

An Abstract Model for Branching and its Application to Mixed Integer Programming

Pierre Le Bodic and George Nemhauser
Georgia Institute of Technology

November 5, 2015

Abstract

The selection of branching variables is a key component of branch-and-bound algorithms for solving Mixed-Integer Programming (MIP) problems since the quality of the selection procedure is likely to have a significant effect on the size of the enumeration tree. State-of-the-art procedures base the selection of variables on their “LP gains”, which is the dual bound improvement obtained after branching on a variable. There are various ways of selecting variables depending on their LP gains. However, all methods are evaluated empirically. In this paper we present a theoretical model for the selection of branching variables. It is based upon an abstraction of MIPs to a simpler setting in which it is possible to analytically evaluate the dual bound improvement of choosing a given variable. We then discuss how the analytical results can be used to choose branching variables for MIPs, and we give experimental results that demonstrate the effectiveness of the method on MIPLIB 2010 “tree” instances where we achieve a 5% geometric average time and node improvement, over the default rule of SCIP, a state-of-the-art MIP solver.

1 Introduction

Branch & Bound (B&B) [10] is currently the most successful and widely used algorithm to solve general Mixed Integer Programs (MIPs). B&B searches the solution space by recursively splitting it, which is traditionally represented by a tree, where the root node is associated with the entire solution space and where sibling nodes represent a partition of the solution space of their parent node. At each node, the subspace is encoded by a MIP, and its Linear Program (LP) relaxation is solved to provide a dual bound. If

the LP is infeasible, or if the LP bound is worse than the primal bound, the node is pruned (the primal bound is the value of the best feasible solution found so far). Otherwise, the subspace at that node is partitioned and the two corresponding children nodes are recursively explored. The part of the B&B algorithm that decides how to partition the solution space is referred to as the *branching rule*. Typically, the branching rule selects one variable among the *candidate variables*, i.e. those that have a fractional value in the LP solution of the current node, but are required to be integer in the original MIP. Formally, suppose that at the current node, the value of an integer variable x in the node's LP solution is $x_{lp} \notin \mathbb{Z}$. Branching on x would result in two children, each encoding a solution subspace, one in which x is upper-bounded by $\lfloor x_{lp} \rfloor$, the other in which x is lower-bounded by $\lceil x_{lp} \rceil$. The PhD thesis [1] provides a practical point of view of the latest advances in MIP solving and branching in particular. For a general overview of MIP solving and B&B, see [12] and [5].

This research is motivated by the desire to understand the fundamentals of state-of-the-art MIP branching rules (many of which are justified experimentally). The main contribution of this paper is the introduction of theoretical decision problems to study the branching component of B&B. Based on an analysis of these models, we introduce new *scoring functions*, and prove their efficiency on both simulated experiments and MIP instances. The paper is organized as follows. After defining the abstract B&B model in Section 2, we study the simplest of our problems in Section 3. More complex problems are introduced and analyzed in Sections 4 and 5, respectively. Section 6 introduces two new scoring functions based on the theory developed in previous sections. Experimental results are presented in Section 7 and conclusions are given in Section 8.

2 Abstract Branch & Bound model

In this section we model an abstract version of B&B through a series of definitions. We start by defining what we consider to be a *variable*.

Definition 1. *A variable x is encoded by a pair of non-negative (left and right) integer gains (l_x, r_x) with $l_x \leq r_x$.*

The l_x and r_x gains model the dual bound changes that occur when branching on the variable x . We suppose that these two gains are *known* and *fixed*.

We now define a *Branch & Bound tree*.

Definition 2. Given a set of variables, a Branch & Bound tree is a binary tree (in the graph sense) with one variable x assigned to each inner node i , such that x is not assigned to any ancestor of i .

We then say that variable x is *branched on* at node i . Note that, in our abstract model, the gains (l_x, r_x) of a variable x do not change with the depth of the node where the variable is branched on, nor do they depend on the dual bound at this node.

These two definitions naturally lead to the definition of *dual gap closed* at a node i in a B&B tree. Note that throughout the article, the gap will always refer to the *absolute* gap (as opposed to the relative gap), i.e. the absolute difference between the primal bound and the dual bound.

Definition 3. The dual gap g_i closed at a node i is given by the recursive formula

$$g_i = \begin{cases} 0 & \text{if } i \text{ is the root node} \\ g_h + l_x & \text{if } i \text{ is the left child of node } h \\ g_h + r_x & \text{if } i \text{ is the right child of node } h. \end{cases}$$

For instance, if at a node i , the dual bound is $g_i = 10$, and variable $(2, 5)$ is branched on, the dual bound closed at the left and right children of i are 12 and 15 respectively.

The gap *closed* by a B&B tree is defined as follows:

Definition 4. A tree closes a gap G if for all leaves i , $g_i \geq G$ holds.

Throughout, the *size* of a tree refers to the number of nodes of the tree. Using these definitions, the following sections define problems that model the B&B algorithm in increasing complexity.

3 The SINGLE VARIABLE BRANCHING problem

The first problem we define using the abstract model set up in Section 2 is the SINGLE VARIABLE BRANCHING (SVB) problem, which is a tree-size measurement problem.

Problem SINGLE VARIABLE BRANCHING

Input: one variable encoded by (l, r) , an integer G and an integer k , all positive.

Question: Is the size of the Branch & Bound tree that closes the gap G , repeatedly using the given variable, at most k ?

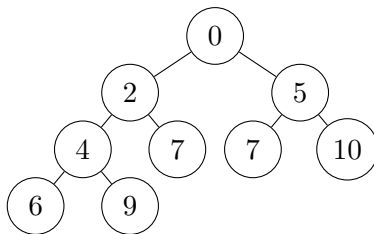


Figure 1: B&B tree with 9 nodes that closes a gap of 6 with variable gains (2, 5). The gap closed at each node is indicated by the value at the center of each node.

In the example given in Figure 1, a B&B tree that closes a gap of $G = 6$ is built using only variables with gains (2, 5). As we will see, the analysis of SVB will prove more complex and insightful than it may seem at first glance.

3.1 Motivation: state-of-the-art scoring functions and example

A state-of-the-art branching rule for MIP solving (as implemented in SCIP [2]) is a hybrid of mainly two methods: *strong branching*, which, given a candidate variable x , computes the LP gains of the would-be children of the current node if x is branched on. The candidate variable which provide the best LP bounds is then branched on. *Pseudo-cost branching* complements strong branching; it keeps track of the LP solved at each node, and, given sufficient historical data, estimates the gains that would be computed by strong branching. Strong branching is computationally expensive, as it requires solving many LP's, while pseudo-cost branching only requires a few arithmetic operations at each node. However, pseudo-cost branching is an estimate, and it requires initialization. The state-of-the-art branching rule essentially consists of using strong branching early in the tree, i.e. at the root node and a few subsequent levels, while pseudo-cost branching is called at lower levels. In practice, there are many refinements to these methods. We direct the reader to Tobias Achterberg's PhD thesis [1] and references therein for an in-depth review of branching rules.

At each node, strong branching (or pseudo-cost branching) provides the (estimated) LP gains (l_x, r_x) resulting from branching up and down for each fractional variable x . Each candidate variable x is then scored according to its gains (l_x, r_x) , and the the highest scoring variable is selected for branching.

The state-of-the-art *scoring function* used for this purpose is:

$$\max(\epsilon, l_x) \times \max(\epsilon, r_x),$$

where $\epsilon > 0$ is chosen close to 0 (e.g. $\epsilon = 10^{-6}$ in [1]) to break the ties if all variables satisfy $\min(l_x, r_x) = 0$, but plays no role otherwise. We simply refer to this scoring function as the *product* function. Prior to the introduction of the *product* function, the *linear* (or convex) function was the standard:

$$(1 - \mu) \times l_x + \mu \times r_x,$$

where μ is a parameter in $[0, 1]$ ($\mu = \frac{1}{6}$ in [1]). In [1], Achterberg reports that switching from the *linear* function to the *product* function yields an improvement of 14% in computing time, and 34% in number of nodes. The rationale behind the *linear* function with $\mu = \frac{1}{6}$ is that it should be preferable to improve the dual bound of both children by a little rather than only one of them. The *product* function prefers equal left and right gains, i.e. it is more likely to produce a balanced tree. However, there are types of instances for which the *linear* function performs better, so the *product* function does not systematically outperform the *linear* one.

To the best of our knowledge, both the *linear* and the *product* function have been established experimentally, and no theoretical evidence supports the use of one over the other, or over any other possible function. We now give a simple example suggesting that more complex functions are required to score variables. If the *product* or the *linear* function are good scoring function for solving MIPs, then they should also perform well on simple models such as SVB. Consider variables (10, 10) and (2, 50): both the *product* and the *linear* function (with default *linear* coefficient $\mu = \frac{1}{6}$) assign these variables an equal score. Observe that variable (2, 49) receives a smaller score than (2, 50) (and therefore than variable (10, 10)). We now consider the tree-sizes of two SVB instances, one that has variable (2, 49) as input, the other having variable (10, 10).

Figure 2 gives two SVB tree-size plots. For gaps no larger than 40, variable (10, 10) requires fewer nodes than (2, 49) to close the gap, but larger gaps are closed faster with (2, 49). For a gap of 1000, using only (10, 10) produces a tree 323 *million* times larger than using only (2, 49), yet both scoring functions assign a higher score to variable (10, 10). It is clear from this example that the state-of-the-art scoring functions need to be improved.

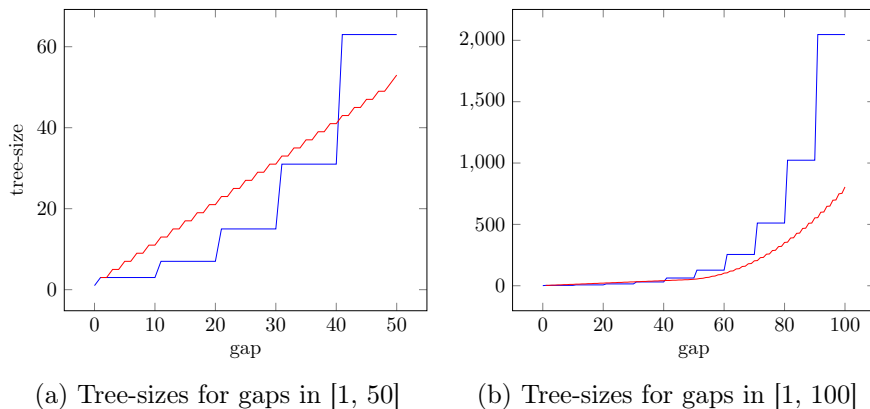


Figure 2: Plot of the sizes of the trees built with variables (10, 10) (in blue) and (2, 49) (in red).

3.2 A polynomial-time algorithm

There exists a simple recurrence relation that solves SVB. For a given gap G , $t(G)$ is defined as the size of the SVB tree that closes G :

$$t(G) = \begin{cases} 1 & \text{if } G \leq 0 \\ 1 + t(G - l) + t(G - r) & \text{otherwise.} \end{cases} \quad (1)$$

Unfortunately, computing $t(G)$ requires $O(G)$ operations. Therefore the running time of this algorithm is pseudo-polynomial in the size of the input. Moreover, observe that we do not have $G = O(\log(k))$ in general. If the given variables are all $(1, G)$, then $k = 2G + 1$ is the tree-size. However, there exists a closed-form formula that can be evaluated in polynomial time:

Theorem 1. *SVB is in P. Furthermore, a formula for the size of the B&B tree is*

$$1 + 2 \times \sum_{k_r=1}^{\lceil \frac{G}{r} \rceil} \left(k_r + \lceil \frac{G - (k_r - 1) \times r}{l} \rceil - 1 \right). \quad (2)$$

Proof. The size of the input is $O(\log(G) + \log(k))$. Let T be the tree that closes the gap G , and let $d = \lceil \frac{G}{r} \rceil$ be its *minimum* depth. If $k < 2^{d+1} - 1$, the answer to the decision problem SVB is necessarily NO, therefore we can assume the opposite throughout the rest of proof. It follows that $d = O(\log(k))$. Starting from the root of T , we turn right at most d times before finding a leaf. For a given path from the root to a leaf, let k_l (resp. k_r) denote

the number of times we turn left (resp. right). Observe that a pair (k_l, k_r) does not uniquely identify a leaf. Furthermore, since the number of leaves is in $O(k)$, it is impossible to iterate over all of them in a time polynomial in the size of the input. However, since all leaves satisfy $k_r \leq d$, it is possible to iterate over all the values of k_r . From now on we suppose that $k_r \leq d$ is fixed, and restrict our attention to those leaves that are reached by turning right exactly k_r times. The inequality

$$k_l \times l + k_r \times r < G + r$$

must hold for each leaf, since the gap closed at each leaf cannot exceed G by more than r , as otherwise the gap would already be closed at the leaf's parent. We therefore have the bound $k_l \leq \lceil \frac{G-r(k_r-1)}{l} \rceil - 1$, and the maximum depth of a leaf is $k_r + \lceil \frac{G-r(k_r-1)}{l} \rceil - 1$. Furthermore, observe that any k_r elements chosen in the set $\{1, \dots, k_r + \lceil \frac{G-r(k_r-1)}{l} \rceil - 1\}$ uniquely determine a path to a leaf, and that each leaf can be encoded using this scheme. The number of such leaves is thus $\binom{k_r + \lceil \frac{G-r(k_r-1)}{l} \rceil - 1}{k_r}$. The total number of leaves is therefore

$$\sum_{k_r=0}^{\lceil \frac{G}{r} \rceil} \binom{k_r + \lceil \frac{G-(k_r-1) \times r}{l} \rceil - 1}{k_r}. \quad (3)$$

Formula (3) can be computed in $O(d^2) = O(\log^2(k))$ operations, which is polynomial in the size of the input. Observe how the formula iterates over the possible k_r 's but avoids iterating over the range of k_l 's, as this may require $O(k)$ steps. Since each inner node has exactly two children, the theorem is proven. \square

Evaluating formula (2) becomes impractical as the gap G becomes large. We present below an asymptotic study of the tree-size that can be used in practice when formula (2) is too expensive to compute.

3.3 Asymptotic study

Analyzing algorithms through a related recurrence relation similar to (1) (and its mathematical properties, such as the asymptotic behavior) is a standard technique (see [6, Section 4.5]). In this section and throughout the article we however systematically provide self-sufficient explanations and analyzes, as our readership may not be familiar with algorithm analysis techniques.

