963	39	87
crudeoil_lee2_06	212	64	30629	130	55	120	294	0	33766	0	55	115	189	89	44592	10164	55	119
crudeoil_lee2_07	246	81	46873	261	64	139	356	0	53951	0	64	150	208	85	59249	10089	64	142

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
crudeoil_lee2_08	304	122	62279	373	74	172	418	0	63449	0	74	178	256	113	74801	17040	74	174
crudeoil_lee2_09	342	117	99020	303	88	210	480	0	94021	0	88	202	298	132	115932	19841	88	202
crudeoil_lee2_10	378	123	110685	434	97	209	542	0	102334	0	97	236	341	151	108023	16769	97	229
crudeoil_lee3_05	289	91	18775	172	120	181	436	0	20743	0	120	200	268	121	23691	5248	120	192
crudeoil_lee3_06	385	125	61367	385	173	256	582	0	59409	0	173	278	386	172	67223	11979	173	257
crudeoil_lee3_07	473	146	65455	388	226	340	728	0	65310	0	226	327	464	180	67697	9114	226	335
crudeoil_lee3_08	556	157	154527	420	287	400	874	0	147336	0	287	403	548	192	155060	15434	287	406
crudeoil_lee3_09	653	177	199420	465	341	472	1020	0	192312	0	341	497	645	215	198085	12122	341	465
crudeoil_lee3_10	743	212	264413	897	397	550	1166	0	271953	0	397	577	721	233	258742	15557	397	537
crudeoil_lee4_05	194	49	22919	137	75	126	294	0	21797	0	75	138	176	73	32441	8285	75	134
crudeoil_lee4_06	247	55	54350	102	100	170	370	0	55073	0	100	176	233	83	57108	7687	100	165
crudeoil_lee4_07	311	85	74047	180	117	204	446	0	70032	0	117	199	275	102	92844	15771	117	201
crudeoil_lee4_08	351	82	95270	146	126	228	522	0	95480	0	126	239	299	92	100573	8046	126	231
crudeoil_lee4_09	417	102	132726	268	150	264	598	0	122505	0	150	267	360	121	142073	13099	150	272
crudeoil_lee4_10	430	107	157458	330	157	295	674	0	146931	0	157	298	382	122	146034	10315	157	289
crudeoil_li01	95	6	2553	8	36	32	120	0	2454	0	36	35	91	22	3669	1056	36	41
crudeoil_li02	32	6	1550	2	13	28	34	0	1550	0	13	29	30	19	2036	730	13	25
crudeoil_li03	254	31	12160	13	21	33	288	0	13074	0	21	27	175	81	19071	9094	21	27
crudeoil_li05	236	22	7467	229	13	22	288	0	7215	0	13	22	107	61	11176	6545	13	21
crudeoil_li06	255	25	11183	16	14	25	288	0	11466	0	14	25	176	67	14803	5176	14	24
crudeoil_li11	257	26	14927	16	19	21	288	0	14820	0	19	21	174	69	20897	7808	19	20
crudeoil_li21	261	31	31300	362	17	11	288	0	32063	0	17	14	188	63	29243	10207	17	13
csched1a	16	1	104	0	3	7	22	0	104	0	3	7	8	9	162	90	3	8
csched1	15	1	130	1	1	4	22	0	130	0	1	4	7	11	163	95	1	5
csched2a	125	28	6817	12	10	70	172	0	6128	0	10	71	109	101	7454	5651	10	66
csched2	152	64	1306	55	1	63	172	0	1481	0	1	66	105	109	1622	1390	1	64
densitymod	316	0	184553	0	315	315	420	0	190555	0	315	315	315	7	188504	3427	315	315
du-opt5	17	0	240	0	4	5	36	0	248	0	4	5	5	9	311	157	4	5
du-opt	17	0	184	0	4	5	40	0	203	0	4	6	5	9	282	176	4	5
edgecross10-060	52	0	2082	0	0	0	56	0	1996	0	0	0	37	43	1368	1368	0	0
edgecross14-039	161	4	7492	10	0	0	168	0	7492	0	0	0	95	100	5040	5040	0	0
elf	11	2	286	5	0	5	12	0	286	0	0	5	7	10	314	281	0	6
eniplac	65	2	369	5	0	23	94	0	383	0	0	23	27	30	418	394	0	21
enpro48pb	42	0	1204	0	10	14	58	0	1341	0	10	14	26	16	1860	773	10	14
enpro56pb	39	0	1667	0	8	17	48	0	1667	0	8	17	32	5	1823	350	8	14
ethanolh	159	30	2729	28	5	73	228	0	2248	0	5	85	143	63	2184	828	5	74
ethanolm	81	10	700	9	7	37	124	0	711	0	7	43	58	34	805	443	7	40
ex1224	8	0	67	0	4	6	8	0	67	0	4	6	8	6	130	70	4	6
ex1233	75	1	191	0	16	26	136	0	289	0	16	33	64	33	624	520	16	36
ex1243	42	4	85	4	0	5	56	0	86	0	0	4	33	23	41	41	0	15
ex1244	59	3	200	0	2	12	80	0	256	0	2	10	48	25	286	200	2	10
ex1252a	32	14	209	7	1	17	58	0	217	0	1	20	24	26	271	262	1	15
ex1252	32	17	500	20	3	16	58	0	287	0	3	25	28	29	413	361	3	21
ex1263a	22	1	112	1	12	16	40	0	108	0	12	15	20	5	102	34	12	15
ex1263	26	2	362	1	11	15	40	0	405	0	11	17	23	7	345	145	11	14
ex1264a	23	0	90	0	9	9	40	0	108	0	9	12	17	5	106	44	9	12
ex1264	24	1	198	0	8	13	40	0	199	0	8	14	17	9	250	140	8	14
ex1265a	40	0	231	0	12	16	60	0	292	0	12	20	27	5	303	83	12	14
ex1265	36	3	251	2	6	15	60	0	410	0	6	16	23	22	216	182	6	17
ex1266a	52	0	223	0	21	25	84	0	234	0	21	23	41	9	213	78	21	23
ex1266	55	1	770	1	19	26	84	0	913	0	19	26	43	25	1382	1016	19	34
ex4	8	0	238	0	4	5	10	0	238	0	4	5	5	6	273	119	4	5
fac1	23	11	70	14	0	8	32	0	93	0	0	9	11	15	115	90	0	9
fac2	48	21	156	25	0	37	108	0	191	0	0	33	46	23	164	58	0	35
fac3	69	17	338	16	0	36	108	0	473	0	0	29	55	14	360	35	0	36

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
feedtray	372	57	8258	71	241	115	412	0	7902	0	241	104	396	136	10509	3443	241	174
fin2bb	84	0	2975	0	61	61	84	0	2975	0	61	61	84	4	3943	1131	61	61
flay02h	3	0	74	0	2	2	4	0	74	0	2	2	2	11	195	181	2	2
flay02m	4	0	61	0	2	2	4	0	61	0	2	2	2	3	67	60	2	2
flay03h	6	0	213	0	3	4	6	0	213	0	3	4	4	14	718	509	3	4
flay03m	6	0	72	0	3	3	6	0	72	0	3	3	3	3	87	68	3	3
flay04h	8	0	234	0	4	1	8	0	234	0	4	1	9	24	314	252	4	0
flay04m	8	0	80	0	0	4	8	0	80	0	0	4	4	8	106	106	0	4
flay05h	10	0	1071	0	0	6	10	0	1071	0	0	6	6	4	1051	626	0	6
flay05m	9	0	103	0	0	5	10	0	103	0	0	5	5	3	151	113	0	5
flay06h	12	0	1776	0	0	5	12	0	1776	0	0	5	6	4	1206	1095	0	4
flay06m	12	0	253	0	0	2	12	0	253	0	0	2	6	10	207	207	0	6
fo7_2	27	4	101	4	0	1	28	0	103	0	0	1	17	7	183	80	0	1
fo7_ar2_1	20	0	98	0	0	14	28	0	118	0	0	14	14	17	159	86	0	14
fo7_ar25_1	19	0	149	0	0	4	28	0	129	0	0	4	9	17	203	108	0	4
fo7_ar3_1	19	0	103	0	0	1	28	0	89	0	0	1	7	15	156	86	0	1
fo7_ar4_1	17	0	94	0	0	5	28	0	96	0	0	7	10	21	128	83	0	5
fo7_ar5_1	23	1	107	0	0	6	28	0	138	0	0	6	17	19	185	107	0	6
fo7	20	1	183	2	0	1	28	0	224	0	0	1	12	3	336	182	0	1
fo8_ar2_1	22	0	99	0	0	16	32	0	91	0	0	16	16	19	146	78	0	16
fo8_ar25_1	25	0	123	0	0	7	32	0	138	0	0	7	12	17	223	109	0	7
fo8_ar3_1	18	0	106	0	0	4	32	0	129	0	0	4	10	18	159	88	0	4
fo8_ar4_1	18	0	114	0	0	7	32	0	120	0	0	8	15	23	221	130	0	11
fo8_ar5_1	25	2	122	0	0	6	32	0	130	0	0	6	16	24	172	86	0	6
fo8	26	0	240	0	0	1	32	0	297	0	0	1	15	6	338	150	0	1
fo9_ar2_1	25	0	58	0	0	18	36	0	63	0	0	18	18	19	178	75	0	18
fo9_ar25_1	24	0	104	0	0	7	36	0	116	0	0	7	11	18	223	115	0	7
fo9_ar3_1	25	0	160	0	0	4	36	0	134	0	0	4	10	27	306	159	0	4
fo9_ar4_1	24	1	178	0	0	7	36	0	178	0	0	8	17	21	311	147	0	7
fo9_ar5_1	30	2	153	0	0	8	36	0	146	0	0	8	20	27	214	97	0	8
fo9	31	0	266	0	0	0	36	0	318	0	0	0	12	6	368	232	0	0
fuel	11	1	192	1	4	8	12	0	227	0	4	8	9	4	225	76	4	9
fuzzy	123	19	5363	5	80	83	158	0	6291	0	80	83	118	31	10540	5868	80	84
gams01	236	8	120856	13	118	141	282	0	121454	0	118	144	240	16	143900	13548	118	139
gasprod_sarawak01	37	11	511	4	6	32	42	0	577	0	6	33	36	21	729	228	6	32
gasprod_sarawak16	572	149	30238	70	113	494	672	0	30937	0	113	543	528	271	29203	3985	113	483
gasprod_sarawak81	2899	813	265274	608	337	2443	3402	0	253840	0	337	2712	2715	1425	258950	24616	337	2456
gastrans	59	2	589	0	35	32	64	0	589	0	35	34	47	12	753	359	35	34
gear2	7	4	44	3	0	4	10	0	65	0	0	5	7	9	71	32	0	4
gear3	6	2	17	0	1	5	10	0	34	0	1	6	8	11	45	26	1	5
gear4	6	2	30	1	2	2	8	0	40	0	2	2	8	10	51	31	2	4
gear	6	2	17	0	1	5	10	0	34	0	1	6	8	11	45	26	1	5
genpooling_lee1	26	6	130	7	0	3	40	0	87	0	0	3	18	15	161	119	0	5
genpooling_lee2	29	3	88	4	0	9	48	0	130	0	0	11	14	12	140	114	0	9
genpooling_meyer04	34	1	1231	11	0	7	56	0	1481	0	0	4	16	7	921	559	0	10
genpooling_meyer10	135	5	6396	19	0	49	260	0	3783	0	0	52	94	14	2498	927	0	55
genpooling_meyer15	270	26	4818	92	0	106	540	0	5344	0	0	107	134	141	4078	3765	0	98
ghg_1veh	52	14	457	7	3	19	78	0	455	0	3	19	33	23	706	387	3	20
ghg_2veh	113	44	1918	21	27	59	150	0	2036	0	27	63	107	60	2741	1184	27	64
ghg_3veh	191	65	4903	45	46	107	256	0	4921	0	46	112	179	80	5614	1556	46	115
gkocis	4	0	19	0	2	3	4	0	19	0	2	3	3	2	26	15	2	3
heatexch_gen1	115	32	610	42	0	58	168	0	731	0	0	67	100	88	826	619	0	64
heatexch_gen3	747	112	1868	114	0	170	1050	0	2025	0	0	221	340	229	1984	1806	0	202
heatexch_spec1	94	31	373	40	1	21	120	0	438	0	1	31	46	51	506	506	1	33
heatexch_spec2	84	16	458	26	10	39	180	0	480	0	10	66	62	49	904	666	10	43

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
heatexch_spec3	442	86	1643	210	0	89	692	0	2844	0	0	116	201	162	2532	2189	0	90
heatexch_trigen	112	31	944	24	16	59	224	0	1001	0	16	92	70	49	805	363	16	55
hybriddynamic_var	52	1	381	1	22	25	58	0	379	0	22	25	33	10	440	151	22	25
hydroenergy1	154	13	1271	14	13	78	184	0	1231	0	13	93	118	59	1845	1109	13	76
hydroenergy2	284	33	2135	23	13	108	368	0	2217	0	13	164	169	93	3541	2402	13	98
hydroenergy3	480	89	5007	57	13	274	644	0	5005	0	13	365	360	197	9246	5012	13	259
jit1	8	0	67	0	6	6	8	0	67	0	6	6	8	9	133	74	6	6
johnall	7378	0	258635	0	7297	7318	7606	0	255913	0	7297	7319	7378	2	275737	4479	7297	7318
kport20	172	27	718	42	8	32	222	0	690	0	8	34	52	50	945	769	8	35
kport40	310	46	3692	34	0	94	470	0	3603	0	0	103	239	175	3959	1988	0	103
lip	8	0	103	0	8	7	8	0	103	0	8	7	8	2	103	0	8	7
m3	9	0	27	0	0	2	12	0	27	0	0	2	6	5	54	31	0	2
m6	18	1	103	1	0	3	24	0	103	0	0	3	15	8	238	110	0	3
m7_ar2_1	14	1	84	0	0	4	28	0	102	0	0	4	11	14	134	82	0	4
m7_ar25_1	18	1	99	0	0	3	28	0	105	0	0	3	11	14	141	81	0	3
m7_ar3_1	22	2	124	0	0	6	28	0	124	0	0	6	18	24	209	100	0	6
m7_ar4_1	19	1	153	1	0	7	28	0	153	0	0	7	17	26	255	156	0	7
m7_ar5_1	23	3	124	0	0	6	28	0	124	0	0	6	21	26	282	123	0	5
m7	16	0	139	0	0	1	28	0	132	0	0	1	12	3	176	91	0	1
meanvarxsc	10	0	46	0	0	8	14	0	46	0	0	10	7	12	42	38	0	7
milinfract	494	0	3759	0	0	1	1002	0	3800	0	0	1	2	61	7543	7543	0	2
minlphix	23	4	36	5	6	8	28	0	44	0	6	8	22	10	96	72	6	8
multiplants_mtg1a	56	6	8610	2	25	41	64	0	8610	0	25	41	55	24	7940	2311	25	47
multiplants_mtg1c	48	8	13566	3	19	27	64	0	10497	0	19	27	42	22	20697	10534	19	23
multiplants_mtg5	70	5	1654	6	33	44	80	0	1936	0	33	50	64	30	2504	1050	33	52
multiplants_stg1a	145	23	8303	34	0	48	234	0	9286	0	0	47	118	78	7042	5157	0	52
multiplants_stg1b	215	70	32291	109	8	37	266	0	38048	0	8	28	150	61	45667	14351	8	89
multiplants_stg1c	126	19	10159	29	33	41	202	0	15472	0	0	56	93	57	15392	10551	0	54
multiplants_stg1	212	75	31854	9483	8	91	330	0	23178	0	8	90	190	79	36124	19710	8	106
multiplants_stg6	217	42	48181	113	9	87	306	0	49075	0	9	82	172	74	101441	34886	9	88
ndcc12	836	5	19748	4	97	319	1172	0	19314	0	97	327	621	21	20442	1720	97	326
ndcc12persp	134	0	2002	0	4	76	184	0	1801	0	4	68	131	6	1807	929	4	80
ndcc13	837	1	20898	0	210	405	1176	0	20675	0	210	441	634	22	22759	3354	210	409
ndcc13persp	129	0	2067	0	72	86	168	0	1854	0	72	88	123	14	2200	1407	72	93
ndcc14	1138	4	37164	4	82	462	1620	0	39805	0	82	433	856	25	40173	6632	82	478
ndcc14persp	160	1	3123	1	2	88	216	0	2970	0	2	87	160	8	3728	1623	2	97
ndcc15	880	1	46475	0	87	361	1220	0	45696	0	87	363	664	31	50879	3580	87	382
ndcc16	1430	1	51437	1	60	535	2040	0	56198	0	60	512	1058	16	59065	6539	60	518
ndcc16persp	177	0	4552	0	0	58	240	0	3954	0	0	65	176	4	4424	1236	0	66
netmod_dol1	10	2	18	0	0	4	12	0	17	0	0	3	6	5	20	7	0	4
netmod_dol2	10	4	3547	5	0	2	12	0	3754	0	0	3	6	8	7705	6098	0	4
netmod_kar1	8	2	14	2	0	1	8	0	14	0	0	1	8	4	14	2	0	1
netmod_kar2	8	2	14	2	0	1	8	0	14	0	0	1	8	4	14	2	0	1
no7_ar2_1	20	0	113	0	0	14	28	0	117	0	0	14	14	17	167	99	0	14
no7_ar25_1	19	0	114	0	0	4	28	0	93	0	0	4	8	17	222	118	0	4
no7_ar3_1	27	0	137	0	0	1	28	0	137	0	0	1	15	18	250	121	0	1
no7_ar4_1	23	0	144	0	0	6	28	0	142	0	0	7	15	21	241	129	0	7
no7_ar5_1	22	2	188	0	0	7	28	0	188	0	0	7	19	20	285	142	0	6
nous1	54	10	285	13	6	34	76	0	300	0	6	39	45	37	351	189	6	34
nous2	61	19	381	20	7	37	76	0	368	0	7	41	55	43	344	167	7	44
nuclear104	2234	39	1764	16	0	29	4362	0	2005	0	0	29	2307	130	1683	461	0	29
nuclear14a	618	162	202427	89805	10	373	784	0	198348	0	10	379	591	209	227582	190963	10	376
nuclear14b	613	95	66415	4155	203	368	784	0	57354	0	203	383	605	122	83825	28942	203	380
nuclear14	417	0	325	0	0	0	820	0	654	0	0	0	410	6	958	61	0	0
nuclear25a	597	148	237521	137860	7	372	816	0	243772	0	7	376	594	216	243450	232252	7	372

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
nuclear25b	763	155	104529	2182	215	409	816	0	88376	0	215	428	767	179	123659	27889	215	411
nuclear25	435	1	386	1	0	0	856	0	947	0	0	0	429	5	810	49	0	0
nuclear49a	1390	361	94723	36372	8	809	1782	0	56127	0	8	775	1306	396	227466	173248	8	798
nuclear49b	1665	311	295948	9365	458	916	1782	0	326493	0	458	940	1865	581	530195	255470	458	1076
nuclear49	951	10	673	5	0	7	1866	0	2600	0	0	7	942	13	2327	98	0	7
nuclearva	189	2	132	2	0	0	366	0	201	0	0	0	185	6	290	20	0	0
nuclearvb	188	1	121	1	0	0	366	0	130	0	0	0	184	5	284	21	0	0
nuclearvc	188	1	121	1	0	0	366	0	150	0	0	0	184	5	274	21	0	0
nuclearvd	188	1	135	1	0	0	366	0	255	0	0	0	184	5	203	21	0	0
nuclearve	188	1	135	1	0	0	366	0	255	0	0	0	184	5	203	21	0	0
nuclearvf	188	1	135	1	0	0	366	0	255	0	0	0	184	5	203	21	0	0
nvs01	11	1	110	1	3	5	14	0	110	0	3	5	11	10	213	107	3	5
nvs02	5	0	21	0	0	0	10	0	20	0	0	0	1	4	39	25	0	0
nvs05	25	1	111	0	2	7	42	0	120	0	2	7	20	6	205	74	2	6
nvs06	9	0	27	0	5	7	18	0	42	0	5	9	8	10	41	30	5	8
nvs08	5	0	36	0	1	1	6	0	36	0	1	1	2	7	48	33	1	1
nvs10	4	0	5	0	4	4	4	0	5	0	4	4	4	2	11	4	4	4
nvs11	6	0	36	0	6	6	6	0	36	0	6	6	6	2	41	17	6	6
nvs12	8	0	51	0	8	8	8	0	51	0	8	8	8	4	63	25	8	8
nvs13	10	0	53	0	8	8	10	0	53	0	8	8	10	2	85	20	8	8
nvs15	4	0	20	0	2	2	4	0	20	0	2	2	3	2	31	22	2	2
nvs16	13	4	42	0	0	0	14	0	51	0	0	0	11	16	65	59	0	0
nvs17	14	0	112	0	11	11	14	0	112	0	11	11	13	2	134	29	11	11
nvs18	12	0	90	0	10	10	12	0	90	0	10	10	11	2	108	23	10	10
nvs19	16	0	176	0	12	12	16	0	176	0	12	12	13	5	207	63	12	12
nvs20	43	0	1660	0	33	33	64	0	1751	0	33	35	34	11	1967	532	33	33
nvs21	5	0	5	0	0	2	16	0	11	0	0	5	3	7	5	5	0	2
nvs23	17	0	249	0	13	13	18	0	249	0	13	13	14	3	272	36	13	13
nvs24	20	0	349	0	12	12	20	0	349	0	12	12	16	3	358	59	12	12
o7_2	23	0	118	0	0	0	28	0	123	0	0	0	13	6	200	118	0	0
o7_ar2_1	20	0	173	0	0	14	28	0	172	0	0	14	14	15	290	147	0	14
o7_ar25_1	26	0	143	0	0	4	28	0	143	0	0	4	15	14	254	110	0	4
o7_ar3_1	27	0	145	0	0	1	28	0	151	0	0	1	14	14	286	135	0	1
o7_ar4_1	23	0	182	0	0	6	28	0	162	0	0	7	14	20	271	120	0	6
o7_ar5_1	25	2	117	0	0	6	28	0	117	0	0	6	19	20	239	117	0	6
o7	21	0	83	0	0	0	28	0	102	0	0	0	8	4	137	87	0	0
o8_ar4_1	22	1	232	0	0	8	32	0	216	0	0	8	14	24	285	162	0	8
o9_ar4_1	27	0	211	0	0	6	36	0	194	0	0	8	17	29	318	143	0	7
oil2	1081	0	10787	0	839	1073	1098	0	10799	0	839	1078	1169	100	12697	2017	839	1163
oil	1370	104	11555	52	863	616	1430	0	12384	0	863	618	1335	190	12863	3843	863	627
ortez	51	12	922	21	8	32	66	0	1022	0	8	39	37	17	1245	386	8	36
pooling_epa1	128	57	5547	34	28	67	170	0	6009	0	28	78	112	73	6177	1475	28	67
pooling_epa2	201	74	8024	86	32	95	296	0	10156	0	32	109	168	102	7774	1926	32	92
pooling_epa3	620	194	91900	235	96	283	984	0	101028	0	96	299	430	252	98390	14640	96	290
portfol_card	17	0	25	0	10	17	18	0	24	0	10	18	19	22	120	74	10	19
portfol_classical050_1	77	0	536	0	59	61	100	0	536	0	59	61	77	2	600	72	59	61
portfol_classical200_2	355	0	3041	0	263	271	400	0	3041	0	263	271	356	2	3248	120	263	271
portfol_robust050_34	178	0	1453	0	168	172	200	0	1453	0	168	172	185	16	1767	297	168	180
portfol_robust100_09	358	0	3493	0	334	337	402	0	3493	0	334	337	357	3	3484	127	334	337
portfol_robust200_03	752	0	6259	0	703	708	802	0	6259	0	703	708	751	2	6544	166	703	708
portfol_shortfall050_68	200	11	605	0	188	200	200	0	605	0	188	200	200	13	641	50	188	200
portfol_shortfall100_04	404	0	2237	0	404	404	404	0	2237	0	404	404	404	6	2255	199	404	404
portfol_shortfall200_05	804	0	5178	0	804	804	804	0	5178	0	804	804	804	2	5182	89	804	804
primary	117	9	1664	2	47	68	192	0	1664	0	47	73	117	26	2137	660	47	66
procsel	2	0	6	0	2	2	4	0	15	0	2	3	2	2	6	6	2	2

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
product2	255	2	2123	1	157	211	318	0	2065	0	157	213	231	23	3318	1671	157	213
product	215	31	2819	20	0	32	326	0	2877	0	0	47	37	47	5047	4997	0	27
qapw	450	0	22954	0	450	450	450	0	22954	0	450	450	450	2	23189	212	450	450
ravempb	41	0	976	0	3	12	56	0	976	0	3	12	22	12	1958	1121	3	11
risk2bpb	6	0	285	0	3	3	6	0	285	0	3	3	3	3	241	223	3	3
routingdelay_bigm	3941	37	426	8	46	174	4110	0	427	0	46	267	3606	142	450	441	46	176
routingdelay_proj	4080	20	351	1	0	135	4246	0	355	0	0	239	3743	142	338	338	0	133
rsyn0805h	18	0	31	0	3	5	24	0	31	0	3	5	8	7	50	46	3	5
rsyn0805m02h	27	4	351	2	4	17	36	0	352	0	4	23	23	13	626	389	4	20
rsyn0805m02m	11	0	116	0	0	4	12	0	116	0	0	4	7	9	99	91	0	5
rsyn0805m03m	18	1	269	1	0	11	18	0	269	0	0	11	14	14	446	312	0	10
rsyn0805m04m	23	1	499	1	0	13	24	0	499	0	0	13	16	12	792	394	0	15
rsyn0805m	5	0	43	0	0	2	6	0	35	0	0	1	2	6	40	40	0	2
rsyn0810m02m	22	0	159	0	0	9	24	0	207	0	0	8	11	9	268	219	0	10
rsyn0810m03m	33	2	470	3	0	16	36	0	437	0	0	13	16	21	433	396	0	10
rsyn0810m04m	45	4	872	5	0	19	48	0	869	0	0	17	26	15	1100	601	0	17
rsyn0810m	9	0	62	0	0	3	12	0	63	0	0	2	6	10	100	94	0	5
rsyn0815m02m	39	3	589	3	0	15	48	0	592	0	0	16	22	19	1045	783	0	13
rsyn0815m03m	61	1	2026	0	2	21	72	0	1917	0	2	20	39	16	2737	1359	2	20
rsyn0815m04m	83	0	2744	0	3	27	96	0	2322	0	3	22	45	21	4672	3035	3	31
rsyn0815m	19	2	139	3	0	4	24	0	157	0	0	2	11	9	340	162	0	8
rsyn0820m02m	50	2	862	1	0	21	60	0	821	0	0	21	31	15	1152	612	0	18
rsyn0820m03m	68	4	1476	10	2	24	90	0	1467	0	2	21	46	15	2368	1175	2	28
rsyn0820m04m	102	1	1917	1	3	29	120	0	1846	0	3	33	58	34	3956	2616	3	33
rsyn0820m	26	1	190	1	0	7	30	0	193	0	0	7	13	5	288	110	0	7
rsyn0830m02m	62	4	830	3	3	25	84	0	794	0	3	23	42	15	1367	774	3	29
rsyn0830m03m	98	8	1428	9	0	34	126	0	1398	0	0	37	59	27	2662	1476	0	41
rsyn0830m04m	136	7	3120	7	4	53	168	0	2719	0	4	48	84	19	3204	1303	4	50
rsyn0830m	29	0	288	0	0	13	42	0	303	0	0	14	21	10	411	237	0	11
rsyn0840m02m	91	3	1115	8	0	36	116	0	1093	0	0	38	61	22	1697	1074	0	34
rsyn0840m03m	145	7	2006	15	1	51	174	0	2281	0	1	53	95	33	3386	1823	1	56
rsyn0840m04m	190	6	3313	6	3	53	232	0	3596	0	3	57	112	22	3766	1479	3	68
rsyn0840m	42	0	366	0	1	15	58	0	369	0	1	16	29	14	607	369	1	17
sep1	10	0	61	0	6	8	10	0	61	0	6	8	10	3	63	24	6	8
sepasequ_complex	653	101	12396	102	53	210	984	0	13587	0	53	267	331	145	18849	4202	53	223
sfacloc1_2_80	162	15	1415	13	8	42	198	0	1385	0	8	39	109	60	1893	1031	8	40
sfacloc1_2_90	152	24	1441	28	8	23	184	0	1563	0	8	28	102	64	1428	819	8	29
sfacloc1_2_95	153	31	1828	61	2	16	178	0	1471	0	2	16	86	94	2080	1560	2	18
sfacloc1_3_80	232	34	2585	38	0	53	306	0	2833	0	0	55	158	90	3226	1755	0	57
sfacloc1_3_90	229	46	3472	87	0	35	300	0	4144	0	0	44	145	96	4583	2089	0	43
sfacloc1_3_95	225	56	2068	90	1	40	312	0	2430	0	1	44	164	174	3136	2398	1	50
sfacloc1_4_80	313	36	4018	33	0	102	426	0	3493	0	0	104	195	157	4233	2966	0	75
sfacloc1_4_90	293	33	4758	37	0	58	436	0	4832	0	0	60	173	125	5487	2377	0	47
sfacloc1_4_95	313	73	3261	114	0	49	426	0	3193	0	0	42	239	209	3452	2292	0	57
sfacloc2_2_80	125	11	1903	10	0	40	150	0	1941	0	0	36	69	74	1506	1090	0	40
sfacloc2_2_90	121	6	1613	8	0	35	150	0	1211	0	0	33	70	75	1332	1118	0	41
sfacloc2_2_95	63	6	1059	4	0	18	80	0	1001	0	0	14	35	40	696	630	0	19
sfacloc2_3_80	60	0	1463	0	0	51	120	0	1610	0	0	50	54	3	1666	395	0	54
sfacloc2_3_90	60	0	1087	0	0	48	120	0	1312	0	0	49	54	6	1324	684	0	54
sfacloc2_3_95	32	0	996	0	0	23	64	0	777	0	0	25	29	3	898	188	0	24
sfacloc2_4_80	75	0	2130	0	0	64	150	0	2448	0	0	67	69	6	3315	480	0	67
sfacloc2_4_90	75	0	1531	0	0	69	150	0	1815	0	0	67	69	3	1522	452	0	69
sfacloc2_4_95	40	0	1107	0	0	33	80	0	1255	0	0	33	37	3	1415	317	0	33
slay04h	14	1	636	5	4	9	16	0	636	0	4	9	11	12	814	751	4	9
slay04m	16	1	231	1	3	4	16	0	231	0	3	4	7	10	299	225	3	4

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
slay05h	18	0	973	0	5	11	20	0	973	0	5	11	19	19	1594	1486	5	19
slay05m	19	0	390	0	0	0	20	0	390	0	0	0	5	6	322	299	0	0
slay06h	24	0	1912	0	6	18	24	0	1912	0	6	18	27	30	4077	3857	6	24
slay06m	22	1	508	1	0	0	24	0	508	0	0	0	0	6	345	345	0	0
slay07h	28	0	3397	0	3	20	28	0	3397	0	3	20	25	23	7459	6995	3	22
slay07m	27	0	653	0	0	0	28	0	653	0	0	0	1	7	560	560	0	1
slay08h	32	1	2707	0	4	27	32	0	2707	0	4	27	34	33	5506	5323	4	34
slay08m	32	1	1191	1	0	1	32	0	1191	0	0	1	0	6	620	620	0	0
slay09h	36	1	5282	0	5	28	36	0	5282	0	5	28	36	38	12896	12731	5	33
slay09m	30	0	1419	0	0	0	36	0	1419	0	0	0	1	7	879	879	0	1
slay10h	40	2	5700	0	5	35	40	0	5700	0	5	35	42	45	30797	30142	5	39
slay10m	38	1	1797	1	0	0	40	0	1797	0	0	0	0	6	1412	1412	0	0
smallinvDAXr1b010-011	13	0	87	0	8	8	16	0	87	0	8	8	12	2	83	19	8	8
smallinvDAXr1b020-022	15	0	76	0	8	8	16	0	76	0	8	8	12	3	85	23	8	8
smallinvDAXr1b050-055	14	0	88	0	12	12	16	0	88	0	12	12	13	2	90	11	12	12
smallinvDAXr1b100-110	15	0	106	0	13	13	16	0	106	0	13	13	14	2	116	22	13	13
smallinvDAXr1b150-165	34	0	131	0	26	28	60	0	131	0	26	28	30	7	138	20	26	28
smallinvDAXr1b200-220	34	0	131	0	26	28	60	0	131	0	26	28	30	7	140	21	26	28
smallinvDAXr2b010-011	14	0	88	0	7	7	16	0	90	0	7	7	10	4	89	28	7	7
smallinvDAXr2b020-022	15	0	92	0	9	9	16	0	92	0	9	9	12	3	97	20	9	9
smallinvDAXr2b050-055	34	0	160	0	27	27	60	0	158	0	27	27	30	5	178	24	27	27
smallinvDAXr2b100-110	14	0	93	0	12	12	16	0	93	0	12	12	13	2	107	14	12	12
smallinvDAXr2b150-165	15	0	85	0	13	13	18	0	85	0	13	13	14	2	103	21	13	13
smallinvDAXr2b200-220	14	0	92	0	13	13	16	0	92	0	13	13	13	2	100	13	13	13
smallinvDAXr3b010-011	15	0	81	0	7	7	16	0	81	0	7	7	12	3	96	25	7	7
smallinvDAXr3b020-022	14	0	100	0	10	10	16	0	100	0	10	10	13	2	128	26	10	10
smallinvDAXr3b050-055	15	0	95	0	13	13	16	0	95	0	13	13	13	3	107	18	13	13
smallinvDAXr3b100-110	15	0	110	0	13	13	16	0	110	0	13	13	13	2	131	21	13	13
smallinvDAXr3b150-165	14	0	101	0	13	13	16	0	101	0	13	13	13	2	111	17	13	13
smallinvDAXr3b200-220	16	0	99	0	13	13	18	0	99	0	13	13	14	2	112	19	13	13
smallinvDAXr4b010-011	14	0	105	0	7	7	16	0	105	0	7	7	12	2	99	26	7	7
smallinvDAXr4b020-022	15	0	113	0	11	11	16	0	113	0	11	11	13	2	134	22	11	11
smallinvDAXr4b050-055	15	0	136	0	13	13	16	0	136	0	13	13	13	3	154	30	13	13
smallinvDAXr4b100-110	14	0	109	0	13	13	16	0	109	0	13	13	13	2	122	15	13	13
smallinvDAXr4b150-165	13	0	97	0	13	13	14	0	97	0	13	13	13	2	100	11	13	13
smallinvDAXr4b200-220	15	0	102	0	13	13	18	0	102	0	13	13	14	2	115	16	13	13
smallinvDAXr5b010-011	13	0	101	0	8	8	16	0	101	0	8	8	11	3	148	48	8	8
smallinvDAXr5b020-022	15	0	128	0	11	11	16	0	126	0	11	11	13	3	131	23	11	11
smallinvDAXr5b050-055	14	0	99	0	13	13	16	0	99	0	13	13	13	2	116	16	13	13
smallinvDAXr5b100-110	16	0	99	0	9	9	18	0	99	0	9	9	14	2	111	14	9	9
smallinvDAXr5b150-165	15	0	98	0	13	13	16	0	98	0	13	13	13	3	105	16	13	13
smallinvDAXr5b200-220	15	0	98	0	13	13	18	0	98	0	13	13	14	2	117	20	13	13
smallinvSNPr1b010-011	114	0	963	0	68	72	198	0	925	0	68	72	98	19	1242	350	68	72
smallinvSNPr1b020-022	112	0	1045	0	67	79	200	0	1025	0	67	79	99	14	1231	210	67	79
smallinvSNPr1b050-055	107	0	1101	0	39	81	200	0	1106	0	39	81	99	8	1180	130	39	81
smallinvSNPr1b100-110	111	0	1232	0	26	79	200	0	1239	0	26	79	99	12	1386	215	26	79
smallinvSNPr1b150-165	112	0	1176	0	31	84	200	0	1173	0	31	84	99	9	1212	118	31	84
smallinvSNPr1b200-220	111	0	1175	0	33	84	200	0	1150	0	33	84	99	15	1370	239	33	84
smallinvSNPr2b010-011	113	0	781	0	75	76	198	0	754	0	75	76	98	16	975	254	75	76
smallinvSNPr2b020-022	111	0	1000	0	47	63	200	0	1074	0	47	63	100	11	1147	166	47	63
smallinvSNPr2b050-055	110	0	1237	0	36	82	200	0	1240	0	36	82	99	11	1390	201	36	82
smallinvSNPr2b100-110	110	0	1106	0	30	81	200	0	1137	0	30	81	99	15	1375	273	30	81
smallinvSNPr2b150-165	109	0	1371	0	22	85	200	0	1350	0	22	85	99	10	1535	188	22	85
smallinvSNPr2b200-220	112	0	1338	0	28	84	200	0	1343	0	28	84	99	17	1570	295	28	84
smallinvSNPr3b010-011	112	0	755	0	68	69	198	0	760	0	68	69	98	14	917	204	68	69