As the gap to close G becomes large, we prove that the tree-size grows essentially linearly, i.e. there exists a fixed *ratio* determining the growth of the tree. This is related to the Fibonacci sequence, for which the golden ratio $\frac{1+\sqrt{5}}{2}$ corresponds to the asymptotic growth rate between two consecutive numbers. In fact, the Fibonacci sequence is given by the recursion formula (1) with variable (1, 2) up to the additive constant 1, which does not intervene in asymptotic results. This section therefore provides convergence results on sequences generalizing the Fibonacci sequence. Unfortunately, even though using the same notion for the ratio as in the Fibonacci sequence would be the most intuitive, the limit of $\frac{t(G+1)}{t(G)}$ when G tends to infinity may not exist in our case. Indeed, consider variable (2, 2), and the limit of these two subsequences:

$$\lim_{\substack{G \rightarrow \infty \\ G \text{ odd}}} \frac{t(G+1)}{t(G)} = 1 \qquad \lim_{\substack{G \rightarrow \infty \\ G \text{ even}}} \frac{t(G+1)}{t(G)} = 2.$$

In this example, the ratio $\frac{t(G+1)}{t(G)}$ does not converge as G tends to infinity. However, there exists another definition of the ratio, that can be shown to converge. We use the following definition:

Definition 5. *The ratio of a variable is*

$$\varphi = \lim_{G \rightarrow \infty} \sqrt[l]{\frac{t(G+l)}{t(G)}}. \tag{4}$$

With this definition, variable (2, 2) has ratio $\varphi = \sqrt{2}$. This means that, asymptotically, the number of nodes doubles for every two additional units of gap to close. We first prove Proposition 2,

Proposition 2. *If the limit φ exists, then $\sqrt[l]{2} \leq \varphi \leq \sqrt[2]{2}$.*

Proof. Let T (resp. T^l, T^r) be the tree that closes the gap G (resp. $G+l, G+r$), and observe that the tree T is a sub-tree of T^l and T^r (all three have the same root). First, each leaf of T is an inner node of T^r . This means that the number of nodes at least asymptotically doubles, thus 2 is a lower bound on φ^r . Second, each leaf of T is either a leaf or the parent of two leaves in T^l , hence 2 is an upper bound on φ^l . \square

The limit φ clearly exists in the case $l = r$, and the bounds in the above property provide the exact value. We now prove that the limit φ exists in general:

Theorem 3. When G tends to infinity, both sequences $\sqrt[l]{\frac{t(G+l)}{t(G)}}$ and $\sqrt[r]{\frac{t(G+r)}{t(G)}}$ converge to φ , which is the unique root greater than 1 of the equation $p(x) = x^r - x^{r-l} - 1 = 0$.

Proof. See Appendix A.1 and Theorem 5. □

Corollary 4. A numerical approximation of φ^r is given by the fixed-point iteration

$$f(x) = 1 + \frac{1}{x^{\frac{l}{r}} - 1}$$

with starting point $x = 2$.

Proof. See Appendix A.1. □

Having established Theorem 3, the definitions of φ and p should now make more sense, as they can be shown to originate more directly from the definition of the sequence:

$$\begin{aligned} t(G) &= 1 + t(G-l) + t(G-r) \\ \Rightarrow \frac{t(G)}{t(G-r)} &= \frac{1}{t(G-r)} + \frac{t(G-l)}{t(G-r)} + 1 \\ \Rightarrow \lim_{G \rightarrow \infty} \frac{t(G)}{t(G-r)} &= \lim_{G \rightarrow \infty} \left(\frac{1}{t(G-r)} + \frac{t(G-l)}{t(G)} \frac{t(G)}{t(G-r)} + 1 \right) \\ \Rightarrow \varphi^r &= \varphi^{r-l} + 1. \end{aligned}$$

The polynomial $p : x \rightarrow x^r - x^{r-l} - 1$ is the so-called *characteristic polynomial* of the recurrence sequence defining t . The Abel–Ruffini theorem [7, p. 264] states that there is no general closed-form formula for roots of polynomials with degrees five or higher. Since we are dealing with a particular trinomial, a formula still might exist, however we have not been able to determine one. Throughout the paper we therefore resort to numerical methods to determine φ (see Section 6.1). The characteristic polynomial for trees with arbitrary degrees has been discussed at length in a SAT setting in [9], where the ratio φ is referred to as the τ -value.

Examples of variables and their respective ratios are given in Table 1. In the case where the sum of the variable gains is fixed (Table 1a), choosing l close to r minimizes the ratio of a variable (l, r). However, if the product is fixed (Table 1b), then the opposite is preferable. Figure 3 gives a plot of the ratio as a function of the left and right gains.

Once the value of φ is determined, it is then easy to circumvent the $O(\log^2(k))$ complexity of the closed-form Formula (2), by evaluating $t(F)$

| Variable | (6, 10) | (5, 11) | (4, 12) | (3, 13) | (2, 14) | (1, 15) |
|-----------|---------|---------|---------|---------|---------|---------|
| φ | 1.0926 | 1.0955 | 1.1002 | 1.107 | 1.1204 | 1.1468 |

(a) Variables that satisfy $l + r = 16$

| Variable | (6, 10) | (5, 12) | (4, 15) | (3, 20) | (2, 30) | (1, 60) |
|-----------|---------|---------|---------|---------|---------|---------|
| φ | 1.0926 | 1.0907 | 1.0873 | 1.0813 | 1.0709 | 1.0515 |

(b) Variables that satisfy $l * r = 60$

Table 1: Ratios for some variables, truncated to 5 significant digits.

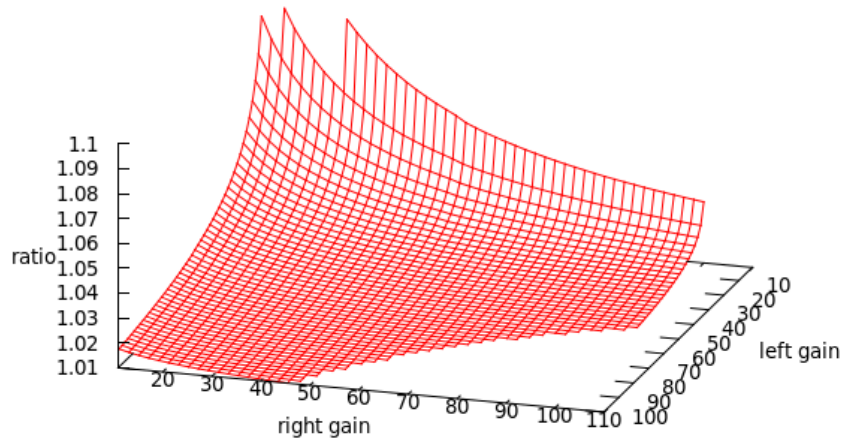


Figure 3: Variable ratio for left and right gains in $[10, 100]$.

for some value F much smaller than G , and approximating $t(G)$ using the formula

$$t(G) \approx \varphi^{G-F} t(F).$$

We now establish some useful properties of p .

Theorem 5. *The characteristic polynomial $p : x \rightarrow x^r - x^{r-l} - 1$ satisfies the following properties.*

- p is monotonically increasing in $[1, \infty)$
- in $[1, \infty)$, p has a single real root $\varphi > 1$

Proof. Consider the sign of $p'(x) = rx^{r-1} - (r-l)x^{r-l-1}$ when $x \in R^+$.

$$\begin{aligned} p'(x) > 0 &\Leftrightarrow x^{r-l-1}(rx^l - r + l) > 0 \\ &\Leftrightarrow rx^l - r + l > 0 \\ &\Leftrightarrow x > \sqrt[l]{1 - \frac{l}{r}} \\ &\Leftrightarrow x \geq 1. \end{aligned}$$

We thus know that p is increasing over $[1, \infty)$. Furthermore, $p(1) = -1$ and $\lim_{x \rightarrow \infty} p(x) = \infty$, and p is continuous, therefore there exists a single root φ in $(1, \infty)$. \square

This analysis is corroborated by the fact that the tree-size increases when incrementing the gap by l or r , thus we should have $\varphi > 1$.

4 The MULTIPLE VARIABLE BRANCHING problem

We consider the MULTIPLE VARIABLE BRANCHING problem (MVB), which naturally extends the SVB problem, defined as follows:

Problem MULTIPLE VARIABLE BRANCHING

Input: n variables encoded by $(l_i, r_i), i \in \{1, \dots, n\}$, an integer $G > 0$, an integer $k > 0$.

Question: Is there a Branch \mathcal{E} Bound tree with at most k nodes that closes the gap G , using each variable as many times as needed?

Figure 4 illustrates the difference between MVB and SVB, given two variables $(2, 5)$ and $(3, 3)$, and $G = 7$. In the MVB tree, variable $(2, 5)$ is

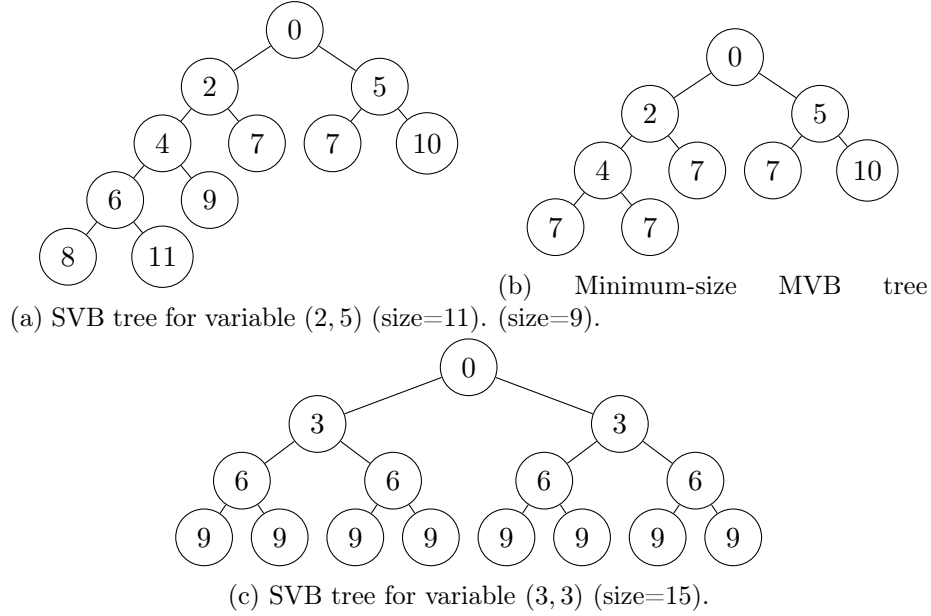


Figure 4: For a gap $G = 7$, SVB trees using variable (2, 5) or (3, 3), and minimum-size MVB tree using both.

branched on at all (inner) nodes except the bottom left one, where variable (3, 3) is branched on.

There is a recursive equation for MVB, similar to (1) for SVB. For a given gap G , $t(G)$ is defined as the minimum size of a MVB tree that closes G :

$$t(G) = \begin{cases} 1 & \text{if } G \leq 0 \\ 1 + \min_{1 \leq i \leq n} (t(G - l_i) + t(G - r_i)) & \text{otherwise.} \end{cases} \quad (5)$$

The time complexity of this algorithm is $O(nG)$, which is pseudo-polynomial in the size of the input. Unlike SVB, we do not know whether MVB can be solved in polynomial time. For a given G , we say that variable i is *branched on* at the corresponding node if $i \in \arg \min_{1 \leq i \leq n} (t(G - l_i) + t(G - r_i))$.

4.1 Motivating example (continued)

Following the example introduced in Section 3.1 for the SVB problem, consider now the plot of the MVB tree-sizes in Figure 5 (the SVB plots are the same as in Figure 2). As anticipated, for a given gap, the MVB tree-size is smaller than each of the SVB tree-sizes. However, the difference in tree-sizes

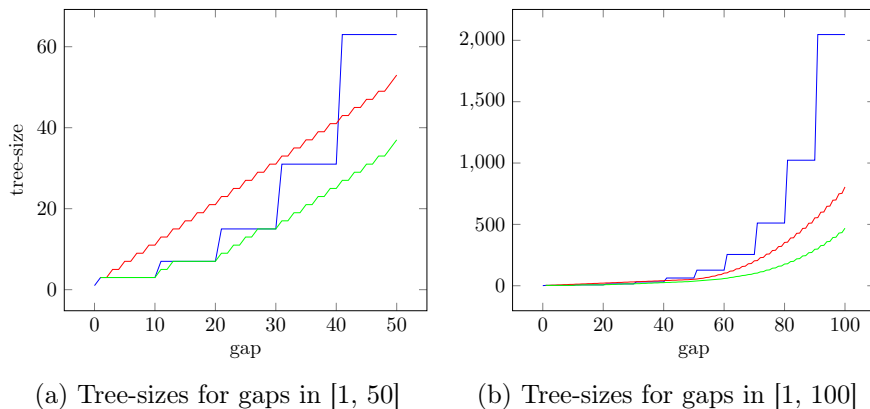


Figure 5: Plot of the sizes of the trees built with variables (10, 10) (in blue) and (2, 49) (in red), and both (in green).

with (2, 49) does not seem to increase drastically. Indeed, for a gap of 1000, the SVB tree-size of (2, 49) is only 1.798 times larger than the MVB tree-size (and this ratio is approximately constant for larger gaps). In fact, for this instance, we can verify experimentally that variable (2, 49) is branched on at every node for which the gap to close is at least 31. This phenomenon is the subject of the next section.

4.2 Asymptotic study

Let φ (resp. φ_i) now denote the ratio associated with the recursive equation (5) of MVB (resp., for SVB, equation (1) for variable i). Furthermore, for a variable i , let p_i be its characteristic polynomial, for $i = 1, \dots, n$. Formally, we define the ratio for a MVB problem to be

$$\varphi = \lim_{G \rightarrow \infty} \sqrt[z]{\frac{t(G+z)}{t(G)}}$$

where z is the least common multiple of all l_i and r_i .

Theorem 6. $\varphi = \min_i \varphi_i$

Proof. See Appendix A.2. □

Essentially, this means that for a large G , the tree-size grows at the growth rate of the *best* variable for SVB. Therefore we also have

$$t(G) \approx \varphi^{G-F} t(F)$$

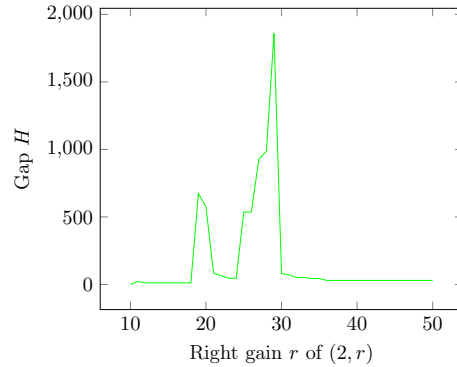


Figure 6: Value of the gap H for the MVB problem with variable $(10, 10)$ and $(2, r)$, for varying values of r .

for large gaps F and G , with $G \geq F$. Furthermore, we propose the following conjecture.

Conjecture 1. *For each instance of MVB, there exist a gap H such that for all gaps greater than H , variable $i = \arg \min_j \varphi_j$ is always branched on at the root node.*

Proving this conjecture (and proving that H is polynomially bounded by the size of the input) may help design a polynomial-time algorithm for MVB. Figure 6 provides the value of H for instances of MVB with two variables, $(10, 10)$ and $(2, r)$, where r is the value on the horizontal axis. The value of H is maximized for $r = 29$, which also minimizes $|\varphi_{(10,10)} - \varphi_{(2,r)}|$.

5 The GENERAL VARIABLE BRANCHING problem

The GENERAL VARIABLE BRANCHING problem (GVB) is defined as follows:

Problem GENERAL VARIABLE BRANCHING

Input: n variables encoded by $(l_i, r_i), i = 1, \dots, n$, an integer $G > 0$, an integer $k > 0$, and a vector of multiplicities $m \in \mathbb{Z}_+^n$.

Question: *Is there a Branch & Bound tree with at most k nodes that closes the gap G , branching on each variable i at most m_i times on each path from the root to a leaf?*

Observe that GVB comprises the input of problem MVB, and additionally stipulates that each variable may not be used more than a given number

of times on each path from the root to a leaf. Consequently, GVB corresponds to MVB whenever the multiplicity of each input variable is large enough, therefore GVB generalizes MVB.

When solving a MIP instance using a B&B search, the minimization version of GVB models the problem of choosing a variable to branch on under the following simplifying hypotheses:

- The up and down gains are known for each variable, and are invariant.
- If the problem is feasible, the optimal value is known.

The first hypothesis is not satisfied in practice. However, as the B&B tree grows, pseudo-costs become better approximations for the up and down gains. Likewise, the second hypothesis becomes true at some point during the B&B search. So, while both hypotheses are not initially satisfied, they become more accurate as the tree grows. As we will see, these hypotheses are not sufficient to render the GVB problem theoretically tractable. A recursion for GVB similar to (5) for MVB is

$$t(G, m) = \begin{cases} 1 & \text{if } G \leq 0 \\ 1 + \min_{1 \leq i \leq n, m_i > 0} (t(G - l_i, m - v_i) + t(G - r_i, m - v_i)) & \text{otherwise} \end{cases} \quad (6)$$

where m and v_i , are both vectors of size n ; m is the vector of multiplicities, and v_i is the indicator vector of the set $\{i\}$. This algorithm keeps track of the variables already branched on, and, in the case where all multiplicities are 1, its running time is $O(2^n G)$.

Figure 7 gives the minimum-size B&B tree for a GVB instance and an MVB instance that has the same variables and gap as input.

5.1 Complexity of GVB

For the complexity results presented in this section, it is sufficient to consider the case where all multiplicities are one. We prove the #P-hardness of GVB by reducing it to a variant of counting knapsack solutions:

Problem #KNAPSACK

Input: N items with weights w_1, \dots, w_N and W the capacity of the knapsack, a given K , all integer positive.

Question: Are there at least K distinct feasible solutions x to the (covering) knapsack constraint $\sum_{i=1}^N w_i x_i \geq W$, with $x_i \in \{0, 1\}, i = 1, \dots, N$?

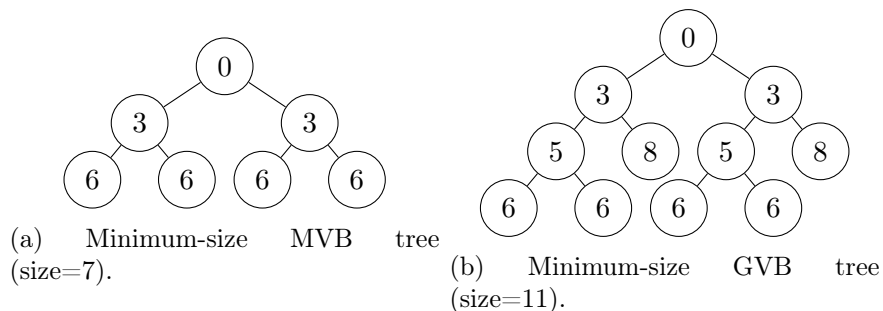


Figure 7: Minimum-size trees for MVB and GVB with input $G = 6$ and variables $(1, 1)$, $(2, 5)$, $(3, 3)$ (in GVB, with multiplicities 1).

Note that in our definition of the $\#\text{KNAPSACK}$ problem, we ask for *at least* K solutions. This variant of the counting problem can solve the exact counting problem by e.g. dichotomy in $O(N)$ calls, hence it is $\#\text{P-hard}$.

Theorem 7. *GVB is (weakly) $\#\text{P-hard}$*

Proof. We build an instance of the GVB problem that embeds an instance of the $\#\text{KNAPSACK}$ problem. Set $n = N + 1$, and for each $i \in \{1, \dots, n - 1\}$, create a variable with gains $(C, C + w_i)$, where $C = \sum_{i \in \{1, \dots, N\}} w_i$; furthermore, create the n^{th} variable (C, C) . Set the target gap $G = (n - 1)C + W$ and the number of allowed nodes $k = 2^{n+1} - 1 - 2K$. Finally, set all multiplicities to one. Suppose, without loss of generality, that the knapsack instance is feasible, and thus $C \geq W$. We prove that the answer to this GVB instance is YES if and only if the answer to the instance of $\#\text{KNAPSACK}$ is YES.

First, observe that a minimum-size tree is obtained by branching on the variable with the biggest right gains, i.e. corresponding to the biggest items (w.r.t. w_i), and finally by branching on the dummy variable with gains (C, C) . Consider a tree T that closes the gap G . Observe that each leaf of T has at least depth $n - 1$, and at most n . In fact, there is a one-to-one mapping between (not necessarily feasible) solutions of the knapsack instance and nodes at depth $n - 1$. Furthermore, there is a one-to-one mapping between *feasible* solutions of the knapsack instance and *leaves* at depth $n - 1$. Therefore, since an infeasible knapsack instance would yield a perfect tree, with $2^{n+1} - 1$ nodes, each feasible solution decreases the number of nodes by 2 exactly. If there are at least K feasible solutions to $\#\text{KNAPSACK}$, there is a tree with at most $2^{n+1} - 1 - 2K$ nodes for their corresponding GVB instance, and vice-versa. \square

Theorem 7 in particular implies, under the widely believed conjecture that the polynomial hierarchy PH [14] is proper to the second level, that GVB is neither in NP nor in co-NP. This is because $PH \subseteq P^{\#P}$ [15].

6 Using the abstract model for scoring branching candidates

6.1 The ratio scoring function

Recall from Section 3.1 that a scoring function combines the left and right LP gains to score candidate variables. The variable with the highest score is then branched on. The `linear` and `product` functions are two scoring functions that were presented in Section 3.1. In this section, we introduce the *ratio* scoring function, based on Definition 5 for the SVB problem. Given two variables i and j and their respective ratios φ_i and φ_j , it selects the variable with the smallest ratio.

To help analyze the behavior of the `ratio` scoring function, we formalize the concept of domination for two variables.

Definition 6. *Given two variables (l_1, r_1) and (l_2, r_2) , we say that (l_1, r_1) dominates (l_2, r_2) if and only if $l_1 \geq l_2$ and $r_1 \geq r_2$, with at least one of the inequalities strict.*

Proposition 8. *Suppose variable (l_1, r_1) dominates variable (l_2, r_2) . Then,*

1. *Both the `product` and the `linear` function assign a better score to (l_1, r_1) than to (l_2, r_2) .*
2. *In the MVB or GVB problem setting, if both (l_1, r_1) and (l_2, r_2) are branched on on a common path, branching on (l_1, r_1) before branching on (l_2, r_2) yields a tree of size no larger than the converse.*

Proof. Part 1 is obvious. Part 2 can be proven by observing that if in a B&B tree, variable (l_2, r_2) is branched on before (l_1, r_1) , then the tree where the variables are switched closes a gap at least as large. \square

In other words, the `linear` and the `product` scoring functions prefer non-dominated variables, as dominated variables necessarily yield larger trees. We prove that the `ratio` scoring function also exhibits this property:

Theorem 9. *Suppose variable (l_1, r_1) dominates variable (l_2, r_2) , then $\varphi_1 < \varphi_2$.*

Proof. Starting with the definition of the characteristic polynomial of variable j , we establish:

$$\begin{aligned} 0 &= p_2(\varphi_2) = \varphi_2^{r_2} - \varphi_2^{r_2-l_2} - 1 = \varphi_2^{r_2}(1 - \varphi_2^{-l_2}) - 1 \\ &< \varphi_2^{r_1}(1 - \varphi_2^{-l_1}) - 1 < p_1(\varphi_2). \end{aligned}$$

Therefore, by Theorem 5, $\varphi_2 > \varphi_1$. □

As a consequence, the ratio of a dominated variable does not have to be computed. Moreover, to perform comparisons between ratios, it is not necessary to compute both of them.

Proposition 10. *Suppose that the ratio φ_1 of variable (l_1, r_1) has been computed, then, given a variable (l_2, r_2) , $\varphi_2 < \varphi_1$ if and only if $p_2(\varphi_1) > 0$.*

Proof. Direct using Theorem 5. □

Hence a strategy to reduce the number of ratio computations is to compute φ for a variable that is believed to be a good candidate for branching (e.g. the best according to the `product` function), and use Proposition 10 to test all other candidates, computing the ratio of a variable only if it is proven to be smaller than the current best one.

The scoring rule that we implement uses the strategy described in Algorithm 1. The computation of the ratio at Steps 2 and 5 is done numerically. We have implemented and tested five different methods, namely the fixed-point iteration method described in Corollary 4, a simple bisection method, Newton and Laguerre’s method [13, Chapter 9]), and a direct method computing (5) using the closed-form formula described in Theorem 1 for a large enough gap. Iterative methods are initialized with starting point $\sqrt[l]{2}$, which is a proven lower bound (see Proposition 2). Experiments have shown that if $\frac{r}{l} \leq 100$, Laguerre’s method was the most effective, otherwise the fixed-point iteration method should be used. Following this rule, computing the ratio of a variable takes around 20 milliseconds of CPU time on a modern computer.

Observe that if the ratio and one of the left (resp. right) gain is fixed, then the right (resp. left) gain can be computed analytically:

Proposition 11. *Given a variable (l, r) and its ratio φ , then*

$$r = -\frac{\log(1 - \varphi^{-l})}{\log \varphi} \qquad l = -\frac{\log(1 - \varphi^{-r})}{\log \varphi}.$$

Algorithm 1 Implementation of the **ratio** scoring function.

- 1: Filter out the branching candidates with dominated gains. \triangleright Theorem 9
 - 2: Compute the ratio φ^* of the best variable according to the **product** function.
 - 3: **for all** remaining branching candidates i **do**
 - 4: **if** $p_i(\varphi^*) < 0$ **then** \triangleright Proposition 10
 - 5: Compute the ratio φ_i and set $\varphi^* = \varphi_i$.
 - 6: **end if**
 - 7: **end for**
 - 8: **return** the variable with ratio φ^* .
-

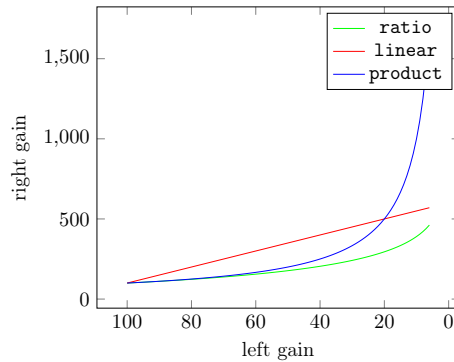


Figure 8: Right gains depending on the left gain such that the score is constant.

Proof. Direct using the definition of the polynomial p . □

Figure 8 has been generated using this property. The leftmost point (corresponding to variable (100, 100)) serves as a reference, and each curve is such that its points have the same score according to one scoring function: the red curve corresponds to the **linear** function with default μ , blue to the **product**, and green to the **ratio**. Unsurprisingly, the red curve is a line and the blue curve is a parabola. Note how the **product** function and the **ratio** function match very closely when l and r are close to each other. This essentially means that if all variables have roughly equal left and right gains, the **product** and **ratio** functions behave similarly. If this is not the case, then the **ratio** function prefers variables with unequal left and right gains compared to the **product** function.

6.2 The svts scoring function

Ranking the variables by ratio only makes sense for large enough gaps. Consider again the example given in Section 3.1, and suppose that variables (10, 10) and (2, 49) are the only candidates for branching. The `ratio` function would score variable (2, 49) higher, independent of the gap. However, in the SVB setting, if the gap is no larger than 40 then (10, 10) is the better variable. We therefore define a more refined scoring function that takes the gap into account. Given the gap at a given node, this scoring rule ranks the variables according to their SVB tree-sizes. We therefore call this scoring function the `svts` function, for “Single Variable Tree-Size”. Algorithm 2 describes the implementation of the `svts` function. We introduce a parameter D , which determines whether an approximation of the tree-size should be used, depending on the smallest depth of a leaf in the SVB B&B tree, i.e. $\lceil \frac{G}{r_i} \rceil$, given a variable (l_i, r_i) and a gap G . Parameter D effectively bounds the number of terms to add in the closed-form formula (2) to compute the SVB tree-size. If $\lceil \frac{G}{r_i} \rceil > D$ (i.e. the gap G is such that the formula (2) would require computing more than D terms), then the exact tree-size is computed for a smaller gap, namely $r_i D$, and an approximation of the tree-size $t(G)$ is computed approximately using the ratio. In our implementation we have set $D = 100$. Deactivating Step 5 increases the average running time of a B&B implementing Algorithm 2 as a scoring function by 4% (in the setup of Section 7.2.1). Deactivating Step 11 (i.e. not restricting the evaluation of formula (2) to “small” trees) increases the average running time by 11%.

Note that for a large enough gap, the SVB function is equivalent to the `ratio` function, i.e. the rankings of variables are equal.

We next evaluate the `ratio` and SVB scoring function in simulations as well as MIP benchmarks.

7 Experimental results

Numerous MIP solving algorithms are developed and parametrized by trial-and-error loop of experiments on MIP benchmarks. The major drawback of this procedure is evidently the computational burden it involves. Another significant flaw is that the instances used to develop the algorithm are often the ones used to benchmark it. This may lead to *overfitting*, which means that the algorithm will perform much better on these instances than on general ones. The magnitude of this problem increases with the number of parameters of a method. In an effort to mitigate these issues, we carry out simulations on a large set of random instances in Section 7.1. The results we

Algorithm 2 Implementation of the `svts` scoring function.

```
1: Compute the absolute gap  $G$  at the current node.
2: if  $G = \infty$  then                                ▷ e.g. no primal solution is known.
3:   return the variable selected by Algorithm 1.      ▷ Theorem 6
4: end if
5: Filter out the branching candidates with dominated gains.
6: for all remaining branching candidates  $i$  with gains  $(l_i, r_i)$  do
7:   Let  $d = \lceil \frac{G}{r_i} \rceil$ .                          ▷ The minimum depth of the SVB tree
8:   if  $d = \infty$  then
9:     Set the tree-size  $t_i(G) = \infty$ .
10:  else
11:    Let  $\tilde{G} = r_i * \min(d, D)$ .                       ▷  $D$  is a parameter
12:    Compute the SVB tree-size  $t_i(\tilde{G})$  using formula (2).
13:    if  $\tilde{G} < G$  then
14:      Compute the ratio  $\varphi_i$ .
15:      Compute the SVB tree-size  $t_i(G) \approx \varphi_i^{G-\tilde{G}} t_i(\tilde{G})$ .  ▷ see (3.3)
16:    end if
17:  end if
18: end for
19: if all variables  $i$  have  $t_i(G) = \infty$  then
20:   return the variable selected by Algorithm 1.
21: else
22:   return the variable  $i$  with minimum  $t_i(G)$ .
23: end if
```

obtain on these simulated instances confirm the improvements we achieve on standard MIP benchmarks in Section 7.2.

7.1 Numerical simulations

In order to evaluate the performance of the `linear`, `product` and `ratio` scoring functions detailed in Section 6, we run simulations in which these scoring functions are used to select variables on MVB and GVB instances. The GVB problem models the B&B algorithm more closely than the MVB problem, but in practice it is only possible to obtain exact tree-sizes for the latter problem. We do not run simulations on the `svts` function, as it requires computing tree-sizes for all variables and each possible gap.

We evaluate the scoring functions on synthetically generated instances. The variable gains are generated in one of four different ways:

- **Balanced** [B] Both left and right gains are integers uniformly drawn in the interval $[1, 1000]$ (if necessary, gains are switched to ensure that the left gain is less than or equal to the right gain).
- **Unbalanced** [U] The left (resp. right) gain is an integer uniformly drawn in the interval $[1, 500]$ (resp. $[501, 1000]$).
- **Very Unbalanced** [V] The left (resp. right) gain is an integer uniformly drawn in the interval $[1, 250]$ (resp. $[251, 1000]$).
- **Extremely Unbalanced** [X] The left (resp. right) gain is an integer uniformly drawn in the interval $[1, 100]$ (resp. $[101, 1000]$).

We will refer to these different ways of generating variables as different *data types*.

7.1.1 Simulation results for the MVB problem

Recall that in the MVB problem, one variable can be used an arbitrary number of times. In this section, two types of methods are evaluated. The first type correspond to *scoring functions* that can be or are actually implemented in MIP solvers to choose which variable to branch on (e.g. the `product` function). The second type are *problem solving algorithms*, which use knowledge of the MVB problem, and cannot directly be implemented in MIP solvers (e.g. an exact recursive algorithm). The methods of the first type are the ones we are really interested in, while the methods of the second type help evaluate the first ones. We compute the tree-sizes that result from the following methods, listed by order of increasing exactness.