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
smallinvSNPr3b020-022	109	0	839	0	61	72	200	0	871	0	61	72	99	12	959	178	61	72
smallinvSNPr3b050-055	109	0	997	0	26	73	200	0	979	0	26	73	99	13	1141	205	26	73
smallinvSNPr3b100-110	110	0	1268	0	32	87	200	0	1239	0	32	87	98	11	1378	172	32	87
smallinvSNPr3b150-165	110	0	1151	0	21	85	200	0	1160	0	21	85	99	14	1347	205	21	85
smallinvSNPr3b200-220	114	0	1143	0	13	78	200	0	1126	0	13	78	99	14	1332	233	13	78
smallinvSNPr4b010-011	110	0	719	0	56	57	198	0	732	0	56	57	98	15	941	225	56	57
smallinvSNPr4b020-022	106	0	876	0	43	61	200	0	890	0	43	61	99	15	1063	227	43	61
smallinvSNPr4b050-055	110	0	1062	0	35	80	200	0	1070	0	35	80	99	13	1148	182	35	80
smallinvSNPr4b100-110	110	0	1197	0	18	82	200	0	1219	0	18	82	99	11	1345	179	18	82
smallinvSNPr4b150-165	109	0	1024	0	18	88	200	0	1062	0	18	88	99	15	1148	192	18	88
smallinvSNPr4b200-220	110	0	1308	0	20	89	200	0	1313	0	20	89	99	11	1478	204	20	89
smallinvSNPr5b010-011	110	0	620	0	66	67	196	0	618	0	66	67	97	14	769	169	66	67
smallinvSNPr5b020-022	110	0	798	0	44	60	198	0	847	0	44	60	98	14	1005	193	44	60
smallinvSNPr5b050-055	108	0	747	0	12	65	200	0	712	0	12	65	99	12	838	174	12	65
smallinvSNPr5b100-110	109	0	709	0	35	89	200	0	710	0	35	89	99	13	877	204	35	89
smallinvSNPr5b150-165	110	0	883	0	15	90	200	0	881	0	15	90	99	11	1038	206	15	90
smallinvSNPr5b200-220	111	0	1044	0	34	92	200	0	1055	0	34	92	98	14	1157	185	34	92
space25a	64	19	81	0	0	28	86	0	106	0	0	35	57	34	64	64	0	29
space25	44	0	80	0	0	2	52	0	80	0	0	2	30	4	87	62	0	0
space960	2494	112	804391	128	12	0	3394	0	769976	0	12	0	417	389	120398	101315	12	0
spectra2	36	0	817	0	26	26	60	0	819	0	26	27	26	5	940	201	26	26
sporttournament14	118	8	496	14	0	2	140	0	425	0	0	2	6	14	390	390	0	1
spring	23	2	117	0	7	12	24	0	117	0	7	12	24	7	178	61	7	13
squfl010-025	449	3	14415	3	0	160	500	0	13166	0	0	157	250	12	13753	841	0	163
squfl010-025persp	705	57	89169	65	267	424	1000	0	89133	0	267	422	580	91	89984	4551	267	421
squfl010-040	726	5	38553	2	0	273	800	0	36779	0	0	257	408	65	31586	2590	0	260
squfl010-040persp	1214	78	325905	105	508	712	1600	0	326615	0	508	706	1130	88	332883	6274	508	710
squfl010-080	1391	32	36181	15	0	493	1600	0	37146	0	0	489	505	525	9573	9573	0	451
squfl010-080persp	2809	136	610351	144	800	1327	3200	0	621749	0	800	1310	1646	230	627460	27848	800	1336
squfl015-060	1718	26	86294	5	0	577	1800	0	96280	0	0	566	930	96	85533	6147	0	556
squfl015-060persp	2909	713	1081371	1657	900	1391	3600	0	1080039	0	900	1423	1878	212	1049697	18371	900	1419
squfl015-080	2236	53	93262	32	0	752	2400	0	92729	0	0	751	1185	133	80626	6777	0	724
squfl015-080persp	3860	322	1377896	380	1200	2008	4800	0	1473240	0	1200	1982	2479	222	1451764	33174	1200	1989
squfl020-040	1551	18	74615	6	0	492	1600	0	68652	0	0	502	824	77	66903	2902	0	510
squfl020-040persp	2265	252	881474	438	1109	1459	3200	0	895950	0	1109	1460	2017	267	921810	20486	1109	1465
squfl020-050	1906	54	138898	8	0	639	2000	0	156945	0	0	627	1012	93	143945	4975	0	632
squfl020-050persp	2877	192	1529480	250	1277	1896	4000	0	1501841	0	1277	1894	2578	263	1526331	52421	1277	1898
squfl020-150persp	9480	1279	4623861	2121	3000	5102	12000	0	4832813	0	3000	5121	6206	402	4896913	109949	3000	5285
squfl025-025	1188	2	59358	3	0	361	1250	0	61586	0	0	374	619	20	53480	1219	0	394
squfl025-025persp	1628	132	616521	276	919	1165	2500	0	603827	0	919	1142	1522	209	608157	8951	919	1162
squfl025-030	1482	13	76202	11	0	441	1500	0	78134	0	0	448	778	40	76950	1847	0	438
squfl025-030persp	1953	240	734069	418	788	1278	3000	0	747281	0	788	1298	1691	219	741719	13266	788	1282
squfl025-040	1939	24	199743	9	0	644	2000	0	184886	0	0	631	1009	85	168250	3927	0	605
squfl025-040persp	2791	302	1024115	419	1353	1845	4000	0	1028077	0	1353	1860	2201	239	1013467	20961	1353	1840
squfl030-100persp	8672	2314	3607947	4100	3000	5138	12000	0	3709486	0	3000	4975	6059	892	3773481	129782	3000	5066
squfl030-150persp	14300	3119	6118433	4888	4500	7839	18000	0	6219033	0	4500	7674	9827	1690	6200935	66619	4500	7780
squfl040-080	5885	159	568778	94	0	1989	6400	0	545366	0	0	1955	3164	152	552823	9776	0	1992
squfl040-080persp	9821	3614	2989762	6934	3200	5337	12800	0	2999354	0	3200	5349	6385	1217	3009751	71812	3200	5490
sssd08-04	14	3	147	0	4	10	16	0	161	0	4	12	10	15	231	129	4	10
sssd08-04persp	27	9	341	19	4	15	32	0	279	0	4	12	24	27	408	260	4	16
sssd12-05	18	4	294	0	0	13	20	0	360	0	0	14	14	16	331	142	0	13
sssd12-05persp	36	11	512	23	5	20	40	0	482	0	5	12	27	28	629	364	5	16
sssd15-04	16	5	293	1	0	12	16	0	293	0	0	12	13	17	375	171	0	12
sssd15-04persp	31	10	424	23	4	15	32	0	468	0	4	10	25	27	625	374	4	17
sssd15-06	20	4	503	0	0	14	24	0	526	0	0	16	17	18	611	223	0	14

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
sssd15-06persp	39	9	705	19	6	22	48	0	747	0	6	20	32	34	874	498	6	22
sssd15-08	26	6	652	1	8	18	32	0	746	0	8	21	21	20	773	247	8	18
sssd15-08persp	49	12	925	17	8	24	64	0	987	0	8	23	43	45	1370	823	8	30
sssd16-07	24	6	484	1	7	17	28	0	499	0	7	19	19	19	574	208	7	17
sssd16-07persp	48	14	774	22	7	25	56	0	739	0	7	22	35	37	995	560	7	24
sssd18-06	20	6	561	2	6	14	24	0	595	0	6	17	16	17	668	233	6	14
sssd18-06persp	40	13	737	19	6	19	48	0	843	0	6	22	34	39	1160	709	6	24
sssd18-08	28	6	840	0	8	20	32	0	783	0	8	22	24	24	991	360	8	20
sssd18-08persp	54	17	1136	31	8	33	64	0	1134	0	8	30	48	48	1376	801	8	34
sssd20-04	14	5	283	4	0	10	16	0	311	0	0	11	11	14	394	184	0	10
sssd20-04persp	27	7	449	17	4	12	32	0	454	0	4	16	22	25	628	428	4	14
sssd20-08	26	7	675	2	8	18	32	0	634	0	8	21	21	20	774	242	8	18
sssd20-08persp	52	12	915	28	8	29	64	0	923	0	8	23	35	36	1259	733	8	22
sssd22-08	28	7	789	1	8	20	32	0	822	0	8	22	22	19	920	207	8	20
sssd22-08persp	55	17	1029	35	8	26	64	0	949	0	8	30	42	44	1460	843	8	27
sssd25-04	14	4	399	1	4	10	16	0	424	0	4	11	11	14	418	172	4	10
sssd25-04persp	28	8	501	17	4	17	32	0	518	0	4	14	22	23	669	371	4	14
sssd25-08	26	6	814	2	8	18	32	0	831	0	8	22	21	20	897	291	8	18
sssd25-08persp	51	13	1097	25	8	28	64	0	1023	0	8	26	42	43	1388	774	8	29
st_e29	8	0	67	0	4	6	8	0	67	0	4	6	8	6	130	70	4	6
st_e31	7	0	28	0	2	2	10	0	36	0	2	2	5	4	32	14	2	2
st_e32	63	2	373	2	17	23	94	0	381	0	17	23	39	10	573	244	17	23
st_e36	13	3	111	1	4	3	16	0	110	0	4	5	12	7	213	94	4	3
st_e38	10	1	47	0	8	7	14	0	47	0	8	7	11	6	67	19	8	7
st_e40	23	4	166	2	5	4	24	0	166	0	5	4	20	13	178	87	5	3
stockcycle	72	0	2522	0	0	0	96	0	2325	0	0	0	48	3	2638	150	0	0
st_test8	17	2	12	2	13	15	22	0	11	0	13	18	17	7	17	14	13	15
st_testgr1	10	0	51	0	10	10	10	0	51	0	10	10	10	2	57	10	10	10
st_testgr3	12	0	47	0	12	12	14	0	50	0	12	12	12	2	60	12	12	12
super1	1129	215	115163	181	341	609	1442	0	110139	0	341	631	1069	306	111473	15828	340	638
super2	1125	199	104966	127	356	609	1444	0	107089	0	355	628	1041	276	104621	11532	356	628
super3	1146	186	130325	131	354	602	1454	0	126445	0	353	637	1070	277	114880	19081	353	606
super3t	853	159	95314	131	256	512	1026	0	85543	0	256	498	804	244	106848	13693	256	532
supplychain	12	1	82	0	4	6	12	0	82	0	4	6	8	4	81	23	4	6
supplychainp1_022020	60	0	13257	0	24	38	80	0	15852	0	24	37	40	3	15417	1173	24	36
supplychainp1_030510	25	0	989	0	10	10	30	0	988	0	10	10	17	7	1329	447	10	12
supplychainp1_053050	130	0	82449	0	50	59	160	0	68027	0	50	76	114	41	123255	55473	50	95
supplychainr1_053050	95	1	189541	0	3	33	160	0	230721	0	3	41	43	29	291416	65292	3	33
syn05h	16	0	69	0	6	11	24	0	72	0	6	10	18	14	60	38	6	14
syn05m02m	10	0	121	0	4	4	12	0	121	0	4	4	6	8	174	92	4	5
syn05m03m	13	0	138	0	5	9	18	0	145	0	5	9	9	11	250	150	5	7
syn05m04m	20	1	179	1	6	9	24	0	139	0	6	7	8	8	355	191	6	8
syn10h	29	7	162	8	1	19	36	0	156	0	1	18	20	18	234	149	1	19
syn10m02m	21	1	254	2	0	7	24	0	253	0	0	8	10	13	398	265	0	9
syn10m03m	31	0	630	0	0	15	36	0	724	0	0	15	18	18	831	488	0	16
syn10m04m	44	1	860	3	0	13	48	0	838	0	0	12	22	23	1415	769	0	18
syn10m	7	0	48	0	6	6	12	0	78	0	6	8	6	2	70	23	6	6
syn15m02m	39	2	738	1	8	12	48	0	722	0	8	16	22	14	1238	532	8	15
syn15m03m	57	4	2157	3	17	23	72	0	2151	0	17	22	32	16	3332	1264	17	22
syn15m04m	78	5	2400	4	2	29	96	0	1991	0	2	26	42	9	2969	803	2	27
syn15m	19	0	138	0	4	7	22	0	138	0	4	7	10	7	247	137	4	7
syn20m02m	49	3	682	2	0	14	56	0	675	0	0	18	31	17	1026	441	0	23
syn20m03m	71	5	836	6	0	23	84	0	938	0	0	21	35	26	2098	1657	0	17
syn20m04m	95	5	1792	6	0	30	112	0	1947	0	0	25	57	28	4517	2687	0	28
syn20m	24	1	295	2	1	5	28	0	295	0	1	5	14	9	374	203	1	7

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
syn30m02m	58	4	709	6	6	22	80	0	632	0	6	22	38	12	1047	449	6	29
syn30m03m	91	3	1141	4	5	37	120	0	1196	0	5	36	58	27	2463	1747	5	37
syn30m04m	131	11	1356	17	10	40	160	0	1316	0	10	47	76	28	1973	1072	10	54
syn30m	29	2	296	2	3	12	42	0	283	0	3	12	21	9	352	172	3	15
syn40m02m	81	7	850	8	0	36	112	0	827	0	0	34	56	20	1820	1180	0	36
syn40m03m	141	6	1458	6	0	47	168	0	1423	0	0	52	82	41	3637	2716	0	55
syn40m04m	192	13	2158	18	0	51	224	0	2210	0	0	53	111	35	6552	5087	0	69
syn40m	39	2	453	3	0	17	56	0	471	0	0	18	31	9	728	312	0	18
synheat	78	22	275	18	1	24	120	0	358	0	1	28	33	38	302	302	1	24
synthes1	3	0	9	0	0	1	4	0	9	0	0	1	1	5	11	11	0	1
synthes2	6	2	33	1	2	4	8	0	33	0	2	3	3	6	44	44	2	3
synthes3	10	2	59	1	0	1	12	0	47	0	0	2	6	10	59	51	0	1
tanksize	26	1	104	2	7	16	34	0	104	0	7	16	19	6	170	73	7	15
tln12	207	2	7240	2	38	64	288	0	7882	0	38	71	108	11	2463	2037	38	60
tln4	30	0	149	0	9	13	40	0	179	0	9	10	15	8	152	81	9	10
tln5	33	1	310	1	15	18	60	0	345	0	15	17	24	8	262	74	15	16
tln6	55	2	384	5	19	22	84	0	386	0	19	21	41	5	323	102	19	21
tln7	89	0	836	0	5	21	112	0	830	0	5	22	41	18	479	337	5	23
tloss	51	0	227	0	21	23	84	0	229	0	21	22	39	7	211	76	21	23
tls12	230	12	1893	1	24	109	436	0	1616	0	24	160	207	22	1774	685	24	88
tls2	9	2	56	2	0	2	12	0	72	0	0	2	7	6	111	57	0	0
tls4	58	13	438	15	15	29	74	0	505	0	15	30	42	31	685	524	15	35
tls5	65	14	1037	54	20	36	86	0	992	0	20	40	62	44	1717	1346	20	45
tls6	88	21	1264	84	20	32	120	0	1306	0	20	40	74	47	1870	1509	20	40
tls7	120	19	2222	23	70	78	182	0	3446	0	70	84	122	43	2344	1054	70	78
tltr	42	3	444	0	6	27	54	0	406	0	6	17	36	25	633	341	6	29
transswitch0009r	80	4	459	5	0	4	94	0	484	0	0	4	18	11	373	299	0	2
transswitch0014r	181	20	1739	23	1	37	206	0	1749	0	1	41	71	69	1688	1508	1	40
transswitch0030r	366	55	4041	18	0	111	396	0	3876	0	0	102	150	116	2649	2233	0	118
transswitch0039r	432	20	5755	30	0	8	500	0	5887	0	0	6	117	37	5388	3292	0	7
transswitch0057r	649	56	7112	57	0	134	736	0	6913	0	0	145	309	238	6837	5665	0	147
transswitch0118r	1551	132	19203	100	0	349	1830	0	19339	0	0	333	732	547	22161	18175	0	337
transswitch0300r	3491	306	39201	235	0	539	4230	0	40442	0	0	572	1358	923	45501	36712	0	565
transswitch2383wpr	25117	1227	179768	623	0	1533	28728	0	181489	0	0	1435	5583	2424	133530	111048	0	1419
transswitch2736spr	32113	1602	236137	699	0	2020	38460	0	241232	0	0	1894	8214	2965	201263	141018	0	1907
tspn05	34	1	707	2	17	15	54	0	870	0	17	15	19	8	925	434	17	15
tspn08	74	2	2001	2	27	34	108	0	1851	0	27	34	34	12	2453	1390	27	34
tspn10	146	1	11546	1	128	133	188	0	12298	0	128	133	147	13	10649	1229	128	133
tspn12	206	4	21790	4	172	182	256	0	23316	0	172	181	195	15	21119	2163	172	181
tspn15	283	11	19453	14	129	142	346	0	20179	0	129	142	150	23	16233	6490	129	144
unitcommit1	300	18	14905	19	1	30	480	0	14792	0	1	42	198	86	18784	8960	1	20
unitcommit2	500	129	855155	131	0	101	812	0	779384	0	0	91	296	235	287186	77177	0	82
util	11	0	62	0	9	9	12	0	62	0	9	9	10	7	73	26	9	9
waste	246	28	2444	26	2	85	450	0	2291	0	2	112	176	104	4444	3722	2	65
water3	88	8	1444	9	23	43	120	0	1506	0	23	43	60	30	1412	568	23	45
watercontamination0202	521	16	57183	81	50	67	1180	0	55465	0	50	96	257	121	40882	10293	50	66
watercontamination0202r	129	45	823	15	0	15	188	0	851	0	0	51	85	64	1330	1108	0	0
watercontamination0303	940	88	32783	415	90	179	2446	0	30429	0	90	285	267	163	21338	9296	90	166
watercontamination0303r	201	44	2978	16	0	19	370	0	2909	0	0	35	65	92	1681	1681	0	0
waternd1	34	3	364	2	0	15	54	0	495	0	0	12	20	13	660	351	0	13
waternd2	112	12	5086	10	0	63	184	0	4403	0	0	61	75	45	3807	1400	0	62
waterno2_01	44	14	561	18	21	33	58	0	583	0	21	37	44	32	913	548	21	34
waterno2_02	100	38	1740	28	13	54	116	0	1891	0	13	58	83	72	3469	2576	13	45
waterno2_03	143	61	3787	88	15	76	174	0	3596	0	15	87	146	130	5788	3976	15	72
waterno2_04	174	64	4771	40	20	83	232	0	4777	0	20	113	165	129	7018	4010	20	83

Table 3 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
waterno2_06	271	119	8017	86	17	139	348	0	7748	0	17	170	255	206	12669	8452	17	139
waterno2_09	406	151	13282	128	28	218	522	0	11547	0	28	259	388	308	21903	13450	28	197
waterno2_12	525	205	17886	194	43	290	696	0	18277	0	43	341	513	403	34596	22624	43	288
waterno2_18	814	324	26333	222	60	412	1044	0	26533	0	60	505	762	610	69999	53301	60	401
waterno2_24	1116	416	38669	310	72	583	1392	0	39617	0	72	677	1097	852	88973	64398	72	596
watertreatnd_conc	59	9	426	1	5	25	110	0	513	0	5	17	29	21	404	252	5	17
watertreatnd_flow	65	16	1794	12	7	48	110	0	1850	0	7	61	61	31	2910	1067	7	46
waterx	101	24	2267	2	2	17	148	0	2267	0	2	17	93	68	1920	822	2	14

Table 4: Detailed results comparing filtering strategies at the root node on test set GO. See Table 1 for aggregated results.

lp — number of LPs solved during OBBT
lp_{filt} — number of LPs solved during OBBT's filtering
iter — number of LP iterations during OBBT
$\text{iter}_{\text{filt}}$ — number of LP iterations of OBBT's filtering LPs
b_{obbt} — number of bounds tightened by OBBT
lvb — number of LVBs found by OBBT

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
alkylation	15	3	87	3	9	10	16	0	87	0	9	10	14	7	116	34	9	10
alkyl	20	0	93	0	13	15	26	0	93	0	13	15	19	6	128	39	13	15
arki0003	4357	1237	16100	566	350	1228	8968	0	20052	0	350	1505	2491	1753	354189	40125	350	2199
arki0004	4983	130	24091	223	0	2235	6474	0	25368	0	0	2323	3905	1736	67760	54431	0	1598
arki0015	2917	626	515010	365	394	1895	3968	0	509218	0	394	1894	2459	794	491857	18746	394	1935
arki0018	19668	2682	191737	2682	329	15021	39216	0	202065	0	329	16808	17053	6997	146956	32702	329	15313
arki0019	1547	0	4858	0	0	708	2038	0	4817	0	0	716	1020	9	5748	2087	0	718
arki0020	3771	1	11474	1	0	1734	5046	0	11817	0	0	1731	2523	7	13565	4379	0	1732
bayes2_10	89	44	638	35	7	62	128	0	609	0	7	66	71	40	690	198	7	67
bayes2_20	79	32	933	18	7	73	128	0	1015	0	7	81	75	33	1164	271	7	75
bayes2_30	94	44	1072	17	7	88	128	0	1178	0	7	90	88	46	1219	161	7	87
bayes2_50	87	24	1627	11	7	75	128	0	1644	0	7	76	83	26	1874	180	7	70
bearing	23	7	233	2	4	16	34	0	240	0	4	19	19	8	328	40	4	16
btest14	95	7	2082	6	16	56	166	0	2269	0	16	58	99	32	2410	506	16	69
camshape100	160	0	483	0	61	98	200	0	483	0	61	99	119	34	584	385	61	108
camshape200	316	0	973	0	113	171	400	0	973	0	113	172	214	68	1337	852	113	184
camshape400	626	0	2030	0	218	335	800	0	2030	0	218	336	470	186	2738	1762	218	432
camshape800	1244	0	4046	0	414	695	1600	0	4046	0	414	696	935	381	5269	3440	414	852
catmix100	501	0	0	0	0	0	600	0	0	0	0	0	400	6	0	0	0	0
catmix200	1001	0	0	0	0	0	1200	0	0	0	0	0	800	6	0	0	0	0
catmix400	2001	0	0	0	0	0	2400	0	0	0	0	0	1600	6	0	0	0	0
catmix800	4001	0	0	0	0	0	4800	0	0	0	0	0	3200	6	0	0	0	0
chain100	507	32	15186	0	68	302	800	0	18932	0	68	302	382	132	21156	11127	68	302
chain200	1010	92	25427	0	155	603	1600	0	30929	0	155	603	682	261	42156	22309	155	603
chain50	263	22	8439	7	51	154	400	0	9836	0	51	154	215	48	20961	9319	51	154
chance	9	0	15	0	8	8	10	0	15	0	8	8	9	4	29	14	8	8
chem	53	6	237	5	13	39	62	0	230	0	13	39	46	15	307	111	13	40
elec200	20995	262	143948	12073	0	0	41000	0	167275	0	0	0	77	82	38811	38811	0	0
elec25	438	34	5647	408	0	0	750	0	4838	0	0	0	23	29	804	804	0	0
elec50	1487	95	20714	3756	0	0	2750	0	20525	0	0	0	58	64	2589	2589	0	0
etamac	56	3	186	9	8	17	88	0	247	0	8	17	41	11	250	134	8	16
ex14_1_1	4	0	30	0	4	4	4	0	30	0	4	4	4	4	49	31	4	4
ex14_1_2	10	0	85	0	4	7	10	0	85	0	4	7	9	6	150	47	4	7
ex14_1_3	4	0	25	0	2	4	4	0	25	0	2	4	4	4	29	8	2	4
ex14_1_5	21	2	42	0	16	1	22	0	42	0	16	1	19	8	45	27	16	1
ex14_1_6	14	0	32	0	4	10	14	0	32	0	4	10	9	6	48	34	4	9
ex14_1_7	76	23	743	29	2	17	84	0	591	0	2	14	64	56	746	421	2	16
ex14_1_8	12	0	41	0	4	9	12	0	41	0	4	9	12	8	56	24	4	9
ex14_1_9	2	0	5	0	0	1	6	0	6	0	0	2	1	4	5	5	0	1
ex14_2_4	82	0	45	0	6	7	86	0	45	0	6	7	81	6	112	62	6	7
ex14_2_7	89	0	47	0	0	4	90	0	56	0	0	4	85	3	48	8	0	4
ex14_2_9	23	0	74	0	13	18	28	0	74	0	13	18	19	5	102	42	13	18
ex2_1_10	22	0	37	0	21	22	40	0	37	0	21	22	22	4	50	12	21	22
ex2_1_1	6	0	5	0	0	1	10	0	5	0	0	1	2	3	6	4	0	1
ex2_1_7	26	0	249	0	20	20	40	0	244	0	20	20	20	5	253	28	20	20
ex2_1_9	19	0	67	0	10	10	20	0	67	0	10	10	20	2	60	0	10	10
ex3_1_1	13	0	35	0	7	3	16	0	35	0	7	3	11	3	38	16	7	3

Table 4 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
ex3_1_2	7	0	15	0	7	7	10	0	15	0	7	7	7	2	21	4	7	7
ex3_1_4	3	0	5	0	0	1	6	0	5	0	0	1	1	3	6	4	0	1
ex3pb	8	0	52	0	0	2	10	0	57	0	0	4	7	5	81	45	0	1
ex4_1_1	2	0	16	0	1	2	2	0	16	0	1	2	2	3	20	20	1	2
ex4_1_2	1	0	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0
ex4_1_3	2	0	8	0	1	1	2	0	8	0	1	1	1	2	19	19	1	1
ex4_1_4	2	0	16	0	0	2	2	0	16	0	0	2	2	4	17	17	0	2
ex4_1_5	3	0	9	0	0	0	4	0	9	0	0	0	2	3	9	1	0	0
ex4_1_6	2	0	11	0	1	1	2	0	11	0	1	1	1	2	7	7	1	1
ex4_1_7	2	0	20	0	2	2	2	0	20	0	2	2	2	2	28	16	2	2
ex4_1_9	2	0	5	0	0	2	2	0	5	0	0	2	2	3	4	4	0	2
ex5_2_2_case1	5	0	14	0	3	3	6	0	14	0	3	3	5	5	26	14	3	4
ex5_2_2_case2	5	0	12	0	1	4	6	0	12	0	1	4	4	4	23	12	1	4
ex5_2_2_case3	5	0	14	0	3	3	6	0	14	0	3	3	5	5	26	14	3	5
ex5_2_4	9	0	32	0	0	1	10	0	32	0	0	1	5	8	42	28	0	3
ex5_2_5	27	7	159	4	0	7	54	0	146	0	0	15	13	19	76	66	0	10
ex5_3_2	13	3	37	2	2	6	16	0	35	0	2	6	10	12	37	33	2	6
ex5_3_3	57	10	324	9	1	29	98	0	392	0	1	44	51	45	411	335	1	27
ex5_4_2	12	0	46	0	6	7	16	0	46	0	6	7	10	3	56	16	6	8
ex5_4_3	14	1	60	1	8	6	20	0	64	0	8	6	11	7	80	52	8	5
ex5_4_4	24	3	148	5	6	4	36	0	131	0	6	6	23	18	222	153	6	9
ex6_1_1	19	4	40	0	4	19	28	0	40	0	4	20	20	15	44	25	4	18
ex6_1_2	9	0	11	0	2	7	10	0	11	0	2	7	8	5	25	11	2	7
ex6_1_4	13	0	65	0	3	10	18	0	65	0	3	10	16	2	102	38	3	9
ex6_2_10	90	28	1173	7	9	37	106	0	1173	0	9	43	77	58	148963	148213	9	45
ex6_2_11	37	1	179	0	6	19	56	0	179	0	6	23	23	18	232	101	6	21
ex6_2_12	42	15	154	0	0	31	60	0	154	0	0	31	35	36	163	105	0	27
ex6_2_13	62	13	231	15	0	28	82	0	246	0	0	29	31	30	168	133	0	27
ex6_2_5	64	38	491	54	0	45	90	0	525	0	0	48	61	55	739	421	0	46
ex6_2_6	27	0	195	0	19	22	36	0	118	0	19	23	27	13	176	84	19	27
ex6_2_7	73	11	1054	21	0	31	96	0	1042	0	0	30	50	23	852	381	0	27
ex6_2_8	28	0	197	0	20	25	32	0	197	0	20	25	30	13	189	66	20	26
ex6_2_9	68	24	660	28	6	33	76	0	476	0	6	40	53	42	495	257	6	38
ex7_2_2	10	0	49	0	3	7	12	0	49	0	3	7	9	3	73	26	3	7
ex7_2_3	27	5	233	1	4	16	36	0	293	0	4	17	22	25	249	161	4	12
ex7_2_4	19	1	169	0	5	11	24	0	169	0	5	11	13	9	265	110	5	11
ex7_3_1	19	0	31	0	10	10	22	0	32	0	10	10	19	2	32	12	10	10
ex7_3_2	10	0	8	0	7	7	10	0	8	0	7	7	11	6	13	7	7	8
ex7_3_3	5	0	4	0	4	4	6	0	5	0	4	4	6	7	6	3	4	6
ex7_3_4	37	2	24	1	15	15	40	0	24	0	15	15	42	15	39	25	15	20
ex7_3_5	36	3	49	2	4	4	46	0	58	0	4	4	36	14	75	31	4	6
ex7_3_6	3	0	3	0	3	3	4	0	3	0	3	3	3	4	11	8	3	3
ex8_1_3	15	0	29	0	0	0	16	0	29	0	0	0	12	5	29	2	0	0
ex8_1_4	4	0	1	0	0	0	4	0	1	0	0	0	4	2	1	0	0	0
ex8_1_5	4	0	0	0	0	0	4	0	0	0	0	0	4	2	0	0	0	0
ex8_1_6	10	0	40	0	0	0	10	0	40	0	0	0	9	2	19	2	0	0
ex8_1_7	10	0	48	0	2	2	12	0	57	0	2	2	3	6	89	54	2	2
ex8_2_1b	74	0	246	0	21	26	114	0	240	0	21	26	33	8	271	110	21	26
ex8_2_2b	6074	93	30636	93	40	40	15044	0	30889	0	40	40	44	15	32262	9112	40	40
ex8_2_3b	13482	0	37854	0	77	85	31282	0	37353	0	77	85	86	51	31186	9403	77	85
ex8_2_4b	82	0	193	0	35	38	122	0	191	0	35	38	41	9	202	84	35	38
ex8_2_5b	2080	3	17797	3	52	54	5068	0	17752	0	52	54	59	15	19166	4765	52	54
ex8_3_11	129	8	261	9	7	40	280	0	328	0	7	42	47	44	224	188	7	41
ex8_3_12	98	6	231	6	7	38	210	0	237	0	7	42	35	38	168	154	7	33
ex8_3_13	149	26	314	20	7	65	290	0	372	0	7	80	94	61	391	160	7	70