- Linear (with different values of the parameter μ), **product** and **ratio** scoring functions. Since the ranking of variables obtained by scoring functions only depends on the left and right gains, the same variable is branched on at every node, which means the tree-size obtained by selecting variables using a scoring function corresponds to the tree-size of the SVB tree for the variable with the best score.
- Lower Bound (LB) for scoring functions: it is the minimum tree-size of all SVB instances, for all variables in the input of the MVB instance.
- The exact recursive function, as described in Section 4: this method produces a minimum-size tree for the instance.

Each branching strategy in this list necessarily produces trees with fewer nodes than those listed higher in the list. Note that the **linear**, **product** and **ratio** scoring functions are listed together, as it is *a priori* not clear whether any of the scoring function dominates the others. These are the scoring functions we are primarily interested in.

The other parameters have been chosen as follows:

- The number of variables for each instance is 100. Increasing the number of variables is possible, however the chances of producing a variable that dominates all others would increase.
- The gap is set at $G = 10^5$. Increasing G by an order of magnitude may cause the tree-sizes to exceed the maximum number that can be encoded by a double-precision floating-point representation ($\approx 10^{307}$).
- The number of instances for each of the B, U, V and X type of variables is 100.

Table 2 provides two different performance measures for each type of data and each branching strategy. The first performance measure, “t-s”, is the percentile change in geometric means of the tree-sizes compared to the (exact) minimum tree-size. For instance, the number 26.2 (at the top-left of the table) indicates that this branching strategy yielded trees with, on average, 26.2% more nodes than the minimum-size trees. For the cases where the difference is in orders of magnitude, only the order is indicated. Note that the tree-sizes obtained in the experiments vary from around 10^{30} for the balanced data to up to 10^{300} for extremely unbalanced data. The second performance measure, “wins”, is the number of times a scoring function produces the smallest tree-size among the **linear**, **product** and **ratio** functions

| Data | $\mu =$ | Linear(μ) | | | | | Product | Ratio | LB |
|------|---------|-----------------|---------------|---------------|---------------|-----------|-------------|-------------|------|
| | | 0 | $\frac{1}{6}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | 1 | | | |
| B | t-s | 26.2 | 8.23 | 6.23 | 3.10 | 10^{17} | 2.84 | 2.84 | 2.83 |
| | wins | 88 | 93 | 95 | 99 | 8 | 100 | 100 | |
| U | t-s | 10^{08} | 96.79 | 6.07 | 8.43 | 10^{23} | 6.60 | 3.35 | 3.14 |
| | wins | 11 | 67 | 93 | 90 | 10 | 93 | 98 | |
| V | t-s | 10^{20} | 57.93 | 24.85 | 282.28 | 10^{23} | 29.82 | 5.75 | 5.75 |
| | wins | 19 | 77 | 86 | 65 | 14 | 83 | 99 | |
| X | t-s | 10^{47} | 6.07 | 731.44 | 10^5 | 10^{31} | 93.86 | 6.07 | 6.07 |
| | wins | 7 | 100 | 72 | 50 | 13 | 82 | 100 | |

Table 2: Simulation results on MVB.

(LB is excluded as it necessarily produces smaller tree-sizes). In case of a tie, multiple scoring functions win.

First, observe that the performance (w.r.t. tree-size) of branching strategies generally decreases as the left and right gains of the variables become less balanced. For some values of μ , the **linear** function performs extremely badly. More interestingly, it appears that the best value of μ heavily depends on the balance of data. The value $\mu = \frac{1}{2}$ is better for balanced data (B), while for extremely unbalanced data (X), $\mu = \frac{1}{6}$ performs best, in a tie with the **ratio** function. For this type of data, the tie may be explained by Figure 8, where the **linear** and **ratio** functions tend to pick similar variables for very unbalanced gains. Similarly, the tie between the **product** function and the **ratio** function for balanced data may be explained by the fact that they pick similar variables for balanced gains. The **ratio** function wins on 397 out of 400 instances, and is very close to the theoretical bound LB that these scoring functions can achieve. Note that both the **ratio** function and LB are within 7% of the actual minimum-tree size, even for the most unbalanced data.

7.1.2 Simulation results for the GVB problem

The experiments we carry out in this section are significantly different than those we did on MVB: indeed, in the GVB problem, each variable can only be used once on a path from the root to a leaf. As a consequence, we are not able to compute the exact minimum tree-sizes, as the state space of the recurrence equation (6) is too large. Note that the algorithms that compute the tree-sizes for scoring functions are also harder to solve than in the previous section, and the gaps G used in our instances are therefore smaller. The parameters of these experiments are presented below.

- As in the previous section, the number of variables for each instance is 100, and the number of instances for each of the B, U, V and X type of variables is also 100.
- The gap takes different values for each type of data. Indeed, while a gap of e.g. $G = 5000$ can be closed with a few hundred nodes in the balanced case, it can only be closed in half of the instances for extremely unbalanced data. We therefore adapt the gap to the data. In each case, the minimum gap tested is chosen such that the tree-sizes are at least 100. The maximum gap tested is chosen such that the depth of the trees is close to 100, i.e. the number of variables, but also such that all 100 instances can be solved. The average between the minimum and the maximum gap is also tested.

Table 3 presents the simulation results, in a format similar to Table 2. Since we do not know the minimum tree-sizes in these experiments, the reference is the **product** function, as it is the state of the art. As a consequence, negative changes may (and do) occur. The observation we made in the previous experiments on MVB, which is that the **ratio** function benefits from unbalancedness, clearly carries over. Furthermore, for a given type of data, it also profits from an increase in the gap. An approximate 7% decrease in the number of nodes occurs when these two facts are taken together. Moreover, the **ratio** function has, by far, the largest number of wins. Note that the **linear** function outperforms the **ratio** or the **product** function in some cases.

Since the **product** function seems to perform slightly better for small gaps, and the **ratio** improves with larger gaps, the idea of changing the scoring function depending on the gap naturally arises. Table 4 provides the results of this hybrid function for the maximum gaps considered for each data type. The notation is the same as in Table 3 (including the fact that the **product** function is taken as a reference). The hybrid **ratio-product** function is parametrized by h , the height at which the scoring function used at a node switches from **ratio** to **product** (the height of a node is the length of the longest path to reach a leaf from this node). For example, $h = 10$ means that at nodes that would have a height of 10 or less with the **ratio** function, the **product** function is used instead. As a consequence, $h = 0$ corresponds to the **ratio** function, and $h = 100$ to the **product** function (since there are 100 variables, the depth of a leaf is at most 100).

First, observe that we do obtain a slight reduction in the number of nodes for any value of h (except for data U and $h = 50$). Second, note that the optimal value for the height parameter increases as the data becomes less

| Data | Gap | $\mu =$ | Linear(μ) | | | | | Product | Ratio |
|------|-------|---------|-----------------|---------------|---------------|---------------|---------|-------------|--------------|
| | | | 0 | $\frac{1}{6}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | 1 | | |
| B | 5000 | t-s | 1.94 | 0.55 | -0.15 | 0.41 | 235.83 | 0.00 | -0.03 |
| | | wins | 67 | 77 | 85 | 76 | 0 | 82 | 83 |
| | 15000 | t-s | 21.34 | 9.37 | 1.18 | 2.36 | 1051.56 | 0.00 | 0.06 |
| | | wins | 2 | 3 | 17 | 14 | 0 | 41 | 45 |
| | 25000 | t-s | 57.70 | 19.45 | 0.89 | 19.23 | 1742.11 | 0.00 | -0.96 |
| | | wins | 0 | 0 | 7 | 1 | 0 | 28 | 64 |
| U | 5000 | t-s | 50.86 | 10.63 | -0.02 | 4.27 | 314.73 | 0.00 | 0.17 |
| | | wins | 0 | 2 | 45 | 3 | 0 | 48 | 44 |
| | 12500 | t-s | 86.38 | 30.30 | 0.75 | 19.99 | 1562.15 | 0.00 | -0.74 |
| | | wins | 0 | 0 | 11 | 0 | 0 | 29 | 60 |
| | 20000 | t-s | 71.91 | 22.99 | 0.30 | 61.17 | 3874.17 | 0.00 | -2.29 |
| | | wins | 0 | 0 | 2 | 0 | 0 | 14 | 84 |
| V | 5000 | t-s | 257.39 | 4.76 | 6.50 | 54.40 | 282.93 | 0.00 | -0.91 |
| | | wins | 0 | 1 | 0 | 0 | 0 | 35 | 65 |
| | 7500 | t-s | 428.67 | 11.88 | 14.85 | 102.26 | 425.88 | 0.00 | -2.29 |
| | | wins | 0 | 0 | 0 | 0 | 0 | 25 | 75 |
| | 10000 | t-s | 528.58 | 24.38 | 25.76 | 135.77 | 525.01 | 0.00 | -5.81 |
| | | wins | 0 | 0 | 0 | 0 | 0 | 4 | 96 |
| X | 2000 | t-s | 445.42 | 8.29 | 42.80 | 63.76 | 89.95 | 0.00 | 3.15 |
| | | wins | 0 | 3 | 0 | 0 | 0 | 76 | 28 |
| | 3000 | t-s | 903.02 | 6.72 | 47.56 | 73.37 | 107.32 | 0.00 | -1.16 |
| | | wins | 0 | 0 | 0 | 0 | 0 | 43 | 58 |
| | 4000 | t-s | 1537.76 | 2.47 | 41.26 | 68.27 | 105.88 | 0.00 | -6.90 |
| | | wins | 0 | 1 | 0 | 0 | 0 | 2 | 97 |

Table 3: Simulation results on GVB.

| Data (& Gap) | $h =$ | Hybrid <code>ratio-product</code> (h) | | | | | | (Product) |
|-----------------|-------|---|--------------|--------------|--------------|---------------|-------|-----------|
| | | (Ratio) | | | | | | |
| | | 0 | 10 | 20 | 30 | 40 | 50 | 100 |
| B (25000) | t-s | -0.96 | -1.31 | -0.49 | -0.02 | 0.00 | 0.00 | 0.00 |
| | wins | 6 | 71 | 28 | 13 | 12 | 12 | 12 |
| U (20000) | t-s | -2.29 | -2.29 | -3.37 | -1.01 | -0.05 | 0.01 | 0.00 |
| | wins | 0 | 7 | 79 | 14 | 0 | 0 | 0 |
| V (10000) | t-s | -5.81 | -5.81 | -6.18 | -8.38 | -4.40 | -0.46 | 0.00 |
| | wins | 0 | 0 | 5 | 81 | 14 | 0 | 0 |
| X (4000) | t-s | -6.90 | -6.95 | -7.38 | -9.03 | -10.57 | -8.27 | 0.00 |
| | wins | 0 | 0 | 0 | 14 | 73 | 13 | 0 |

Table 4: Simulation results on GVB using a hybrid `ratio-product` function, parametrized by a height h .

balanced. Third, observe how for each data type, the relative change compared to the `product` function is unimodal, which suggests a low variability of the results and backs up our analysis.

7.2 Experiments on MIP instances

We have modified the solver SCIP 3.1.1 [2] to use the `ratio` or the `svts` function, alternatively to the default `product` function. Note that in SCIP, the selection of the branching variable does not solely rely on the `product`. There are a number of “soft” tie-breakers that can come into play if the `product` scores of multiple variables are almost equal. This is referred to as hybrid branching [3]. If at a given node, the variable that the hybrid branching rule selects is not the one that maximizes the `product` function, then we keep that variable and do not use the `ratio` or `svts` function. Furthermore, for the `ratio`, we use the parameter $h = 10$ as defined in Section 7.1.2, following the results of Table 4. This essentially means that whenever a node is believed to be “close” to the leaves, the `product` function is used rather than the `ratio` function.

7.2.1 Benchmark instances

The benchmark test set comprises the instances from MIPLIB 3.0, MIPLIB 2003 [4], and MIPLIB 2010 Benchmark [8]. These instances are the state of the art in MIP benchmarking. The instances that are at the time of writing classified as “open” have been removed, which leaves 159 instances. For feasible instances, the optimal value is provided as a primal bound, and primal heuristics are disabled (for all instances). Cuts after the root node are also disabled to reduce performance variability [11]. The disconnected compo-

nents presolver is disabled as it can lead to different transformed problems, and thus possibly increased variability. Furthermore, for each instance, ten different seeds are used to create random permutations of the input (using SCIP’s internal procedure). A time limit of two hours is specified. In this setting, the testbed comprises 1590 instances, and the total running time that this experiment required on a single machine is 4 months.

Three different scoring functions are tested. The first one is the `product` function, which is the default in SCIP. The second one is the `ratio` function, defined in Section 6.1. The third one is the `svts` function defined in Section 6.2.

Table 5 gives the result for these three scoring functions, with `product` as the reference. Each line corresponds to the 10 permutations of each instance of the test set. There are three columns for each scoring function. The columns of the `product` scoring function gives absolute performance measures, while the two others give measures relative to `product`. For `product`, the first column provides the number of permutations solved, and, for the other two functions, the difference from the reference. Note that all 10 permutations of a given instance may not be solved the same number of times depending on the scoring function used. For each instance, we determine for each scoring function the number of permutations solved. Let N be the minimum and note that N can be as large as 6 and we could still have no single permutation solved by all three scoring functions. After having determined N for a given instance, we take the N best results for each scoring function and compute the arithmetic averages. These averages of time (in seconds) and nodes are directly displayed for the `product` function. In the nodes column, we use the letters k and m as a shorthand for thousands and millions. For the `ratio` and `svts` functions, we display the ratio of averages for both time and nodes, compared to the `product`. Considering the N best permutations rather than the intersection of the sets of solved permutations enables more data to be used, and reflects in the averages the fact that some scoring functions solve more permutations, giving them a fair advantage.

The instances for which at least one permutation can be solved at the root node by any setting, or for which no permutation is solved by all settings (i.e. $N = 0$), are not displayed. We similarly exclude instances that can be solved in less than a second by at least one setting and permutation. Note that these instances are accounted in the total number of instances solved, but not in the different average measures.

The total, geometric mean and shifted geometric mean (with shifts 10 and 100 for time and nodes, respectively) [1, p. 321] are provided at the bottom of the table. These are not computed on the averages given at each

line, but on each value that is used to compute the line averages (i.e. an instance solved 10 times by all settings counts more than an instance solved fewer than 10 times by all settings).

Function `ratio` solves marginally more instances than the `product` function. Both functions `ratio` and `svts` slightly outperform `product`, in terms of time and nodes used. The difference in performance is especially apparent when considering the total resources used, as it mostly reflects the performance on large instances, on which the scoring functions we introduce perform better. In Section 7.2.2, we will consider a set of instances that contains instances that experimentally require large B&B trees, and on which this observation is confirmed.

| Instance | product | | | ratio | | | svts | | |
|-----------------|---------|----------|--------|-------|-------------|-------------|------|-------------|-------------|
| | # | time (s) | nodes | # | time (s) | nodes | # | time (s) | nodes |
| 30n20b8 | 10 | 521 | 10 | +0 | 1.02 | 1.49 | +0 | 1.01 | 1.34 |
| aflow30a | 10 | 36 | 2.5k | +0 | 0.98 | 0.99 | +0 | 0.96 | 0.89 |
| aflow40b | 10 | 3483 | 210.9k | +0 | 0.68 | 0.65 | +0 | 0.70 | 0.63 |
| air04 | 10 | 30 | 7 | +0 | 0.96 | 1.00 | +0 | 1.03 | 1.00 |
| app1-2 | 10 | 845 | 565 | +0 | 0.90 | 1.01 | +0 | 0.89 | 1.14 |
| arki001 | 2 | 4862 | 1.3m | +0 | 0.73 | 0.74 | +1 | 0.85 | 0.80 |
| ash608gpia-3col | 10 | 87 | 5 | +0 | 1.04 | 1.00 | +0 | 1.01 | 1.00 |
| bell5 | 10 | 1 | 1.1k | +0 | 0.99 | 1.00 | +0 | 0.99 | 1.00 |
| biella1 | 10 | 217 | 2.8k | +0 | 0.98 | 1.06 | +0 | 0.99 | 1.00 |
| bienst2 | 10 | 288 | 112.6k | +0 | 1.05 | 0.98 | +0 | 1.11 | 1.09 |
| binkar10_1 | 10 | 431 | 135.8k | +0 | 1.09 | 1.03 | +0 | 1.01 | 0.98 |
| blend2 | 10 | 2 | 222 | +0 | 0.99 | 1.00 | +0 | 1.01 | 1.01 |
| cap6000 | 10 | 8 | 1.8k | +0 | 1.00 | 1.04 | +0 | 1.01 | 1.05 |
| csched010 | 9 | 6035 | 721.2k | -1 | 0.86 | 0.86 | +0 | 0.90 | 0.84 |
| danoint | 2 | 7047 | 1.2m | -1 | 1.00 | 0.96 | -1 | 1.02 | 0.97 |
| dcmulti | 10 | 2 | 9 | +0 | 1.00 | 1.00 | +0 | 1.00 | 1.00 |
| dfn-gwin-UUM | 10 | 256 | 62.9k | +0 | 1.03 | 1.07 | +0 | 1.05 | 0.96 |
| eil33-2 | 10 | 103 | 496 | +0 | 0.97 | 0.99 | +0 | 1.11 | 0.97 |
| eilB101 | 10 | 134 | 4.4k | +0 | 0.82 | 0.84 | +0 | 0.88 | 0.76 |
| enlight13 | 0 | - | - | +10 | - | - | +0 | - | - |
| fast0507 | 10 | 129 | 494 | +0 | 0.98 | 1.18 | +0 | 0.99 | 1.22 |
| fiber | 10 | 5 | 12 | +0 | 1.00 | 1.00 | +0 | 1.01 | 1.10 |
| fixnet6 | 10 | 5 | 9 | +0 | 1.00 | 1.00 | +0 | 1.01 | 1.02 |
| gesa2-o | 10 | 4 | 4 | +0 | 1.01 | 1.00 | +0 | 1.01 | 1.00 |
| gesa2_o | 10 | 4 | 5 | +0 | 0.99 | 1.00 | +0 | 1.00 | 1.00 |
| gesa3 | 10 | 4 | 20 | +0 | 0.99 | 1.00 | +0 | 0.99 | 0.97 |
| gesa3_o | 10 | 4 | 9 | +0 | 0.99 | 1.00 | +0 | 1.00 | 1.00 |
| glass4 | 10 | 29 | 11.5k | +0 | 1.26 | 1.34 | +0 | 1.05 | 1.03 |
| iis-100-0-cov | 10 | 2286 | 83.6k | +0 | 0.97 | 0.99 | +0 | 0.97 | 0.99 |
| iis-pima-cov | 10 | 806 | 6.7k | +0 | 1.00 | 0.96 | +0 | 1.00 | 0.96 |
| khh05250 | 10 | 1 | 7 | +0 | 0.99 | 1.00 | +0 | 0.98 | 1.00 |
| l152lav | 10 | 2 | 16 | +0 | 1.01 | 1.05 | +0 | 1.01 | 1.00 |
| lseu | 10 | 2 | 1.0k | +0 | 0.92 | 0.83 | +0 | 1.09 | 1.11 |
| map18 | 10 | 462 | 260 | +0 | 1.03 | 0.99 | +0 | 1.05 | 1.00 |
| map20 | 10 | 422 | 300 | +0 | 1.06 | 1.00 | +0 | 1.00 | 0.98 |
| mas74 | 10 | 1810 | 3.3m | +0 | 0.87 | 0.88 | +0 | 0.92 | 0.90 |

Table 5: Comparison of scoring function on the benchmark test set (continues in the next page).

| Instance | product | | | ratio | | | svts | | |
|------------------|---------|-------------|---------------|-------|-------------|-------------|------|-------------|-------------|
| | # | time (s) | nodes | # | time (s) | nodes | # | time (s) | nodes |
| mas76 | 10 | 200 | 484.2k | +0 | 0.83 | 0.83 | +0 | 0.98 | 0.98 |
| mcsched | 10 | 268 | 18.1k | +0 | 0.99 | 1.01 | +0 | 0.98 | 0.95 |
| mik-250-1-100-1 | 10 | 1832 | 829.9k | +0 | 1.18 | 1.17 | +0 | 1.30 | 1.31 |
| mine-166-5 | 10 | 30 | 356 | +0 | 1.01 | 0.95 | +0 | 1.05 | 1.13 |
| mine-90-10 | 10 | 1228 | 116.9k | +0 | 0.92 | 0.96 | +0 | 1.13 | 1.18 |
| misc03 | 10 | 2 | 125 | +0 | 0.98 | 1.02 | +0 | 0.98 | 0.97 |
| misc06 | 10 | 1 | 7 | +0 | 0.99 | 1.00 | +0 | 1.01 | 1.00 |
| misc07 | 10 | 45 | 23.8k | +0 | 1.24 | 1.28 | +0 | 1.06 | 0.99 |
| mod008 | 10 | 5 | 12 | +0 | 0.99 | 1.07 | +0 | 0.99 | 1.00 |
| mod011 | 10 | 74 | 886 | +0 | 0.97 | 0.89 | +0 | 0.98 | 0.96 |
| modglob | 10 | 1 | 20 | +0 | 1.00 | 1.04 | +0 | 1.00 | 1.01 |
| momentum2 | 1 | 4417 | 13.6k | +1 | 1.30 | 1.33 | +1 | 1.15 | 1.13 |
| msc98-ip | 10 | 1959 | 6.3k | +0 | 0.90 | 1.00 | +0 | 0.77 | 0.75 |
| mzzv11 | 10 | 732 | 233 | +0 | 0.99 | 0.89 | +0 | 1.00 | 0.91 |
| mzzv42z | 10 | 554 | 11 | +0 | 1.01 | 1.04 | +0 | 0.97 | 1.04 |
| n4-3 | 10 | 1213 | 46.6k | +0 | 1.04 | 0.99 | +0 | 1.03 | 0.95 |
| neos-1109824 | 10 | 395 | 23.2k | +0 | 0.52 | 0.37 | +0 | 0.93 | 0.85 |
| neos-1396125 | 10 | 486 | 69.9k | +0 | 1.00 | 1.01 | +0 | 0.81 | 0.81 |
| neos-476283 | 10 | 431 | 131 | +0 | 0.88 | 0.99 | +0 | 1.00 | 0.94 |
| neos-686190 | 10 | 142 | 1.8k | +0 | 1.13 | 1.22 | +0 | 1.13 | 1.18 |
| neos13 | 10 | 275 | 12 | +0 | 0.95 | 0.98 | +0 | 1.00 | 0.98 |
| neos18 | 10 | 76 | 5.3k | +0 | 1.27 | 1.59 | +0 | 1.12 | 1.16 |
| net12 | 10 | 2011 | 2.9k | +0 | 1.03 | 1.07 | +0 | 1.08 | 1.08 |
| netdiversion | 10 | 1463 | 13 | +0 | 0.97 | 0.92 | +0 | 0.95 | 1.04 |
| newdano | 5 | 5636 | 1.3m | -1 | 1.02 | 0.96 | +0 | 0.95 | 0.97 |
| noswot | 10 | 573 | 679.8k | +0 | 1.54 | 1.57 | +0 | 1.43 | 1.36 |
| ns1208400 | 10 | 414 | 174 | +0 | 1.10 | 1.20 | +0 | 1.13 | 1.33 |
| ns1688347 | 10 | 81 | 92 | +0 | 1.02 | 1.14 | +0 | 1.05 | 0.92 |
| ns1766074 | 10 | 5740 | 925.7k | +0 | 0.98 | 0.98 | +0 | 0.99 | 0.98 |
| ns1830653 | 10 | 492 | 18.6k | +0 | 0.98 | 1.03 | +0 | 0.91 | 0.92 |
| nsrand-ipx | 3 | 5004 | 584.7k | +7 | 0.33 | 0.30 | +2 | 0.50 | 0.49 |
| nw04 | 10 | 2719 | 10 | +0 | 0.84 | 1.00 | +0 | 0.93 | 1.00 |
| opm2-z7-s2 | 10 | 858 | 977 | +0 | 1.03 | 1.00 | +0 | 0.92 | 0.79 |
| p0201 | 10 | 3 | 10 | +0 | 0.99 | 1.02 | +0 | 1.00 | 1.02 |
| p2756 | 10 | 4 | 10 | +0 | 1.00 | 1.00 | +0 | 1.01 | 1.00 |
| pg5_34 | 10 | 1397 | 116.0k | +0 | 0.93 | 0.93 | +0 | 0.96 | 0.94 |
| pk1 | 10 | 123 | 325.8k | +0 | 1.13 | 1.14 | +0 | 1.09 | 1.06 |
| pp08a | 10 | 2 | 237 | +0 | 1.00 | 1.01 | +0 | 0.98 | 1.01 |
| pp08aCUTS | 10 | 2 | 153 | +0 | 0.99 | 1.02 | +0 | 0.99 | 1.05 |
| pw-myciel4 | 0 | - | - | +7 | - | - | +4 | - | - |
| qiu | 10 | 106 | 12.5k | +0 | 1.00 | 1.02 | +0 | 0.98 | 0.98 |
| qnet1 | 10 | 11 | 4 | +0 | 0.98 | 1.00 | +0 | 1.00 | 1.00 |
| qnet1_o | 10 | 6 | 4 | +0 | 0.97 | 1.00 | +0 | 0.98 | 1.00 |
| rail507 | 10 | 130 | 507 | +0 | 1.07 | 1.17 | +0 | 1.08 | 1.19 |
| ran16x16 | 10 | 544 | 349.3k | +0 | 0.82 | 0.82 | +0 | 0.81 | 0.78 |
| rd-rplusc-21 | 0 | - | - | +1 | - | - | +0 | - | - |
| reblock67 | 10 | 250 | 47.9k | +0 | 0.98 | 1.02 | +0 | 0.93 | 0.94 |
| rmatr100-p10 | 10 | 109 | 799 | +0 | 1.03 | 1.00 | +0 | 1.02 | 0.93 |
| rmatr100-p5 | 10 | 140 | 387 | +0 | 0.96 | 0.99 | +0 | 0.95 | 0.99 |
| rmine6 | 3 | 3540 | 380.6k | +1 | 1.10 | 1.05 | -2 | 2.00 | 2.39 |
| rocII-4-11 | 10 | 419 | 4.7k | +0 | 2.43 | 5.91 | +0 | 1.27 | 1.80 |
| rococoC10-001000 | 1 | 4677 | 323.9k | +1 | 0.83 | 0.81 | +1 | 0.44 | 0.50 |
| roll3000 | 9 | 3781 | 429.3k | -2 | 1.15 | 1.18 | -3 | 0.84 | 0.83 |
| rout | 10 | 111 | 44.6k | +0 | 0.73 | 0.73 | +0 | 0.94 | 0.94 |

Table 5: Comparison of scoring function on the benchmark test set (continues in the next page).

| Instance | product | | | ratio | | | svts | | |
|----------------|-----------|-------------|---------------|-------|-------------|-------------|------|-------------|-------------|
| | # | time (s) | nodes | # | time (s) | nodes | # | time (s) | nodes |
| satellites1-25 | 10 | 1590 | 3.4k | +0 | 1.03 | 0.98 | +0 | 1.02 | 1.01 |
| set1ch | 10 | 2 | 5 | +0 | 1.00 | 0.92 | +0 | 1.00 | 0.92 |
| sp98ic | 10 | 4736 | 276.2k | +0 | 0.49 | 0.42 | +0 | 0.70 | 0.58 |
| sp98ir | 10 | 71 | 1.6k | +0 | 0.97 | 1.16 | +0 | 0.98 | 1.15 |
| stein27 | 10 | 2 | 4.2k | +0 | 0.98 | 0.98 | +0 | 1.00 | 0.99 |
| stein45 | 10 | 34 | 51.5k | +0 | 0.92 | 0.98 | +0 | 0.97 | 0.99 |
| tanglegram1 | 7 | 1749 | 289 | -1 | 1.13 | 1.06 | -2 | 0.90 | 0.58 |
| tanglegram2 | 10 | 10 | 4 | +0 | 0.98 | 1.00 | +0 | 1.00 | 1.00 |
| timtab1 | 10 | 1184 | 774.5k | +0 | 1.04 | 1.04 | +0 | 1.01 | 1.02 |
| tr12-30 | 10 | 3644 | 780.2k | +0 | 1.02 | 1.08 | +0 | 1.00 | 1.02 |
| unitcal_7 | 9 | 3415 | 32.0k | +1 | 0.76 | 0.75 | +0 | 0.91 | 0.87 |
| vpm2 | 10 | 2 | 180 | +0 | 1.00 | 1.05 | +0 | 0.98 | 1.01 |
| zib54-UUE | 3 | 4914 | 413.0k | +0 | 1.20 | 1.18 | +0 | 1.09 | 1.13 |
| Total | 1225 | 705680 | 121.4m | +24 | 0.92 | 0.97 | +1 | 0.94 | 0.97 |
| Geo. mean | | 98 | 1.2k | | 0.98 | 1.00 | | 0.99 | 0.98 |
| Sh. geo. mean | | 148 | 2.8k | | 0.98 | 1.00 | | 0.99 | 0.98 |

Table 5: Comparison of scoring function on the benchmark test set.

7.2.2 Instances with large B&B trees

MIPLIB 2010 also has a so-called Tree set with 52 instances, which “contains instances that (empirically) lead to large enumeration trees” [8]. These instances have been selected because the scoring functions `ratio` and `svts` defined in Section 6 have specifically been designed to solve instances with large B&B trees. We have tested our scoring functions on these instances in the same setup as the experiments of Section 7.2.1, but we gave a time limit of 12 hours to allow for a more significant number of instances to be solved. Including permutations, the testbed comprises 520 instances, and required the equivalent of 19 months of running time on a single machine.