Table 4 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
ex8_3_14	183	26	583	25	7	52	330	0	536	0	7	64	103	69	389	367	7	47
ex8_3_1	127	10	237	9	7	38	280	0	311	0	7	47	65	34	233	130	7	46
ex8_3_5	88	4	233	5	7	33	180	0	204	0	7	36	37	37	180	170	7	32
ex8_3_7	123	23	259	30	7	48	214	0	253	0	7	48	63	67	241	221	7	44
ex8_3_8	100	4	339	9	7	37	210	0	329	0	7	47	44	26	303	176	7	32
ex8_3_9	70	3	166	1	7	38	136	0	162	0	7	36	39	40	109	92	7	35
ex8_4_1	22	0	205	0	6	7	42	0	240	0	6	7	12	12	364	207	6	8
ex8_4_2	54	25	472	28	1	31	86	0	468	0	1	31	45	47	545	519	1	31
ex8_4_4	43	2	327	2	17	22	48	0	377	0	17	22	33	9	384	88	17	23
ex8_4_5	25	1	183	0	3	3	30	0	183	0	3	3	6	12	382	280	3	4
ex8_4_6	57	3	113	2	1	3	92	0	209	0	1	4	24	17	136	134	1	3
ex8_4_7	110	10	5196	4	48	93	184	0	6344	0	48	94	110	31	7430	1324	48	100
ex8_4_8_bnd	202	31	5245	8	71	159	274	0	4929	0	71	177	202	64	5143	1187	71	167
ex8_5_1	22	4	38	0	3	8	24	0	38	0	3	8	20	11	51	28	3	8
ex8_5_2	25	1	72	0	8	8	26	0	72	0	8	8	22	5	104	29	8	6
ex8_5_4	15	1	67	1	2	6	18	0	67	0	2	6	13	12	69	44	2	6
ex8_5_5	18	2	63	2	3	4	20	0	63	0	3	4	15	11	72	32	3	5
ex8_5_6	28	3	17	1	11	13	32	0	16	0	11	13	26	9	42	19	11	13
ex8_6_1	160	3	57	7	8	8	228	0	59	0	8	8	95	6	238	224	8	8
ex9_2_2	3	0	5	0	2	3	4	0	5	0	2	3	3	4	10	5	2	3
ex9_2_4	4	0	16	0	0	0	4	0	16	0	0	0	0	4	18	18	0	0
ex9_2_5	4	0	18	0	1	3	4	0	18	0	1	3	4	6	21	17	1	3
ex9_2_6	6	0	20	0	0	2	8	0	20	0	0	2	2	6	20	20	0	2
ex9_2_7	3	0	11	0	2	3	4	0	11	0	2	3	3	3	10	5	2	3
glider100	3394	598	3318	199	1	701	4198	0	3884	0	1	701	3094	901	4542	2989	1	601
glider200	6794	1198	6777	399	1	1401	8398	0	7693	0	1	1401	6194	1801	9066	5986	1	1201
glider400	13594	2398	12991	799	1	2801	16798	0	15201	0	1	2801	12394	3601	18776	11980	1	2401
glider50	1694	298	1713	99	1	351	2098	0	1906	0	1	351	1544	451	2301	1489	1	301
gsg_0001	100	23	854	52	50	90	120	0	1119	0	50	90	92	28	1105	313	50	90
hhfair	28	3	35	2	6	10	42	0	42	0	6	10	24	12	35	20	6	10
himmel11	7	0	11	0	7	7	10	0	11	0	7	7	7	2	19	4	7	7
himmel16	18	0	91	0	0	0	18	0	91	0	0	0	6	8	76	62	0	0
house	8	0	9	0	6	8	10	0	9	0	6	8	8	8	14	5	6	8
hs62	9	0	22	0	8	9	12	0	31	0	8	9	9	4	40	18	8	9
hybriddynamic_fixedcc	46	0	70	0	26	27	64	0	74	0	26	29	43	7	83	18	26	27
hybriddynamic_varcc	107	3	1353	2	37	40	146	0	1368	0	37	40	84	21	1795	550	37	40
infeas1	2847	31	191666	18	2631	2710	2948	0	192980	0	2631	2710	2869	56	108394	12208	2631	2708
jbearing100	5001	0	2500	0	0	0	10000	0	2500	0	0	0	5000	5	2501	1	0	0
jbearing25	1251	0	625	0	0	0	2500	0	625	0	0	0	1250	5	626	1	0	0
jbearing50	2501	0	1250	0	0	0	5000	0	1250	0	0	0	2500	5	1251	1	0	0
jbearing75	3751	0	1875	0	0	0	7500	0	1875	0	0	0	3750	5	1876	1	0	0
kall_circles_c6a	25	3	94	2	7	9	32	0	110	0	7	13	16	15	68	68	7	14
kall_circles_c6b	25	3	107	2	7	10	32	0	99	0	7	9	18	18	61	61	7	16
kall_circles_c6c	27	1	98	0	8	11	36	0	95	0	8	11	19	15	58	58	8	17
kall_circles_c7a	25	1	109	1	3	10	36	0	126	0	3	10	17	7	188	72	3	12
kall_circles_c8a	29	3	151	7	9	11	40	0	162	0	9	10	22	9	161	75	9	14
kall_circlespolygons_c1p12	28	4	164	4	2	2	30	0	173	0	2	2	24	24	168	118	2	2
kall_circlespolygons_c1p13	27	3	163	2	2	2	30	0	173	0	2	2	20	21	183	134	2	2
kall_circlespolygons_c1p5a	100	39	700	59	1	1	124	0	741	0	1	2	76	71	893	607	1	1
kall_circlespolygons_c1p5b	587	197	5343	244	0	1	694	0	6254	0	0	1	449	418	5121	3353	0	1
kall_circlespolygons_c1p6a	876	307	8928	428	0	1	988	0	9905	0	0	1	677	645	7785	5393	0	1
kall_circlesrectangles_c1r12	33	4	129	4	2	2	34	0	120	0	2	3	23	22	147	111	2	2
kall_circlesrectangles_c1r13	31	4	133	3	0	2	34	0	157	0	0	3	21	24	186	133	0	0
kall_circlesrectangles_c6r1	154	42	1016	51	0	4	168	0	1151	0	0	3	119	110	942	707	0	3
kall_circlesrectangles_c6r29	313	77	2181	78	0	5	344	0	2625	0	0	4	257	228	2316	1712	0	3

Table 4 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
kall_circlesrectangles_c6r39	512	142	3781	152	0	6	556	0	4353	0	0	2	438	404	4062	2924	0	5
kall_congruentcircles_c31	11	2	26	2	7	8	16	0	27	0	7	8	8	4	15	12	7	7
kall_congruentcircles_c32	11	1	39	1	1	2	16	0	40	0	1	5	6	5	44	19	1	4
kall_congruentcircles_c41	7	0	9	0	3	1	8	0	9	0	3	1	6	6	12	7	3	3
kall_congruentcircles_c42	14	4	42	2	2	8	20	0	42	0	2	6	11	10	60	33	2	8
kall_congruentcircles_c51	19	5	47	5	5	8	24	0	61	0	5	7	20	21	88	59	5	14
kall_congruentcircles_c52	15	4	53	4	3	3	24	0	58	0	3	4	13	14	68	41	3	8
kall_congruentcircles_c61	22	5	64	5	6	10	28	0	60	0	6	9	16	14	50	40	6	15
kall_congruentcircles_c62	15	4	73	6	4	2	28	0	73	0	4	6	15	15	105	62	4	9
kall_congruentcircles_c63	20	2	86	1	9	9	28	0	80	0	9	9	18	10	88	32	9	13
kall_congruentcircles_c71	26	5	109	5	0	7	32	0	132	0	0	9	17	12	149	51	0	14
kall_congruentcircles_c72	21	4	105	8	5	4	32	0	100	0	5	4	17	16	96	54	5	11
kall_diffcircles_10	29	2	78	3	1	4	38	0	85	0	1	5	9	6	93	55	1	3
kall_diffcircles_5a	17	2	46	2	5	12	24	0	48	0	5	10	13	10	65	42	5	11
kall_diffcircles_5b	18	0	49	0	6	9	24	0	43	0	6	9	10	8	44	36	6	9
kall_diffcircles_6	18	2	35	3	2	4	28	0	45	0	2	6	5	9	41	36	2	4
kall_diffcircles_7	20	3	54	3	2	6	32	0	61	0	2	8	7	11	54	50	2	7
kall_diffcircles_8	21	0	73	0	8	8	30	0	75	0	8	8	11	4	58	45	8	8
kall_diffcircles_9	24	2	67	2	1	3	34	0	66	0	1	4	11	8	82	46	1	6
knp3-12	64	30	426	174	0	0	72	0	416	0	0	0	1	7	250	250	0	0
knp4-24	187	1	2548	1	0	0	192	0	2548	0	0	0	7	13	827	827	0	0
knp5-40	395	1	7223	1	0	0	400	0	7223	0	0	0	116	121	4065	4065	0	0
knp5-41	401	6	7873	148	0	0	410	0	8456	0	0	0	38	45	3228	3228	0	0
knp5-42	412	1	9755	3	0	0	420	0	9755	0	0	0	25	31	3466	3466	0	0
knp5-43	424	0	9888	0	0	0	430	0	9552	0	0	0	21	28	3162	3162	0	0
knp5-44	434	0	12542	0	0	0	440	0	12542	0	0	0	49	56	4503	4503	0	0
launch	85	1	170	2	49	62	140	0	161	0	49	74	69	15	157	81	49	59
least	21	3	58	0	0	6	28	0	60	0	0	6	10	12	55	46	0	6
like	697	32	661	31	0	1	790	0	1051	0	0	91	667	37	716	92	0	1
linear	30	0	75	0	10	16	40	0	75	0	10	16	27	4	71	24	10	16
mathopt5_7	2	0	8	0	0	1	2	0	8	0	0	1	1	2	15	15	0	1
mathopt5_8	2	0	7	0	0	1	2	0	7	0	0	1	1	2	18	18	0	1
maxmin	148	39	709	95	66	10	204	0	653	0	66	8	180	102	1370	957	66	11
methanol100	7590	0	199	0	0	0	7594	0	199	0	0	0	7589	5	199	0	0	0
methanol50	3966	0	224	0	0	0	3970	0	224	0	0	0	3965	5	190	0	0	0
minlphi	18	5	44	1	7	5	20	0	53	0	7	5	18	9	73	33	7	5
oaer	4	0	13	0	2	4	4	0	13	0	2	4	3	2	16	10	2	3
pindyck	174	26	2862	0	112	124	190	0	2862	0	112	124	174	34	3178	416	112	124
pinene50	1226	0	1233	0	0	0	1230	0	1233	0	0	0	1225	3	1233	0	0	0
pointpack04	14	0	27	0	2	4	16	0	36	0	2	4	8	5	29	18	2	3
pointpack06	21	3	38	1	1	8	24	0	65	0	1	10	7	10	67	55	1	4
pointpack08	25	6	61	3	0	9	32	0	62	0	0	9	8	14	52	52	0	7
pointpack10	31	13	62	9	0	12	40	0	61	0	0	13	11	17	85	85	0	9
pointpack12	40	19	97	16	0	13	48	0	79	0	0	17	15	21	81	81	0	13
pointpack14	45	22	107	23	0	19	56	0	116	0	0	17	17	23	93	93	0	16
portfol_buyin	13	1	73	2	7	10	18	0	59	0	7	11	12	10	140	67	7	12
powerflow0009r	86	3	315	3	0	2	94	0	310	0	0	2	38	14	472	329	0	2
powerflow0014r	148	14	1198	6	1	19	206	0	1505	0	1	23	55	42	1658	1278	1	18
powerflow0030r	291	2	1334	3	0	7	396	0	1376	0	0	6	90	17	1578	1154	0	7
powerflow0039r	441	2	1720	1	0	4	480	0	1740	0	0	4	151	28	2191	1760	0	4
powerflow0057r	522	36	4962	11	0	40	736	0	5748	0	0	46	142	116	7759	6756	0	37
powerflow0118r	1277	54	17492	21	0	105	1826	0	17656	0	0	114	352	228	27827	23735	0	92
powerflow0300r	3101	166	39841	47	0	257	4146	0	41902	0	0	278	1085	463	57435	41895	0	229
powerflow2383wpr	23076	280	74025	168	0	538	26676	0	77489	0	0	539	4481	625	79578	65937	0	493
powerflow2736spr	27638	553	70238	283	0	911	31780	0	71843	0	0	920	8540	927	89057	72114	0	855

Table 4 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
prob06	3	0	6	0	2	2	4	0	6	0	2	2	2	2	7	6	2	2
process	16	1	101	0	8	10	16	0	101	0	8	10	15	5	131	45	8	10
procsyn	32	0	88	0	1	0	40	0	92	0	1	0	12	7	68	48	1	0
prolog	9	0	50	0	4	4	12	0	52	0	4	4	6	5	86	45	4	4
qp3	145	1	258	0	114	131	200	0	254	0	114	129	147	8	251	13	114	133
rocket100	1114	108	2782	192	1	104	1800	0	2637	0	1	107	1133	640	3272	2507	1	395
rocket200	2443	439	4550	205	1	432	3600	0	4826	0	1	323	2249	1256	6527	5036	1	790
rocket400	4748	745	10075	948	1	714	7200	0	9231	0	1	658	4516	2523	12995	10186	1	1586
rocket50	560	54	1359	55	1	50	900	0	1381	0	1	50	550	307	1617	1242	1	195
shiporig	21	0	19	0	0	0	30	0	17	0	0	0	20	5	17	0	0	0
st_bpaf1a	14	0	19	0	13	13	20	0	22	0	13	13	14	2	24	4	13	13
st_bsj2	5	0	9	0	4	4	6	0	9	0	4	4	5	2	13	4	4	4
st_bsj4	7	0	6	0	6	6	12	0	6	0	6	6	6	4	9	5	6	6
st_e03	16	2	126	1	8	10	16	0	126	0	8	10	15	5	170	58	8	10
st_e04	7	1	83	1	2	5	10	0	83	0	2	5	6	9	120	70	2	5
st_e05	8	2	10	0	5	5	8	0	10	0	5	5	8	6	16	7	5	5
st_e07	5	0	17	0	1	3	6	0	17	0	1	3	4	4	25	11	1	2
st_e08	4	0	7	0	4	4	4	0	7	0	4	4	4	2	9	1	4	4
st_e09	4	0	2	0	4	4	4	0	2	0	4	4	4	4	4	2	4	4
st_e11	4	1	2	0	0	0	4	0	2	0	0	0	4	3	3	1	0	0
st_e16	16	0	72	0	9	6	20	0	59	0	9	6	14	6	95	36	9	6
st_e19	4	0	17	0	0	0	4	0	17	0	0	0	0	4	17	17	0	0
st_e22	4	0	3	0	3	3	4	0	3	0	3	3	4	2	5	2	3	3
st_e24	3	0	2	0	3	3	4	0	4	0	3	3	3	4	5	4	3	3
st_e25	8	0	18	0	6	2	8	0	18	0	6	2	7	4	18	8	6	2
st_e26	4	1	2	0	2	4	4	0	2	0	2	4	4	5	2	2	2	4
st_e28	7	0	11	0	7	7	10	0	11	0	7	7	7	2	19	4	7	7
st_e30	7	0	29	0	2	2	10	0	32	0	2	2	5	4	38	21	2	2
st_e33	5	0	14	0	1	3	6	0	14	0	1	3	4	4	27	11	1	3
st_e41	17	1	52	0	11	15	24	0	52	0	11	18	16	10	69	38	11	15
st_fp7a	28	0	235	0	20	20	40	0	236	0	20	20	20	5	253	24	20	20
st_fp7b	24	0	224	0	20	20	40	0	227	0	20	20	20	5	245	24	20	20
st_fp7c	28	0	235	0	20	20	40	0	236	0	20	20	20	5	242	26	20	20
st_fp7d	28	0	235	0	20	20	40	0	236	0	20	20	20	5	249	24	20	20
st_fp7e	26	0	249	0	20	20	40	0	244	0	20	20	20	5	253	28	20	20
st_glmp_fp1	4	0	5	0	4	2	4	0	5	0	4	2	4	2	10	4	4	2
st_glmp_fp2	3	0	12	0	3	2	4	0	14	0	3	2	3	4	16	6	3	2
st_glmp_fp3	4	0	13	0	3	0	4	0	13	0	3	0	4	4	23	13	3	0
st_glmp_kk90	3	0	5	0	3	3	4	0	5	0	3	3	3	4	8	4	3	3
st_glmp_kk92	4	0	18	0	2	3	4	0	18	0	2	3	4	2	22	6	2	3
st_glmp_kky	3	0	6	0	3	3	4	0	8	0	3	3	3	2	8	4	3	3
st_glmp_ss1	3	0	7	0	2	2	4	0	11	0	2	2	2	2	7	5	2	2
st_glmp_ss2	3	0	5	0	3	3	4	0	5	0	3	3	3	2	6	3	3	3
st_iqpbk1	15	0	99	0	2	6	16	0	99	0	2	6	7	7	71	36	2	7
st_iqpbk2	16	0	66	0	2	7	16	0	66	0	2	7	7	5	75	41	2	7
st_jcbpaf2	16	1	57	1	12	12	20	0	53	0	12	12	16	6	79	30	12	12
st_pan1	5	0	6	0	1	4	6	0	6	0	1	4	4	4	12	5	1	4
st_ph14	4	0	4	0	4	4	6	0	4	0	4	4	4	2	6	2	4	4
st_ph15	7	0	13	0	4	4	8	0	13	0	4	4	6	2	19	5	4	4
st_ph1	7	0	10	0	4	6	12	0	10	0	4	7	6	4	13	7	4	6
st_ph20	3	0	7	0	3	2	4	0	7	0	3	2	3	4	19	11	3	2
st_ph2	7	0	10	0	4	6	12	0	10	0	4	7	6	3	16	7	4	6
st_ph3	5	0	7	0	5	5	8	0	7	0	5	6	5	2	10	3	5	5
st_phex	4	0	4	0	4	2	4	0	4	0	4	2	4	2	10	6	4	2
st_qpc-m0	3	0	3	0	2	0	4	0	3	0	2	0	3	7	3	0	2	1

Table 4 continued

	greedy (base)						greedy+filt. off						greedy+filt. aggr.					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
st_qpc-m1	9	0	33	0	6	6	10	0	33	0	6	6	8	2	32	7	6	6
st_qpc-m3a	13	0	98	0	1	6	20	0	101	0	1	6	10	4	84	26	1	10
st_qpc-m3b	17	6	148	36	7	9	20	0	128	0	7	9	10	3	160	88	7	9
st_qpk1	3	0	4	0	2	0	4	0	4	0	2	0	4	8	5	1	2	0
st_qpk2	10	1	27	0	0	6	12	0	19	0	0	6	6	3	44	13	0	6
st_qpk3	21	1	80	0	0	11	22	0	88	0	0	11	11	3	115	23	0	11
st_rv1	12	0	25	0	10	12	20	0	23	0	10	12	12	5	35	12	10	12
st_rv2	24	0	130	0	21	21	40	0	127	0	21	21	23	4	152	19	21	21
st_rv3	28	1	174	0	20	23	40	0	174	0	20	23	26	5	226	40	20	23
st_rv7	37	0	294	0	29	32	60	0	291	0	29	32	35	5	310	24	29	32
st_rv8	47	0	637	0	39	43	80	0	631	0	39	43	44	6	719	63	39	43
st_rv9	58	0	632	0	42	50	100	0	641	0	42	50	51	4	663	48	42	49
st_z	6	0	10	0	5	4	6	0	10	0	5	4	5	2	11	0	5	4
supplychainp1_020306	15	0	262	0	7	8	18	0	262	0	7	8	13	12	358	243	7	12
syn05m	4	0	27	0	3	4	6	0	27	0	3	4	4	2	44	17	3	4
tricp	71	0	108	0	20	20	108	0	115	0	20	20	20	7	125	59	20	20
wall	16	0	1	0	0	0	16	0	1	0	0	0	16	2	1	0	0	0
wastewater02m1	13	0	67	0	9	11	20	0	74	0	9	12	13	5	54	17	9	12
wastewater02m2	15	3	95	0	1	15	20	0	91	0	1	16	15	15	148	107	1	15
wastewater04m1	18	4	109	0	6	15	24	0	94	0	6	16	20	12	116	46	6	18
wastewater04m2	23	5	97	2	10	18	24	0	95	0	10	18	30	19	195	82	10	20
wastewater05m1	26	2	199	0	3	13	48	0	169	0	3	15	24	16	239	106	3	16
wastewater05m2	43	9	1089	27	6	30	48	0	1152	0	6	33	45	42	1507	960	6	37
wastewater11m1	75	3	1750	5	58	65	140	0	1873	0	58	69	69	12	1483	181	58	67
wastewater11m2	89	16	3081	6	6	66	140	0	3138	0	6	77	90	90	2125	1264	6	81
wastewater12m1	133	3	6463	4	112	117	260	0	6684	0	112	126	124	10	6945	577	112	120
wastewater12m2	154	21	6559	14	10	131	260	0	5345	0	10	129	149	139	5015	2544	10	134
wastewater13m1	276	4	12608	2	246	258	540	0	14539	0	246	266	265	16	8607	1171	246	262
wastewater13m2	282	10	20279	12	13	228	540	0	26621	0	13	244	301	253	15320	8019	13	285
wastewater14m1	51	3	971	0	30	42	90	0	917	0	30	44	44	13	1091	267	30	44
wastewater14m2	51	6	3228	7	13	40	90	0	2181	0	13	69	52	34	2784	810	13	49
wastewater15m1	28	3	381	1	12	22	48	0	443	0	12	26	22	12	372	131	12	22
wastewater15m2	37	6	1232	7	13	33	48	0	1261	0	13	43	38	23	1574	550	13	36
waterund01	41	10	495	21	14	18	54	0	503	0	14	19	29	21	666	393	14	23
waterund08	98	21	2877	37	34	50	124	0	2794	0	34	49	59	26	3009	830	34	53
waterund11	54	9	904	8	20	27	78	0	803	0	20	30	36	21	817	430	20	28
waterund14	140	19	3434	25	42	67	160	0	3844	0	41	64	83	45	3986	1257	41	68
waterund17	78	10	1601	22	25	42	96	0	1504	0	25	41	57	34	1812	619	25	47
waterund18	62	12	876	12	21	31	78	0	884	0	21	32	42	22	976	390	21	34
waterund22	128	32	2698	64	32	49	160	0	2737	0	32	52	71	33	2091	742	32	51
waterund25	103	15	4021	23	34	47	140	0	4016	0	34	48	70	26	4886	1163	34	50
waterund27	321	91	61193	198	105	148	400	0	57524	0	105	144	205	64	49595	5702	105	148
waterund28	950	57	923371	48	254	488	1320	0	906130	0	255	524	658	194	768548	32680	254	489
waterund32	741	94	150635	159	180	360	1160	0	151030	0	180	381	567	107	91337	15635	182	377
waterund36	334	84	54394	252	101	133	432	0	52570	0	101	133	193	53	54552	8883	101	143
weapons	182	37	2143	38	0	69	310	0	2177	0	0	78	157	57	3017	1032	0	78

Table 5: Detailed results comparing ordering strategies at the root node on test set INT. See Table 1 for aggregated results.

lp — number of LPs solved during OBBT
lp_{filt} — number of LPs solved during OBBT's filtering
iter — number of LP iterations during OBBT
$\text{iter}_{\text{filt}}$ — number of LP iterations of OBBT's filtering LPs
b_{obbt} — number of bounds tightened by OBBT
lvb — number of LVBs found by OBBT

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
4stufen	54	9	1130	12	8	28	49	11	1018	7	8	30	54	9	649	3	8	31	52	14	1244	9	8	30
alan	5	0	39	0	5	5	5	0	23	0	5	5	6	0	33	0	5	5	6	0	33	0	5	5
batch0812_nc	77	14	1310	5	40	55	78	13	1535	6	40	55	76	9	1723	1	40	54	79	20	1430	11	40	55
batch0812	49	4	964	3	34	43	51	3	1050	4	34	43	51	2	672	0	34	44	49	5	1020	4	34	42
batchdes	15	3	161	1	9	12	15	2	146	0	9	12	15	0	104	0	9	11	15	3	224	1	9	12
batch_nc	54	14	2145	4	20	39	57	11	2417	3	20	39	56	10	1947	3	20	36	53	14	2031	3	20	38
batch	34	1	791	0	16	24	33	2	787	0	16	23	36	0	638	0	16	27	33	1	1155	0	16	23
batchs101006m	68	4	2116	2	19	24	74	2	1962	0	19	27	86	1	1710	1	19	32	71	7	2306	0	19	30
batchs121208m	96	11	9279	0	23	39	98	9	9415	1	23	41	105	0	7776	0	23	39	96	14	10556	1	23	35
batchs151208m	89	12	3306	1	23	42	90	10	3030	0	23	46	108	0	3504	0	23	48	94	19	3724	0	23	45
batchs201210m	96	14	4703	0	23	48	93	13	3934	2	23	51	119	2	4635	0	23	47	87	14	5128	6	23	45
bchoco05	80	17	2586	50	37	75	80	14	2712	39	37	74	84	6	1848	12	37	80	83	19	3136	53	37	77
bchoco06	143	59	8138	245	6	120	144	63	7499	401	6	118	142	42	6622	90	6	115	138	59	6667	130	6	104
blend029	33	6	377	16	2	16	38	8	509	17	2	19	36	5	469	4	2	22	35	7	520	20	2	15
blend146	93	26	2119	150	0	45	91	23	2851	127	0	53	108	13	3729	53	0	54	98	27	3508	183	0	45
blend480	130	36	4294	56	0	83	128	31	3553	37	0	87	142	26	5335	78	0	88	135	34	4444	33	0	68
blend531	104	23	2217	56	0	65	105	23	2365	33	0	59	118	15	2311	48	0	57	110	29	2843	43	0	57
blend718	86	15	2101	61	1	49	80	12	2012	50	1	46	90	6	2087	20	1	43	91	21	2355	61	1	39
blend721	95	19	2709	20	0	57	94	25	2347	27	0	56	104	17	2903	95	0	58	104	31	2794	52	0	47
blend852	129	33	3203	42	0	71	125	33	2711	47	0	62	145	14	3981	55	0	72	130	29	3204	52	0	70
carton7	40	11	1414	64	4	12	38	9	1740	50	4	15	39	1	1644	2	4	16	40	13	1888	69	4	13
carton9	57	18	2317	24	11	18	60	21	1927	23	11	19	60	7	1945	19	11	21	62	24	2870	71	11	18
casctanks	431	101	49363	75	185	347	439	114	42794	74	185	357	452	53	22032	22	185	352	489	102	53955	67	185	354
cecil_13	101	11	21454	0	43	78	109	15	9790	3	43	79	108	2	3864	2	43	79	93	15	18557	1	43	77
chp_partload	1487	384	357337	169	399	982	1488	380	338716	208	399	984	1561	297	146966	137	399	995	1501	408	423902	218	399	998
clay0203h	258	148	9376	133	0	229	262	140	12132	88	0	227	264	140	3895	100	0	235	268	145	18737	112	0	223
clay0203m	9	0	85	0	0	1	9	0	105	0	0	0	12	0	135	0	0	0	9	1	91	1	0	0
clay0204h	357	177	23796	110	0	319	353	176	19114	186	0	315	357	176	5427	121	0	321	353	174	26741	117	0	312
clay0204m	12	0	180	0	0	1	14	0	148	0	0	1	15	0	158	0	0	1	9	0	171	0	0	1
clay0205h	478	317	26540	205	0	359	490	317	20862	219	0	397	489	267	16457	162	0	430	516	312	41173	149	0	403
clay0205m	18	0	448	0	0	1	18	0	381	0	0	1	19	0	394	0	0	3	19	0	600	0	0	3
clay0303h	426	277	10015	246	0	313	439	288	9339	227	0	316	454	249	4629	16	0	332	424	264	10831	115	0	324
clay0303m	9	0	134	0	0	0	10	0	159	0	0	1	12	0	174	0	0	1	10	0	217	0	0	0
clay0304h	567	366	12543	128	0	422	548	336	12907	176	0	361	589	323	9242	82	0	425	566	341	16607	320	0	443
clay0304m	14	0	225	0	0	2	14	1	235	0	0	1	16	0	236	0	0	1	12	0	223	0	0	1
clay0305h	716	460	19102	197	0	548	712	464	17464	364	0	536	757	413	15207	269	0	558	722	463	18244	339	0	565
clay0305m	18	0	485	0	0	1	18	0	406	0	0	0	20	1	288	1	0	0	17	0	522	0	0	1
crudeoil_lee1_05	90	39	5967	159	34	48	86	37	5518	82	33	50	91	34	3787	70	32	50	92	42	6786	108	33	52
crudeoil_lee1_06	113	65	9892	175	22	59	115	67	9291	204	21	55	117	48	8147	150	21	59	107	57	11720	145	22	52
crudeoil_lee1_07	121	63	14835	141	25	73	132	76	12646	206	25	80	135	50	10008	116	25	72	131	74	15285	183	24	80
crudeoil_lee1_08	163	97	20275	297	28	87	154	87	20567	235	30	87	176	69	12957	140	28	95	162	93	21835	268	31	90
crudeoil_lee1_09	181	101	27229	318	36	97	181	98	26518	291	34	109	192	83	18964	201	36	104	180	102	30227	204	35	105
crudeoil_lee1_10	194	105	33570	472	44	109	193	113	31059	456	41	117	195	64	18502	161	42	115	192	104	32176	331	44	117
crudeoil_lee2_05	148	66	29494	101	39	86	148	68	27636	159	39	83	158	48	20051	86	39	86	143	65	30231	112	39	87
crudeoil_lee2_06	200	92	49423	195	55	119	211	102	51343	191	55	118	212	64	30629	130	55	120	201	96	52271	245	55	116
crudeoil_lee2_07	239	111	75143	577	64	137	241	109	81824	458	64	137	246	81	46873	261	64	139	236	112	81417	459	64	142
crudeoil_lee2_08	284	138	103947	473	74	176	280	129	102900	336	74	166	304	122	62279	373	74	172	292	149	106808	543	74	167

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
crudeoil_lee2_09	318	147	156297	452	88	199	314	142	163736	517	88	198	342	117	99020	303	88	210	325	159	163760	703	88	190
crudeoil_lee2_10	370	185	192113	775	97	212	365	181	184391	1038	97	221	378	123	110685	434	97	209	366	177	190908	909	97	218
crudeoil_lee3_05	287	126	24112	298	120	186	283	118	27882	279	120	195	289	91	18775	172	120	181	289	118	28354	252	120	195
crudeoil_lee3_06	384	161	90551	401	173	253	392	169	86420	401	173	261	385	125	61367	385	173	256	375	154	92173	340	173	258
crudeoil_lee3_07	470	180	103216	436	226	338	464	179	103795	463	226	325	473	146	65455	388	226	340	472	193	93411	646	226	333
crudeoil_lee3_08	549	185	239891	602	287	384	535	187	225806	522	287	383	556	157	154527	420	287	400	543	201	226734	599	287	363
crudeoil_lee3_09	633	229	309283	763	341	466	639	236	300295	575	341	463	653	177	199420	465	341	472	650	248	307962	1039	341	458
crudeoil_lee3_10	729	265	418387	939	397	512	715	250	423729	949	397	520	743	212	264413	897	397	550	731	272	413710	1175	397	524
crudeoil_lee4_05	188	62	42158	72	75	127	186	60	40577	89	75	126	194	49	22919	137	75	126	185	62	38900	125	75	136
crudeoil_lee4_06	240	79	83813	208	100	168	244	81	92723	199	100	171	247	55	54350	102	100	170	245	83	87453	231	100	173
crudeoil_lee4_07	296	106	108887	284	117	203	288	99	124704	249	117	192	311	85	74047	180	117	204	286	98	116459	279	117	199
crudeoil_lee4_08	332	123	150556	559	126	219	328	114	153321	330	126	222	351	82	95270	146	126	228	330	122	155091	347	126	208
crudeoil_lee4_09	401	159	209233	518	150	269	398	155	213011	563	150	272	417	102	132726	268	150	264	404	162	224760	622	150	271
crudeoil_lee4_10	408	140	252133	394	157	286	406	150	227715	428	157	287	430	107	157458	330	157	295	409	149	235760	501	157	281
crudeoil_li01	103	15	3404	20	36	36	100	8	3257	6	36	37	95	6	2553	8	36	32	98	16	4062	21	36	35
crudeoil_li02	31	7	2871	1	13	26	30	8	2636	2	13	25	32	6	1550	2	13	28	31	9	3064	3	13	26
crudeoil_li03	238	69	14974	130	21	30	242	56	14280	76	21	31	254	31	12160	13	21	33	226	85	14994	169	21	28
crudeoil_li05	202	35	8888	514	13	20	212	34	8826	206	13	20	236	22	7467	229	13	22	189	47	10114	234	13	21
crudeoil_li06	246	64	13636	72	14	23	250	64	11239	94	14	23	255	25	11183	16	14	25	233	75	13159	114	14	30
crudeoil_li11	240	59	16730	75	19	19	237	60	15701	57	19	25	257	26	14927	16	19	21	240	78	16610	176	19	21
crudeoil_li21	235	69	35065	3338	17	13	249	77	30150	91	17	16	261	31	31300	362	17	11	234	85	32682	1116	17	18
csched1a	14	1	86	0	3	7	15	1	123	0	3	7	16	1	104	0	3	7	11	1	109	1	3	7
csched1	15	2	155	1	1	6	14	1	149	1	1	5	15	1	130	1	1	4	10	0	181	0	1	5
csched2a	112	62	4445	31	10	61	107	59	6793	31	10	58	125	28	6817	12	10	70	111	66	6769	45	10	62
csched2	152	84	1563	71	1	64	151	79	1577	65	1	59	152	64	1306	55	1	63	162	93	1333	83	1	64
densitymod	315	0	191478	0	315	315	315	0	188197	0	315	315	316	0	184553	0	315	315	315	0	186328	0	315	315
du-opt5	12	0	229	0	4	5	14	1	231	1	4	5	17	0	240	0	4	5	9	1	232	1	4	5
du-opt	18	0	224	0	4	4	12	0	212	0	4	4	17	0	184	0	4	5	11	0	241	0	4	5
edgecross10-060	44	0	3113	0	0	0	42	0	3291	0	0	0	52	0	2082	0	0	0	54	0	7045	0	0	0
edgecross14-039	143	2	16262	2	0	0	126	14	11717	23	0	0	161	4	7492	10	0	0	151	3	23834	127	0	0
elf	12	3	209	7	0	6	12	1	120	3	0	4	11	2	286	5	0	5	11	3	259	8	0	5
eniplac	69	15	432	21	0	18	67	11	431	19	0	21	65	2	369	5	0	23	70	21	405	23	0	20
enpro48pb	36	0	1174	0	10	14	37	0	1282	0	10	14	42	0	1204	0	10	14	32	0	1441	0	10	14
enpro56pb	35	1	2052	1	8	16	35	1	2044	1	8	15	39	0	1667	0	8	17	30	1	2470	0	8	18
ethanolh	149	46	5997	46	5	74	157	56	3953	56	5	73	159	30	2729	28	5	73	145	54	6916	49	5	61
ethanolm	64	16	1130	14	7	34	74	16	906	10	7	36	81	10	700	9	7	37	66	21	1171	16	7	39
ex1224	8	0	90	0	4	7	8	0	128	0	4	6	8	0	67	0	4	6	8	1	71	0	4	6
ex1233	71	20	262	43	16	27	70	16	319	32	16	26	75	1	191	0	16	26	67	22	226	44	16	27
ex1243	43	5	81	6	0	5	42	3	58	5	0	4	42	4	85	4	0	5	42	3	70	7	0	4
ex1244	59	5	260	0	2	14	59	3	294	0	2	9	59	3	200	0	2	12	59	5	200	7	2	11
ex1252a	24	10	250	3	1	12	26	6	272	5	1	9	32	14	209	7	1	17	36	22	325	25	1	13
ex1252	36	22	274	18	3	18	30	16	326	11	3	16	32	17	500	20	3	16	41	30	471	32	3	21
ex1263a	21	1	135	1	12	14	22	2	97	1	12	17	22	1	112	1	12	16	22	2	103	1	12	16
ex1263	27	5	353	16	11	16	27	5	323	11	11	14	26	2	362	1	11	15	24	3	291	3	11	14
ex1264a	21	0	106	0	9	10	22	1	86	1	9	9	23	0	90	0	9	9	21	1	105	1	9	11
ex1264	21	0	220	0	8	13	22	0	232	0	8	13	24	1	198	0	8	13	22	1	228	0	8	15
ex1265a	33	2	282	2	12	17	32	1	249	1	12	15	40	0	231	0	12	16	33	3	171	3	12	16
ex1265	32	5	189	4	6	17	32	4	182	2	6	16	36	3	251	2	6	15	30	3	211	4	6	14
ex1266a	44	3	202	2	21	21	44	3	175	2	21	21	52	0	223	0	21	25	42	3	195	2	21	22
ex1266	44	4	617	17	19	26	42	1	721	7	19	29	55	1	770	1	19	26	41	3	621	8	19	26
ex4	8	1	205	0	4	5	9	1	155	0	4	5	8	0	238	0	4	5	7	1	256	0	4	5
fac1	22	9	101	10	0	10	19	10	87	13	0	8	23	11	70	14	0	8	18	13	75	13	0	7
fac2	41	14	168	24	0	23	45	16	172	20	0	31	48	21	156	25	0	37	45	16	172	20	0	31
fac3	59	17	413	14	0	29	60	24	388	29	0	36	69	17	338	16	0	36	61	25	371	24	0	37
feedtray	354	75	14256	32	241	112	359	64	12653	69	241	131	372	57	8258	71	241	115	336	75	22129	35	241	109