Table 6 provides the result for the tree test set with the same notation as Table 5 in Section 7.2.1. Both functions `ratio` and `svts` largely outperform `product`, both in terms of time and node, and solve slightly more instances. Note that some instances in this test set do not require larger trees than those in the benchmark test set (e.g. `glass4`, `ns894788` and `pg`). This is probably due to the fact that MIP solvers have made significant progress for these instances since this test set was designed. A more extensive report of the numerical experiments is provided in the appendices, Section B.

| Instance | product | | | ratio | | | svts | | |
|---------------|-----------|--------------|---------------|-------|-------------|-------------|------|-------------|-------------|
| | # | time (s) | nodes | # | time (s) | nodes | # | time (s) | nodes |
| csched007 | 10 | 31000 | 5.7m | +0 | 0.78 | 0.75 | +0 | 0.82 | 0.76 |
| csched008 | 10 | 1440 | 112.4k | +0 | 1.21 | 1.03 | +0 | 1.14 | 1.06 |
| glass4 | 10 | 28 | 11.5k | +0 | 1.37 | 1.34 | +0 | 1.03 | 1.03 |
| gmu-35-40 | 0 | - | - | +2 | - | - | +1 | - | - |
| k16x240 | 10 | 23835 | 23.2m | +0 | 0.88 | 0.89 | +0 | 0.86 | 0.85 |
| neos-1616732 | 10 | 9182 | 2.5m | +0 | 0.85 | 0.94 | +0 | 0.82 | 0.84 |
| neos-942830 | 10 | 9184 | 1.5m | +0 | 0.95 | 0.95 | +0 | 1.00 | 0.94 |
| neos15 | 1 | 34352 | 12.3m | +0 | 0.91 | 0.94 | +0 | 0.87 | 0.91 |
| neos858960 | 10 | 3997 | 2.8m | +0 | 0.92 | 0.93 | +0 | 0.92 | 0.93 |
| noswot | 10 | 556 | 679.8k | +0 | 1.62 | 1.57 | +0 | 1.44 | 1.36 |
| ns1766074 | 10 | 5747 | 925.7k | +0 | 0.95 | 0.98 | +0 | 0.98 | 0.98 |
| pg | 10 | 24 | 203 | +0 | 1.01 | 0.97 | +0 | 1.01 | 1.10 |
| ran14x18 | 10 | 23039 | 18.4m | +0 | 0.83 | 0.82 | +0 | 0.87 | 0.81 |
| reblock166 | 4 | 16858 | 1.6m | +2 | 0.98 | 1.00 | +2 | 0.79 | 0.95 |
| timtab1 | 10 | 1142 | 774.5k | +0 | 1.05 | 1.04 | +0 | 1.02 | 1.02 |
| umts | 10 | 8726 | 1.5m | +0 | 0.69 | 0.71 | +0 | 0.64 | 0.65 |
| wachplan | 10 | 15334 | 324.2k | +0 | 1.02 | 1.01 | +0 | 1.01 | 1.01 |
| Total | 155 | 1434131 | 602.1m | +4 | 0.87 | 0.87 | +3 | 0.87 | 0.84 |
| Geo. mean | | 2879 | 618.6k | | 0.97 | 0.96 | | 0.95 | 0.95 |
| Sh. geo. mean | | 3021 | 636.4k | | 0.97 | 0.96 | | 0.95 | 0.95 |

Table 6: Comparison of scoring function on the tree test set.

8 Conclusions

We developed one of the first models for the branching component of the Branch & Bound algorithm, over fifty years after its introduction. We proved that these models are relevant by theoretically establishing a new scoring function that is efficient for MIP solving. Numerous questions naturally arise regarding these models.

One type of questions concerns the computational complexity of the decision problems we have defined. For instance, can MVB be solved in polynomial time (even for two variables)? Is there an approximation algorithm for the minimization version of GVB?

Scoring functions are the only components of B&B that we numerically analyze through the theory we develop in this paper. Are there other elements of B&B that can be studied via the current models? There certainly exists decision problems that can model the B&B algorithm more accurately than GVB (e.g. if the gains of a variable are not fixed). Would more general models allow for a broader study of B&B, and for additional MIP solving improvements?

Acknowledgments

The authors would like to thank Eduardo Uchoa for pointing out reference [9] and Graham Farr for helping with the discussion at the end of Section 5.1. This research was funded by AFOSR grant FA9550-12-1-0151 of the Air Force Office of Scientific Research and the National Science Foundation Grant CCF-1415460 to the Georgia Institute of Technology.

Appendices

A Proofs

A.1 Proof of Theorem 3 and Corollary 4

Theorem. *When G tends to infinity, both sequences $\sqrt[l]{\frac{t(G+l)}{t(G)}}$ and $\sqrt[r]{\frac{t(G+r)}{t(G)}}$ converge to φ , which is the unique root greater than 1 of the equation $p(x) = x^r - x^{r-l} - 1 = 0$.*

Proof. Proposition 2 proves the case $l = r$, therefore we suppose throughout the proof that $r > l$. First, we define the notation

$$\begin{aligned} \underline{L} &= \liminf_{G \rightarrow \infty} \frac{t(G+l)}{t(G)} & \bar{L} &= \limsup_{G \rightarrow \infty} \frac{t(G+l)}{t(G)} \\ \underline{R} &= \liminf_{G \rightarrow \infty} \frac{t(G+r)}{t(G)} & \bar{R} &= \limsup_{G \rightarrow \infty} \frac{t(G+r)}{t(G)}. \end{aligned}$$

Using the recursive definition 1 of t between the first and the second lines, we can easily establish:

$$\begin{aligned} \underline{L} &= \left(\limsup_{G \rightarrow \infty} \frac{t(G)}{t(G+l)} \right)^{-1} = \left(\limsup_{G \rightarrow \infty} \frac{t(G-l)}{t(G)} \right)^{-1} \\ &= \left(\limsup_{G \rightarrow \infty} \frac{t(G) - t(G-r)}{t(G)} \right)^{-1} = \left(1 - \limsup_{G \rightarrow \infty} \frac{t(G-r)}{t(G)} \right)^{-1} \\ &= \left(1 - \limsup_{G \rightarrow \infty} \frac{t(G)}{t(G+r)} \right)^{-1} = \left(1 - \left(\liminf_{G \rightarrow \infty} \frac{t(G+r)}{t(G)} \right)^{-1} \right)^{-1} \\ &= (1 - \underline{R}^{-1})^{-1} = 1 + \frac{1}{\underline{R} - 1}. \end{aligned}$$

Similarly, $\bar{L} = 1 + \frac{1}{\bar{R}-1}$. Next, we obtain

$$\begin{aligned} \liminf_{G \rightarrow \infty} \frac{t(G+lr)}{t(G)} &= \liminf_{G \rightarrow \infty} \frac{t(G+lr)}{t(G+(l-1)r)} \frac{t(G+(l-1)r)}{t(G+(l-2)r)} \cdots \frac{t(G+r)}{t(G)} \\ &\geq \liminf_{G \rightarrow \infty} \frac{t(G+lr)}{t(G+(l-1)r)} \liminf_{G \rightarrow \infty} \frac{t(G+(l-1)r)}{t(G+(l-2)r)} \cdots \liminf_{G \rightarrow \infty} \frac{t(G+r)}{t(G)} \\ &\geq \left(\liminf_{G \rightarrow \infty} \frac{t(G+r)}{t(G)} \right)^l = \underline{R}^l. \end{aligned}$$

Also, we have

$$\begin{aligned} \limsup_{G \rightarrow \infty} \frac{t(G+lr)}{t(G)} &= \limsup_{G \rightarrow \infty} \frac{t(G+lr)}{t(G+l(r-1))} \cdots \frac{t(G+l)}{t(G)} \\ &\leq \limsup_{G \rightarrow \infty} \frac{t(G+lr)}{t(G+l(r-1))} \cdots \limsup_{G \rightarrow \infty} \frac{t(G+l)}{t(G)} \\ &\leq \left(\limsup_{G \rightarrow \infty} \frac{t(G+l)}{t(G)} \right)^r = \bar{L}^r. \end{aligned}$$

Together, these yield

$$\bar{L}^r \geq \limsup_{G \rightarrow \infty} \frac{t(G+lr)}{t(G)} \geq \liminf_{G \rightarrow \infty} \frac{t(G+lr)}{t(G)} \geq \underline{R}^l.$$

Following the same steps, we can show $\bar{R}^l \geq \underline{L}^r$. Starting from this inequality, we prove

$$\bar{R}^{\frac{l}{r}} \geq \underline{L} = 1 + \frac{1}{\underline{R}-1} \Rightarrow \underline{R} \geq 1 + \frac{1}{\bar{R}^{\frac{l}{r}}-1}.$$

Likewise, we can show that the inequality $\bar{R} \leq 1 + \frac{1}{\bar{R}^{\frac{l}{r}}-1}$ holds.

We now introduce two monotonic sequences α_n and ω_n that respectively bound \underline{R} from below and bound \bar{R} from above, and we prove that they converge to the same limit. For all non-negative n , let α_n and ω_n be defined as follows:

$$\alpha_n = \begin{cases} 2 & \text{if } n = 0 \\ f(\omega_{n-1}) & \text{otherwise} \end{cases} \quad \omega_n = f(\alpha_n)$$

where $f(x) = 1 + \frac{1}{x^{\frac{l}{r}}-1}$ for all $x \in (1, \infty)$. We first prove by induction that for all non-negative integer n , α_n and ω_n satisfy $\alpha_n \leq \underline{R} \leq \bar{R} \leq \omega_n$. Proposition 2 ensures that $\alpha_0 = 2$ is a lower bound on \underline{R} . Suppose that for

a given non-negative n , the inequality $\alpha_n \leq \underline{R}$ holds. We prove that $\bar{R} \leq \omega_n$ holds too:

$$\begin{aligned}\alpha_n \leq \underline{R} &\Rightarrow 1 + \frac{1}{\alpha_n^{\frac{l}{r}} - 1} \geq 1 + \frac{1}{\underline{R}^{\frac{l}{r}} - 1} \geq \bar{R} \\ &\Rightarrow f(\alpha_n) \geq \bar{R} \\ &\Rightarrow \omega_n \geq \bar{R}.\end{aligned}$$

The same reasoning also proves that for all non-negative integers n , $\omega_n \geq \bar{R}$ implies $\alpha_{n+1} \leq \underline{R}$. The sequence α_n thus bounds \underline{R} from below, and ω_n bounds \bar{R} from above.

We now prove that α_n is monotonically increasing. For a given non-negative n , consider the inequality:

$$\begin{aligned}\alpha_1 > \alpha_0 &\Leftrightarrow f(f(2)) > 2 \Leftrightarrow \frac{1}{f(2)^{\frac{l}{r}} - 1} > 1 \Leftrightarrow f(2) < 2^{\frac{r}{l}} \\ &\Leftrightarrow 1 + \frac{1}{2^{\frac{l}{r}} - 1} < 2^{\frac{r}{l}} \Leftrightarrow 2^{\frac{l}{r}} < 2^{\frac{r}{l}}(2^{\frac{l}{r}} - 1) \Leftrightarrow r > l.\end{aligned}$$

Suppose now that for a given non-negative n , $\alpha_n > \alpha_{n-1}$. This implies that $\omega_n < \omega_{n-1}$:

$$\omega_n - \omega_{n-1} = \frac{1}{\alpha_n^{\frac{l}{r}} - 1} - \frac{1}{\alpha_{n-1}^{\frac{l}{r}} - 1} < 0 \Leftrightarrow \alpha_n^{\frac{l}{r}} > \alpha_{n-1}^{\frac{l}{r}}.$$

In turn, for that given n , $\omega_n < \omega_{n-1}$ implies $\alpha_{n+1} > \alpha_n$. The sequence α_n and ω_n are thus increasing and decreasing, respectively. Since they are bounded, each of them converges to one of the solutions of the fixed-point equation $x = f(x)$. We now prove that there is a unique solution to that equation:

$$\begin{aligned}x = f(x) &\Leftrightarrow x = 1 + \frac{1}{x^{\frac{l}{r}} - 1} \Leftrightarrow (x^{\frac{l}{r}} - 1)(x - 1) = 1 \\ &\Leftrightarrow x^{\frac{l+r}{r}} - x^{\frac{l}{r}} - x = 0 \\ &\Leftrightarrow p(X) = X^r - X^{r-l} - 1 = 0\end{aligned}$$

where $X = x^{\frac{1}{r}}$. We establish in Theorem 5 that the polynomial p has a unique root in $(1, \infty)$, hence the fixed-point equation $x = f(x)$ also has a unique solution. Consequently, it is necessary that both sequences α_n and ω_n converge to this unique fixed point, and thus $\underline{R} = \bar{R}$. Furthermore, the sequence $\sqrt[r]{\frac{t(G+r)}{t(G)}}$ converges to the root $\varphi > 1$ of the polynomial p .

Since we have established that $\underline{L} = 1 + \frac{1}{R-1}$ and $\bar{L} = 1 + \frac{1}{R-1}$, it follows that $\underline{L} = \bar{L}$. Since $p(\varphi) = 0$, φ equivalently satisfies $\varphi^l = 1 + \frac{1}{\varphi^{r-1}}$, therefore $\underline{L} = \bar{L} = \varphi^l$. \square

Corollary. *A numerical approximation of φ^r is given by the fixed-point iteration*

$$f(x) = 1 + \frac{1}{x^{\frac{l}{r}} - 1}$$

with the starting point $x = 2$.

Proof. Recall the definition of the sequences α_n and ω_n as given in the proof of Theorem 3 (notice that function f has the same definition). The sequence f_n generated by the fixed-point equation is $(2, f(2), f(f(2)), f(f(f(2))), \dots)$, which is equal to $(\alpha_0, \omega_0, \alpha_1, \omega_1, \dots)$. Formally, the sequence f_n generated by the fixed-point equation satisfies

$$f_n = \begin{cases} \alpha_{\frac{n}{2}} & \text{if } n \text{ is even} \\ \omega_{\frac{n+1}{2}} & \text{if } n \text{ is odd.} \end{cases}$$

In the proof of Theorem 3, we prove that both α_n and ω_n converge to φ^r when n tends to infinity, therefore f_n also converges to φ^r . \square

A.2 Proof of Theorem 6

Recall that z is the least common multiple of all l_i and r_i .

Theorem. $\varphi = \min_i \varphi_i$

Proof. Let α and ω be such that for all positive integers $x \leq z$,

$$\alpha^x \leq \liminf_{G \rightarrow \infty} \frac{t(G+x)}{t(G)} \leq \limsup_{G \rightarrow \infty} \frac{t(G+x)}{t(G)} \leq \omega^x. \quad (7)$$

Possible values are $\alpha = 1$ and $\omega = 2$. Similar to the proof of Theorem 3, we use the notation

$$\underline{Z} = \liminf_{G \rightarrow \infty} \frac{t(G+z)}{t(G)} \quad \bar{Z} = \limsup_{G \rightarrow \infty} \frac{t(G+z)}{t(G)}.$$

For all variables i , we have:

$$\begin{aligned}
\bar{Z} &= \limsup_{G \rightarrow \infty} \left(\frac{t(G+z)}{t(G+z-l_i)} \frac{t(G+z-l_i)}{t(G+z-2l_i)} \cdots \frac{t(G+z-(\frac{z}{l_i}-1)l_i)}{t(G)} \right) \\
&\leq \left(\limsup_{G \rightarrow \infty} \frac{t(G+l_i)}{t(G)} \right)^{\frac{z}{l_i}} \\
&\leq \left(\limsup_{G \rightarrow \infty} \frac{1 + \min_j (t(G+l_i-l_j) + t(G+l_i-r_j))}{t(G)} \right)^{\frac{z}{l_i}} \\
&\leq \left(1 + \limsup_{G \rightarrow \infty} \frac{t(G+l_i-r_i)}{t(G)} \right)^{\frac{z}{l_i}} \\
&\leq \left(1 + \left(\liminf_{G \rightarrow \infty} \frac{t(G)}{t(G+l_i-r_i)} \right)^{-1} \right)^{\frac{z}{l_i}} \\
&\leq \left(1 + \left(\liminf_{G \rightarrow \infty} \frac{t(G+r_i-l_i)}{t(G)} \right)^{-1} \right)^{\frac{z}{l_i}} \\
&\leq \left(1 + \alpha^{l_i-r_i} \right)^{\frac{z}{l_i}}
\end{aligned}$$

where the last line follows from the lower bound on α^{-x} from (7). Suppose there exists a variable i such that $\alpha > \varphi_i$, then we establish, using Theorem 5:

$$\begin{aligned}
\alpha > \varphi_i &\Rightarrow p_i(\alpha) > 0 \\
&\Rightarrow \alpha^{r_i} - \alpha^{r_i-l_i} - 1 > 0 \\
&\Rightarrow 1 + \alpha^{l_i-r_i} < \alpha^{l_i} \\
&\Rightarrow (1 + \alpha^{l_i-r_i})^{\frac{z}{l_i}} < (\alpha^{l_i})^{\frac{z}{l_i}} \\
&\Rightarrow \bar{Z} < \alpha^z.
\end{aligned}$$