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
fin2bb	84	0	7436	0	61	61	84	0	4756	0	61	61	84	0	2975	0	61	61	84	0	3636	0	61	61
flay02h	3	0	68	0	2	2	3	0	74	0	2	2	3	0	74	0	2	2	4	0	82	0	2	2
flay02m	4	0	107	0	2	2	4	0	75	0	2	2	4	0	61	0	2	2	4	0	124	0	2	2
flay03h	6	0	316	0	3	4	6	0	300	0	3	4	6	0	213	0	3	4	6	0	357	0	3	4
flay03m	6	0	106	0	3	3	6	0	122	0	3	3	6	0	72	0	3	3	6	0	168	0	3	3
flay04h	8	0	277	0	4	1	8	0	254	0	4	1	8	0	234	0	4	1	8	0	317	0	4	1
flay04m	8	1	158	0	0	4	8	0	111	0	0	4	8	0	80	0	0	4	7	0	144	0	0	4
flay05h	10	0	1334	0	0	5	10	0	1366	0	0	6	10	0	1071	0	0	6	10	0	1653	0	0	6
flay05m	9	0	184	0	0	5	9	0	193	0	0	5	9	0	103	0	0	5	9	0	212	0	0	5
flay06h	12	0	2070	0	0	5	12	0	2064	0	0	5	12	0	1776	0	0	5	12	0	2609	0	0	5
flay06m	12	1	310	1	0	4	12	0	251	0	0	5	12	0	253	0	0	2	11	0	394	0	0	5
fo7_2	21	1	142	1	0	1	24	1	121	1	0	1	27	4	101	4	0	1	19	2	118	3	0	1
fo7_ar2_1	18	1	145	1	0	14	19	1	129	1	0	14	20	0	98	0	0	14	16	1	151	1	0	14
fo7_ar25_1	16	0	177	0	0	4	18	0	109	0	0	4	19	0	149	0	0	4	14	0	149	0	0	4
fo7_ar3_1	15	0	92	0	0	1	17	0	100	0	0	1	19	0	103	0	0	1	15	2	107	1	0	1
fo7_ar4_1	14	0	108	0	0	5	19	1	100	0	0	5	17	0	94	0	0	5	15	2	128	0	0	5
fo7_ar5_1	19	1	159	0	0	6	23	0	171	0	0	6	23	1	107	0	0	6	18	2	176	0	0	7
fo7	17	1	317	2	0	1	18	0	263	0	0	1	20	1	183	2	0	1	16	1	220	4	0	1
fo8_ar2_1	19	0	108	0	0	16	22	0	126	0	0	16	22	0	99	0	0	16	17	0	177	0	0	16
fo8_ar25_1	22	0	119	0	0	7	24	0	164	0	0	7	25	0	123	0	0	7	22	0	215	0	0	7
fo8_ar3_1	17	0	136	0	0	4	19	0	164	0	0	4	18	0	106	0	0	4	16	0	198	0	0	4
fo8_ar4_1	20	5	100	0	0	8	22	5	128	0	0	8	18	0	114	0	0	7	21	3	211	0	0	8
fo8_ar5_1	21	2	180	0	0	6	24	1	163	0	0	6	25	2	122	0	0	6	19	0	186	0	0	6
fo8	22	0	346	0	0	1	23	0	325	0	0	1	26	0	240	0	0	1	21	0	331	0	0	1
fo9_ar2_1	22	0	115	0	0	18	24	0	119	0	0	18	25	0	58	0	0	18	19	0	151	0	0	18
fo9_ar25_1	23	0	115	0	0	7	23	0	127	0	0	7	24	0	104	0	0	7	21	0	208	0	0	7
fo9_ar3_1	18	0	251	0	0	4	23	0	186	0	0	4	25	0	160	0	0	4	19	0	225	0	0	4
fo9_ar4_1	24	1	243	0	0	7	24	0	200	0	0	7	24	1	178	0	0	7	21	3	260	1	0	7
fo9_ar5_1	23	1	222	0	0	8	28	1	205	1	0	8	30	2	153	0	0	8	24	5	279	3	0	8
fo9	19	0	256	0	0	0	24	5	311	4	0	0	31	0	266	0	0	0	18	1	365	1	0	0
fuel	10	2	231	0	4	9	10	1	263	0	4	9	11	1	192	1	4	8	11	3	276	3	4	9
fuzzy	120	15	10377	7	80	84	119	22	8657	8	80	84	123	19	5363	5	80	83	116	15	8858	6	80	84
gams01	216	25	271251	23	118	143	212	27	233527	24	118	138	236	8	120856	13	118	141	216	27	296185	22	118	142
gasprod_sarawak01	37	9	547	2	6	32	39	11	635	2	6	34	37	11	511	4	6	32	37	13	657	8	6	34
gasprod_sarawak16	566	193	52080	91	113	479	565	166	40639	58	113	486	572	149	30238	70	113	494	564	213	48142	88	113	494
gasprod_sarawak81	2846	1123	607509	620	337	2432	2882	1026	428958	523	337	2420	2899	813	265274	608	337	2443	2819	1227	453014	543	337	2393
gastrans	57	5	616	3	35	34	56	5	774	3	35	34	59	2	589	0	35	32	55	6	760	4	35	32
gear2	7	4	46	3	0	4	7	2	57	1	0	4	7	4	44	3	0	4	7	1	73	0	0	4
gear3	6	1	22	0	1	5	7	2	20	0	1	5	6	2	17	0	1	5	8	1	37	0	1	5
gear4	6	2	48	1	2	2	6	1	46	1	2	2	6	2	30	1	2	2	6	1	48	0	2	2
gear	6	1	22	0	1	5	7	2	20	0	1	5	6	2	17	0	1	5	8	1	37	0	1	5
genpooling_lee1	27	6	115	10	0	6	26	6	164	8	0	5	26	6	130	7	0	3	26	6	149	22	0	4
genpooling_lee2	32	9	109	21	0	8	31	4	126	5	0	8	29	3	88	4	0	9	30	9	140	25	0	10
genpooling_meyer04	31	5	1824	43	0	5	31	5	1224	28	0	10	34	1	1231	11	0	7	33	7	1402	45	0	7
genpooling_meyer10	128	19	3150	49	0	39	131	16	3481	41	0	36	135	5	6396	19	0	49	128	15	3970	29	0	53
genpooling_meyer15	252	43	4446	113	0	99	268	45	4378	118	0	96	270	26	4818	92	0	106	260	48	4301	110	0	113
ghg_1veh	55	16	656	12	3	21	56	14	568	9	3	19	52	14	457	7	3	19	54	20	795	15	3	20
ghg_2veh	117	43	3685	32	27	58	118	44	3380	29	27	58	113	44	1918	21	27	59	123	40	4627	27	27	60
ghg_3veh	196	74	10064	49	46	104	192	73	7425	80	46	108	191	65	4903	45	46	107	199	72	11585	31	46	111
gkocis	4	0	19	0	2	3	4	0	12	0	2	3	4	0	19	0	2	3	4	0	26	0	2	3
heatexch_gen1	115	55	636	64	0	60	115	49	713	83	0	56	115	32	610	42	0	58	110	47	892	94	0	59
heatexch_spec1	95	47	450	59	1	30	97	45	379	61	1	27	94	31	373	40	1	21	93	50	372	53	1	26
heatexch_spec2	87	35	745	48	10	40	84	30	847	52	10	44	84	16	458	26	10	39	69	25	608	42	10	45
heatexch_spec3	424	139	2560	221	0	86	373	134	2156	202	0	101	442	86	1643	210	0	89	408	177	2268	294	0	93
heatexch_trigen	103	39	1232	33	16	57	96	36	1249	36	16	56	112	31	944	24	16	59	92	39	1163	32	16	59

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
hybriddynamic_var	37	1	849	1	22	25	40	1	428	0	22	25	52	1	381	1	22	25	35	5	948	7	22	25
hydroenergy1	133	58	1756	55	13	78	132	57	1579	47	13	80	154	13	1271	14	13	78	135	50	1873	52	13	73
hydroenergy2	229	82	3240	41	13	114	223	84	2931	49	13	115	284	33	2135	23	13	108	237	81	3490	53	13	111
hydroenergy3	436	198	8333	132	13	270	429	195	8446	85	13	285	480	89	5007	57	13	274	472	195	8871	134	13	279
jit1	8	0	84	0	6	6	8	0	56	0	6	6	8	0	67	0	6	6	8	0	43	0	6	6
kport20	152	44	679	31	8	31	148	45	669	41	8	33	172	27	718	42	8	32	142	45	733	35	8	33
kport40	284	102	5042	113	0	93	287	96	4274	101	0	94	310	46	3692	34	0	94	282	94	4271	81	0	99
lip	8	0	81	0	8	7	8	0	71	0	8	7	8	0	103	0	8	7	8	0	92	0	8	7
m3	10	1	37	0	0	2	9	0	21	0	0	2	9	0	27	0	0	2	9	0	56	0	0	2
m6	14	0	177	0	0	3	17	0	169	0	0	3	18	1	103	1	0	3	17	3	184	2	0	3
m7_ar2_1	18	1	113	1	0	4	18	1	83	0	0	4	14	1	84	0	0	4	15	0	93	0	0	4
m7_ar25_1	16	3	110	1	0	3	18	1	121	0	0	3	18	1	99	0	0	3	18	2	138	0	0	3
m7_ar3_1	22	4	150	0	0	6	23	2	127	1	0	6	22	2	124	0	0	6	21	6	211	0	0	6
m7_ar4_1	20	4	216	0	0	7	19	3	206	0	0	7	19	1	153	1	0	7	20	7	224	2	0	7
m7_ar5_1	25	4	193	0	0	5	23	4	215	2	0	5	23	3	124	0	0	6	25	7	292	2	0	5
m7	18	0	131	0	0	1	17	0	129	0	0	1	16	0	139	0	0	1	16	0	142	0	0	1
meanvarxsc	9	2	44	0	0	7	8	0	61	0	0	7	10	0	46	0	0	8	9	0	66	0	0	8
milinfract	416	0	4394	0	0	1	431	0	4401	0	0	1	494	0	3759	0	0	1	393	0	4504	0	0	1
minlphix	24	3	68	4	6	9	23	3	62	4	6	9	23	4	36	5	6	8	24	3	88	1	6	10
multiplants_mtg1c	49	15	16991	14	19	24	53	16	13501	6	19	21	48	8	13566	3	19	27	51	15	17934	11	19	20
multiplants_mtg5	65	10	2671	5	33	49	65	8	2216	2	33	48	70	5	1654	6	33	44	68	13	3281	14	33	51
multiplants_stg1b	151	73	33250	171	8	36	130	49	35611	162	8	30	215	70	32291	109	8	37	155	71	35078	90	8	19
multiplants_stg1c	91	29	29701	41	0	15	88	29	39740	85	0	16	126	19	10159	29	33	41	128	64	21437	101	0	14
multiplants_stg6	147	87	80288	195	9	29	143	44	50059	233	8	40	217	42	48181	113	9	87	188	91	75262	251	9	35
ndcc12	613	33	28770	19	97	317	601	27	22051	6	97	331	836	5	19748	4	97	319	612	36	15637	25	97	325
ndcc12persp	130	1	2818	0	4	73	129	2	1962	0	4	77	134	0	2002	0	4	76	129	0	2701	0	4	68
ndcc13	624	21	32094	8	210	408	607	13	23716	3	210	403	837	1	20898	0	210	405	604	11	14640	5	210	403
ndcc13persp	127	0	1901	0	72	90	127	0	1667	0	72	84	129	0	2067	0	72	86	126	1	1732	1	72	86
ndcc14	844	38	62684	17	82	446	841	41	44662	18	82	457	1138	4	37164	4	82	462	826	23	26092	3	82	427
ndcc14persp	159	4	4083	1	2	96	157	4	3092	1	2	88	160	1	3123	1	2	88	160	2	4411	1	2	85
ndcc15	643	16	64087	6	87	354	648	15	46671	9	87	381	880	1	46475	0	87	361	639	6	27119	4	87	328
ndcc16	1047	39	83759	13	60	532	1050	49	51773	16	60	543	1430	1	51437	1	60	535	1039	22	34194	8	60	460
ndcc16persp	177	3	4952	0	0	70	175	4	4168	0	0	64	177	0	4552	0	0	58	176	0	5348	0	0	58
netmod_dol1	8	1	22	1	0	2	7	0	19	0	0	2	10	2	18	0	0	4	6	0	22	0	0	4
netmod_dol2	10	4	5288	5	0	1	9	3	3856	4	0	1	10	4	3547	5	0	2	8	0	9094	0	0	3
netmod_kar1	6	1	13	1	0	2	6	1	12	1	0	2	8	2	14	2	0	1	5	1	13	1	0	2
netmod_kar2	6	1	13	1	0	2	6	1	12	1	0	2	8	2	14	2	0	1	5	1	13	1	0	2
no7_ar2_1	19	0	130	0	0	14	24	0	139	0	0	14	20	0	113	0	0	14	16	0	170	0	0	14
no7_ar25_1	16	0	178	0	0	4	19	0	143	0	0	4	19	0	114	0	0	4	18	0	205	0	0	4
no7_ar3_1	19	1	158	0	0	1	21	0	176	0	0	1	27	0	137	0	0	1	18	2	200	3	0	1
no7_ar4_1	18	2	196	0	0	6	20	2	162	0	0	6	23	0	144	0	0	6	18	3	171	0	0	6
no7_ar5_1	24	3	211	2	0	6	27	0	250	0	0	6	22	2	188	0	0	7	22	3	281	0	0	6
nous1	50	22	300	21	6	32	50	17	352	12	6	33	54	10	285	13	6	34	51	24	304	23	6	33
nous2	60	29	387	26	7	37	55	26	344	19	7	41	61	19	381	20	7	37	55	25	391	20	7	34
nuclear104	2228	40	2142	17	0	29	2232	40	2303	17	0	29	2234	39	1764	16	0	29	2227	40	2680	17	0	29
nuclear14a	593	195	370944	85466	10	376	592	187	239133	79491	10	383	618	162	202427	89805	10	373	565	175	201612	53606	10	376
nuclear14b	610	113	140665	3054	203	375	618	112	78886	2044	203	365	613	95	66415	4155	203	368	605	113	107550	3248	203	377
nuclear14	417	0	657	0	0	0	417	0	611	0	0	0	417	0	325	0	0	0	416	1	861	1	0	0
nuclear25a	574	185	85266	11054	7	374	578	181	65180	12629	7	377	597	148	237521	137860	7	372	569	187	219967	138747	7	371
nuclear25b	759	174	166037	7933	215	417	765	175	113147	6249	215	411	763	155	104529	2182	215	409	769	190	167498	9644	215	421
nuclear25	433	1	651	1	0	0	433	1	682	1	0	0	435	1	386	1	0	0	432	1	790	1	0	0
nuclear49a	1256	401	324252	60214	8	794	1242	388	225194	62799	8	787	1390	361	94723	36372	8	809	1217	395	266001	41856	8	783
nuclear49b	1664	390	971746	27547	458	939	1662	392	618448	45659	458	919	1665	311	295948	9365	458	916	1658	390	767076	50106	458	913
nuclear49	949	11	2241	6	0	8	947	10	1489	5	0	7	951	10	673	5	0	7	948	11	2576	6	0	7
nuclearva	189	2	220	2	0	0	188	2	259	2	0	0	189	2	132	2	0	0	188	2	260	2	0	0

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
nuclearvb	188	1	168	1	0	0	187	1	199	1	0	0	188	1	121	1	0	0	187	1	248	1	0	0
nuclearvc	188	1	168	1	0	0	187	1	197	1	0	0	188	1	121	1	0	0	187	1	248	1	0	0
nuclearvd	188	2	148	3	0	0	187	1	264	1	0	0	188	1	135	1	0	0	188	2	262	2	0	0
nuclearve	188	2	148	3	0	0	187	1	264	1	0	0	188	1	135	1	0	0	188	2	262	2	0	0
nuclearvf	188	2	148	3	0	0	187	1	264	1	0	0	188	1	135	1	0	0	188	2	262	2	0	0
nvs01	14	1	138	0	3	5	12	0	161	0	3	5	11	1	110	1	3	5	13	4	207	4	3	6
nvs02	5	0	21	0	0	0	5	0	28	0	0	0	5	0	21	0	0	0	7	1	54	3	0	0
nvs05	23	2	136	1	2	6	24	3	149	2	2	7	25	1	111	0	2	7	27	6	125	5	2	7
nvs06	8	1	29	1	5	7	9	1	30	0	5	7	9	0	27	0	5	7	8	1	27	0	5	7
nvs08	4	0	48	0	1	1	5	0	19	0	1	1	5	0	36	0	1	1	4	0	31	0	1	1
nvs10	4	0	10	0	4	4	4	0	7	0	4	4	4	0	5	0	4	4	4	0	7	0	4	4
nvs11	6	0	41	0	6	6	6	0	49	0	6	6	6	0	36	0	6	6	6	0	41	0	6	6
nvs12	8	0	51	0	8	8	8	0	49	0	8	8	8	0	51	0	8	8	8	0	63	0	8	8
nvs13	9	0	72	0	8	8	9	0	65	0	8	8	10	0	53	0	8	8	8	0	69	0	8	8
nvs15	4	0	20	0	2	2	4	0	18	0	2	2	4	0	20	0	2	2	4	0	22	0	2	2
nvs16	10	2	53	2	1	1	12	2	49	2	0	0	13	4	42	0	0	0	12	4	45	5	0	0
nvs17	11	0	102	0	11	11	11	0	117	0	11	11	14	0	112	0	11	11	11	0	132	0	11	11
nvs18	10	0	96	0	10	10	10	0	82	0	10	10	12	0	90	0	10	10	10	0	108	0	10	10
nvs19	13	0	189	0	12	12	13	0	193	0	12	12	16	0	176	0	12	12	12	0	200	0	12	12
nvs20	40	2	1977	2	33	33	37	2	2076	2	33	33	43	0	1660	0	33	33	37	3	1780	3	33	33
nvs21	5	1	5	0	0	2	5	1	5	0	0	2	5	0	5	0	0	2	6	2	5	0	0	2
nvs23	15	0	238	0	13	13	15	0	227	0	13	13	17	0	249	0	13	13	14	0	260	0	13	13
nvs24	17	0	301	0	12	12	17	0	330	0	12	12	20	0	349	0	12	12	16	0	322	0	12	12
o7_2	21	1	108	1	0	0	22	0	129	0	0	0	23	0	118	0	0	0	18	2	143	1	0	0
o7_ar2_1	17	0	208	0	0	14	22	0	319	0	0	14	20	0	173	0	0	14	15	0	289	0	0	14
o7_ar25_1	19	1	133	0	0	4	20	0	198	0	0	4	26	0	143	0	0	4	25	0	261	0	0	4
o7_ar3_1	21	1	205	1	0	1	19	0	198	0	0	1	27	0	145	0	0	1	18	0	174	0	0	1
o7_ar4_1	18	2	190	0	0	6	22	4	186	1	0	4	23	0	182	0	0	6	18	3	168	0	0	6
o7_ar5_1	23	4	225	0	0	6	25	1	210	2	0	6	25	2	117	0	0	6	21	3	215	2	0	6
o7	16	0	105	0	0	0	16	0	111	0	0	0	21	0	83	0	0	0	16	0	137	0	0	0
o8_ar4_1	21	3	251	0	0	8	24	3	266	0	0	8	22	1	232	0	0	8	21	6	247	0	0	8
o9_ar4_1	25	2	222	0	0	7	28	4	318	0	0	7	27	0	211	0	0	6	27	4	275	0	0	7
oil2	1078	1	67858	1	839	1073	1082	1	47390	1	839	1073	1081	0	10787	0	839	1073	1083	1	16682	1	839	1073
oil	1360	126	63281	63	863	639	1367	134	34025	46	863	610	1370	104	11555	52	863	616	1366	138	22809	55	863	748
ortez	51	16	1535	39	8	34	49	12	1223	35	8	34	51	12	922	21	8	32	51	18	1519	49	8	33
pooling_epa1	125	52	6299	32	28	66	123	52	6882	45	28	63	128	57	5547	34	28	67	126	49	7939	46	28	55
pooling_epa2	183	72	8491	76	32	90	190	89	9670	123	32	91	201	74	8024	86	32	95	199	90	11410	105	32	91
pooling_epa3	581	200	115046	260	96	261	593	220	128257	284	96	262	620	194	91900	235	96	283	581	228	127675	395	96	251
portfol_card	17	4	53	1	10	17	17	5	38	0	10	17	17	0	25	0	10	17	17	5	69	2	10	17
portfol_classical050_1	87	0	785	0	59	61	98	0	893	0	59	61	77	0	536	0	59	61	88	0	841	0	59	61
portfol_classical200_2	354	0	4217	0	263	271	400	0	4506	0	263	271	355	0	3041	0	263	271	323	0	4398	0	263	271
portfol_robust050_34	177	0	2053	0	168	172	184	0	2102	0	168	172	178	0	1453	0	168	172	174	0	2371	0	168	172
portfol_robust100_09	354	0	3940	0	334	337	365	0	3803	0	334	337	358	0	3493	0	334	337	337	0	4810	0	334	337
portfol_robust200_03	727	0	7991	0	703	708	758	0	8534	0	703	708	752	0	6259	0	703	708	708	0	9329	0	703	708
portfol_shortfall050_68	200	11	829	2	188	200	200	12	967	3	188	200	200	11	605	0	188	200	200	12	1012	4	188	200
portfol_shortfall100_04	404	0	2563	0	404	404	404	0	2599	0	404	404	404	0	2237	0	404	404	404	0	2832	0	404	404
portfol_shortfall200_05	804	0	6276	0	804	804	804	0	6746	0	804	804	804	0	5178	0	804	804	804	0	7062	0	804	804
primary	117	18	3140	11	47	69	118	21	2668	17	47	64	117	9	1664	2	47	68	117	19	2480	12	47	64
procsel	2	0	6	0	2	2	2	0	6	0	2	2	2	0	6	0	2	2	2	0	6	0	2	2
product2	243	7	6129	3	157	211	238	8	6524	0	157	212	255	2	2123	1	157	211	238	6	5950	2	157	213
product	183	51	3333	28	0	51	191	45	2997	58	0	40	215	31	2819	20	0	32	178	57	2663	39	0	56
qapw	450	0	25320	0	450	450	450	0	25427	0	450	450	450	0	22954	0	450	450	450	0	31382	0	450	450
ravempb	32	1	1472	0	3	10	39	1	1591	0	3	10	41	0	976	0	3	12	40	7	1930	4	3	10
risk2bpb	6	0	383	0	3	3	6	0	328	0	3	3	6	0	285	0	3	3	6	0	273	0	3	3
routingdelay_bigm	3940	73	440	4	46	171	3941	65	443	2	46	155	3941	37	426	8	46	174	3932	118	440	5	46	177

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
routingdelay_proj	4075	68	367	0	0	127	4081	53	377	0	0	97	4080	20	351	1	0	135	4073	122	345	2	0	139
rsyn0805h	17	0	29	0	3	5	16	0	29	0	3	5	18	0	31	0	3	5	15	0	25	0	3	5
rsyn0805m02h	26	4	563	2	4	18	26	5	394	2	4	18	27	4	351	2	4	17	26	6	664	2	4	17
rsyn0805m02m	8	1	124	2	0	5	10	4	145	3	0	6	11	0	116	0	0	4	7	1	133	1	0	6
rsyn0805m03m	16	5	282	7	0	9	18	6	185	7	0	7	18	1	269	1	0	11	15	4	284	4	0	9
rsyn0805m04m	21	4	459	3	0	12	21	5	606	5	0	14	23	1	499	1	0	13	22	7	683	10	0	14
rsyn0805m	5	0	61	0	0	2	5	0	41	0	0	2	5	0	43	0	0	2	4	0	32	0	0	2
rsyn0810m02m	18	4	311	5	0	8	16	1	256	1	0	9	22	0	159	0	0	9	17	3	303	5	0	9
rsyn0810m03m	26	5	554	7	0	12	28	7	636	11	0	9	33	2	470	3	0	16	30	9	673	14	0	12
rsyn0810m04m	37	7	863	9	0	18	38	10	1042	10	0	15	45	4	872	5	0	19	38	14	953	13	0	18
rsyn0810m	10	0	83	0	0	3	8	0	71	0	0	5	9	0	62	0	0	3	8	0	63	0	0	3
rsyn0815m02m	33	7	731	5	0	17	32	7	790	8	0	17	39	3	589	3	0	15	32	10	503	13	0	12
rsyn0815m03m	48	11	2441	14	2	18	49	17	2161	25	2	18	61	1	2026	0	2	21	52	19	2210	25	2	17
rsyn0815m04m	66	15	2961	20	3	27	70	18	3221	30	3	26	83	0	2744	0	3	27	65	23	3409	35	3	27
rsyn0815m	19	4	216	5	0	9	16	6	216	7	0	5	19	2	139	3	0	4	15	3	219	8	0	5
rsyn0820m02m	42	12	1099	12	0	18	40	11	1069	17	0	18	50	2	862	1	0	21	43	14	1106	32	0	19
rsyn0820m03m	57	12	2074	11	2	25	56	12	1944	26	2	22	68	4	1476	10	2	24	57	15	1809	29	2	24
rsyn0820m04m	84	20	2433	36	3	28	83	20	2288	25	3	28	102	1	1917	1	3	29	88	34	2614	57	3	29
rsyn0820m	21	3	196	3	0	9	25	7	258	9	0	5	26	1	190	1	0	7	21	5	291	5	0	7
rsyn0830m02m	54	13	1146	12	3	25	55	11	1126	10	3	31	62	4	830	3	3	25	54	9	1414	11	3	27
rsyn0830m03m	92	26	1653	53	0	34	89	28	1880	55	0	35	98	8	1428	9	0	34	82	24	1634	50	0	31
rsyn0830m04m	117	31	3951	38	4	52	111	26	4412	41	4	50	136	7	3120	7	4	53	118	37	2787	42	4	53
rsyn0830m	25	2	398	1	0	10	24	2	329	2	0	14	29	0	288	0	0	13	27	8	332	8	0	14
rsyn0840m02m	75	16	1320	22	0	32	78	21	1356	33	0	33	91	3	1115	8	0	36	74	24	1214	26	0	37
rsyn0840m03m	120	26	3420	44	1	55	122	29	2951	30	1	63	145	7	2006	15	1	51	112	27	2535	28	1	62
rsyn0840m04m	150	30	5005	35	3	65	152	32	5461	27	3	64	190	6	3313	6	3	53	145	39	4392	42	3	59
rsyn0840m	37	4	543	5	1	14	37	5	551	5	1	11	42	0	366	0	1	15	36	9	512	15	1	16
sep1	10	0	69	0	6	9	10	0	67	0	6	8	10	0	61	0	6	8	10	1	75	1	6	7
sepasequ_complex	570	147	22152	118	53	222	602	164	23069	234	53	219	653	101	12396	102	53	210	535	173	16099	181	53	222
sepasequ_convent	667	220	110455	13421	258	387	678	212	115685	14890	258	398	743	208	116229	12022	258	386	667	218	99377	13783	258	404
sfacloc1_2_80	147	67	1515	182	8	28	154	55	1497	139	8	36	162	15	1415	13	8	42	140	72	1479	122	8	35
sfacloc1_2_90	146	52	1552	85	8	27	146	39	1660	51	8	30	152	24	1441	28	8	23	130	66	1527	71	8	23
sfacloc1_2_95	148	55	2089	83	2	16	147	54	2051	77	2	17	153	31	1828	61	2	16	140	80	1762	126	2	20
sfacloc1_3_80	220	99	2980	169	0	48	222	98	2584	172	0	57	232	34	2585	38	0	53	223	125	2293	199	0	60
sfacloc1_3_90	204	78	4031	116	0	45	208	90	3944	144	0	44	229	46	3472	87	0	35	215	113	3556	159	0	38
sfacloc1_3_95	203	90	2687	141	1	37	216	99	2557	175	1	44	225	56	2068	90	1	40	225	131	2189	179	1	44
sfacloc1_4_80	297	130	3700	144	0	85	293	113	3400	142	0	86	313	36	4018	33	0	102	283	145	3245	179	0	100
sfacloc1_4_90	272	98	4888	121	0	51	265	112	4795	121	0	53	293	33	4758	37	0	58	265	125	4420	135	0	48
sfacloc1_4_95	303	156	3767	226	0	58	299	147	3900	191	0	49	313	73	3261	114	0	49	285	158	3543	232	0	51
sfacloc2_2_80	115	28	2366	30	0	38	106	31	1890	31	0	33	125	11	1903	10	0	40	105	38	1914	36	0	31
sfacloc2_2_90	104	30	1650	24	0	31	107	20	1711	14	0	36	121	6	1613	8	0	35	96	25	1792	33	0	31
sfacloc2_2_95	55	7	1358	4	0	21	57	12	1160	17	0	20	63	6	1059	4	0	18	55	15	1262	16	0	19
sfacloc2_3_80	60	0	1516	0	0	51	60	0	1583	0	0	50	60	0	1463	0	0	51	60	0	1401	0	0	51
sfacloc2_3_90	60	0	1020	0	0	48	60	0	1194	0	0	49	60	0	1087	0	0	48	60	0	1079	0	0	50
sfacloc2_3_95	32	0	829	0	0	26	32	0	779	0	0	24	32	0	996	0	0	23	32	0	666	0	0	26
sfacloc2_4_80	75	0	3264	0	0	66	75	0	3154	0	0	67	75	0	2130	0	0	64	75	0	3033	0	0	68
sfacloc2_4_90	75	0	1737	0	0	62	75	0	1785	0	0	67	75	0	1531	0	0	69	75	0	1564	0	0	66
sfacloc2_4_95	40	0	1331	0	0	35	40	0	1327	0	0	33	40	0	1107	0	0	33	40	0	1709	0	0	35
slay04h	13	2	593	1	4	9	15	2	675	4	4	8	14	1	636	5	4	9	14	2	756	1	4	8
slay04m	12	0	325	0	3	4	12	2	331	5	3	4	16	1	231	1	3	4	12	1	297	1	3	4
slay05h	18	1	1050	4	5	11	19	1	989	0	5	11	18	0	973	0	5	11	19	5	1248	1	5	10
slay05m	14	0	346	0	0	0	15	1	404	1	0	0	19	0	390	0	0	0	17	2	475	3	0	0
slay06h	23	1	2221	0	6	18	23	6	2068	9	6	17	24	0	1912	0	6	18	24	3	2637	1	6	19
slay06m	19	2	477	2	0	0	15	2	457	3	0	0	22	1	508	1	0	0	14	0	634	0	0	0
slay07h	27	4	3032	2	3	18	26	2	3687	0	3	19	28	0	3397	0	3	20	25	6	3235	1	3	19

Table 5 continued

Instance	random						min–max						greedy (base)						reverse greedy					
	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
slay07m	21	3	843	5	0	0	23	4	768	5	0	0	27	0	653	0	0	0	20	2	1131	2	0	1
slay08h	31	5	3323	1	4	25	28	7	3150	7	4	23	32	1	2707	0	4	27	30	9	4201	8	4	25
slay08m	26	1	1094	2	0	0	25	4	1318	5	0	0	32	1	1191	1	0	1	28	2	1713	6	0	0
slay09h	34	5	5370	0	5	28	34	8	6526	4	5	29	36	1	5282	0	5	28	33	10	5984	7	5	26
slay09m	28	1	1702	1	0	0	26	2	2141	2	0	0	30	0	1419	0	0	0	29	4	2521	5	0	0
slay10h	38	6	8052	1	5	32	37	4	7136	2	5	34	40	2	5700	0	5	35	38	7	8806	0	5	34
slay10m	32	4	1979	6	0	0	29	1	2168	1	0	0	38	1	1797	1	0	0	31	1	2947	1	0	0
smallinvDAXr1b010-011	11	0	77	0	8	8	12	0	72	0	8	8	13	0	87	0	8	8	10	0	80	0	8	8
smallinvDAXr1b020-022	13	0	71	0	8	8	13	0	79	0	8	8	15	0	76	0	8	8	11	0	86	0	8	8
smallinvDAXr1b050-055	14	0	108	0	12	12	14	0	96	0	12	12	14	0	88	0	12	12	14	0	113	0	12	12
smallinvDAXr1b100-110	14	0	106	0	13	13	14	0	105	0	13	13	15	0	106	0	13	13	14	0	117	0	13	13
smallinvDAXr1b150-165	31	0	132	0	26	28	32	0	130	0	26	28	34	0	131	0	26	28	30	0	127	0	26	28
smallinvDAXr1b200-220	31	0	137	0	26	28	32	0	133	0	26	28	34	0	131	0	26	28	30	0	129	0	26	28
smallinvDAXr2b010-011	10	0	90	0	7	7	10	0	81	0	7	7	14	0	88	0	7	7	8	0	79	0	7	7
smallinvDAXr2b020-022	12	0	82	0	9	9	14	0	89	0	9	9	15	0	92	0	9	9	12	0	96	0	9	9
smallinvDAXr2b050-055	31	0	159	0	27	27	32	0	152	0	27	27	34	0	160	0	27	27	30	0	165	0	27	27
smallinvDAXr2b100-110	13	0	86	0	12	12	13	0	87	0	12	12	14	0	93	0	12	12	13	0	123	0	12	12
smallinvDAXr2b150-165	15	0	115	0	13	13	15	0	113	0	13	13	15	0	85	0	13	13	15	0	122	0	13	13
smallinvDAXr2b200-220	14	0	105	0	13	13	14	0	100	0	13	13	14	0	92	0	13	13	14	0	113	0	13	13
smallinvDAXr3b010-011	11	0	92	0	7	7	11	0	81	0	7	7	15	0	81	0	7	7	11	0	86	0	7	7
smallinvDAXr3b020-022	14	0	119	0	10	10	14	0	103	0	10	10	14	0	100	0	10	10	14	0	117	0	10	10
smallinvDAXr3b050-055	14	0	136	0	13	13	14	0	116	0	13	13	15	0	95	0	13	13	14	0	133	0	13	13
smallinvDAXr3b100-110	14	0	128	0	13	13	14	0	110	0	13	13	15	0	110	0	13	13	14	0	135	0	13	13
smallinvDAXr3b150-165	14	0	111	0	13	13	14	0	115	0	13	13	14	0	101	0	13	13	14	0	122	0	13	13
smallinvDAXr3b200-220	15	0	118	0	13	13	15	0	108	0	13	13	16	0	99	0	13	13	15	0	118	0	13	13
smallinvDAXr4b010-011	12	0	86	0	7	7	12	0	85	0	7	7	14	0	105	0	7	7	12	0	96	0	7	7
smallinvDAXr4b020-022	13	0	113	0	11	11	14	0	127	0	11	11	15	0	113	0	11	11	13	0	121	0	11	11
smallinvDAXr4b050-055	14	0	128	0	13	13	14	0	134	0	13	13	15	0	136	0	13	13	14	0	147	0	13	13
smallinvDAXr4b100-110	14	0	123	0	13	13	14	0	121	0	13	13	14	0	109	0	13	13	14	0	127	0	13	13
smallinvDAXr4b150-165	13	0	108	0	13	13	13	0	110	0	13	13	13	0	97	0	13	13	13	0	113	0	13	13
smallinvDAXr4b200-220	15	0	113	0	13	13	15	0	111	0	13	13	15	0	102	0	13	13	15	0	137	0	13	13
smallinvDAXr5b010-011	12	0	93	0	8	8	12	0	93	0	8	8	13	0	101	0	8	8	11	0	91	0	8	8
smallinvDAXr5b020-022	13	0	107	0	11	11	12	0	88	0	11	11	15	0	128	0	11	11	12	0	118	0	11	11
smallinvDAXr5b050-055	14	0	97	0	13	13	14	0	99	0	13	13	14	0	99	0	13	13	14	0	114	0	13	13
smallinvDAXr5b100-110	14	0	132	0	9	9	14	0	122	0	9	9	16	0	99	0	9	9	12	0	133	0	9	9
smallinvDAXr5b150-165	14	0	102	0	13	13	14	0	105	0	13	13	15	0	98	0	13	13	14	0	120	0	13	13
smallinvDAXr5b200-220	15	0	110	0	13	13	15	0	115	0	13	13	15	0	98	0	13	13	15	0	114	0	13	13
smallinvSNPr1b010-011	100	0	1037	0	68	72	100	0	1068	0	68	72	114	0	963	0	68	72	99	0	885	0	68	72
smallinvSNPr1b020-022	101	0	1275	0	67	79	101	0	1195	0	67	79	112	0	1045	0	67	79	99	0	1041	0	67	79
smallinvSNPr1b050-055	101	0	1407	0	39	81	101	0	1242	0	39	81	107	0	1101	0	39	81	99	0	1195	0	39	81
smallinvSNPr1b100-110	101	0	1655	0	26	79	101	0	1509	0	26	79	111	0	1232	0	26	79	99	0	1282	0	26	79
smallinvSNPr1b150-165	101	0	1352	0	31	84	101	0	1231	0	31	84	112	0	1176	0	31	84	99	0	1104	0	31	84
smallinvSNPr1b200-220	101	0	1430	0	33	84	101	0	1367	0	33	84	111	0	1175	0	33	84	99	0	1150	0	33	84
smallinvSNPr2b010-011	100	0	818	0	75	76	101	0	826	0	75	76	113	0	781	0	75	76	100	0	806	0	75	76
smallinvSNPr2b020-022	101	0	1255	0	47	63	102	0	1204	0	47	63	111	0	1000	0	47	63	101	0	1053	0	47	63
smallinvSNPr2b050-055	101	0	1595	0	36	82	101	0	1389	0	36	82	110	0	1237	0	36	82	100	0	1265	0	36	82
smallinvSNPr2b100-110	101	0	1470	0	30	81	101	0	1374	0	30	81	110	0	1106	0	30	81	100	0	1198	0	30	81
smallinvSNPr2b150-165	101	0	1763	0	22	85	101	0	1541	0	22	85	109	0	1371	0	22	85	100	0	1356	0	22	85
smallinvSNPr2b200-220	101	0	1579	0	28	84	101	0	1565	0	28	84	112	0	1338	0	28	84	100	0	1359	0	28	84
smallinvSNPr3b010-011	100	0	928	0	68	69	101	0	954	0	68	69	112	0	755	0	68	69	100	0	781	0	68	69
smallinvSNPr3b020-022	101	0	906	0	61	72	100	0	950	0	61	72	109	0	839	0	61	72	99	0	828	0	61	72
smallinvSNPr3b050-055	101	0	1181	0	26	73	100	0	1161	0	26	73	109	0	997	0	26	73	100	0	969	0	26	73
smallinvSNPr3b100-110	101	0	1461	0	32	87	100	0	1472	0	32	87	110	0	1268	0	32	87	100	0	1217	0	32	87
smallinvSNPr3b150-165	102	0	1404	0	21	85	100	0	1393	0	21	85	110	0	1151	0	21	85	100	0	1260	0	21	85
smallinvSNPr3b200-220	101	0	1388	0	13	78	101	0	1399	0	13	78	114	0	1143	0	13	78	100	0	1138	0	13	78

Table 5 continued

Instance	random						min–max						greedy (base)						reverse greedy					
	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
smallinvSNPr4b010-011	101	0	891	0	56	57	101	0	855	0	56	57	110	0	719	0	56	57	99	0	787	0	56	57
smallinvSNPr4b020-022	101	0	1192	0	43	61	101	0	1070	0	43	61	106	0	876	0	43	61	98	0	815	0	43	61
smallinvSNPr4b050-055	101	0	1270	0	35	80	101	0	1226	0	35	80	110	0	1062	0	35	80	99	0	1083	0	35	80
smallinvSNPr4b100-110	101	0	1593	0	18	82	102	1	1400	1	18	82	110	0	1197	0	18	82	99	0	1174	0	18	82
smallinvSNPr4b150-165	101	0	1366	0	18	88	101	0	1380	0	18	88	109	0	1024	0	18	88	99	0	1098	0	18	88
smallinvSNPr4b200-220	101	0	1706	0	20	89	100	0	1629	0	20	89	110	0	1308	0	20	89	99	0	1279	0	20	89
smallinvSNPr5b010-011	99	0	748	0	66	67	100	0	730	0	66	67	110	0	620	0	66	67	98	0	662	0	66	67
smallinvSNPr5b020-022	100	0	1074	0	44	60	100	0	1111	0	44	60	110	0	798	0	44	60	98	0	857	0	44	60
smallinvSNPr5b050-055	100	0	887	0	12	65	101	0	1028	0	12	65	108	0	747	0	12	65	98	0	694	0	12	65
smallinvSNPr5b100-110	101	0	987	0	35	89	101	0	938	0	35	89	109	0	709	0	35	89	98	0	720	0	35	89
smallinvSNPr5b150-165	101	0	1062	0	15	90	100	0	1038	0	15	90	110	0	883	0	15	90	98	0	792	0	15	90
smallinvSNPr5b200-220	100	0	1229	0	34	92	101	0	1302	0	34	92	111	0	1044	0	34	92	98	0	995	0	34	92
space25a	64	23	79	0	0	31	62	22	68	0	0	30	64	19	81	0	0	28	67	26	71	0	0	30
space25	43	1	89	2	0	1	43	1	77	1	0	1	44	0	80	0	0	2	42	1	125	1	0	2
space960	2089	330	547345	263	12	1	2092	318	639268	251	12	0	2494	112	804391	128	12	0	2149	336	734197	287	12	0
spectra2	31	0	857	0	26	26	31	0	773	0	26	26	36	0	817	0	26	26	30	0	794	0	26	27
sporttournament14	75	10	501	14	0	2	66	11	427	14	0	2	118	8	496	14	0	2	56	16	533	27	0	2
spring	23	2	197	0	7	12	23	2	249	0	7	12	23	2	117	0	7	12	23	2	309	0	7	12
squfl010-025	290	72	18950	93	0	166	249	28	20060	30	0	171	449	3	14415	3	0	160	248	40	16503	49	0	185
squfl010-025persp	564	66	123215	70	267	444	612	94	118245	123	267	430	705	57	89169	65	267	424	527	59	133231	62	267	437
squfl010-040	410	45	53820	25	0	265	401	40	52032	16	0	273	726	5	38553	2	0	273	392	28	56904	8	0	278
squfl010-040persp	913	84	516367	132	508	724	949	82	478019	157	508	712	1214	78	325905	105	508	712	854	81	526724	105	508	726
squfl010-080	694	186	39984	39	0	477	716	201	40205	82	0	471	1391	32	36181	15	0	493	753	241	42969	115	0	479
squfl010-080persp	1917	451	942637	452	800	1347	1933	460	941096	510	800	1331	2809	136	610351	144	800	1327	1746	438	923404	481	800	1326
squfl015-060	855	136	148368	20	0	640	878	134	141633	52	0	637	1718	26	86294	5	0	577	852	110	145907	13	0	623
squfl015-060persp	2313	835	2031133	2210	900	1561	2506	969	2078376	2365	900	1563	2909	713	1081371	1657	900	1391	2072	753	2101965	2271	900	1518
squfl015-080	1069	381	136465	162	0	815	1098	391	128955	215	0	802	2236	53	93262	32	0	752	1066	316	134552	91	0	829
squfl015-080persp	3085	543	2272852	634	1200	2058	3179	572	2193950	710	1200	2035	3860	322	1377896	380	1200	2008	2949	548	2247949	671	1200	2096
squfl020-040	809	81	110149	30	0	572	844	108	103812	69	0	560	1551	18	74615	6	0	492	841	127	113882	69	0	545
squfl020-040persp	1964	279	2079652	366	1109	1498	2008	308	1865504	460	1109	1479	2265	252	881474	438	1109	1459	1873	267	2137165	337	1109	1504
squfl020-050	1061	266	243097	119	0	727	1023	186	232174	67	0	687	1906	54	138898	8	0	639	1062	245	248620	107	0	725
squfl020-050persp	2349	273	3777672	422	1277	1909	2373	261	3272307	361	1277	1891	2877	192	1529480	250	1277	1896	2175	211	3974302	213	1277	1924
squfl020-150persp	7873	1072	6440077	1503	3000	5230	8309	1086	6405817	1614	3000	5176	9480	1279	4623861	2121	3000	5102	6847	767	5515301	998	3000	5227
squfl025-025	647	29	91617	22	0	405	724	114	94988	103	0	409	1188	2	59358	3	0	361	669	69	82316	51	0	432
squfl025-025persp	1426	164	1125663	330	919	1204	1465	170	988580	355	919	1186	1628	132	616521	276	919	1165	1360	140	1119077	267	919	1201
squfl025-030	826	87	120114	77	0	477	792	58	121324	51	0	508	1482	13	76202	11	0	441	820	83	116761	78	0	525
squfl025-030persp	1716	280	1624054	653	788	1379	1786	314	1368587	527	788	1367	1953	240	734069	418	788	1278	1631	229	1604636	359	788	1391
squfl025-040	1069	199	351241	103	0	674	1070	167	300238	91	0	698	1939	24	199743	9	0	644	1026	146	342754	58	0	676
squfl025-040persp	2457	475	2379379	665	1353	1839	2489	473	1987106	624	1353	1851	2791	302	1024115	419	1353	1845	2435	526	2520972	631	1353	1860
squfl030-100persp	8868	3224	4451696	5660	3000	5037	8903	3213	4799764	5946	3000	4906	8672	2314	3607947	4100	3000	5138	7613	2435	4198360	4813	3000	4866
squfl030-150persp	13733	4167	7279649	6351	4500	7468	13606	3934	7091007	5717	4500	7536	14300	3119	6118433	4888	4500	7839	11661	2844	6301574	4253	4500	7526
squfl040-080	3644	1402	1146707	961	0	2129	3333	945	1069174	578	0	2074	5885	159	568778	94	0	1989	3489	1142	1045691	745	0	2112
squfl040-080persp	9083	3814	3095437	6997	3200	5128	8982	3714	3116829	7314	3200	5117	9821	3614	2989762	6934	3200	5337	8798	3826	2975984	5985	3200	5249
sssd08-04	14	4	181	1	4	10	13	6	160	2	4	10	14	3	147	0	4	10	13	7	167	3	4	10
sssd08-04persp	28	11	298	20	4	12	27	12	280	14	4	14	27	9	341	19	4	15	25	12	248	17	4	15
sssd12-05	18	6	308	1	0	13	18	9	281	4	0	13	18	4	294	0	0	13	18	11	259	6	0	13
sssd12-05persp	32	10	513	20	5	17	34	14	474	33	5	14	36	11	512	23	5	20	33	16	401	26	5	21
sssd15-04	16	6	319	1	0	12	16	6	364	5	0	12	16	5	293	1	0	12	16	10	317	4	0	12
sssd15-04persp	28	14	439	29	4	12	25	11	390	15	4	16	31	10	424	23	4	15	28	15	391	21	4	16
sssd15-06	19	5	467	1	0	14	20	4	450	0	0	14	20	4	503	0	0	14	20	11	460	6	0	14
sssd15-06persp	36	8	742	9	6	22	36	11	608	14	6	21	39	9	705	19	6	22	35	14	599	16	6	25
sssd15-08	26	7	769	3	8	18	26	10	771	5	8	18	26	6	652	1	8	18	26	13	693	8	8	18
sssd15-08persp	45	6	1258	8	8	23	48	14	1061	22	8	30	49	12	925	17	8	24	45	16	855	8	8	33
sssd16-07	24	9	515	3	7	17	24	6	518	1	7	16	24	6	484	1	7	17	24	14	438	7	7	17
sssd16-07persp	45	14	837	19	7	23	46	20	647	25	7	27	48	14	774	22	7	25	46	22	622	25	7	26

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
sssd18-06	20	6	514	2	6	14	20	9	540	6	6	13	20	6	561	2	6	14	20	11	462	6	6	14
sssd18-06persp	37	16	647	21	6	21	37	12	708	20	6	19	40	13	737	19	6	19	37	16	669	20	6	20
sssd18-08	28	10	937	4	8	20	28	11	804	4	8	20	28	6	840	0	8	20	28	15	820	8	8	20
sssd18-08persp	50	13	1330	19	8	31	51	19	977	29	8	33	54	17	1136	31	8	33	52	23	1125	23	8	31
sssd20-04	14	8	237	4	0	10	14	5	250	1	0	10	14	5	283	4	0	10	14	8	275	4	0	10
sssd20-04persp	27	11	400	12	4	16	24	10	363	12	4	14	27	7	449	17	4	12	27	13	420	23	4	16
sssd20-08	26	10	706	4	8	18	26	9	567	3	8	18	26	7	675	2	8	18	26	15	590	8	8	18
sssd20-08persp	47	17	947	28	8	26	47	13	968	15	8	22	52	12	915	28	8	29	50	23	718	25	8	30
sssd22-08	28	12	813	11	8	20	28	10	843	9	8	20	28	7	789	1	8	20	28	16	770	8	8	20
sssd22-08persp	47	16	1086	34	8	26	46	13	1122	22	8	29	55	17	1029	35	8	26	51	24	975	32	8	29
sssd25-04	14	5	370	2	4	10	14	5	397	5	4	10	14	4	399	1	4	10	14	8	362	6	4	10
sssd25-04persp	26	10	510	15	4	13	28	13	520	20	4	15	28	8	501	17	4	17	20	9	430	19	4	14
sssd25-08	26	8	754	3	8	18	26	11	680	8	8	18	26	6	814	2	8	18	26	14	735	11	8	18
sssd25-08persp	48	11	1107	15	8	29	48	14	1125	19	8	25	51	13	1097	25	8	28	50	23	968	25	8	30
st_e29	8	0	90	0	4	7	8	0	128	0	4	6	8	0	67	0	4	6	8	1	71	0	4	6
st_e31	7	0	26	0	2	2	7	0	41	0	2	2	7	0	28	0	2	2	6	0	26	0	2	2
st_e32	56	4	514	4	17	23	54	7	551	5	17	23	63	2	373	2	17	23	56	5	587	3	17	23
st_e36	13	2	101	1	4	3	13	3	142	1	4	3	13	3	111	1	4	3	13	4	99	1	4	3
st_e38	10	0	43	0	8	7	10	0	53	0	8	7	10	1	47	0	8	7	10	1	60	0	8	7
st_e40	23	6	225	3	5	4	24	10	169	3	5	3	23	4	166	2	5	4	23	10	184	2	5	1
stockcycle	49	0	1947	0	0	0	50	0	2205	0	0	0	72	0	2522	0	0	0	48	0	2782	0	0	0
st_test8	17	2	24	1	13	15	17	2	18	1	13	15	17	2	12	2	13	15	17	3	31	3	13	15
st_testgr1	10	0	47	0	10	10	10	0	55	0	10	10	10	0	51	0	10	10	10	0	55	0	10	10
st_testgr3	12	0	57	0	12	12	12	0	56	0	12	12	12	0	47	0	12	12	12	0	65	0	12	12
super1	1081	256	232483	170	341	611	1057	269	220784	168	340	595	1129	215	115163	181	341	609	1094	301	221829	234	340	593
super2	1081	257	245036	208	355	603	1089	266	209761	177	356	607	1125	199	104966	127	356	609	1084	281	220676	184	355	604
super3	1105	284	250326	176	353	603	1102	279	213212	207	354	602	1146	186	130325	131	354	602	1095	300	257448	185	353	588
super3t	793	204	192428	134	256	504	796	210	157683	154	256	499	853	159	95314	131	256	512	805	213	197118	134	256	500
supplychain	12	2	83	1	4	6	12	2	97	0	4	6	12	1	82	0	4	6	12	2	93	2	4	6
supplychainp1_022020	60	0	14327	0	24	37	60	0	19196	0	24	35	60	0	13257	0	24	38	60	0	13231	0	24	37
supplychainp1_030510	25	0	879	0	10	10	25	0	867	0	10	10	25	0	989	0	10	10	24	1	847	1	10	10
supplychainp1_053050	122	3	88505	4	50	63	130	0	78188	0	50	63	130	0	82449	0	50	59	80	0	59891	0	50	63
supplychainr1_053050	48	6	218824	8	3	26	45	5	205098	13	3	23	95	1	189541	0	3	33	34	4	222844	6	3	20
syn05h	16	4	70	1	6	9	16	3	55	1	6	10	16	0	69	0	6	11	16	5	68	0	6	10
syn05m02m	7	0	120	0	4	5	8	0	123	0	4	5	10	0	121	0	4	4	7	0	128	0	4	5
syn05m03m	13	1	174	1	5	9	12	1	218	1	5	9	13	0	138	0	5	9	10	0	138	0	5	8
syn05m04m	17	1	401	1	6	8	16	1	268	1	6	12	20	1	179	1	6	9	15	0	419	0	6	10
syn10h	29	7	209	7	1	15	29	8	187	11	1	14	29	7	162	8	1	19	28	6	226	2	1	15
syn10m02m	18	3	323	4	0	10	18	5	241	5	0	12	21	1	254	2	0	7	19	5	227	8	0	8
syn10m03m	29	5	931	3	0	18	27	4	848	5	0	14	31	0	630	0	0	15	27	5	623	6	0	12
syn10m04m	34	3	1245	4	0	19	34	4	1300	2	0	18	44	1	860	3	0	13	33	9	919	9	0	16
syn10m	7	0	46	0	6	6	7	0	44	0	6	6	7	0	48	0	6	6	7	0	47	0	6	7
syn15m02m	34	8	831	7	8	15	33	6	878	5	8	15	39	2	738	1	8	12	30	4	796	4	8	15
syn15m03m	48	8	2509	13	17	21	48	7	2862	9	17	21	57	4	2157	3	17	23	47	7	2229	7	17	26
syn15m04m	69	14	2978	13	2	29	63	11	2812	9	2	29	78	5	2400	4	2	29	66	17	2247	21	2	32
syn15m	17	3	187	3	4	7	14	1	175	1	4	6	19	0	138	0	4	7	16	5	160	7	4	8
syn20m02m	39	10	737	11	0	14	41	13	846	14	0	14	49	3	682	2	0	14	41	15	715	15	0	14
syn20m03m	64	21	1232	33	0	19	63	19	1191	27	0	23	71	5	836	6	0	23	57	19	1062	26	0	22
syn20m04m	83	22	2457	21	0	35	78	22	2648	25	0	23	95	5	1792	6	0	30	76	28	2437	31	0	34
syn20m	22	5	286	6	1	5	19	2	311	3	1	8	24	1	295	2	1	5	18	2	312	3	1	7
syn30m02m	49	14	925	16	6	29	50	10	780	12	6	22	58	4	709	6	6	22	48	10	896	16	6	22
syn30m03m	77	16	1819	35	5	35	76	22	1875	47	5	35	91	3	1141	4	5	37	78	25	1372	35	5	31
syn30m04m	113	26	2269	42	10	52	111	37	2041	51	10	48	131	11	1356	17	10	40	109	33	1722	65	10	46
syn30m	24	4	342	6	3	12	26	5	416	6	3	14	29	2	296	2	3	12	28	6	325	8	3	12
syn40m02m	69	13	1383	10	0	38	66	13	1384	16	0	36	81	7	850	8	0	36	70	19	1050	23	0	33

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
syn40m03m	113	24	1984	29	0	57	112	28	1790	30	0	52	141	6	1458	6	0	47	112	31	1685	30	0	59
syn40m04m	159	36	3556	62	0	68	150	43	3443	57	0	71	192	13	2158	18	0	51	148	41	2728	59	0	63
syn40m	34	5	621	7	0	19	33	7	552	11	0	15	39	2	453	3	0	17	35	10	489	14	0	10
synheat	79	31	387	30	1	28	87	41	317	40	1	25	78	22	275	18	1	24	77	41	311	46	1	26
synthes1	3	0	13	0	0	1	3	1	10	1	0	1	3	0	9	0	0	1	3	1	10	1	0	1
synthes2	6	1	18	1	2	3	5	1	17	1	2	3	6	2	33	1	2	4	4	1	37	1	2	2
synthes3	10	1	76	0	0	1	7	1	43	0	0	2	10	2	59	1	0	1	7	2	53	1	0	2
tanksize	27	4	177	8	7	15	25	2	120	4	7	16	26	1	104	2	7	16	23	2	166	1	7	16
tln12	148	3	4243	6	38	59	150	4	4350	4	38	63	207	2	7240	2	38	64	164	26	1514	83	38	62
tln4	26	3	163	3	9	10	24	2	150	2	9	12	30	0	149	0	9	13	23	3	174	3	9	10
tln5	28	1	245	1	15	16	27	1	295	1	15	18	33	1	310	1	15	18	28	3	302	2	15	16
tln6	47	5	304	11	19	25	44	5	365	10	19	22	55	2	384	5	19	22	44	6	333	11	19	22
tln7	61	1	525	1	5	22	58	1	593	1	5	20	89	0	836	0	5	21	60	4	542	5	5	16
tloss	44	4	230	4	21	24	44	4	187	4	21	21	51	0	227	0	21	23	42	4	191	4	21	22
tls12	229	12	1142	3	24	103	229	14	1295	7	24	106	230	12	1893	1	24	109	229	14	2092	6	24	97
tls2	9	2	88	2	0	2	9	1	57	1	0	1	9	2	56	2	0	2	7	0	81	0	0	1
tls4	56	16	385	21	15	28	50	11	467	18	15	27	58	13	438	15	15	29	49	12	552	22	15	31
tls5	66	17	1076	36	20	37	64	15	959	58	20	34	65	14	1037	54	20	36	61	15	1084	91	20	37
tls6	83	15	1407	91	20	33	80	12	1067	89	20	32	88	21	1264	84	20	32	86	19	1184	101	20	38
tls7	121	26	1844	26	70	78	123	24	2553	36	70	79	120	19	2222	23	70	78	121	25	1589	21	70	83
tltr	36	7	550	5	6	16	33	5	525	4	6	17	42	3	444	0	6	27	33	5	585	3	6	23
transswitch0009r	64	6	574	14	0	3	66	8	585	10	0	4	80	4	459	5	0	4	60	10	431	12	0	3
transswitch0014r	149	27	2107	37	1	40	158	37	1802	43	1	42	181	20	1739	23	1	37	148	37	2611	60	1	36
transswitch0030r	322	59	4913	15	0	122	325	57	4559	16	0	125	366	55	4041	18	0	111	319	69	5978	26	0	105
transswitch0039r	358	27	6066	33	0	6	360	27	6085	33	0	7	432	20	5755	30	0	8	346	27	7814	38	0	8
transswitch0057r	588	116	8566	159	0	129	560	114	6705	150	0	130	649	56	7112	57	0	134	556	128	10067	173	0	131
transswitch0118r	1344	262	22221	277	0	329	1327	238	21731	238	0	308	1551	132	19203	100	0	349	1279	282	27930	293	0	326
transswitch0300r	3030	483	47650	415	0	544	2976	484	45604	429	0	528	3491	306	39201	235	0	539	2803	526	55430	470	0	538
transswitch2383wpr	20478	1792	212206	1215	0	1498	20621	1697	214110	1126	0	1523	25117	1227	179768	623	0	1533	18994	2060	260358	1384	0	1502
transswitch2736spr	26853	2226	280462	1163	0	2059	26944	2201	290641	1204	0	2024	32113	1602	236137	699	0	2020	24563	2672	335567	1668	0	1969
tspn05	35	2	781	2	17	15	36	4	775	4	17	15	34	1	707	2	17	15	35	4	1264	4	17	15
tspn08	71	4	1678	4	27	34	71	6	1896	5	27	34	74	2	2001	2	27	34	68	4	3662	2	27	34
tspn10	148	2	14757	2	128	133	150	7	12121	9	128	133	146	1	11546	1	128	133	144	2	18926	0	128	133
tspn12	211	11	27560	12	172	181	210	10	23758	18	172	181	206	4	21790	4	172	182	216	20	34942	17	172	181
tspn15	257	32	27389	50	129	142	281	33	25056	41	129	142	283	11	19453	14	129	142	240	35	34519	29	129	142
unitcommit1	246	44	20230	50	1	25	263	54	15191	55	1	23	300	18	14905	19	1	30	214	51	15400	42	1	30
unitcommit2	422	169	1117364	218	0	80	459	153	1403553	183	0	86	500	129	855155	131	0	101	442	203	1319684	243	0	89
util	10	0	65	0	9	9	10	0	55	0	9	9	11	0	62	0	9	9	10	1	74	0	9	9
waste	248	38	3329	55	2	94	240	36	2908	36	2	99	246	28	2444	26	2	85	245	39	3131	31	2	88
watercontamination0202	439	132	130903	1036	50	64	452	149	123455	910	50	66	521	16	57183	81	50	67	431	132	82242	1118	50	62
watercontamination0202r	128	46	1145	17	0	21	130	47	1001	20	0	4	129	45	823	15	0	15	132	50	929	12	0	0
watercontamination0303	837	220	52651	1685	90	178	864	216	53317	1774	90	177	940	88	32783	415	90	179	811	279	35224	1792	90	180
watercontamination0303r	189	39	5652	5	0	25	194	30	4108	2	0	19	201	44	2978	16	0	19	166	11	4220	0	0	0
waterndl	28	5	463	4	0	13	27	9	468	9	0	13	34	3	364	2	0	15	31	8	629	14	0	11
waterndl2	103	32	4455	28	0	59	107	34	5241	53	0	60	112	12	5086	10	0	63	100	34	5718	37	0	62
waterno2_01	46	17	1339	27	21	34	44	14	1092	27	21	33	44	14	561	18	21	33	44	16	1319	36	21	30
waterno2_02	98	56	3246	47	13	48	98	55	2871	38	13	52	100	38	1740	28	13	54	99	63	3773	76	13	52
waterno2_03	144	86	5643	130	15	72	142	85	4691	164	15	73	143	61	3787	88	15	76	138	85	5921	208	15	73
waterno2_04	170	95	7388	89	20	82	173	99	8557	153	20	85	174	64	4771	40	20	83	168	100	8331	100	20	85
waterno2_06	259	152	14046	123	17	130	267	163	13398	205	17	126	271	119	8017	86	17	139	258	165	13099	119	17	140
waterno2_09	402	221	24408	181	28	216	399	223	22416	143	28	221	406	151	13282	128	28	218	397	241	20470	160	28	215
waterno2_12	532	304	31068	225	43	273	527	301	31638	177	43	284	525	205	17886	194	43	290	526	328	29869	244	43	274
waterno2_18	801	456	52333	344	60	426	802	460	49503	280	60	432	814	324	26333	222	60	412	787	489	47066	417	60	435
waterno2_24	1105	620	70969	444	72	576	1090	621	64758	489	72	586	1116	416	38669	310	72	583	1095	680	68732	558	72	601
watersym1	110	18	6045	3	86	92	114	24	5601	3	86	89	116	12	5259	1	86	90	113	23	5332	9	86	87

Table 5 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
watertreatnd_conc	59	12	384	10	5	23	54	11	398	8	5	24	59	9	426	1	5	25	50	14	284	8	5	22
watertreatnd_flow	64	21	2003	35	7	47	64	22	1692	45	7	46	65	16	1794	12	7	48	64	22	2048	42	7	48
waterx	91	29	3788	16	2	15	89	35	3602	10	2	20	101	24	2267	2	2	17	89	44	4537	33	2	16
waterz	76	27	839	44	2	29	77	29	722	32	2	21	84	15	825	19	2	28	78	37	881	50	2	26

Table 6: Detailed results comparing ordering strategies at the root node on test set GO. See Table 1 for aggregated results.

lp — number of LPs solved during OBBT
lp_{filt} — number of LPs solved during OBBT's filtering
iter — number of LP iterations during OBBT
$\text{iter}_{\text{filt}}$ — number of LP iterations of OBBT's filtering LPs
b_{obbt} — number of bounds tightened by OBBT
lvb — number of LVBs found by OBBT