This contradicts expression (7), hence for all variables i , $\alpha \leq \varphi_i$. Moreover, since $\alpha^z > \varphi_i^z$ implies $\alpha^z > \bar{Z}$, then $\varphi_i^z \geq \bar{Z}$ must hold for all variables i . In addition, for each variable i , the following is satisfied:

$$\begin{aligned}
\underline{Z} &\geq \left(\liminf_{G \rightarrow \infty} \frac{t(G+r_i)}{t(G)} \right)^{\frac{z}{r_i}} \\
&\geq \left(\liminf_{G \rightarrow \infty} \frac{1 + \min_j (t(G+r_i-l_j) + t(G+r_i-r_j))}{t(G)} \right)^{\frac{z}{r_i}}
\end{aligned}$$

Hence there must exist a variable j such that

$$\begin{aligned} \underline{Z} &\geq \left(1 + \liminf_{G \rightarrow \infty} \frac{t(G + r_j - l_j)}{t(G)}\right)^{\frac{z}{r_j}} \\ &\geq \left(1 + \alpha^{r_j - l_j}\right)^{\frac{z}{r_j}}. \end{aligned}$$

Suppose $\alpha < \varphi_j$, then, using Theorem 5 in the first line:

$$\begin{aligned} \alpha < \varphi_j &\Rightarrow p_j(\alpha) < 0 \\ &\Rightarrow 1 + \alpha^{r_j - l_j} > \alpha^{r_j} \\ &\Rightarrow (1 + \alpha^{r_j - l_j})^{\frac{z}{r_j}} > (\alpha^{r_j})^{\frac{z}{r_j}} \\ &\Rightarrow \underline{Z} > \alpha^z. \end{aligned}$$

Since $\varphi_j^z > \alpha^z$ implies $\underline{Z} > \alpha^z$, then $\underline{Z} \geq \varphi_j^z$ must be true. Since $\varphi_i^z \geq \bar{Z}$ holds for all variables i , then variable j satisfies $\varphi_j^z \leq \underline{Z} \leq \bar{Z} \leq \varphi_i^z$ for all variables i . We can finally conclude that

$$\lim_{G \rightarrow \infty} \sqrt[z]{\frac{t(G + z)}{t(G)}} = \min_i \varphi_i^z.$$

□

B Detailed experimental results

We present in Section B.1 and B.2 the detailed results for the benchmark and tree test sets, as described in Section 7.2.1 and 7.2.2, respectively. There are three tables in each section, one for each of the scoring function tested, namely `product`, `ratio`, and `svts`. In each table, each line corresponds to all 10 permutations for each instance of the test set. The second column indicates the number of permuted instances solved. We provide statistics in terms of time, number of nodes and LP iterations. For each of these three measures, we give, from left to right in the table, the minimum, average over solved instances (and average over all 10 instances), and maximum. In the node and LP iterations columns, we use the letters k , m and b as a shorthand for thousands, millions and billions. Total and averages are provided at the end of each table. Note that it would not be fair to compare the measures over solved instances in this setting. Indeed, the set of solved instances differ depending on the scoring function considered. Function `ratio` generally solves more hard instances, thus the averages on solved instances have higher values.

B.1 Benchmark test set results

Tables 7, 8 and 9 give the results on the benchmark test set for the scoring functions `product`, `ratio`, and `svts`, respectively.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| 10teams | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 1.0k | 1.2k (1.2k) | 1.6k |
| 30n20b8 | 10 | 483 | 521 (521) | 544 | 7 | 10 (10) | 15 | 31.7k | 37.5k (37.5k) | 48.6k |
| alc1s1 | 0 | 7200 | - (7200) | 7200 | 308.8k | - (433.7k) | 577.7k | 11.9m | - (13.0m) | 14.6m |
| acc-tight5 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| aflow30a | 10 | 33 | 36 (36) | 42 | 1.8k | 2.5k (2.5k) | 4.2k | 34.5k | 45.1k (45.1k) | 64.9k |
| aflow40b | 10 | 2592 | 3483 (3483) | 4952 | 142.9k | 210.9k (210.9k) | 289.1k | 2.9m | 4.4m (4.4m) | 6.0m |
| air04 | 10 | 28 | 30 (30) | 40 | 6 | 7 (7) | 8 | 11.6k | 13.0k (13.0k) | 14.6k |
| app1-2 | 10 | 603 | 845 (845) | 1362 | 29 | 565 (565) | 2.0k | 8.5k | 28.3k (28.3k) | 84.0k |
| arki001 | 2 | 3738 | 4862 (6732) | 7200 | 1.0m | 1.3m (1.7m) | 1.9m | 5.3m | 6.8m (11.5m) | 17.0m |
| ash608gpia-3col | 10 | 80 | 87 (87) | 111 | 5 | 5 (5) | 9 | 8.0k | 10.3k (10.3k) | 19.1k |
| atlanta-ip | 0 | 7200 | - (7200) | 7200 | 4.1k | - (6.8k) | 9.4k | 1.9m | - (2.4m) | 2.9m |
| bab5 | 0 | 7200 | - (7200) | 7200 | 10.8k | - (16.2k) | 21.9k | 721.2k | - (946.2k) | 1.3m |
| beasleyC3 | 0 | 7200 | - (7200) | 7200 | 790.1k | - (1.1m) | 1.5m | 28.5m | - (32.8m) | 36.6m |
| bell5 | 10 | 1 | 1 (1) | 1 | 1.1k | 1.1k (1.1k) | 1.1k | 1.3k | 1.4k (1.4k) | 1.6k |
| biella1 | 10 | 163 | 217 (217) | 286 | 2.0k | 2.8k (2.8k) | 5.3k | 266.8k | 355.8k (355.8k) | 518.3k |
| bienst2 | 10 | 208 | 288 (288) | 414 | 82.9k | 112.6k (112.6k) | 167.1k | 2.1m | 2.9m (2.9m) | 3.9m |
| binkar10_1 | 10 | 268 | 431 (431) | 592 | 85.4k | 135.8k (135.8k) | 196.2k | 633.3k | 1.0m (1.0m) | 1.4m |
| blend2 | 10 | 1 | 2 (2) | 2 | 166 | 222 (222) | 305 | 780 | 960 (960) | 1.2k |
| bley_xl1 | 10 | 1155 | 2182 (2182) | 4539 | 1 | 1 (1) | 1 | 2.6k | 14.4k (14.4k) | 97.8k |
| bnatt350 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| cap6000 | 10 | 8 | 8 (8) | 8 | 1.7k | 1.8k (1.8k) | 1.8k | 2.6k | 2.7k (2.7k) | 2.7k |
| core2536-691 | 10 | 9 | 11 (11) | 13 | 1 | 1 (1) | 1 | 9.1k | 11.5k (11.5k) | 15.5k |
| cov1075 | 0 | 7200 | - (7200) | 7200 | 1.1m | - (1.2m) | 1.3m | 37.5m | - (40.3m) | 44.0m |
| csched010 | 9 | 5048 | 6151 (6256) | 7200 | 561.5k | 738.0k (752.7k) | 884.3k | 28.9m | 37.7m (38.4m) | 45.0m |
| danoint | 2 | 7047 | 7073 (7175) | 7200 | 1.2m | 1.2m (1.2m) | 1.3m | 51.7m | 51.9m (53.8m) | 57.0m |
| dcmulti | 10 | 2 | 2 (2) | 3 | 6 | 9 (9) | 10 | 1.5k | 1.6k (1.6k) | 1.7k |
| dfn-gwin-UUM | 10 | 239 | 256 (256) | 291 | 62.9k | 62.9k (62.9k) | 62.9k | 1.0m | 1.0m (1.0m) | 1.0m |
| disctom | 10 | 0 | 1 (1) | 2 | 0 | 1 (1) | 1 | 0 | 358 (358) | 1.1k |
| dsbmip | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 974 | 1.2k (1.2k) | 1.3k |
| egout | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 37 | 39 (39) | 42 |
| eil33-2 | 10 | 93 | 103 (103) | 123 | 442 | 496 (496) | 598 | 24.3k | 26.8k (26.8k) | 31.0k |
| eilB101 | 10 | 101 | 134 (134) | 204 | 2.7k | 4.4k (4.4k) | 8.6k | 235.0k | 318.5k (318.5k) | 480.9k |
| enigma | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| enlight13 | 0 | 7200 | - (7200) | 7200 | 4.1m | - (4.6m) | 4.9m | 18.7m | - (20.1m) | 21.2m |

Table 7: Detailed results for the product scoring function on the benchmark test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| enlight14 | 0 | 7200 | - (7200) | 7200 | 4.7m | - (5.1m) | 5.4m | 20.6m | - (22.1m) | 22.9m |
| ex9 | 10 | 2 | 2 (2) | 3 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| fast0507 | 10 | 116 | 129 (129) | 140 | 412 | 494 (494) | 534 | 48.1k | 54.1k (54.1k) | 58.7k |
| fiber | 10 | 4 | 5 (5) | 5 | 1 | 12 (12) | 51 | 735 | 1.1k (1.1k) | 1.6k |
| fixnet6 | 10 | 5 | 5 (5) | 5 | 7 | 9 (9) | 12 | 1.4k | 1.7k (1.7k) | 2.2k |
| flugpl | 10 | 0 | 0 (0) | 0 | 220 | 221 (221) | 221 | 188 | 188 (188) | 188 |
| gen | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 139 | 143 (143) | 146 |
| gesa2 | 10 | 3 | 4 (4) | 4 | 2 | 3 (3) | 3 | 1.2k | 1.3k (1.3k) | 1.4k |
| gesa2-o | 10 | 4 | 4 (4) | 4 | 2 | 4 (4) | 5 | 1.1k | 1.4k (1.4k) | 1.5k |
| gesa2_o | 10 | 4 | 4 (4) | 4 | 4 | 5 (5) | 5 | 1.4k | 1.6k (1.6k) | 1.7k |
| gesa3 | 10 | 4 | 4 (4) | 4 | 18 | 20 (20) | 22 | 1.6k | 1.7k (1.7k) | 1.7k |
| gesa3_o | 10 | 4 | 4 (4) | 4 | 7 | 9 (9) | 11 | 1.4k | 1.9k (1.9k) | 2.3k |
| glass4 | 10 | 15 | 29 (29) | 45 | 5.4k | 11.5k (11.5k) | 18.9k | 36.3k | 75.2k (75.2k) | 130.2k |
| gmu-35-40 | 0 | 7200 | - (7200) | 7200 | 8.1m | - (10.1m) | 12.4m | 24.9m | - (28.2m) | 33.0m |
| gt2 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 91 | 91 (91) | 91 |
| harp2 | 0 | 7200 | - (7200) | 7200 | 4.5m | - (5.0m) | 5.9m | 5.0m | - (5.7m) | 7.0m |
| iis-100-0-cov | 10 | 2194 | 2286 (2286) | 2461 | 80.4k | 83.6k (83.6k) | 86.9k | 3.3m | 3.4m (3.4m) | 3.5m |
| iis-bupa-cov | 0 | 7200 | - (7200) | 7200 | 132.0k | - (134.2k) | 137.1k | 7.8m | - (8.0m) | 8.2m |
| iis-pima-cov | 10 | 737 | 806 (806) | 926 | 6.1k | 6.7k (6.7k) | 7.6k | 375.4k | 411.6k (411.6k) | 469.6k |
| khb05250 | 10 | 1 | 1 (1) | 1 | 2 | 7 (7) | 9 | 323 | 554 (554) | 837 |
| l152lav | 10 | 1 | 2 (2) | 2 | 15 | 16 (16) | 17 | 1.5k | 1.8k (1.8k) | 2.0k |
| lectsched-4-obj | 10 | 1 | 1 (1) | 1 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| lseu | 10 | 2 | 2 (2) | 2 | 971 | 1.0k (1.0k) | 1.1k | 3.5k | 3.6k (3.6k) | 3.7k |
| m100n500k4r1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 166 | 203 (203) | 260 |
| macrophage | 0 | 7200 | - (7200) | 7200 | 955.7k | - (1.1m) | 1.2m | 17.1m | - (19.0m) | 22.1m |
| manna81 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 3.1k | 3.2k (3.2k) | 3.2k |
| map18 | 10 | 391 | 462 (462) | 578 | 225 | 260 (260) | 289 | 84.6k | 91.1k (91.1k) | 98.9k |
| map20 | 10 | 367 | 422 (422) | 480 | 277 | 300 (300) | 329 | 83.5k | 89.9k (89.9k) | 95.7k |
| markshare1 | 0 | 7200 | - (7200) | 7200 | 22.2m | - (23.1m) | 23.7m | 45.9m | - (48.0m) | 49.3m |
| markshare2 | 0 | 7200 | - (7200) | 7200 | 26.4m | - (26.8m) | 27.2m | 63.7m | - (64.8m) | 65.7m |
| mas74 | 10 | 1729 | 1810 (1810) | 1982 | 3.2m | 3.3m (3.3m) | 3.3m | 12.5m | 12.9m (12.9m) | 12.9m |
| mas76 | 10 | 197 | 200 (200) | 203 | 484.2k | 484.2k (484.2k) | 484.2k | 1.6m | 1.6m (1.6m) | 1.6m |
| mcsched | 10 | 209 | 268 (268) | 369 | 12.7k | 18.1k (18.1k) | 25.9k | 469.7k | 674.1k (674.1k) | 1.0m |
| mik-250-1-100-1 | 10 | 1732 | 1832 (1832) | 1995 | 787.0k | 829.9k (829.9k) | 904.1k | 2.5m | 2.7m (2.7m) | 2.9m |
| mine-166-5 | 10 | 28 | 30 (30) | 33 | 186 | 356 (356) | 722 | 3.9k | 4.8k (4.8k) | 5.7k |