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$\text{iter}_{\text{filt}}$	b_{obbt}	lvb
alkylation	15	2	104	0	9	11	15	1	106	0	9	11	15	3	87	3	9	10	15	3	98	4	9	11
alkyl	20	2	108	1	13	15	21	2	105	1	13	15	20	0	93	0	13	15	20	3	109	2	13	15
arki0015	2883	979	1539968	498	394	1940	2947	990	1042378	959	394	1948	2917	626	515010	365	394	1895	3083	1132	1517341	2752	394	1862
arki0018	20130	4894	1390320	1305	329	14039	17949	4746	1326615	1199	329	14039	19668	2682	191737	2682	329	15021	23942	3543	1466425	6	329	14173
arki0019	1274	169	3709	364	0	727	1233	212	3609	508	0	741	1547	0	4858	0	0	708	1060	71	1768	138	0	554
arki0020	3464	410	8552	856	0	1792	3327	497	7655	1010	0	1788	3771	1	11474	1	0	1734	2346	627	7004	1618	0	1755
bayes2_10	82	38	944	14	7	63	81	40	818	26	7	60	89	44	638	35	7	62	82	49	845	21	7	61
bayes2_20	83	39	1486	11	7	74	84	44	1379	11	7	74	79	32	933	18	7	73	82	45	1509	19	7	75
bayes2_30	96	46	1956	15	7	87	98	52	1885	22	7	88	94	44	1072	17	7	88	98	56	1741	17	7	88
bayes2_50	84	27	3237	17	7	77	84	27	3233	6	7	78	87	24	1627	11	7	75	86	31	3379	35	7	72
bearing	23	8	310	9	4	16	21	8	380	8	4	15	23	7	233	2	4	16	22	6	418	2	4	13
btest14	91	11	5888	2	16	55	87	2	6287	0	16	56	95	7	2082	6	16	56	86	14	7850	28	16	54
camshape100	129	43	2893	1	61	117	131	46	783	0	61	120	160	0	483	0	61	98	128	44	4028	0	61	120
camshape200	246	96	7634	6	113	223	249	95	2193	1	113	227	316	0	973	0	113	171	240	94	12777	2	113	227
camshape400	468	201	24414	2	218	447	479	199	7484	6	218	444	626	0	2030	0	218	335	473	200	42185	1	218	447
camshape800	930	407	87721	10	414	874	936	401	22068	18	414	864	1244	0	4046	0	414	695	946	411	164359	11	414	875
catmix100	501	0	0	0	0	0	501	0	0	0	0	0	501	0	0	0	0	0	501	0	0	0	0	0
catmix200	1001	0	0	0	0	0	1001	0	0	0	0	0	1001	0	0	0	0	0	1001	0	0	0	0	0
catmix400	2001	0	0	0	0	0	2001	0	0	0	0	0	2001	0	0	0	0	0	2001	0	0	0	0	0
catmix800	4001	0	0	0	0	0	4001	0	0	0	0	0	4001	0	0	0	0	0	4001	0	0	0	0	0
chain100	447	109	59201	0	68	302	486	100	40307	0	68	302	507	32	15186	0	68	302	388	115	45107	0	68	302
chain200	904	219	151272	2	155	603	971	206	104730	1	155	603	1010	92	25427	0	155	603	838	220	138530	0	155	603
chain50	240	55	25460	14	51	154	236	46	17228	8	51	154	263	22	8439	7	51	154	228	62	32074	13	51	154
chance	9	0	22	0	8	8	9	0	23	0	8	8	9	0	15	0	8	8	8	0	21	0	8	8
chem	50	4	451	2	13	39	50	4	321	1	13	40	53	6	237	5	13	39	49	5	476	1	13	40
elec200	20893	412	99162	12909	0	0	20782	492	47218	14781	0	0	20995	262	143948	12073	0	0	20677	625	20739	7597	0	0
elec25	423	48	5097	650	0	0	424	72	3889	928	0	0	438	34	5647	408	0	0	395	71	1306	485	0	0
elec50	1481	104	17299	3256	0	0	1454	123	12398	3566	0	0	1487	95	20714	3756	0	0	1423	106	3998	1009	0	0
etamac	53	4	181	12	8	17	57	4	200	12	8	17	56	3	186	9	8	17	53	4	211	12	8	17
ex14_1_1	4	0	38	0	4	4	4	0	35	0	4	4	4	0	30	0	4	4	4	0	46	0	4	4
ex14_1_2	10	0	108	0	4	7	10	1	118	0	4	7	10	0	85	0	4	7	10	1	119	0	4	7
ex14_1_3	4	0	27	0	2	4	4	0	25	0	2	4	4	0	25	0	2	4	4	1	13	0	2	4
ex14_1_5	20	2	68	0	16	1	21	2	44	0	16	1	21	2	42	0	16	1	22	3	56	1	16	1
ex14_1_6	13	1	56	2	4	7	12	1	49	2	4	7	14	0	32	0	4	10	13	0	94	0	4	9
ex14_1_7	72	35	1171	52	2	15	72	36	831	46	2	14	76	23	743	29	2	17	74	37	1127	39	2	13
ex14_1_8	12	0	40	0	4	9	11	0	46	0	4	9	12	0	41	0	4	9	11	0	73	0	4	9
ex14_1_9	2	0	4	0	0	0	2	0	4	0	0	0	2	0	5	0	0	1	2	0	4	0	0	0
ex14_2_4	83	0	56	0	6	7	83	0	82	0	6	7	82	0	45	0	6	7	82	0	92	0	6	7
ex14_2_7	86	0	33	0	0	4	87	0	32	0	0	4	89	0	47	0	0	4	85	0	40	0	0	4
ex14_2_9	24	0	256	0	13	18	26	1	258	1	13	18	23	0	74	0	13	18	23	0	258	0	13	18
ex2_1_10	23	0	47	0	21	22	23	0	51	0	21	22	22	0	37	0	21	22	22	0	44	0	21	22
ex2_1_1	6	0	5	0	0	1	6	0	6	0	0	1	6	0	5	0	0	1	5	0	4	0	0	1
ex2_1_7	22	0	244	0	20	20	22	0	228	0	20	20	26	0	249	0	20	20	20	0	227	0	20	20
ex2_1_9	11	0	44	0	10	10	12	1	48	1	10	10	19	0	67	0	10	10	10	0	48	0	10	10
ex3_1_1	12	1	39	1	7	3	13	0	38	0	7	2	13	0	35	0	7	3	12	0	40	0	7	3
ex3_1_2	7	0	13	0	7	7	7	0	13	0	7	7	7	0	15	0	7	7	7	0	15	0	7	7
ex3_1_4	2	0	5	0	0	1	3	0	4	0	0	1	3	0	5	0	0	1	3	0	4	0	0	1

Table 6 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
ex3pb	8	0	82	0	0	2	8	0	57	0	0	2	8	0	52	0	0	2	8	1	98	0	0	2
ex4_1_1	2	0	20	0	1	2	2	0	20	0	1	2	2	0	16	0	1	2	2	0	20	0	1	2
ex4_1_2	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
ex4_1_3	2	0	19	0	1	1	2	0	19	0	1	1	2	0	8	0	1	1	2	0	19	0	1	1
ex4_1_4	2	0	17	0	0	2	2	0	17	0	0	2	2	0	16	0	0	2	2	0	17	0	0	2
ex4_1_5	3	0	9	0	0	0	3	0	9	0	0	0	3	0	9	0	0	0	4	0	13	0	0	0
ex4_1_6	2	0	7	0	1	1	2	0	7	0	1	1	2	0	11	0	1	1	2	0	7	0	1	1
ex4_1_7	2	0	16	0	2	2	2	0	16	0	2	2	2	0	20	0	2	2	2	0	16	0	2	2
ex4_1_9	2	0	4	0	0	2	2	0	4	0	0	2	2	0	5	0	0	2	2	0	4	0	0	2
ex5_2_2_case1	5	0	20	0	3	4	5	0	19	0	3	5	5	0	14	0	3	3	4	0	17	0	3	4
ex5_2_2_case2	5	1	16	0	1	4	5	1	15	0	1	3	5	0	12	0	1	4	5	2	13	1	1	3
ex5_2_2_case3	5	0	21	0	3	4	5	0	20	0	3	5	5	0	14	0	3	3	4	0	18	0	3	4
ex5_2_4	9	2	27	2	0	1	7	1	23	2	0	2	9	0	32	0	0	1	6	0	31	0	0	3
ex5_2_5	24	4	133	3	0	12	27	8	129	9	0	6	27	7	159	4	0	7	27	6	124	5	0	9
ex5_3_2	12	6	40	9	2	6	13	5	33	4	2	7	13	3	37	2	2	6	14	5	46	8	2	5
ex5_3_3	58	29	346	79	1	31	59	28	351	55	1	25	57	10	324	9	1	29	57	29	447	80	1	30
ex5_4_2	13	2	49	2	6	8	13	0	42	0	6	8	12	0	46	0	6	7	11	0	43	0	6	7
ex5_4_3	13	1	58	1	8	6	14	2	53	2	8	7	14	1	60	1	8	6	15	4	64	5	8	4
ex5_4_4	22	4	146	3	6	7	24	6	136	9	6	8	24	3	148	5	6	4	23	8	151	12	6	6
ex6_1_2	10	1	25	0	2	7	10	0	26	0	2	7	9	0	11	0	2	7	9	2	15	1	2	7
ex6_1_4	14	1	88	0	3	10	15	1	97	0	3	8	13	0	65	0	3	10	14	2	94	0	3	9
ex6_2_10	83	38	1663	6	9	45	88	34	1530	17	9	36	90	28	1173	7	9	37	71	33	1579	11	9	50
ex6_2_11	36	4	272	1	6	19	35	5	349	2	6	18	37	1	179	0	6	19	34	3	249	2	6	16
ex6_2_12	45	13	206	2	0	25	47	18	149	2	0	25	42	15	154	0	0	31	44	16	158	5	0	27
ex6_2_13	64	21	181	23	0	28	60	26	196	25	0	27	62	13	231	15	0	28	59	22	317	19	0	29
ex6_2_5	68	46	460	35	0	44	66	43	970	43	0	46	64	38	491	54	0	45	59	40	509	30	0	43
ex6_2_6	26	0	304	0	19	22	28	0	157	0	19	22	27	0	195	0	19	22	27	0	370	0	19	22
ex6_2_7	71	21	1874	61	0	28	68	20	1752	60	0	27	73	11	1054	21	0	31	74	19	2559	27	0	29
ex6_2_8	25	0	350	0	20	24	26	0	310	0	20	24	28	0	197	0	20	25	27	1	410	0	20	24
ex6_2_9	61	27	687	34	6	37	60	24	873	339	6	38	68	24	660	28	6	33	57	29	482	39	6	35
ex7_2_2	9	0	68	0	3	7	9	0	65	0	3	7	10	0	49	0	3	7	9	0	88	0	3	7
ex7_2_3	26	9	173	11	4	12	26	10	153	1	4	13	27	5	233	1	4	16	26	10	225	4	4	11
ex7_2_4	17	1	184	0	5	11	16	2	186	0	5	10	19	1	169	0	5	11	18	2	211	0	5	10
ex7_3_1	19	0	76	0	10	10	19	0	49	0	10	10	19	0	31	0	10	10	19	0	50	0	10	10
ex7_3_2	10	0	9	0	7	7	10	0	7	0	7	7	10	0	8	0	7	7	10	0	7	0	7	7
ex7_3_3	5	0	4	0	4	4	5	0	3	0	4	4	5	0	4	0	4	4	5	0	3	0	4	4
ex7_3_4	37	2	23	0	15	15	37	2	29	1	15	15	37	2	24	1	15	15	37	2	28	1	15	15
ex7_3_5	35	3	53	1	4	4	35	3	47	1	4	4	36	3	49	2	4	4	35	3	53	2	4	4
ex7_3_6	3	0	3	0	3	3	3	0	6	0	3	3	3	0	3	0	3	3	3	0	14	0	3	3
ex8_1_3	15	0	36	0	0	0	15	0	28	0	0	0	15	0	29	0	0	0	15	0	51	0	0	0
ex8_1_4	4	0	1	0	0	0	4	0	1	0	0	0	4	0	1	0	0	0	4	0	1	0	0	0
ex8_1_5	4	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0	0	4	0	0	0	0	0
ex8_1_6	9	0	47	0	0	0	10	0	37	0	0	0	10	0	40	0	0	0	10	0	51	0	0	0
ex8_1_7	8	0	53	0	2	2	8	0	49	0	2	2	10	0	48	0	2	2	6	0	70	0	2	2
ex8_2_1b	71	0	342	0	21	26	73	0	298	0	21	26	74	0	246	0	21	26	70	0	372	0	21	26
ex8_2_2b	6332	88	40680	88	40	40	6652	104	29662	104	40	40	6074	93	30636	93	40	40	7317	62	33567	62	40	40
ex8_2_3b	13259	0	83246	0	77	85	15528	0	54682	0	77	85	13482	0	37854	0	77	85	13421	3	64821	3	77	85
ex8_2_4b	82	0	296	0	35	38	81	0	245	0	35	38	82	0	193	0	35	38	78	0	338	0	35	38
ex8_2_5b	2113	3	27002	3	52	54	2401	1	22645	1	52	54	2080	3	17797	3	52	54	2597	1	26910	1	52	54
ex8_3_11	126	18	246	15	7	34	125	15	239	14	7	35	129	8	261	9	7	40	118	21	235	17	7	32
ex8_3_12	98	13	158	17	7	30	97	10	195	15	7	34	98	6	231	6	7	38	95	7	182	8	7	26
ex8_3_13	142	26	341	20	7	68	141	38	345	37	7	66	149	26	314	20	7	65	133	46	331	44	7	64
ex8_3_1	122	13	223	16	7	37	126	14	264	16	7	33	127	10	237	9	7	38	127	15	262	14	7	30
ex8_3_5	85	11	228	12	7	34	92	22	228	26	7	39	88	4	233	5	7	33	90	14	231	13	7	34
ex8_3_7	117	24	203	32	7	48	121	28	174	36	7	45	123	23	259	30	7	48	125	39	212	48	7	48

Table 6 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
ex8_3_8	101	19	282	29	7	35	99	17	333	27	7	35	100	4	339	9	7	37	107	22	343	32	7	30
ex8_3_9	72	12	127	9	7	36	67	12	145	11	7	37	70	3	166	1	7	38	67	15	166	13	7	39
ex8_4_1	21	0	249	0	6	7	17	0	242	0	6	7	22	0	205	0	6	7	22	0	284	0	6	7
ex8_4_2	54	30	542	20	1	30	53	34	534	22	1	33	54	25	472	28	1	31	66	41	883	34	1	27
ex8_4_4	39	4	425	3	17	22	36	4	464	2	17	22	43	2	327	2	17	22	37	6	573	4	17	22
ex8_4_5	19	3	260	1	3	3	17	2	203	0	3	3	25	1	183	0	3	3	17	3	214	0	3	3
ex8_4_6	51	5	154	4	1	2	54	9	197	13	1	2	57	3	113	2	1	3	50	8	179	12	1	2
ex8_4_7	105	12	7119	4	48	93	113	18	5500	6	48	94	110	10	5196	4	48	93	111	22	12595	2	48	94
ex8_4_8_bnd	184	45	14545	13	71	158	187	46	13982	15	71	160	202	31	5245	8	71	159	190	50	19312	19	71	158
ex8_5_1	20	2	44	0	3	7	21	3	51	0	3	9	22	4	38	0	3	8	19	3	27	1	3	8
ex8_5_2	21	1	96	0	8	7	21	2	92	2	8	6	25	1	72	0	8	8	21	3	76	3	8	6
ex8_5_4	13	1	54	1	2	6	14	2	60	1	2	4	15	1	67	1	2	6	14	4	51	2	2	6
ex8_5_6	27	2	23	1	11	12	27	2	25	0	11	13	28	3	17	1	11	13	26	2	39	0	11	13
ex8_6_1	159	2	58	6	8	8	160	4	54	9	8	8	160	3	57	7	8	8	152	4	54	9	8	8
ex9_2_2	4	1	5	0	2	3	3	0	5	0	2	3	3	0	5	0	2	3	4	1	6	0	2	3
ex9_2_4	3	0	11	0	0	0	4	0	13	0	0	0	4	0	16	0	0	0	3	0	10	0	0	0
ex9_2_5	4	1	16	1	1	3	4	1	18	1	1	3	4	0	18	0	1	3	4	0	30	0	1	3
ex9_2_6	4	0	20	0	0	2	4	0	13	0	0	2	6	0	20	0	0	2	3	0	11	0	0	2
ex9_2_7	3	0	11	0	2	3	3	0	11	0	2	3	3	0	11	0	2	3	3	0	17	0	2	3
glider100	3363	674	6553	319	1	603	3391	603	4771	162	1	701	3394	598	3318	199	1	701	3181	625	5072	145	1	606
glider200	6712	1348	11449	1005	1	1208	6785	1216	9216	324	1	1401	6794	1198	6777	399	1	1401	6372	1246	10043	364	1	1203
glider400	13460	2699	21717	1517	1	2403	13561	2412	20306	643	1	2801	13594	2398	12991	799	1	2801	12742	2506	20239	919	1	2403
glider50	1673	324	2753	135	1	303	1685	298	2396	79	1	351	1694	298	1713	99	1	351	1592	308	2493	95	1	304
gsg_0001	104	30	2071	73	50	90	103	27	2517	68	50	90	100	23	854	52	50	90	108	32	3209	79	50	90
hhfair	25	5	28	1	6	11	25	5	27	1	6	10	28	3	35	2	6	10	26	5	22	0	6	13
himmel11	7	0	11	0	7	7	7	0	11	0	7	7	7	0	11	0	7	7	7	0	14	0	7	7
himmel16	15	3	85	5	0	0	16	3	88	6	0	0	18	0	91	0	0	0	13	0	104	0	0	0
house	9	0	14	0	6	8	8	0	8	0	6	8	8	0	9	0	6	8	8	1	6	0	6	8
hs62	9	0	28	0	8	9	9	0	20	0	8	9	9	0	22	0	8	9	9	1	28	0	8	9
hybriddynamic_fixedcc	46	1	110	0	26	27	45	1	76	0	26	27	46	0	70	0	26	27	44	1	118	0	26	27
hybriddynamic_varcc	102	9	2770	10	37	40	94	2	2452	1	37	40	107	3	1353	2	37	40	98	7	2834	24	37	40
infeas1	2830	55	3307309	19	2631	2707	2839	53	805051	12	2631	2709	2847	31	191666	18	2631	2710	2846	55	7056532	13	2631	2709
jbearing100	5001	0	2500	0	0	0	5001	0	2500	0	0	0	5001	0	2500	0	0	0	5001	0	2501	0	0	0
jbearing25	1251	0	625	0	0	0	1251	0	625	0	0	0	1251	0	625	0	0	0	1251	0	626	0	0	0
jbearing50	2501	0	1250	0	0	0	2501	0	1250	0	0	0	2501	0	1250	0	0	0	2501	0	1251	0	0	0
jbearing75	3751	0	1875	0	0	0	3751	0	1875	0	0	0	3751	0	1875	0	0	0	3751	0	1876	0	0	0
kall_circles_c6a	22	2	62	1	7	12	23	3	78	3	7	13	25	3	94	2	7	9	21	3	55	3	7	9
kall_circles_c6b	21	2	43	1	7	11	23	3	81	3	7	11	25	3	107	2	7	10	22	2	46	1	7	12
kall_circles_c6c	26	4	76	2	8	16	25	4	87	4	8	11	27	1	98	0	8	11	24	1	67	0	8	14
kall_circles_c7a	24	3	163	1	3	13	25	3	204	2	3	12	25	1	109	1	3	10	25	4	184	1	3	11
kall_circles_c8a	28	3	225	10	9	12	31	2	222	9	9	11	29	3	151	7	9	11	27	5	183	11	9	12
kall_circlespolygons_c1p12	27	7	142	2	2	2	29	4	181	2	2	2	28	4	164	4	2	2	28	5	190	0	2	2
kall_circlespolygons_c1p13	27	5	179	2	2	2	27	5	191	6	2	2	27	3	163	2	2	2	27	2	258	0	2	2
kall_circlespolygons_c1p5a	93	22	882	20	1	1	99	31	840	70	1	1	100	39	700	59	1	1	96	13	1199	5	1	1
kall_circlespolygons_c1p5b	576	112	7315	97	0	1	592	89	6661	140	0	1	587	197	5343	244	0	1	579	92	10195	76	0	0
kall_circlespolygons_c1p6a	835	210	11086	184	0	1	881	133	9950	144	0	0	876	307	8928	428	0	1	857	126	15048	98	0	0
kall_circlesrectangles_c1r12	32	4	145	5	2	2	30	6	115	7	2	2	33	4	129	4	2	2	31	6	195	2	2	2
kall_circlesrectangles_c1r13	32	8	184	7	0	1	32	6	186	6	0	1	31	4	133	3	0	2	32	3	201	4	0	1
kall_circlesrectangles_c6r1	140	25	1134	25	0	2	142	29	1183	20	0	6	154	42	1016	51	0	4	139	19	1460	6	0	2
kall_circlesrectangles_c6r29	297	62	2650	61	0	4	309	57	2435	67	0	4	313	77	2181	78	0	5	304	37	3492	27	0	3
kall_circlesrectangles_c6r39	485	84	4927	64	0	4	506	95	4504	133	0	7	512	142	3781	152	0	6	493	68	6169	36	0	6
kall_congruentcircles_c31	11	2	27	2	7	8	11	2	30	2	7	8	11	2	26	2	7	8	11	3	21	3	7	8
kall_congruentcircles_c32	11	0	35	0	1	3	10	0	26	0	1	3	11	1	39	1	1	2	10	3	42	1	1	4
kall_congruentcircles_c41	7	0	11	0	3	1	7	1	11	1	3	1	7	0	9	0	3	1	7	1	13	0	3	3
kall_congruentcircles_c42	14	3	65	4	2	6	14	3	50	4	2	4	14	4	42	2	2	8	13	3	57	4	2	5

Table 6 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
kall_congruentcircles_c51	19	5	53	6	5	7	19	5	57	7	5	8	19	5	47	5	5	8	19	6	45	6	5	8
kall_congruentcircles_c52	15	3	83	4	3	5	16	5	70	6	3	7	15	4	53	4	3	3	15	6	79	8	3	5
kall_congruentcircles_c61	21	4	65	3	6	12	22	7	61	8	6	12	22	5	64	5	6	10	21	6	49	4	6	10
kall_congruentcircles_c62	19	5	87	6	4	5	20	8	91	21	4	1	15	4	73	6	4	2	18	7	112	8	4	5
kall_congruentcircles_c63	20	2	123	1	9	10	21	2	63	1	9	13	20	2	86	1	9	9	20	3	120	3	9	10
kall_congruentcircles_c71	24	4	182	4	0	13	24	6	142	4	0	13	26	5	109	5	0	7	24	6	156	5	0	12
kall_congruentcircles_c72	22	7	147	6	5	8	22	6	115	7	5	3	21	4	105	8	5	4	21	8	129	9	5	2
kall_diffcircles_10	25	2	99	2	1	2	23	1	85	0	1	5	29	2	78	3	1	4	25	1	121	1	1	6
kall_diffcircles_5a	15	2	45	2	5	10	15	2	44	2	5	12	17	2	46	2	5	12	15	2	59	2	5	10
kall_diffcircles_5b	16	0	49	0	6	10	17	0	48	0	6	9	18	0	49	0	6	9	15	2	38	1	6	8
kall_diffcircles_6	15	1	28	1	2	3	14	0	48	0	2	7	18	2	35	3	2	4	16	1	51	0	2	5
kall_diffcircles_7	17	1	38	0	2	8	21	2	51	2	2	3	20	3	54	3	2	6	18	1	72	1	2	5
kall_diffcircles_8	18	0	40	0	8	8	15	0	34	0	8	8	21	0	73	0	8	8	17	1	50	1	8	8
kall_diffcircles_9	21	0	95	0	1	5	20	0	75	0	1	4	24	2	67	2	1	3	21	1	87	1	1	4
knp3-12	58	18	587	68	0	0	58	35	526	220	0	0	64	30	426	174	0	0	51	17	575	91	0	0
knp4-24	128	5	3499	48	0	0	130	4	3626	10	0	0	187	1	2548	1	0	0	140	6	5594	20	0	0
knp5-40	258	3	12162	4	0	0	273	9	13009	218	0	0	395	1	7223	1	0	0	249	20	14436	312	0	0
knp5-41	269	11	12676	73	0	0	266	16	13655	394	0	0	401	6	7873	148	0	0	245	31	14948	746	0	0
knp5-42	285	3	16648	14	0	0	291	5	17439	204	0	0	412	1	9755	3	0	0	309	20	24269	674	0	0
knp5-43	281	2	16310	86	0	0	192	9	11075	51	0	0	424	0	9888	0	0	0	206	12	14736	254	0	0
knp5-44	222	7	16463	55	0	0	301	3	21749	4	0	0	434	0	12542	0	0	0	222	7	23891	162	0	0
launch	80	10	416	12	49	59	83	8	257	16	49	57	85	1	170	2	49	62	76	11	193	5	49	59
least	18	3	50	0	0	6	16	2	53	1	0	4	21	3	58	0	0	6	15	2	49	1	0	5
like	697	32	769	31	0	1	697	31	645	30	0	2	697	32	661	31	0	1	757	91	991	0	0	92
linear	26	0	127	0	10	16	23	0	86	0	10	16	30	0	75	0	10	16	25	0	160	0	10	16
mathopt5_7	2	0	15	0	0	1	2	0	15	0	0	1	2	0	8	0	0	1	2	0	15	0	0	1
mathopt5_8	2	0	18	0	0	1	2	0	18	0	0	1	2	0	7	0	0	1	2	0	18	0	0	1
maxmin	150	48	886	195	66	6	154	54	909	146	66	8	148	39	709	95	66	10	174	22	815	46	66	11
methanol100	7590	0	1336	0	0	0	7590	0	1526	0	0	0	7590	0	199	0	0	0	7590	0	2170	0	0	0
methanol50	3966	0	794	0	0	0	3966	0	976	0	0	0	3966	0	224	0	0	0	3966	0	961	0	0	0
minlphi	19	3	63	1	7	5	18	4	61	0	7	4	18	5	44	1	7	5	19	3	88	1	7	5
oaer	4	0	12	0	2	4	4	0	7	0	2	4	4	0	13	0	2	4	4	0	6	0	2	4
pindyck	174	26	10199	0	112	124	174	26	7222	0	112	124	174	26	2862	0	112	124	174	26	8538	0	112	124
pinene50	1226	0	1233	0	0	0	1226	0	1233	0	0	0	1226	0	1233	0	0	0	1226	0	1233	0	0	0
pointpack04	12	5	25	5	2	5	13	3	31	4	2	4	14	0	27	0	2	4	12	4	32	4	2	4
pointpack06	18	7	39	6	1	5	20	5	53	6	1	6	21	3	38	1	1	8	16	5	67	4	1	5
pointpack08	22	3	75	1	0	6	22	11	59	5	0	8	25	6	61	3	0	9	23	9	90	3	0	10
pointpack10	29	10	86	9	0	8	26	12	70	6	0	8	31	13	62	9	0	12	33	18	126	29	0	6
pointpack12	34	17	95	16	0	14	36	20	94	10	0	15	40	19	97	16	0	13	36	17	109	19	0	12
pointpack14	40	17	105	10	0	18	41	21	105	16	0	16	45	22	107	23	0	19	39	22	94	24	0	9
portfol_buyin	12	1	69	0	7	12	13	2	84	2	7	10	13	1	73	2	7	10	11	1	98	0	7	11
powerflow0009r	64	1	350	1	0	2	63	3	374	3	0	2	86	3	315	3	0	2	56	4	509	4	0	2
powerflow0014r	124	16	1600	22	1	17	129	11	1616	2	1	16	148	14	1198	6	1	19	108	17	1523	7	1	16
powerflow0030r	218	4	1488	3	0	6	216	2	1525	1	0	8	291	2	1334	3	0	7	203	5	1937	2	0	8
powerflow0039r	339	0	2218	0	0	3	332	3	1969	3	0	4	441	2	1720	1	0	4	273	4	2526	4	0	3
powerflow0057r	457	47	6459	40	0	42	437	44	6288	22	0	41	522	36	4962	11	0	40	413	60	7746	39	0	36
powerflow0118r	1041	91	19254	34	0	96	1038	77	19913	30	0	87	1277	54	17492	21	0	105	976	95	25292	39	0	95
powerflow0300r	2498	229	45803	95	0	238	2487	218	44621	87	0	243	3101	166	39841	47	0	257	2345	249	55851	129	0	238
powerflow2383wpr	17314	305	85369	201	0	502	17140	273	92481	156	0	526	23076	280	74025	168	0	538	13706	385	119174	268	0	494
powerflow2736spr	20515	556	84288	356	0	872	20498	460	92504	263	0	888	27638	553	70238	283	0	911	16556	744	105515	575	0	850
prob06	2	0	6	0	2	2	2	0	6	0	2	2	3	0	6	0	2	2	2	0	2	0	2	2
process	16	2	119	0	8	10	16	2	164	0	8	10	16	1	101	0	8	10	16	2	195	0	8	10
procsyn	23	0	51	0	1	0	28	1	56	1	1	0	32	0	88	0	1	0	29	0	87	0	1	0
prolog	8	0	60	0	4	4	8	0	61	0	4	4	9	0	50	0	4	4	7	0	42	0	4	4
qp3	147	1	350	1	114	129	149	1	351	1	114	133	145	1	258	0	114	131	146	1	366	0	114	129

Table 6 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
rocket100	1143	530	4368	428	1	283	1327	409	2030	257	1	320	1114	108	2782	192	1	104	1047	466	2642	140	1	369
rocket200	2338	1018	8855	601	1	703	2338	710	5161	824	1	431	2443	439	4550	205	1	432	2025	945	5486	297	1	744
rocket400	4450	2266	12444	1194	1	1313	5623	1934	7578	1102	1	1292	4748	745	10075	948	1	714	4196	2054	10893	697	1	1495
rocket50	611	237	2375	155	1	150	656	197	1028	127	1	144	560	54	1359	55	1	50	508	215	1296	60	1	181
shiporig	21	0	16	0	0	0	21	0	15	0	0	0	21	0	19	0	0	0	24	2	29	5	0	0
st_bpaf1a	13	0	19	0	13	13	14	0	24	0	13	13	14	0	19	0	13	13	13	0	22	0	13	13
st_bsj2	5	0	12	0	4	4	5	0	8	0	4	4	5	0	9	0	4	4	5	0	11	0	4	3
st_bsj4	7	0	9	0	6	6	7	0	8	0	6	6	7	0	6	0	6	6	6	0	7	0	6	6
st_e03	16	3	167	1	8	10	16	2	146	1	8	10	16	2	126	1	8	10	15	2	212	0	8	10
st_e04	7	1	64	1	2	5	7	2	69	2	2	5	7	1	83	1	2	5	9	3	53	2	2	5
st_e05	8	1	11	0	5	5	8	1	11	0	5	5	8	2	10	0	5	5	8	1	15	0	5	5
st_e07	5	0	16	0	1	3	5	0	12	0	1	3	5	0	17	0	1	3	6	1	21	1	1	4
st_e08	4	0	8	0	4	4	4	0	7	0	4	4	4	0	7	0	4	4	4	0	10	0	4	4
st_e09	4	0	3	0	4	4	4	0	2	0	4	4	4	0	2	0	4	4	4	0	4	0	4	4
st_e11	4	1	4	0	0	0	4	1	3	0	0	0	4	1	2	0	0	0	4	1	3	0	0	0
st_e16	15	1	53	0	9	7	14	1	57	0	9	7	16	0	72	0	9	6	15	1	85	0	9	6
st_e19	4	1	15	1	0	0	4	1	17	1	0	0	4	0	17	0	0	0	4	1	15	1	0	0
st_e22	4	0	5	0	3	3	4	0	4	0	3	3	4	0	3	0	3	3	4	0	6	0	3	3
st_e24	3	0	3	0	3	3	3	0	3	0	3	3	3	0	2	0	3	3	3	0	3	0	3	3
st_e25	7	0	13	0	6	2	7	0	16	0	6	2	8	0	18	0	6	2	7	0	22	0	6	2
st_e26	4	1	3	0	2	4	4	1	2	0	2	4	4	1	2	0	2	4	4	1	3	0	2	4
st_e28	7	0	11	0	7	7	7	0	11	0	7	7	7	0	11	0	7	7	7	0	14	0	7	7
st_e30	7	0	25	0	2	2	7	0	36	0	2	2	7	0	29	0	2	2	6	0	28	0	2	2
st_e33	5	0	14	0	1	3	5	0	10	0	1	4	5	0	14	0	1	3	5	0	19	0	1	3
st_e41	17	3	59	3	11	15	16	4	67	5	11	15	17	1	52	0	11	15	16	3	105	3	11	15
st_fp7a	21	0	225	0	20	20	22	0	239	0	20	20	28	0	235	0	20	20	20	0	220	0	20	20
st_fp7b	21	0	235	0	20	20	22	0	235	0	20	20	24	0	224	0	20	20	20	0	228	0	20	20
st_fp7c	21	0	225	0	20	20	22	0	239	0	20	20	28	0	235	0	20	20	20	0	220	0	20	20
st_fp7d	21	0	225	0	20	20	22	0	239	0	20	20	28	0	235	0	20	20	20	0	220	0	20	20
st_fp7e	22	0	244	0	20	20	22	0	228	0	20	20	26	0	249	0	20	20	20	0	227	0	20	20
st_glmp_fp1	4	0	13	0	4	2	4	0	12	0	4	2	4	0	5	0	4	2	4	0	13	0	4	2
st_glmp_fp2	3	0	10	0	3	2	3	0	10	0	3	2	3	0	12	0	3	2	3	0	10	0	3	2
st_glmp_fp3	4	0	35	0	3	0	4	0	27	0	3	0	4	0	13	0	3	0	4	0	35	0	3	0
st_glmp_kk90	3	0	3	0	3	3	3	0	3	0	3	3	3	0	5	0	3	3	3	0	3	0	3	3
st_glmp_kk92	4	0	26	0	2	3	4	0	19	0	2	3	4	0	18	0	2	3	4	0	26	0	2	3
st_glmp_kky	3	0	4	0	3	3	3	0	4	0	3	3	3	0	6	0	3	3	3	0	4	0	3	3
st_glmp_ss1	3	0	7	0	2	2	3	0	7	0	2	2	3	0	7	0	2	2	3	0	4	0	2	2
st_glmp_ss2	3	0	3	0	3	3	3	0	3	0	3	3	3	0	5	0	3	3	3	0	8	0	3	3
st_iqpbk1	16	0	175	0	2	7	15	1	122	0	2	7	15	0	99	0	2	6	15	0	228	0	2	7
st_iqpbk2	16	0	151	0	2	7	16	0	119	0	2	7	16	0	66	0	2	7	14	0	185	0	2	7
st_jcbpaf2	13	0	52	0	12	12	13	0	77	0	12	12	16	1	57	1	12	12	12	0	75	0	12	12
st_pan1	5	0	13	0	1	4	5	0	10	0	1	4	5	0	6	0	1	4	5	0	13	0	1	4
st_ph14	4	0	4	0	4	4	4	0	4	0	4	4	4	0	4	0	4	4	4	0	4	0	4	4
st_ph15	7	0	23	0	4	4	7	0	19	0	4	4	7	0	13	0	4	4	6	0	14	0	4	4
st_ph1	7	0	13	0	4	6	7	0	14	0	4	6	7	0	10	0	4	6	8	0	16	0	4	6
st_ph20	3	0	9	0	3	2	3	0	7	0	3	2	3	0	7	0	3	2	3	0	7	0	3	2
st_ph2	7	0	13	0	4	5	7	0	14	0	4	6	7	0	10	0	4	6	8	0	15	0	4	6
st_ph3	5	0	7	0	5	5	5	0	8	0	5	5	5	0	7	0	5	5	5	0	10	0	5	5
st_phex	4	0	7	0	4	2	4	0	6	0	4	2	4	0	4	0	4	2	4	0	8	0	4	2
st_qpc-m0	3	0	3	0	2	0	3	0	3	0	2	0	3	0	3	0	2	0	4	0	5	0	2	0
st_qpc-m1	8	0	33	0	6	6	8	0	41	0	6	6	9	0	33	0	6	6	7	0	35	0	6	6
st_qpc-m3a	11	0	158	0	1	8	13	1	118	2	1	7	13	0	98	0	1	6	14	3	120	9	1	5
st_qpc-m3b	17	6	160	48	7	10	17	6	164	48	7	10	17	6	148	36	7	9	11	0	107	0	7	9
st_qpk1	3	0	4	0	2	0	3	0	4	0	2	0	3	0	4	0	2	0	4	0	7	0	2	0
st_qpk2	7	0	27	0	0	6	7	0	15	0	0	6	10	1	27	0	0	6	6	0	25	0	0	6

Table 6 continued

	random						min–max						greedy (base)						reverse greedy					
Instance	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb	lp	lp_{filt}	iter	$iter_{filt}$	b_{obbt}	lvb
st_qpk3	12	0	64	0	0	11	12	0	90	0	0	11	21	1	80	0	0	11	12	1	88	0	0	11
st_rv1	12	0	35	0	10	12	12	0	33	0	10	12	12	0	25	0	10	12	12	0	31	0	10	12
st_rv2	23	0	148	0	21	21	22	0	157	0	21	21	24	0	130	0	21	21	21	0	139	0	21	21
st_rv3	25	1	223	0	20	23	24	1	226	0	20	23	28	1	174	0	20	23	24	1	220	0	20	23
st_rv7	35	0	329	0	29	32	33	0	333	0	29	32	37	0	294	0	29	32	33	0	344	0	29	32
st_rv8	44	0	697	0	39	43	43	0	668	0	39	43	47	0	637	0	39	43	43	0	763	0	39	43
st_rv9	52	0	690	0	42	50	53	0	776	0	42	50	58	0	632	0	42	50	51	0	718	0	42	50
st_z	5	0	12	0	5	4	5	0	8	0	5	4	6	0	10	0	5	4	5	0	8	0	5	4
supplychainp1_020306	15	0	255	0	7	7	15	0	264	0	7	7	15	0	262	0	7	8	15	0	225	0	7	7
syn05m	4	0	29	0	3	4	4	0	31	0	3	4	4	0	27	0	3	4	4	0	21	0	3	4
tricp	65	1	255	1	20	20	62	1	203	1	20	20	71	0	108	0	20	20	54	0	457	0	20	20
wall	16	0	1	0	0	0	16	0	1	0	0	0	16	0	1	0	0	0	16	0	1	0	0	0
wastewater02m1	12	1	36	0	9	12	12	0	45	0	9	12	13	0	67	0	9	11	12	1	46	0	9	12
wastewater02m2	15	6	110	0	1	13	15	6	78	0	1	14	15	3	95	0	1	15	15	8	66	0	1	15
wastewater04m1	18	5	61	1	6	13	18	6	88	6	6	16	18	4	109	0	6	15	18	6	72	4	6	14
wastewater04m2	23	7	108	3	10	16	23	6	110	1	10	19	23	5	97	2	10	18	23	6	139	0	10	17
wastewater05m1	25	5	124	4	3	12	26	4	123	2	3	14	26	2	199	0	3	13	25	4	98	2	3	12
wastewater05m2	43	18	1329	22	6	32	42	20	1185	12	6	33	43	9	1089	27	6	30	44	24	1170	36	6	33
wastewater11m1	74	4	1517	12	58	65	74	5	1703	21	58	64	75	3	1750	5	58	65	74	4	1579	22	58	65
wastewater11m2	88	25	3232	7	6	68	89	25	2253	9	6	70	89	16	3081	6	6	66	89	39	2267	48	6	74
wastewater12m1	135	4	3669	6	112	118	135	3	3162	3	112	120	133	3	6463	4	112	117	135	5	3028	6	112	121
wastewater12m2	153	38	4630	48	10	130	155	36	7079	18	10	120	154	21	6559	14	10	131	154	56	5080	89	10	133
wastewater13m1	277	7	10020	28	246	255	277	7	9872	12	246	253	276	4	12608	2	246	258	275	5	9531	14	246	257
wastewater13m2	284	38	16925	26	13	226	283	32	15941	48	13	222	282	10	20279	12	13	228	280	63	10043	136	13	261
wastewater14m1	50	6	978	24	30	42	52	7	932	13	30	42	51	3	971	0	30	42	50	5	902	9	30	44
wastewater14m2	51	11	2766	13	13	47	51	10	2406	8	13	43	51	6	3228	7	13	40	51	13	2553	8	13	48
wastewater15m1	28	3	276	0	12	21	28	3	309	0	12	22	28	3	381	1	12	22	29	7	378	23	12	21
wastewater15m2	37	11	1168	4	13	35	38	12	1061	7	13	34	37	6	1232	7	13	33	37	13	1373	16	13	32
waterund01	37	11	440	16	14	18	40	16	499	31	14	20	41	10	495	21	14	18	39	17	663	30	14	15
waterund08	88	23	4013	40	34	52	87	27	4045	36	34	54	98	21	2877	37	34	50	88	31	3751	55	34	45
waterund11	52	13	918	17	20	27	54	16	836	14	20	30	54	9	904	8	20	27	50	14	1128	14	20	24
waterund14	124	45	4303	84	41	62	134	54	4162	152	42	60	140	19	3434	25	42	67	112	45	4406	80	42	60
waterund17	68	26	1647	24	25	41	72	24	1648	35	25	40	78	10	1601	22	25	42	72	35	1580	52	25	41
waterund18	52	14	814	14	21	28	53	15	1000	20	21	27	62	12	876	12	21	31	51	20	1086	22	21	26
waterund22	120	46	2412	58	32	53	124	50	2361	100	32	54	128	32	2698	64	32	49	116	50	2717	78	32	51
waterund25	89	28	3789	47	34	44	101	34	4154	57	34	46	103	15	4021	23	34	47	96	36	5075	68	34	47
waterund27	289	107	57259	286	105	144	305	121	61276	293	105	141	321	91	61193	198	105	148	302	126	59765	496	105	140
waterund28	751	175	1103181	117	254	484	748	186	1121567	279	254	481	950	57	923371	48	254	488	759	193	1065053	147	254	494
waterund32	668	140	179095	184	180	374	663	150	147944	206	180	367	741	94	150635	159	180	360	675	128	180056	184	180	336
waterund36	289	103	57248	274	101	129	284	97	46912	279	101	135	334	84	54394	252	101	133	291	101	53513	210	101	133
weapons	183	41	2933	58	0	80	169	32	2777	20	0	78	182	37	2143	38	0	69	166	37	2503	28	0	67

A.2 Tree Experiments

Table 7: Detailed results comparing SCIP without OBBT, SCIP with OBBT only, and SCIP with OBBT and LVB propagation over MINLPLib2 instances with integral variables after presolving. See Table 2 for aggregated results.

n — number of branch-and-bound nodes
t — total running time
t_{obbt} — time used by OBBT
lp — LPs solved by OBBT
b_{obbt} — bounds tightened by OBBT
lvb — LVBs found by OBBT
t_{lvb} — time used by LVB propagation
b_{lvb} — bounds tightened by LVB propagation

	SCIP plain		SCIP+OBBT					SCIP+OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
alan	3	0.1	4	0.1	0.00	34	8	4	0.0	0.00	0.00	8	0	9
batch	8	0.2	16	0.3	0.01	259	16	16	0.2	0.02	0.01	16	72	24
batch0812	4	0.3	4	0.4	0.03	449	82	3	0.5	0.02	0.00	74	30	85
batch0812_nc	911	2.2	792	1.9	0.13	717	77	592	2.0	0.15	0.00	69	906	111
batchdes	1	0.1	1	0.0	0.00	55	9	1	0.1	0.01	0.00	9	7	14
batchs101006m	1631	9.8	2008	10.3	0.14	608	19	1584	9.2	0.14	0.04	19	963	25
batchs121208m	3852	17.6	3378	14.8	0.59	2629	23	3094	13.0	0.59	0.01	23	1897	48
batchs151208m	3932	28.2	2797	19.2	0.47	4439	35	5151	31.1	0.54	0.02	35	2036	75
batchs201210m	11277	60.8	6267	33.4	0.73	6477	33	7189	39.4	0.73	0.02	33	3593	78
blend029	29046	12.0	21241	9.6	0.02	305	2	33150	13.1	0.02	0.06	2	3974	18
blend721	1073801	717.4	1536191	1013.7	0.22	1739	0	580639	375.0	0.21	0.95	0	25950	61
carton7	595797	459.1	499975	367.8	0.14	2031	4	318913	229.8	0.13	0.53	4	2761	11
clay0203h	102	2.3	626	3.5	0.44	3469	0	2991	5.7	0.60	0.01	0	4472	233
clay0203m	20	0.2	20	0.2	0.01	128	0	20	0.2	0.01	0.01	0	0	0
clay0204h	1172	13.4	1172	13.6	0.75	3342	0	1021	11.9	0.75	0.00	0	4326	328
clay0204m	1086	1.2	1143	1.1	0.01	176	0	1145	1.2	0.01	0.00	0	1	1
clay0205h	12475	103.2	14493	112.0	1.31	6806	0	13605	131.5	1.51	0.10	0	49273	381
clay0205m	8107	3.9	8635	4.2	0.01	255	0	8717	4.0	0.01	0.01	0	0	1
clay0303h	395	6.8	395	7.1	0.46	4162	0	395	7.4	0.45	0.01	0	890	315
clay0303m	23	0.3	23	0.3	0.00	133	0	23	0.3	0.01	0.00	0	2	1
clay0304h	30328	412.8	26806	411.2	1.90	16058	0	18802	275.4	2.00	0.07	0	54488	414
clay0304m	535	1.0	610	1.1	0.01	195	0	652	0.9	0.01	0.00	0	3	4
clay0305h	9917	201.0	9415	216.8	3.73	22875	0	12022	276.9	3.67	0.06	0	5733	532
clay0305m	9493	5.3	9527	5.2	0.02	280	0	9537	5.3	0.01	0.00	0	0	1
crudeoil_lee1_05	8	1.1	12	1.2	0.24	1613	34	3	1.3	0.39	0.00	34	3	52
crudeoil_lee1_06	43	2.0	42	2.8	0.52	2310	21	45	2.9	0.63	0.00	21	53	64
crudeoil_lee1_07	56	4.0	40	5.5	0.75	3235	25	46	5.5	0.67	0.00	25	59	78
crudeoil_lee1_08	176	7.8	153	8.8	1.06	3416	32	161	9.0	1.06	0.00	32	175	96
crudeoil_lee1_09	52	7.3	33	8.3	1.39	4531	35	35	9.0	1.42	0.01	35	87	100
crudeoil_lee1_10	66	8.7	165	17.2	2.30	6101	43	165	18.6	2.46	0.02	43	257	117
crudeoil_lee2_05	12	2.0	12	4.9	2.53	11746	92	18	4.6	2.50	0.00	103	25	248
crudeoil_lee2_06	157	10.0	250	14.1	3.27	10239	55	257	14.1	3.55	0.00	55	337	119
crudeoil_lee2_07	192	11.0	330	17.8	4.95	10292	63	301	19.3	4.99	0.00	63	242	141
crudeoil_lee2_08	543	22.9	345	26.9	7.45	13794	74	455	27.9	7.69	0.02	74	229	174
crudeoil_lee2_09	517	28.6	730	39.6	13.14	17313	86	666	39.5	13.07	0.02	86	827	199
crudeoil_lee2_10	685	35.4	596	40.5	13.87	16925	97	775	52.4	13.83	0.01	97	483	229
crudeoil_lee3_05	409961	506.2	335361	430.6	1.48	5369	120	535451	692.0	1.50	7.57	120	1453824	192
crudeoil_lee4_05	43	4.0	66	6.4	2.23	8398	75	99	6.9	2.35	0.00	75	47	134
crudeoil_lee4_06	83	10.2	5	16.5	5.33	7834	100	5	15.7	5.13	0.00	100	3	165
crudeoil_lee4_07	88	9.8	195	26.8	9.24	13056	116	412	28.4	9.26	0.01	116	729	201
crudeoil_lee4_08	91	19.8	218	36.2	10.85	8190	109	303	33.6	10.15	0.01	109	190	213
crudeoil_lee4_09	128	24.9	83	42.5	16.56	13255	113	101	44.8	16.52	0.00	113	98	235
crudeoil_lee4_10	451	43.5	243	58.2	22.67	20971	156	383	59.1	23.02	0.00	156	491	294
crudeoil_li06	92305	966.2	51531	526.2	1.67	5289	14	44145	512.5	1.67	0.02	14	5949	24

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
du-opt	1808	2.8	1364	2.1	0.01	181	4	1323	2.6	0.00	0.02	4	295	5
du-opt5	64	0.5	66	0.5	0.00	163	4	57	0.5	0.00	0.01	4	35	5
edgecross10-060	42081	79.0	42081	79.1	0.09	1368	0	42081	79.6	0.09	0.02	0	0	0
edgecross14-039	28597	505.2	28579	508.6	0.42	5040	0	28579	505.2	0.42	0.04	0	0	0
elf	477	0.8	477	1.0	0.01	283	0	350	0.8	0.00	0.00	0	283	6
eniplac	232	1.2	232	1.1	0.01	397	0	232	1.2	0.01	0.00	0	54	21
enpro48pb	20	1.0	34	1.1	0.10	1127	32	32	1.2	0.14	0.01	33	25	44
enpro56pb	1318	3.0	546	2.1	0.08	547	8	563	2.3	0.09	0.01	8	93	14
ex1224	4	0.1	1	0.2	0.00	76	4	1	0.1	0.00	0.00	4	5	6
ex1243	131	1.1	131	1.1	0.00	57	0	135	1.1	0.00	0.00	0	81	15
ex1244	4154	4.4	3473	5.6	0.01	228	2	3306	5.8	0.01	0.01	2	855	10
ex1263	276	0.7	467	0.8	0.02	401	12	371	0.6	0.03	0.00	12	335	30
ex1263a	67	0.1	728	0.9	0.00	53	12	188	0.3	0.00	0.00	12	178	15
ex1264	253	0.6	153	0.4	0.00	154	8	103	0.2	0.00	0.00	8	112	14
ex1264a	66	0.1	137	0.1	0.00	61	9	145	0.2	0.01	0.01	9	118	12
ex1265	98	0.3	224	0.7	0.01	190	6	294	0.9	0.01	0.00	6	167	17
ex1265a	40	0.1	78	0.2	0.01	109	12	74	0.1	0.00	0.00	12	57	14
ex1266	339	1.0	289	1.2	0.08	1340	23	99	1.0	0.08	0.03	23	145	49
ex1266a	5	0.1	121	0.2	0.00	115	21	153	0.5	0.00	0.00	21	138	23
ex4	14	0.7	14	0.7	0.03	287	8	11	0.7	0.02	0.00	7	11	19
fac1	2	0.1	2	0.1	0.00	149	0	2	0.1	0.00	0.00	0	0	20
fac3	9	0.3	9	0.3	0.01	75	0	9	0.3	0.01	0.00	0	0	36
fin2bb	5000	230.5	18473	637.1	0.21	1215	61	5570	238.1	0.18	0.00	61	403	61
flay02h	7	0.1	7	0.1	0.00	156	2	7	0.1	0.02	0.00	2	3	2
flay02m	7	0.1	7	0.1	0.00	62	2	7	0.1	0.00	0.00	2	1	2
flay03h	111	0.9	103	0.7	0.04	512	3	103	0.8	0.03	0.00	3	2	4
flay03m	103	0.2	107	0.2	0.00	71	3	107	0.2	0.00	0.00	3	68	3
flay04h	2196	4.1	2335	4.1	0.02	256	4	2335	4.3	0.03	0.00	4	0	0
flay04m	2363	1.8	2385	2.0	0.00	106	0	2385	2.0	0.01	0.00	0	0	4
flay05h	106721	191.2	106443	191.0	0.06	631	0	99444	182.7	0.06	0.39	0	933	6
flay05m	95853	54.6	98881	55.3	0.01	118	0	98107	55.8	0.00	0.38	0	10317	5
flay06m	3609871	1793.5	3596393	1777.4	0.01	207	0	3814781	1878.0	0.01	9.60	0	2272908	6
fo7	168006	85.0	207840	98.1	0.01	194	0	207840	98.9	0.02	0.35	0	0	1
fo7_2	55500	24.5	53727	24.4	0.00	93	0	53727	24.5	0.00	0.03	0	0	1
fo7_ar25_1	36189	18.1	42734	19.1	0.00	116	0	42734	19.3	0.01	0.10	0	0	4
fo7_ar2_1	113648	51.2	142494	61.6	0.01	100	0	142494	62.1	0.00	0.22	0	0	14
fo7_ar3_1	52706	23.9	51514	22.4	0.01	92	0	51514	22.4	0.01	0.10	0	0	1
fo7_ar4_1	62643	31.3	62404	30.6	0.01	88	0	62404	30.6	0.01	0.12	0	0	5
fo7_ar5_1	35545	20.2	36539	23.4	0.01	120	0	36539	23.5	0.00	0.06	0	0	6
fo8	211923	120.9	220300	136.2	0.02	162	0	210539	124.4	0.01	0.26	0	1010	1
fo8_ar25_1	614176	249.1	605817	250.2	0.01	121	0	605817	242.8	0.00	0.85	0	0	7
fo8_ar2_1	443504	184.0	275619	113.0	0.01	94	0	275619	113.9	0.01	0.33	0	0	16
fo8_ar3_1	198579	106.4	162891	83.8	0.01	98	0	162891	83.1	0.01	0.19	0	0	4
fo8_ar4_1	180458	87.9	134526	70.6	0.00	139	0	134526	71.4	0.01	0.20	0	0	11
fo8_ar5_1	75792	41.9	110941	59.5	0.01	98	0	110941	59.5	0.01	0.18	0	0	6
fo9	2358400	1243.6	2078934	1127.7	0.02	242	0	2078934	1131.4	0.03	1.79	0	0	0
fo9_ar2_1	3667495	1263.0	3653351	1256.3	0.01	93	0	3653351	1261.5	0.01	4.12	0	0	18
fo9_ar3_1	330999	171.7	524507	242.0	0.01	168	0	524507	244.5	0.01	0.57	0	0	4
fo9_ar4_1	1909806	1097.2	1102537	581.4	0.02	161	0	1102537	574.0	0.02	1.36	0	0	7
fo9_ar5_1	1013002	640.5	775573	442.9	0.01	113	0	775573	444.2	0.02	1.18	0	0	8
fuel	1	0.1	1	0.0	0.00	83	4	1	0.0	0.00	0.00	4	3	9
gasprod_sarawak01	652	1.5	28	1.6	0.05	529	10	16	1.3	0.06	0.00	10	27	64
gastrans	112	0.7	6	0.3	0.05	931	46	6	0.2	0.07	0.00	46	1	59
gear	1	0.0	1	0.0	0.00	26	1	1	0.0	0.00	0.00	1	0	5
gear2	1	0.0	1	0.1	0.00	32	0	1	0.1	0.00	0.00	0	0	4

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
gear3	1	0.0	1	0.0	0.00	26	1	1	0.0	0.00	0.00	1	0	5
genpooling_lee1	4928	6.3	4928	6.4	0.00	126	0	5199	6.9	0.01	0.01	0	749	5
genpooling_lee2	155441	123.7	155441	128.2	0.00	120	0	86231	94.7	0.01	0.25	0	16751	9
ghg_1veh	7481	38.5	9901	45.3	0.01	315	3	10451	52.4	0.02	0.04	3	4804	20
gkocis	4	0.1	3	0.1	0.00	20	2	3	0.1	0.00	0.00	2	0	4
hybriddynamic_var	9011	8.2	3391	3.2	0.01	179	22	3131	3.1	0.02	0.00	22	1946	25
jit1	10	0.0	11	0.0	0.00	68	6	9	0.0	0.00	0.00	6	12	6
johnall	771	179.6	7	435.7	188.62	9799	5158	7	450.3	196.16	0.01	5158	1355	5259
kport20	187383	159.3	310503	274.6	0.02	816	8	268805	270.5	0.03	1.45	8	482559	35
lip	5	0.2	1	0.2	0.00	8	8	1	0.2	0.00	0.00	8	0	7
m3	19	0.2	19	0.2	0.01	35	0	19	0.2	0.00	0.00	0	0	2
m6	3817	2.4	3817	2.4	0.01	120	0	3817	2.4	0.01	0.00	0	0	3
m7	4499	4.3	4499	4.5	0.01	103	0	4499	4.7	0.01	0.01	0	0	1
m7_ar25_1	1689	2.0	1674	1.8	0.00	89	0	1674	1.9	0.01	0.01	0	0	3
m7_ar2_1	11338	6.6	10546	6.0	0.01	90	0	10546	6.1	0.01	0.04	0	0	4
m7_ar3_1	7729	4.8	7729	4.8	0.01	110	0	8349	5.8	0.02	0.01	0	33	6
m7_ar4_1	2679	3.7	2643	3.8	0.01	167	0	2643	3.7	0.02	0.02	0	0	7
m7_ar5_1	4616	5.0	10723	9.2	0.01	135	0	10723	9.5	0.01	0.01	0	0	5
meanvarxsc	6	0.2	5	0.2	0.00	70	14	5	0.2	0.00	0.00	14	1	26
netmod_dol1	52477	1813.5	56979	1900.8	0.01	12	0	57905	1914.9	0.01	0.06	0	419	4
netmod_dol2	209	71.0	209	71.8	0.70	5698	0	259	74.6	0.82	0.01	0	217	4
netmod_kar1	336	3.5	336	3.3	0.00	6	0	336	3.9	0.00	0.00	0	1	1
netmod_kar2	336	3.6	336	3.3	0.00	6	0	336	3.5	0.00	0.00	0	1	1
no7_ar25_1	246825	120.2	250005	116.7	0.01	125	0	250005	116.8	0.01	0.34	0	0	4
no7_ar2_1	68666	36.9	68395	34.4	0.01	113	0	68395	36.2	0.01	0.09	0	0	14
no7_ar3_1	613840	259.6	514058	221.1	0.00	135	0	514058	222.3	0.01	0.53	0	0	1
no7_ar4_1	279454	138.4	305424	152.5	0.01	140	0	305424	154.4	0.01	0.32	0	0	7
no7_ar5_1	115855	68.7	114540	62.1	0.01	156	0	114540	62.9	0.02	0.21	0	0	6
nous2	8651	6.7	6161	5.4	0.01	187	7	13231	10.7	0.01	0.03	7	4956	44
nvs01	9	0.1	8	0.1	0.00	114	3	9	0.1	0.01	0.00	3	4	5
nvs02	4	0.0	4	0.0	0.00	26	0	4	0.0	0.01	0.00	0	0	0
nvs06	17	0.1	11	0.1	0.00	35	5	11	0.1	0.00	0.00	5	4	8
nvs08	3	0.1	5	0.1	0.00	34	1	5	0.1	0.00	0.00	1	2	1
nvs09	17365951	3600.0	1	0.1	0.01	244	1	1	0.1	0.02	0.00	1	0	7
nvs10	1	0.0	1	0.0	0.00	8	4	1	0.0	0.00	0.00	4	0	4
nvs11	5	0.0	1	0.0	0.00	23	6	1	0.0	0.00	0.00	6	0	6
nvs12	5	0.1	5	0.1	0.00	33	8	5	0.0	0.00	0.00	8	0	8
nvs13	21	0.1	11	0.1	0.01	30	8	11	0.1	0.00	0.00	8	4	8
nvs15	6	0.0	6	0.0	0.00	25	2	6	0.0	0.00	0.00	2	0	2
nvs16	6	0.1	6	0.1	0.00	62	0	6	0.1	0.00	0.00	0	0	0
nvs17	45	0.2	47	0.2	0.00	42	11	43	0.2	0.00	0.00	11	55	11
nvs18	33	0.1	31	0.2	0.00	34	10	33	0.1	0.00	0.00	10	22	10
nvs19	105	0.4	99	0.3	0.00	59	12	81	0.4	0.01	0.00	12	97	12
nvs21	255	0.5	255	0.6	0.00	5	0	255	0.5	0.00	0.00	0	91	2
nvs23	117	0.6	109	0.6	0.00	50	13	101	0.5	0.00	0.00	13	67	13
nvs24	139	0.7	144	0.5	0.01	75	12	138	0.6	0.01	0.00	12	97	12
o7	4697648	1971.1	4752796	2020.9	0.00	95	0	4752796	2022.9	0.00	4.08	0	0	0
o7_2	1556312	625.6	1399982	560.8	0.01	129	0	1399982	560.8	0.01	1.05	0	0	0
o7_ar25_1	735468	333.7	811159	364.8	0.02	124	0	811159	366.3	0.01	1.30	0	0	4
o7_ar2_1	448027	206.7	398089	185.1	0.01	161	0	398089	185.8	0.02	0.65	0	0	14
o7_ar3_1	1549801	690.3	1425964	634.3	0.01	148	0	1425964	621.2	0.01	1.67	0	0	1
o7_ar4_1	2494812	1214.3	2284976	1072.1	0.01	131	0	2284976	1077.5	0.02	3.48	0	0	6
o7_ar5_1	736393	335.6	688302	340.6	0.01	131	0	688302	337.1	0.01	0.84	0	0	6
oil2	1140	11.3	2	5.0	2.19	3096	839	2	5.0	2.17	0.01	839	8	1163
ortez	9	0.5	4	0.4	0.05	477	27	4	0.4	0.03	0.00	8	13	36

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
portfol_card	8	0.3	6	0.3	0.00	130	18	5	0.3	0.00	0.00	18	15	32
portfol_classical050_1	21882	73.9	20437	67.7	0.01	192	59	19195	66.5	0.03	0.22	59	33442	61
portfol_robust050_34	272	8.7	310	6.4	0.13	658	314	324	6.4	0.09	0.00	314	322	348
portfol_robust100_09	3913	69.5	4237	73.0	0.31	570	334	4655	77.6	0.25	0.33	334	31939	338
portfol_shortfall050_68	1609	12.8	1085	10.9	0.05	277	188	1062	10.8	0.02	0.03	188	11355	200
procsel	1	0.0	1	0.0	0.00	8	2	1	0.0	0.00	0.00	2	0	2
product	2681	16.6	2681	17.3	0.22	5002	0	2331	16.2	0.22	0.00	0	148	27
product2	362	9.6	2	3.4	0.14	1665	157	2	3.4	0.15	0.00	157	4	214
ravempb	31	1.2	19	0.8	0.08	1064	3	19	0.8	0.08	0.00	3	6	11
risk2bpb	7	0.3	6	0.3	0.01	226	3	6	0.3	0.01	0.00	3	0	3
routingdelay_bigm	1041210	3600.1	4699	75.0	0.86	4288	138	145491	528.3	0.94	0.27	138	107747	360
rsyn0805h	7926888	3600.0	14	0.8	0.00	53	3	20	1.1	0.01	0.00	3	0	5
rsyn0805m	645	1.0	645	1.2	0.00	40	0	645	1.1	0.01	0.00	0	16	2
rsyn0805m02h	1375	3.3	1392	3.4	0.07	517	4	1277	3.5	0.07	0.00	4	108	20
rsyn0805m02m	7680	9.0	7680	9.0	0.01	93	0	7680	9.2	0.00	0.03	0	1520	5
rsyn0805m03m	7421	18.9	6839	16.9	0.04	601	0	6839	17.2	0.04	0.03	0	289	22
rsyn0805m04m	4483	17.3	4043	16.6	0.04	402	0	4043	16.4	0.04	0.01	0	230	15
rsyn0810h	108	0.8	90	0.7	0.02	265	5	90	0.7	0.02	0.00	5	19	14
rsyn0810m	921	1.6	921	1.6	0.01	96	0	921	1.4	0.01	0.00	0	32	5
rsyn0810m02m	34147	39.8	34147	40.0	0.02	225	0	33663	37.5	0.01	0.15	0	17398	10
rsyn0810m03m	30636	56.2	30636	56.5	0.02	398	0	30796	57.8	0.02	0.04	0	1000	10
rsyn0810m04m	44515	104.0	60082	135.2	0.12	993	0	59924	129.2	0.11	0.05	0	206	34
rsyn0815m	383	0.9	383	1.0	0.01	170	0	383	1.0	0.01	0.00	0	6	8
rsyn0815m02m	54903	61.6	45159	54.5	0.05	794	0	45900	55.6	0.04	0.09	0	10889	13
rsyn0815m03m	52362	75.8	55757	76.9	0.36	2905	2	53327	77.7	0.35	0.16	2	8092	42
rsyn0815m04m	1893032	3329.5	2060091	3479.4	0.32	3067	3	1959755	3252.5	0.32	3.18	3	410497	31
rsyn0820m	3125	3.8	3125	3.7	0.02	122	0	3125	3.8	0.01	0.00	0	356	7
rsyn0820m02m	56626	82.3	56626	81.8	0.05	634	0	49906	76.1	0.05	0.08	0	9073	18
rsyn0830m	2310	3.8	2310	4.0	0.03	254	0	2310	4.2	0.02	0.00	0	140	11
rsyn0830m02m	141891	216.4	148153	227.4	0.07	806	3	146332	235.0	0.07	0.27	3	6072	29
rsyn0840m	2269	4.4	2097	4.2	0.03	391	1	2097	4.0	0.03	0.02	1	91	17
rsyn0840m02m	185484	314.6	211288	392.9	0.08	1124	0	202303	373.3	0.08	0.35	0	49444	34
sep1	37	0.6	21	0.5	0.01	33	6	19	0.5	0.00	0.00	6	1	8
sepasequ_convent	504	5.0	2000	24.9	7.57	26534	239	3197	24.1	7.56	0.12	239	433	378
sfacloc2_2_80	748	5.8	872	5.8	0.11	1103	0	872	5.6	0.11	0.00	0	487	40
sfacloc2_2_90	409	1.6	409	1.5	0.06	1131	0	409	1.5	0.09	0.00	0	496	41
sfacloc2_2_95	195	0.6	195	0.7	0.04	635	0	195	0.7	0.04	0.00	0	146	19
sfacloc2_3_90	230215	130.7	229749	131.9	0.08	737	0	256360	144.9	0.10	0.42	0	45632	54
sfacloc2_3_95	15839	10.4	15839	10.5	0.06	217	0	15573	10.4	0.05	0.04	0	3940	24
sfacloc2_4_90	1613184	1163.6	1698010	1219.0	0.11	521	0	1655381	1193.6	0.11	3.04	0	518857	69
sfacloc2_4_95	103694	61.7	108256	66.7	0.10	354	0	104671	63.4	0.10	0.13	0	20550	33
slay04h	52	0.8	55	1.0	0.05	758	4	55	0.9	0.04	0.00	4	0	9
slay04m	38	0.9	40	0.7	0.01	227	3	42	0.7	0.00	0.00	3	23	4
slay05h	342	2.7	289	3.4	0.11	1495	5	279	2.9	0.10	0.00	5	2	19
slay05m	33	1.0	33	0.9	0.01	344	0	33	0.8	0.01	0.00	0	0	0
slay06h	1533	8.8	1337	7.8	0.22	3868	6	679	7.0	0.20	0.01	6	15	24
slay06m	75	1.7	73	1.7	0.01	345	0	73	1.7	0.00	0.00	0	0	0
slay07h	8863	46.6	2841	27.3	0.45	7008	3	4884	33.6	0.43	0.01	3	165	22
slay07m	358	2.4	236	2.0	0.02	560	0	236	2.0	0.02	0.00	0	2	1
slay08h	4740	42.9	5087	43.6	0.52	7008	4	4270	35.1	0.39	0.00	4	353	34
slay08m	667	6.6	667	6.8	0.04	620	0	667	6.6	0.03	0.00	0	0	0
slay09h	8091	77.2	280805	1179.7	1.03	12596	5	10346	127.0	1.04	0.07	5	414	33
slay09m	5063	24.6	3972	22.2	0.06	879	0	3972	21.4	0.05	0.02	0	0	1
slay10m	30497	134.3	55465	225.1	0.04	1412	0	55465	221.1	0.05	0.03	0	0	0
smallinvDAXr1b010-011	5119	0.9	4343	0.9	0.00	51	8	1096	0.5	0.00	0.02	8	417	8

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
smallinvDAXr1b020-022	96	0.2	110	0.2	0.00	53	8	99	0.2	0.00	0.00	8	97	8
smallinvDAXr1b050-055	280	0.4	236	0.4	0.00	46	13	181	0.3	0.00	0.00	13	101	13
smallinvDAXr1b100-110	9302382	3332.0	678	0.8	0.00	49	14	582	0.8	0.00	0.02	14	285	14
smallinvDAXr1b150-165	1020	0.9	1326	1.2	0.00	50	26	1326	1.1	0.00	0.00	26	49	28
smallinvDAXr1b200-220	780	1.0	311	0.8	0.01	50	26	311	0.9	0.00	0.02	26	44	28
smallinvDAXr2b010-011	5095	1.1	9838	1.7	0.00	54	7	2434	0.8	0.00	0.02	7	613	7
smallinvDAXr2b020-022	110	0.2	107	0.3	0.01	51	9	101	0.3	0.00	0.00	9	28	9
smallinvDAXr2b050-055	212	0.6	193	0.5	0.00	53	25	293	0.7	0.00	0.00	25	56	25
smallinvDAXr2b100-110	2843962	958.9	512	0.7	0.00	45	13	462	0.5	0.01	0.01	13	119	13
smallinvDAXr2b150-165	2907046	3600.0	3550365	3600.0	0.01	54	13	2401412	864.8	0.00	7.33	13	176575	13
smallinvDAXr2b200-220	111	0.3	281	0.6	0.00	39	13	317	0.5	0.01	0.00	13	25	13
smallinvDAXr3b010-011	4881	1.0	9575	1.3	0.02	51	7	5194	0.9	0.00	0.01	7	1241	7
smallinvDAXr3b020-022	32273	4.7	41816	6.0	0.00	52	10	4326	1.1	0.00	0.00	10	405	10
smallinvDAXr3b050-055	227	0.3	262	0.4	0.00	52	13	202	0.2	0.00	0.00	13	52	13
smallinvDAXr3b100-110	329	0.7	492	0.6	0.01	57	13	472	0.6	0.01	0.00	13	61	13
smallinvDAXr3b150-165	9352132	1299.6	993502	100.8	0.00	46	13	65682	11.8	0.00	0.15	13	8205	13
smallinvDAXr4b010-011	5346	1.2	9210	1.5	0.00	51	7	3308	0.8	0.00	0.00	7	624	7
smallinvDAXr4b020-022	31409	4.7	28331	4.3	0.00	56	11	1149	0.6	0.00	0.04	11	218	11
smallinvDAXr4b050-055	218	0.3	289	0.6	0.00	63	13	232	0.4	0.00	0.00	13	70	13
smallinvDAXr4b100-110	3913031	702.9	475322	124.1	0.00	39	13	12150	3.0	0.00	0.05	13	3151	13
smallinvDAXr4b150-165	5952282	2040.1	10212	2.5	0.01	28	13	872	0.7	0.00	0.00	13	68	13
smallinvDAXr5b010-011	5874	1.1	4706	1.1	0.00	72	8	132	0.2	0.00	0.00	8	36	8
smallinvDAXr5b020-022	75	0.2	101	0.2	0.00	48	11	90	0.2	0.00	0.00	11	71	11
smallinvDAXr5b050-055	257	0.4	228	0.3	0.00	45	13	224	0.3	0.00	0.00	13	83	13
smallinvDAXr5b100-110	7000991	3600.0	12248804	3600.1	0.01	46	9	438	0.6	0.00	0.00	9	71	9
smallinvDAXr5b150-165	14209156	3600.0	16272742	1776.2	0.00	42	13	1955490	333.3	0.00	3.39	13	124188	13
smallinvDAXr5b200-220	4347891	3600.0	8592721	3600.0	0.00	48	13	4114605	905.6	0.00	6.64	13	220065	13
smallinvSNPr1b010-011	794	6.5	806	7.1	0.01	362	68	735	5.9	0.01	0.04	68	139	72
smallinvSNPr1b020-022	28178	117.9	16605	72.3	0.01	335	67	26358	112.4	0.02	0.43	67	388	79
smallinvSNPr1b050-055	710662	3600.0	681172	3600.0	0.01	236	39	610118	3016.5	0.01	9.91	39	1309	81
smallinvSNPr2b010-011	511	5.3	564	5.7	0.01	346	75	695	5.8	0.01	0.04	75	516	76
smallinvSNPr2b020-022	4382	22.1	5993	29.3	0.01	240	47	2995	17.5	0.02	0.08	47	405	63
smallinvSNPr2b050-055	48883	232.3	194470	878.8	0.02	342	36	175781	809.0	0.02	3.25	36	758	82
smallinvSNPr3b010-011	297	3.7	402	3.5	0.01	290	68	317	3.7	0.01	0.00	68	451	69
smallinvSNPr3b020-022	930	7.8	824	8.5	0.02	281	61	1929	12.6	0.01	0.04	61	925	72
smallinvSNPr3b050-055	17318	83.4	10035	51.3	0.01	304	26	10144	50.7	0.01	0.10	26	537	73
smallinvSNPr3b100-110	208403	1012.1	559743	2855.6	0.02	274	32	114885	531.0	0.01	2.03	32	3920	87
smallinvSNPr4b010-011	294	2.7	263	3.0	0.01	404	56	427	3.1	0.02	0.01	56	106	57
smallinvSNPr4b020-022	436	5.5	441	5.2	0.01	329	43	659	6.2	0.02	0.03	43	214	61
smallinvSNPr4b050-055	1575	14.1	2127	14.2	0.02	328	35	1800	13.9	0.01	0.06	35	569	80
smallinvSNPr4b100-110	58741	271.8	18046	83.4	0.02	273	18	7022	35.5	0.01	0.19	18	1398	82
smallinvSNPr4b150-165	49636	242.9	48629	228.1	0.01	293	18	106884	461.7	0.03	1.54	18	13691	88
smallinvSNPr4b200-220	50809	232.6	80304	384.0	0.01	437	20	160943	682.5	0.02	2.33	20	24679	89
smallinvSNPr5b010-011	212	2.4	155	1.9	0.00	271	66	217	2.3	0.00	0.01	66	119	67
smallinvSNPr5b020-022	314	3.8	230	3.6	0.02	290	44	243	3.2	0.01	0.05	44	241	60
smallinvSNPr5b050-055	998	10.3	964	8.7	0.01	287	12	831	9.7	0.01	0.06	12	501	65
smallinvSNPr5b100-110	4308	28.6	4856	32.2	0.02	314	35	3065	20.6	0.01	0.07	35	1199	89
smallinvSNPr5b150-165	8152	41.4	4326	26.1	0.01	334	15	12756	72.8	0.01	0.27	15	3458	90
smallinvSNPr5b200-220	24188	117.0	29683	156.6	0.01	254	34	264735	1240.0	0.01	4.70	34	48933	92
spectra2	23	1.0	33	1.3	0.03	227	26	33	1.3	0.03	0.00	26	0	26
spring	36	0.3	23	0.2	0.00	137	17	22	0.4	0.01	0.00	17	28	24
squfl010-025	706	2294.3	710	2424.7	1.25	1082	0	710	2400.8	1.25	0.01	0	2427	163
squfl010-025persp	10	2.1	8	12.1	9.88	7480	578	6	11.2	9.18	0.02	394	480	479
squfl010-040persp	6	2.4	8	19.6	17.13	12668	450	6	18.0	15.53	0.01	476	124	807
squfl010-080persp	133	24.4	133	61.3	36.31	28039	422	133	59.1	35.10	0.01	424	307	535

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
squfl015-060persp	215	36.8	211	82.9	43.23	18428	243	211	82.5	42.88	0.01	243	243	243
squfl015-080persp	541	77.5	537	129.5	45.02	33311	265	537	129.6	45.15	0.01	265	0	265
squfl020-040persp	39	10.9	28	125.3	115.42	37136	986	28	127.8	117.38	0.16	986	1237	1405
squfl020-050persp	90	21.7	83	265.1	247.38	64516	1285	83	262.8	245.42	0.17	1285	2728	1875
squfl020-150persp	6355	2083.7	6261	2597.6	516.17	112533	1458	5989	2559.0	523.54	0.23	1458	3551	1467
squfl025-025persp	18	7.3	18	82.2	74.35	34997	1274	25	70.0	61.96	0.20	1022	307	1115
squfl025-030persp	35	12.8	45	58.9	45.58	18678	746	39	58.6	45.94	0.06	746	333	760
squfl025-040persp	271	31.0	291	127.7	101.39	22712	997	299	129.1	101.28	0.17	997	1700	1334
sssd08-04	34140	9.7	31928	9.7	0.01	78	4	34932	10.7	0.00	0.23	4	48271	10
sssd08-04persp	56521	13.8	55164	13.3	0.01	264	4	49989	12.5	0.01	0.20	4	10294	16
st_e29	4	0.1	1	0.1	0.00	76	4	1	0.2	0.00	0.00	4	5	6
st_e31	451	1.6	381	1.5	0.00	18	2	381	1.6	0.00	0.00	2	6	2
st_e32	2713	5.1	1467	3.5	0.01	280	17	1596	3.4	0.01	0.01	17	3232	23
st_e35	18551	22.9	13571	15.8	0.06	521	18	6021	8.5	0.04	0.03	18	4165	44
st_e36	1135	1.1	833	1.2	0.00	105	4	833	1.1	0.00	0.00	4	0	3
st_e38	11	0.2	11	0.2	0.00	28	8	11	0.2	0.00	0.00	8	3	7
st_e40	22	0.1	15	0.1	0.00	98	5	15	0.1	0.00	0.00	5	5	3
st_test8	1	0.0	1	0.0	0.00	27	13	1	0.0	0.00	0.00	13	0	15
st_testgr1	38	0.1	27	0.0	0.00	40	10	22	0.0	0.00	0.00	10	18	10
st_testgr3	12	0.1	4	0.0	0.00	38	15	4	0.0	0.00	0.00	15	2	16
stockcycle	29590	150.7	29590	150.1	0.04	198	0	29590	149.9	0.04	0.04	0	0	0
supplychain	141	0.5	121	0.5	0.00	30	4	143	0.5	0.00	0.00	4	21	6
supplychainp1_022020	2531189	3600.1	1165327	2225.4	1.90	1213	24	157251	537.7	1.90	2.87	24	8399	36
supplychainp1_030510	2211	4.6	347	2.0	0.11	462	10	347	2.0	0.07	0.01	10	2	12
supplychainr1_022020	116764	124.9	1804365	3600.0	1.51	9978	32	2379500	3600.0	1.50	17.82	32	95834	64
syn05h	35592722	3600.0	4	0.1	0.00	46	6	4	0.1	0.00	0.00	6	0	14
syn05m02m	3	0.2	3	0.3	0.00	139	18	3	0.2	0.00	0.00	18	4	19
syn05m03m	4	0.4	4	0.3	0.02	225	23	2	0.2	0.02	0.00	17	6	22
syn05m04m	3	0.4	4	0.2	0.01	242	34	4	0.5	0.01	0.00	34	7	36
syn10h	1	0.2	1	0.2	0.01	156	1	1	0.2	0.01	0.00	1	0	19
syn10m	1	0.1	1	0.1	0.00	29	6	1	0.1	0.00	0.00	6	0	6
syn10m02h	13	0.5	5	0.6	0.06	829	4	5	0.5	0.05	0.00	4	0	50
syn10m02m	13	0.4	13	0.4	0.03	568	14	13	0.4	0.04	0.00	14	3	34
syn10m03h	34	0.9	34	0.9	0.04	717	3	35	0.8	0.03	0.00	3	5	41
syn10m03m	54	0.7	40	0.7	0.06	805	13	54	0.7	0.05	0.00	13	1	39
syn10m04m	18	0.8	18	0.9	0.11	1149	18	18	0.8	0.11	0.00	16	3	39
syn15h	10	0.4	2	0.3	0.03	351	7	2	0.3	0.02	0.00	7	0	25
syn15m	11	0.3	11	0.2	0.02	277	13	11	0.3	0.01	0.00	13	5	17
syn15m02h	63	0.8	63	0.9	0.06	559	22	63	0.8	0.06	0.00	22	23	58
syn15m02m	66	0.6	60	0.6	0.07	869	28	35	0.6	0.10	0.00	28	24	38
syn15m03m	186	1.0	169	1.0	0.21	2214	49	170	0.9	0.14	0.01	46	161	60
syn15m04m	615	2.0	577	2.1	0.10	844	2	577	2.2	0.10	0.00	2	168	27
syn20m	70	0.5	58	0.7	0.03	375	13	58	0.7	0.02	0.00	13	10	21
syn20m02h	447	2.6	85	1.2	0.12	1870	17	75	1.3	0.16	0.00	17	41	164
syn20m02m	345	1.5	345	1.6	0.06	462	0	345	1.6	0.06	0.00	0	63	23
syn20m03m	1585	3.5	1547	3.7	0.16	2814	10	1581	3.9	0.23	0.01	10	202	47
syn20m04h	18316	87.7	16520	80.5	0.33	2194	20	18816	91.0	0.32	0.22	20	21502	160
syn20m04m	10791	12.9	11147	14.7	0.41	4399	7	10905	13.8	0.36	0.06	7	2606	66
syn30m	15	0.4	14	0.4	0.01	190	3	14	0.3	0.01	0.00	3	0	15
syn30m02m	397	2.4	423	1.8	0.02	482	6	423	2.1	0.06	0.00	6	80	29
syn30m03m	1318	4.3	1334	4.9	0.10	1791	5	1328	5.0	0.09	0.00	5	477	37
syn30m04m	75167	191.0	72732	189.3	0.07	1126	10	72261	180.6	0.09	0.12	10	27054	54
syn40m	110	0.7	110	0.6	0.05	339	0	110	0.8	0.03	0.00	0	8	18
syn40m02m	5100	9.3	5388	9.1	0.07	1227	0	5235	8.9	0.07	0.01	0	2115	36
syn40m03m	492145	798.0	464931	655.7	0.16	2773	0	484423	689.7	0.19	1.01	0	253290	55

Table 7 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
syn40m04m	1639582	3335.2	1711034	3309.7	0.43	5180	0	1623399	3319.2	0.41	3.83	0	726420	69
synthes1	5	0.1	5	0.1	0.00	11	0	5	0.1	0.00	0.00	0	0	1
synthes2	4	0.1	4	0.1	0.00	61	8	4	0.1	0.00	0.00	8	0	9
synthes3	3	0.1	3	0.1	0.00	53	0	3	0.1	0.00	0.00	0	0	1
tanksize	2211	1.9	2805	2.5	0.00	92	7	4381	2.9	0.00	0.03	7	2101	15
tln4	3487	1.7	4385	1.6	0.00	93	9	2370	1.4	0.00	0.00	9	1151	10
tln5	623529	304.4	4107464	1953.6	0.01	94	15	875200	447.0	0.00	2.79	15	555519	16
tloss	65	0.2	49	0.1	0.00	113	21	68	0.1	0.00	0.01	21	89	23
tls2	10	0.1	10	0.1	0.00	63	0	10	0.2	0.00	0.00	0	0	0
tls4	23124	23.1	21975	24.4	0.02	363	15	20011	24.3	0.01	0.06	15	18952	35
tltr	5	0.4	8	0.4	0.02	608	6	8	0.5	0.02	0.00	6	12	38
tspn05	19941	53.9	14401	32.6	0.01	453	17	15451	37.2	0.02	0.05	17	4883	15
unitcommit1	2	1.6	4	3.3	1.66	11757	309	4	3.7	1.76	0.01	305	2	419
util	6	0.4	17	0.3	0.00	51	9	119	0.4	0.00	0.00	9	366	9
watercontamination0202	35	429.0	35	1299.8	1.81	10432	50	35	1291.4	1.69	0.00	50	17	66
watercontamination0202r	6	651.3	6	1224.4	0.07	1142	0	6	1225.9	0.08	0.00	0	0	0
waternd1	84391	68.1	84391	69.3	0.02	367	0	54171	48.2	0.01	0.04	0	15217	13
waterno2_01	21	0.8	11	0.3	0.03	569	21	11	0.4	0.04	0.00	21	1	34
waterno2_02	261	5.8	181	7.0	0.25	2336	13	191	7.2	0.26	0.00	13	70	45
waterno2_03	66813	195.3	58551	164.5	0.42	3707	15	65964	182.6	0.45	0.32	15	21243	72
watertreatnd_conc	11301	11.6	16361	16.5	0.01	268	5	7221	9.9	0.01	0.02	5	2260	17
watertreatnd_flow	51391	61.1	154361	181.7	0.09	947	2	203903	223.4	0.07	0.29	2	6432	42

Table 8: Detailed results comparing SCIP without OBBT, SCIP with OBBT only, and SCIP with OBBT and LVB propagation over MINLPLib2 instances with only continuous variables after presolving. See Table 2 for aggregated results.

n — number of branch-and-bound nodes
t — total running time
t_{obbt} — time used by OBBT
lp — LPs solved by OBBT
b_{obbt} — bounds tightened by OBBT
lvb — LVBs found by OBBT
t_{lvb} — time used by LVB propagation
b_{lvb} — bounds tightened by LVB propagation

	SCIP plain		SCIP+OBBT					SCIP+OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
alkyl	101	0.3	48	0.3	0.01	57	13	48	0.3	0.00	0.00	13	1	15
bearing	7361	10.3	7051	11.2	0.01	52	4	8404	15.7	0.00	0.11	4	2831	16
chance	11	0.1	7	0.1	0.00	23	8	7	0.1	0.00	0.00	8	2	8
chem	1370091	3429.6	1897422	3600.0	0.01	150	13	1313341	2543.8	0.01	11.19	13	2439604	40
ex14_1_1	61	0.4	82	0.4	0.00	35	4	82	0.4	0.00	0.00	4	0	4
ex14_1_2	94	0.6	94	0.6	0.00	55	4	94	0.6	0.00	0.00	4	0	7
ex14_1_3	1	0.1	1	0.1	0.00	11	2	1	0.1	0.00	0.00	2	0	4
ex14_1_5	387	0.5	260	0.4	0.00	43	16	260	0.4	0.00	0.00	16	0	1
ex14_1_6	55	0.3	38	0.3	0.00	41	4	38	0.3	0.00	0.00	4	0	9
ex14_1_7	111551	125.8	36724	111.0	0.03	505	2	339431	338.0	0.03	0.33	2	1520	15
ex14_1_8	11	0.1	1	0.1	0.00	34	4	1	0.1	0.00	0.00	4	0	9
ex2_1_1	17	0.2	17	0.1	0.00	6	0	17	0.1	0.00	0.00	0	8	1
ex2_1_10	9	0.1	5	0.1	0.00	34	21	5	0.1	0.00	0.00	21	0	22
ex2_1_7	5415	1.5	7877	1.8	0.00	24	6	7710	1.7	0.00	0.00	6	6550	6
ex2_1_9	3697	3.8	3601	3.4	0.00	20	10	3681	3.6	0.00	0.00	10	283	10
ex3_1_1	2631	0.8	2901	1.0	0.00	27	7	2901	1.0	0.00	0.00	7	0	3
ex3_1_2	3	0.1	1	0.1	0.00	11	7	1	0.1	0.00	0.00	7	0	7
ex3_1_4	29	0.2	29	0.3	0.00	5	0	29	0.3	0.00	0.00	0	1	1
ex3pb	4	0.3	3	0.2	0.00	64	5	3	0.2	0.00	0.00	5	1	6
ex4_1_1	31	0.3	31	0.3	0.00	21	1	31	0.3	0.00	0.00	1	0	2
ex4_1_2	42	0.3	42	0.3	0.00	0	0	42	0.3	0.00	0.00	0	0	0
ex4_1_3	29	0.2	35	0.2	0.00	20	1	35	0.2	0.00	0.00	1	0	1
ex4_1_4	45	0.3	45	0.3	0.00	17	0	45	0.3	0.00	0.00	0	0	2
ex4_1_6	69	0.3	69	0.3	0.00	8	1	69	0.3	0.00	0.00	1	0	1
ex4_1_7	27	0.2	29	0.3	0.00	18	2	29	0.2	0.00	0.00	2	0	2
ex4_1_9	35	0.2	35	0.2	0.00	5	0	35	0.2	0.00	0.00	0	0	2
ex5_2_2_case1	65	0.3	181	0.3	0.00	18	3	181	0.3	0.00	0.00	3	24	4
ex5_2_2_case2	47	0.2	129	0.2	0.00	16	1	419	0.4	0.00	0.00	1	66	4
ex5_2_2_case3	41	0.1	77	0.3	0.00	18	3	85	0.3	0.00	0.00	3	14	5
ex5_2_4	591	0.6	571	0.7	0.00	31	0	571	0.6	0.00	0.00	0	11	3
ex5_3_2	901	0.5	869	0.7	0.00	35	2	876	0.7	0.00	0.01	2	181	6
ex5_4_2	771	0.5	511	0.4	0.00	26	6	511	0.4	0.00	0.00	6	30	8
ex5_4_3	39	0.3	16	0.2	0.00	62	8	16	0.2	0.00	0.00	8	0	5
ex5_4_4	2757336	3600.0	1021821	1084.7	0.01	166	6	869231	1046.1	0.00	2.61	6	27736	9
ex6_1_1	45101	26.8	36381	22.2	0.00	35	4	39081	24.1	0.00	0.12	4	2501	18
ex6_1_2	100	0.4	96	0.4	0.00	19	2	81	0.4	0.00	0.00	2	10	7
ex6_1_3	332351	317.1	1251081	1216.3	0.00	56	3	575561	464.5	0.00	1.18	3	3876	19
ex6_1_4	802	1.0	1240	1.1	0.00	54	3	1511	1.0	0.00	0.00	3	168	9
ex6_2_14	954461	1464.2	758871	1520.0	0.02	255	6	715921	3600.0	0.03	1.17	6	4916	32
ex6_2_6	31911	29.1	34501	28.3	0.00	106	19	17981	16.9	0.00	0.20	19	128652	27
ex7_2_2	101	0.4	71	0.3	0.00	35	3	71	0.3	0.00	0.00	3	14	7
ex7_2_4	7351	6.5	7361	6.2	0.00	120	5	7921	6.6	0.00	0.01	5	815	11
ex7_3_1	5	0.1	1	0.1	0.00	31	10	1	0.1	0.00	0.00	10	0	10
ex7_3_2	9	0.1	3	0.1	0.00	17	7	3	0.1	0.00	0.00	7	0	8

Table 8 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
ex7_3_3	27	0.2	23	0.1	0.00	7	4	23	0.2	0.00	0.00	4	4	6
ex7_3_6	7	0.1	7	0.1	0.00	11	3	7	0.1	0.01	0.00	3	0	3
ex8_1_6	1	0.1	1	0.1	0.00	11	0	1	0.1	0.00	0.00	0	0	0
ex8_1_7	106	0.8	111	0.7	0.00	57	2	57	0.7	0.00	0.00	2	18	2
ex8_2_3b	208623	3600.2	1	42.7	22.80	9489	77	1	40.6	22.22	0.01	77	0	85
ex8_2_4b	12713279	3600.0	39691	11.7	0.01	125	35	39691	12.0	0.00	0.06	35	1	38
ex8_4_1	17511	268.3	15761	266.7	0.00	218	6	16701	269.3	0.01	0.16	6	2431	8
ex8_4_4	35101	65.0	29441	56.0	0.01	118	17	26991	56.2	0.02	0.09	17	31560	23
ex8_4_5	131	2.0	175	2.3	0.00	283	3	211	2.2	0.01	0.00	3	8	4
ex8_4_6	460501	2783.9	518431	3133.3	0.00	150	1	482731	2921.2	0.00	1.66	1	5250	3
ex9_2_2	3	0.1	3	0.1	0.00	7	2	3	0.0	0.00	0.00	2	17	3
ex9_2_4	3	0.1	3	0.1	0.00	18	0	3	0.1	0.00	0.00	0	0	0
ex9_2_5	5	0.1	5	0.1	0.00	18	1	5	0.1	0.00	0.00	1	0	3
ex9_2_6	7	0.1	7	0.1	0.00	20	0	7	0.1	0.00	0.00	0	3	2
ex9_2_7	7	0.1	5	0.1	0.00	7	2	5	0.1	0.00	0.00	2	0	3
gsg_0001	733501	901.5	151361	338.2	0.04	216	50	66601	61.9	0.04	0.23	50	10391	90
himmel11	3	0.0	1	0.0	0.00	11	7	1	0.0	0.00	0.00	7	0	7
himmel16	2755	2.4	2709	2.4	0.00	66	0	2709	2.4	0.00	0.00	0	0	0
house	40001599	3600.0	71	0.3	0.01	11	6	71	0.3	0.00	0.01	6	5	8
hs62	861	1.3	701	1.2	0.00	27	8	691	1.0	0.00	0.01	8	491	9
hybriddynamic_fixedcc	1	0.1	1	0.1	0.00	59	26	1	0.0	0.00	0.00	26	0	27
hybriddynamic_varcc	9781	9.1	18035	15.6	0.03	625	37	16221	14.2	0.03	0.04	37	32	40
kall_circles_c6a	346931	121.7	413896	143.4	0.00	75	7	356091	122.8	0.00	0.59	7	88640	14
kall_circles_c6b	183845	66.0	152473	50.8	0.00	68	7	150795	55.0	0.00	0.32	7	53639	16
kall_circles_c7a	265491	84.6	280786	91.2	0.00	128	2	206724	74.0	0.01	0.42	2	83590	12
kall_circles_c8a	1321535	527.7	4637793	1790.6	0.00	92	9	874238	373.4	0.01	1.56	9	373611	14
kall_congruentcircles_c31	149	0.4	109	0.4	0.00	19	7	109	0.4	0.00	0.00	7	3	7
kall_congruentcircles_c32	215	0.3	199	0.4	0.00	23	1	197	0.3	0.00	0.00	1	25	4
kall_congruentcircles_c41	11	0.3	11	0.3	0.00	10	3	11	0.3	0.00	0.00	3	0	3
kall_congruentcircles_c42	194	0.4	141	0.4	0.00	39	2	158	0.4	0.00	0.00	2	417	8
kall_congruentcircles_c51	7191	2.9	9406	3.3	0.00	61	5	9662	3.5	0.00	0.08	5	4523	14
kall_congruentcircles_c52	1247	0.8	959	1.0	0.00	45	3	995	0.9	0.00	0.00	3	627	8
kall_congruentcircles_c61	53881	17.1	54788	17.8	0.00	47	6	64917	21.4	0.00	0.10	6	28855	15
kall_congruentcircles_c62	4000	1.8	3461	1.8	0.00	67	4	3055	1.7	0.00	0.00	4	3201	9
kall_congruentcircles_c63	3011	1.5	2155	1.4	0.00	44	9	1993	1.3	0.01	0.01	9	4598	13
kall_congruentcircles_c71	326242	113.3	326242	112.8	0.00	60	0	648184	223.7	0.00	1.31	0	212731	14
kall_congruentcircles_c72	18849	6.9	17809	6.7	0.00	60	5	17631	6.6	0.01	0.05	5	9435	11
kall_diffcircles_10	11986760	3600.0	3021901	862.9	0.01	62	1	2170571	622.5	0.00	2.84	1	13361	3
kall_diffcircles_5a	2251	1.0	2031	1.0	0.00	50	5	2461	1.0	0.00	0.01	5	1156	11
kall_diffcircles_5b	14051	3.6	20551	4.6	0.00	43	6	15941	3.9	0.00	0.02	6	5460	9
kall_diffcircles_6	24151	6.7	25431	5.9	0.00	38	2	14161	3.9	0.00	0.03	2	86	4
kall_diffcircles_7	72461	14.9	28187	7.9	0.01	52	2	27374	7.5	0.00	0.01	2	270	7
kall_diffcircles_8	5773801	1344.1	5621451	1287.7	0.01	56	8	5621451	1286.7	0.00	6.38	8	1	8
kall_diffcircles_9	1053901	259.2	12988887	3600.0	0.01	53	1	13756590	3600.0	0.00	15.29	1	13901	6
linear	1	2.8	1	2.7	0.00	51	10	1	2.8	0.00	0.00	10	0	16
mathopt5_7	33	0.2	33	0.2	0.00	16	0	33	0.2	0.00	0.00	0	0	1
mathopt5_8	33	0.3	33	0.2	0.00	19	0	33	0.2	0.00	0.00	0	0	1
oaer	2	0.1	1	0.0	0.00	13	2	1	0.0	0.00	0.00	2	0	3
pointpack04	13	0.2	5	0.1	0.01	25	2	5	0.1	0.00	0.00	2	1	3
pointpack06	10161	3.6	6263	2.3	0.00	56	1	6515	2.3	0.00	0.03	1	833	4
pointpack08	456455	175.3	456455	174.8	0.01	52	0	405110	156.9	0.00	0.83	0	195310	7
portfol_buyin	4	0.1	4	0.2	0.01	93	13	4	0.2	0.00	0.00	13	0	20
prob06	1	0.0	1	0.0	0.00	8	2	1	0.0	0.00	0.00	2	0	2
prob07	139911	269.3	24081	51.1	0.01	151	48	30061	73.6	0.01	0.29	48	24072	52
process	100	0.4	131	0.5	0.00	59	8	131	0.5	0.00	0.00	8	0	10

Table 8 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
procsyn	1	3.1	1	3.2	0.00	60	1	1	3.2	0.00	0.00	1	0	0
st_bpaf1a	11	0.1	3	0.0	0.00	21	13	1	0.0	0.00	0.00	13	100	13
st_bsj2	11	0.1	7	0.1	0.00	9	4	7	0.1	0.00	0.00	4	2	4
st_bsj4	3	0.0	3	0.0	0.00	11	6	3	0.0	0.00	0.00	6	0	6
st_e04	17	0.2	21	0.2	0.00	73	2	21	0.2	0.00	0.00	2	4	5
st_e05	64	0.3	66	0.2	0.00	12	5	66	0.2	0.00	0.00	5	0	5
st_e07	89	0.3	85	0.3	0.00	15	1	85	0.3	0.00	0.00	1	0	2
st_e08	1	0.0	1	0.0	0.00	5	4	1	0.0	0.00	0.00	4	0	4
st_e09	11	0.1	11	0.1	0.00	6	4	11	0.1	0.00	0.00	4	0	4
st_e11	6	0.2	6	0.2	0.00	4	0	6	0.2	0.00	0.00	0	0	0
st_e16	35	0.3	23	0.3	0.00	49	9	21	0.2	0.01	0.00	9	7	6
st_e19	145	0.3	145	0.3	0.00	17	0	145	0.3	0.00	0.00	0	0	0
st_e22	1	0.0	1	0.0	0.00	6	3	1	0.0	0.00	0.00	3	0	3
st_e24	3	0.0	3	0.0	0.00	7	3	3	0.0	0.00	0.00	3	0	3
st_e25	15	0.1	11	0.1	0.00	15	6	11	0.1	0.00	0.00	6	0	2
st_e26	1	0.0	1	0.0	0.00	4	2	1	0.0	0.01	0.00	2	0	4
st_e28	3	0.0	1	0.0	0.00	11	7	1	0.0	0.00	0.00	7	0	7
st_e30	97	0.4	97	0.4	0.00	25	2	97	0.4	0.00	0.00	2	2	2
st_e33	89	0.3	85	0.3	0.00	15	1	87	0.3	0.00	0.00	1	12	3
st_e41	13	0.2	11	0.2	0.00	50	11	11	0.2	0.00	0.00	11	12	15
st_fp7a	3815	1.2	213	0.3	0.00	47	19	227	0.4	0.02	0.00	19	381	19
st_fp7b	1885	0.9	1667	0.8	0.00	21	7	1320	0.7	0.00	0.00	7	1141	7
st_fp7c	3347	1.1	857	0.6	0.00	26	10	901	0.6	0.00	0.00	10	1655	10
st_fp7d	3777	1.1	507	0.4	0.01	51	19	413	0.5	0.01	0.00	19	804	19
st_fp7e	5415	1.4	7877	1.9	0.00	24	6	7710	1.8	0.00	0.04	6	6550	6
st_glmp_fp1	5	0.1	1	0.0	0.00	8	4	1	0.0	0.01	0.00	4	0	2
st_glmp_fp2	9	0.1	5	0.1	0.00	9	3	5	0.1	0.00	0.00	3	0	2
st_glmp_fp3	3	0.0	3	0.0	0.00	17	3	3	0.0	0.00	0.00	3	0	0
st_glmp_kk90	9	0.0	7	0.0	0.00	7	3	7	0.1	0.00	0.00	3	0	3
st_glmp_kk92	3	0.0	1	0.0	0.00	10	2	1	0.0	0.00	0.00	2	0	3
st_glmp_kky	1	0.0	1	0.0	0.00	7	3	1	0.0	0.00	0.00	3	0	3
st_glmp_ss1	21	0.1	21	0.1	0.00	7	2	21	0.1	0.00	0.00	2	2	2
st_glmp_ss2	1	0.0	1	0.0	0.00	6	3	1	0.0	0.00	0.00	3	0	3
st_iqpbk1	37	0.3	31	0.3	0.00	42	2	31	0.3	0.00	0.00	2	0	7
st_iqpbk2	39	0.3	29	0.3	0.00	48	2	29	0.3	0.00	0.00	2	0	7
st_jcbpaf2	105	0.2	7	0.1	0.00	45	12	7	0.1	0.00	0.00	12	1	12
st_pan1	7	0.1	7	0.1	0.00	9	1	7	0.1	0.00	0.00	1	2	4
st_ph1	3	0.0	1	0.0	0.00	12	4	1	0.0	0.00	0.00	4	0	6
st_ph14	3	0.0	1	0.0	0.00	6	4	1	0.0	0.00	0.00	4	0	4
st_ph15	11	0.1	5	0.1	0.00	11	4	5	0.1	0.00	0.00	4	40	4
st_ph2	1	0.1	1	0.1	0.00	13	4	1	0.1	0.00	0.00	4	0	6
st_ph20	5	0.0	3	0.0	0.00	14	3	3	0.0	0.00	0.00	3	0	2
st_ph3	3	0.0	1	0.0	0.00	8	5	1	0.0	0.00	0.00	5	0	5
st_phex	1	0.0	1	0.0	0.00	10	4	1	0.0	0.00	0.00	4	0	2
st_qpc-m0	15	0.1	15	0.1	0.00	0	0	15	0.1	0.00	0.00	0	0	0
st_qpc-m1	23	0.1	9	0.1	0.00	15	6	7	0.1	0.00	0.00	6	1	6
st_qpc-m3a	171	0.4	181	0.4	0.00	35	1	181	0.4	0.00	0.00	1	33	10
st_qpc-m3b	9	0.1	7	0.1	0.00	98	7	7	0.1	0.00	0.00	7	1	9
st_qpk1	13	0.1	13	0.1	0.00	0	0	13	0.1	0.00	0.00	0	0	0
st_qpk2	35	0.3	35	0.3	0.00	19	0	35	0.3	0.00	0.00	0	39	6
st_qpk3	234	0.2	234	0.3	0.00	34	0	234	0.3	0.00	0.00	0	472	11
st_rv1	19	0.1	17	0.1	0.00	24	10	17	0.1	0.00	0.00	10	8	12
st_rv2	63	0.4	70	0.3	0.00	42	21	69	0.4	0.00	0.00	21	49	21
st_rv3	341	0.4	301	0.4	0.00	66	20	297	0.4	0.00	0.00	20	86	23
st_rv7	297	0.4	255	0.4	0.00	70	28	255	0.4	0.00	0.01	28	111	31

Table 8 continued

	SCIP plain		SCIP +OBBT					SCIP +OBBT+LVB						
Instance	n	t	n	t	t_{obbt}	lp	b_{obbt}	n	t	t_{obbt}	t_{lvb}	b_{obbt}	b_{lvb}	lvb
st_rv8	573	0.6	463	0.5	0.01	82	27	321	0.5	0.01	0.00	27	120	31
st_rv9	1361	1.1	1931	1.0	0.00	88	31	1691	0.9	0.01	0.02	31	2886	37
st_z	11	0.1	7	0.1	0.00	5	5	7	0.1	0.00	0.00	5	0	4
supplychainp1_020306	21	0.7	25	0.9	0.02	276	16	31	0.8	0.02	0.00	16	131	34
supplychainr1_020306	14	0.3	14	0.6	0.01	8	4	14	0.5	0.00	0.00	4	104	11
supplychainr1_030510	4	0.1	4	0.2	0.04	463	22	4	0.2	0.04	0.00	22	0	28
syn05m	3	0.2	2	0.1	0.01	21	3	2	0.1	0.00	0.00	3	0	4
syn05m03h	3	0.4	4	0.5	0.01	241	33	4	0.5	0.01	0.00	33	37	44
wastewater02m1	95	0.3	151	0.3	0.00	29	9	121	0.3	0.00	0.00	9	59	12
wastewater02m2	127	0.6	137	0.7	0.01	113	1	139	0.6	0.01	0.00	1	75	15
wastewater04m1	111	0.4	1141	0.6	0.00	61	6	151	0.4	0.00	0.00	6	52	18
wastewater04m2	108	0.7	69	0.7	0.01	99	10	69	0.7	0.01	0.00	10	53	20
wastewater05m1	478	3600.0	7591	5.0	0.00	122	3	6191	4.7	0.01	0.04	3	9730	16