Table 7: Detailed results for the product scoring function on the benchmark test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|--------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| mine-90-10 | 10 | 338 | 1228 (1228) | 2701 | 37.9k | 116.9k (116.9k) | 208.2k | 236.6k | 681.9k (681.9k) | 1.1m |
| misc03 | 10 | 1 | 2 (2) | 2 | 59 | 125 (125) | 239 | 1.9k | 2.8k (2.8k) | 3.9k |
| misc06 | 10 | 1 | 1 (1) | 2 | 3 | 7 (7) | 10 | 1.3k | 1.8k (1.8k) | 2.3k |
| misc07 | 10 | 36 | 45 (45) | 62 | 17.9k | 23.8k (23.8k) | 29.6k | 140.6k | 180.7k (180.7k) | 223.9k |
| mitre | 10 | 45 | 48 (48) | 54 | 1 | 1 (1) | 1 | 1.3k | 1.4k (1.4k) | 1.4k |
| mod008 | 10 | 5 | 5 (5) | 5 | 11 | 12 (12) | 12 | 544 | 594 (594) | 607 |
| mod010 | 10 | 1 | 1 (1) | 1 | 2 | 3 (3) | 4 | 905 | 1.0k (1.0k) | 1.3k |
| mod011 | 10 | 61 | 74 (74) | 88 | 601 | 886 (886) | 1.5k | 49.6k | 67.1k (67.1k) | 96.1k |
| modglob | 10 | 1 | 1 (1) | 2 | 10 | 20 (20) | 27 | 675 | 768 (768) | 1.0k |
| momentum1 | 0 | 7200 | - (7200) | 7200 | 18.6k | - (21.1k) | 26.2k | 892.2k | - (1.5m) | 1.9m |
| momentum2 | 1 | 4417 | 4417 (6922) | 7200 | 13.6k | 13.6k (22.9k) | 27.8k | 426.1k | 426.1k (733.2k) | 859.9k |
| msc98-ip | 10 | 934 | 1959 (1959) | 4088 | 2.3k | 6.3k (6.3k) | 16.3k | 783.4k | 1.9m (1.9m) | 4.3m |
| mspp16 | 0 | 7200 | - (7200) | 7200 | 0 | - (0) | 0 | 0 | - (0) | 0 |
| mzzv11 | 10 | 672 | 732 (732) | 832 | 12 | 233 (233) | 776 | 48.6k | 60.0k (60.0k) | 76.0k |
| mzzv42z | 10 | 493 | 554 (554) | 620 | 2 | 11 (11) | 20 | 32.9k | 40.5k (40.5k) | 46.5k |
| n3div36 | 0 | 7200 | - (7200) | 7200 | 96.8k | - (147.7k) | 173.9k | 985.7k | - (1.2m) | 1.5m |
| n3seq24 | 0 | 7200 | - (7200) | 7200 | 82.4k | - (116.2k) | 190.0k | 1.3m | - (2.1m) | 3.3m |
| n4-3 | 10 | 831 | 1213 (1213) | 1846 | 30.8k | 46.6k (46.6k) | 70.9k | 1.3m | 1.7m (1.7m) | 2.5m |
| neos-1109824 | 10 | 234 | 395 (395) | 632 | 10.1k | 23.2k (23.2k) | 40.7k | 103.8k | 207.0k (207.0k) | 353.5k |
| neos-1337307 | 0 | 7200 | - (7200) | 7200 | 231.6k | - (268.3k) | 307.0k | 3.5m | - (3.7m) | 4.0m |
| neos-1396125 | 10 | 236 | 486 (486) | 884 | 32.1k | 69.9k (69.9k) | 129.8k | 911.0k | 1.8m (1.8m) | 3.4m |
| neos-1601936 | 10 | 7 | 7 (7) | 8 | 1 | 1 (1) | 1 | 10.4k | 11.0k (11.0k) | 11.4k |
| neos-476283 | 10 | 314 | 431 (431) | 584 | 67 | 131 (131) | 201 | 8.0k | 11.3k (11.3k) | 16.5k |
| neos-686190 | 10 | 97 | 142 (142) | 173 | 1.6k | 1.8k (1.8k) | 2.0k | 34.2k | 37.2k (37.2k) | 39.9k |
| neos-849702 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| neos-916792 | 0 | 7200 | - (7200) | 7200 | 197.5k | - (239.2k) | 280.2k | 2.3m | - (2.8m) | 3.3m |
| neos-934278 | 0 | 7200 | - (7200) | 7200 | 7.8k | - (11.3k) | 18.1k | 5.6m | - (6.8m) | 7.8m |
| neos13 | 10 | 246 | 275 (275) | 294 | 12 | 12 (12) | 12 | 3.0k | 3.0k (3.0k) | 3.0k |
| neos18 | 10 | 58 | 76 (76) | 119 | 3.6k | 5.3k (5.3k) | 8.1k | 118.6k | 177.5k (177.5k) | 296.9k |
| net12 | 10 | 1279 | 2011 (2011) | 2973 | 2.1k | 2.9k (2.9k) | 3.7k | 1.1m | 2.0m (2.0m) | 3.3m |
| netdiversion | 10 | 1032 | 1463 (1463) | 2132 | 5 | 13 (13) | 57 | 30.5k | 46.3k (46.3k) | 86.3k |
| newdano | 5 | 4722 | 5939 (6569) | 7200 | 1.1m | 1.3m (1.5m) | 1.9m | 54.7m | 69.5m (73.1m) | 85.7m |
| noswot | 10 | 283 | 573 (573) | 1136 | 367.0k | 679.8k (679.8k) | 1.0m | 1.2m | 2.1m (2.1m) | 3.2m |
| ns1208400 | 10 | 225 | 414 (414) | 626 | 1 | 174 (174) | 533 | 142.1k | 286.5k (286.5k) | 433.0k |
| ns1688347 | 10 | 59 | 81 (81) | 105 | 6 | 92 (92) | 438 | 5.8k | 23.8k (23.8k) | 38.2k |

Table 7: Detailed results for the product scoring function on the benchmark test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| ns1758913 | 6 | 1651 | 3272 (4843) | 7200 | 1 | 1 (1) | 1 | 29.3k | 44.1k (50.4k) | 74.6k |
| ns1766074 | 10 | 5324 | 5740 (5740) | 6120 | 895.3k | 925.7k (925.7k) | 949.9k | 3.4m | 3.7m (3.7m) | 3.8m |
| ns1830653 | 10 | 402 | 492 (492) | 726 | 15.9k | 18.6k (18.6k) | 24.6k | 1.1m | 1.3m (1.3m) | 1.9m |
| nsrand-ipx | 3 | 2411 | 5004 (6541) | 7200 | 248.9k | 584.7k (903.1k) | 1.1m | 1.6m | 3.6m (5.4m) | 6.8m |
| nw04 | 10 | 1979 | 2719 (2719) | 3271 | 7 | 10 (10) | 12 | 1.3k | 1.6k (1.6k) | 1.9k |
| opm2-z7-s2 | 10 | 679 | 858 (858) | 1106 | 707 | 977 (977) | 1.4k | 43.2k | 64.4k (64.4k) | 86.8k |
| opt1217 | 5 | 2 | 2 (3601) | 7200 | 1 | 1 (1.7m) | 4.4m | 747 | 798 (9.3m) | 29.5m |
| p0033 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 57 | 57 (57) | 57 |
| p0201 | 10 | 2 | 3 (3) | 3 | 4 | 10 (10) | 39 | 1.1k | 1.4k (1.4k) | 2.8k |
| p0282 | 10 | 1 | 2 (2) | 2 | 2 | 3 (3) | 3 | 459 | 649 (649) | 890 |
| p0548 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 331 | 331 (331) | 331 |
| p2756 | 10 | 4 | 4 (4) | 4 | 10 | 10 (10) | 10 | 538 | 547 (547) | 555 |
| pg5_34 | 10 | 1257 | 1397 (1397) | 1824 | 116.0k | 116.0k (116.0k) | 116.0k | 2.2m | 2.2m (2.2m) | 2.2m |
| pigeon-10 | 0 | 7200 | - (7200) | 7200 | 929.9k | - (1.1m) | 1.1m | 4.2m | - (6.5m) | 9.1m |
| pk1 | 10 | 120 | 123 (123) | 126 | 325.8k | 325.8k (325.8k) | 325.8k | 1.9m | 1.9m (1.9m) | 1.9m |
| pp08a | 10 | 2 | 2 (2) | 3 | 107 | 237 (237) | 350 | 2.0k | 3.3k (3.3k) | 5.1k |
| pp08aCUTS | 10 | 2 | 2 (2) | 3 | 75 | 153 (153) | 245 | 1.8k | 2.5k (2.5k) | 3.8k |
| protfold | 0 | 7200 | - (7200) | 7200 | 15.6k | - (30.4k) | 48.7k | 15.0m | - (21.9m) | 26.5m |
| pw-myciel4 | 0 | 7200 | - (7200) | 7200 | 482.7k | - (561.1k) | 624.2k | 19.4m | - (21.9m) | 24.7m |
| qiu | 10 | 78 | 106 (106) | 125 | 9.6k | 12.5k (12.5k) | 14.3k | 304.7k | 410.6k (410.6k) | 495.1k |
| qnet1 | 10 | 9 | 11 (11) | 12 | 3 | 4 (4) | 6 | 3.7k | 4.0k (4.0k) | 4.5k |
| qnet1_o | 10 | 5 | 6 (6) | 6 | 3 | 4 (4) | 6 | 2.5k | 2.8k (2.8k) | 3.3k |
| rail507 | 10 | 114 | 130 (130) | 156 | 450 | 507 (507) | 540 | 51.6k | 54.6k (54.6k) | 56.3k |
| ran16x16 | 10 | 491 | 544 (544) | 606 | 322.5k | 349.3k (349.3k) | 397.2k | 2.6m | 2.9m (2.9m) | 3.2m |
| rd-rplusc-21 | 0 | 7200 | - (7200) | 7200 | 42.8k | - (70.9k) | 174.8k | 450.1k | - (729.0k) | 1.6m |
| reblock67 | 10 | 203 | 250 (250) | 312 | 41.6k | 47.9k (47.9k) | 57.5k | 459.6k | 504.7k (504.7k) | 618.0k |
| rentacar | 10 | 5 | 6 (6) | 7 | 2 | 2 (2) | 2 | 2.8k | 3.2k (3.2k) | 3.8k |
| rgn | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 2 | 327 | 392 (392) | 556 |
| rmatr100-p10 | 10 | 106 | 109 (109) | 116 | 789 | 799 (799) | 805 | 64.1k | 64.2k (64.2k) | 64.3k |
| rmatr100-p5 | 10 | 128 | 140 (140) | 160 | 387 | 387 (387) | 387 | 67.5k | 67.6k (67.6k) | 67.6k |
| rmine6 | 3 | 3540 | 4297 (6329) | 7200 | 380.6k | 461.5k (780.9k) | 985.0k | 2.6m | 3.1m (5.6m) | 7.0m |
| rocII-4-11 | 10 | 352 | 419 (419) | 522 | 3.4k | 4.7k (4.7k) | 7.0k | 71.6k | 84.6k (84.6k) | 110.5k |
| rococoC10-001000 | 1 | 4677 | 4677 (6947) | 7200 | 37.1k | 323.9k (309.5k) | 562.8k | 13.2m | 13.2m (22.6m) | 32.2m |
| roll3000 | 9 | 1543 | 4724 (4971) | 7200 | 150.3k | 529.1k (551.7k) | 804.1k | 2.3m | 6.9m (7.3m) | 10.8m |
| rout | 10 | 57 | 111 (111) | 186 | 18.8k | 44.6k (44.6k) | 68.9k | 263.8k | 614.3k (614.3k) | 930.2k |

Table 7: Detailed results for the product scoring function on the benchmark test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|----------------|---------|------------------|-------------|------|-----------------|-----------------|--------|----------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| satellites1-25 | 10 | 925 | 1590 (1590) | 2011 | 1.1k | 3.4k (3.4k) | 5.7k | 149.4k | 765.6k (765.6k) | 1.5m |
| set1ch | 10 | 2 | 2 (2) | 3 | 3 | 5 (5) | 7 | 987 | 1.1k (1.1k) | 1.5k |
| seymour | 0 | 7200 | - (7200) | 7200 | 101.9k | - (111.6k) | 120.6k | 7.3m | - (7.9m) | 8.4m |
| sp97ar | 0 | 7200 | - (7200) | 7200 | 118.5k | - (155.3k) | 179.1k | 4.0m | - (4.8m) | 5.4m |
| sp98ic | 10 | 2283 | 4736 (4736) | 6527 | 123.0k | 276.2k (276.2k) | 389.0k | 2.0m | 4.3m (4.3m) | 6.3m |
| sp98ir | 10 | 57 | 71 (71) | 85 | 1.2k | 1.6k (1.6k) | 1.9k | 30.6k | 35.6k (35.6k) | 42.2k |
| stein27 | 10 | 2 | 2 (2) | 2 | 3.7k | 4.2k (4.2k) | 4.5k | 13.7k | 14.4k (14.4k) | 14.9k |
| stein45 | 10 | 32 | 34 (34) | 38 | 48.6k | 51.5k (51.5k) | 54.3k | 260.0k | 267.4k (267.4k) | 272.8k |
| swath | 0 | 7200 | - (7200) | 7200 | 107.7k | - (147.6k) | 194.0k | 1.4m | - (1.8m) | 2.4m |
| tanglegram1 | 7 | 833 | 2464 (3885) | 7200 | 21 | 745 (1.6k) | 3.8k | 118.3k | 596.8k (848.4k) | 1.7m |
| tanglegram2 | 10 | 7 | 10 (10) | 17 | 3 | 4 (4) | 7 | 6.0k | 6.8k (6.8k) | 9.1k |
| timtab1 | 10 | 767 | 1184 (1184) | 1512 | 532.7k | 774.5k (774.5k) | 1.0m | 4.4m | 6.0m (6.0m) | 8.0m |
| timtab2 | 0 | 7200 | - (7200) | 7200 | 3.0m | - (3.3m) | 3.6m | 34.6m | - (36.6m) | 39.1m |
| tr12-30 | 10 | 3006 | 3644 (3644) | 4240 | 685.0k | 780.2k (780.2k) | 900.4k | 3.3m | 3.7m (3.7m) | 4.1m |
| triptim1 | 10 | 178 | 203 (203) | 232 | 1 | 1 (1) | 1 | 30.7k | 33.8k (33.8k) | 35.9k |
| unitcal_7 | 9 | 1576 | 3415 (3794) | 7200 | 11.8k | 32.0k (35.9k) | 71.3k | 204.8k | 499.0k (547.7k) | 1.2m |
| vpm1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 59 | 60 (60) | 60 |
| vpm2 | 10 | 2 | 2 (2) | 2 | 151 | 180 (180) | 207 | 1.6k | 1.8k (1.8k) | 2.1k |
| vpphard | 0 | 7200 | - (7200) | 7200 | 814 | - (8.3k) | 21.9k | 1.3m | - (1.5m) | 1.7m |
| zib54-UUE | 3 | 3911 | 4914 (6514) | 7200 | 378.8k | 413.0k (592.5k) | 734.1k | 17.1m | 20.7m (27.9m) | 32.9m |
| Total | 1225 | 809064 (3437062) | | | 128.2m (1.0b) | | | 1.7b (7.6b) | | |
| Arit. mean | | 660 (2162) | | | 104.7k (660.0k) | | | 1.4m (4.8m) | | |
| Sh. geo. mean | | 87 (251) | | | 1.3k (4.4k) | | | 33.4k (111.4k) | | |

Table 7: Detailed results for the **product** scoring function on the benchmark test set.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------|---------|----------|-----------|------|--------|------------|--------|---------------|---------------|-------|
| | | min | avg | max | min | avg | max | min | avg | max |
| 10teams | 10 | 0 | 1 (1) | 1 | 1 | 1 (1) | 1 | 1.0k | 1.2k (1.2k) | 1.6k |
| 30n20b8 | 10 | 485 | 529 (529) | 571 | 7 | 15 (15) | 34 | 32.6k | 37.1k (37.1k) | 42.9k |
| a1cls1 | 0 | 7200 | - (7200) | 7200 | 277.0k | - (430.3k) | 545.7k | 10.7m | - (13.4m) | 15.1m |
| acc-tight5 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |

Table 8: Detailed results for the **ratio** scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| aflow30a | 10 | 34 | 35 (35) | 39 | 1.7k | 2.5k (2.5k) | 4.4k | 34.2k | 43.9k (43.9k) | 61.4k |
| aflow40b | 10 | 1596 | 2372 (2372) | 2974 | 93.8k | 137.2k (137.2k) | 171.5k | 2.0m | 2.9m (2.9m) | 3.6m |
| air04 | 10 | 27 | 29 (29) | 34 | 6 | 7 (7) | 8 | 11.6k | 13.0k (13.0k) | 14.6k |
| app1-2 | 10 | 619 | 758 (758) | 986 | 29 | 572 (572) | 2.0k | 8.4k | 23.6k (23.6k) | 65.9k |
| arki001 | 2 | 3386 | 3566 (6473) | 7200 | 954.3k | 971.4k (1.7m) | 2.1m | 5.1m | 5.1m (10.2m) | 15.8m |
| ash608gpia-3col | 10 | 80 | 91 (91) | 108 | 5 | 5 (5) | 9 | 8.0k | 10.3k (10.3k) | 19.1k |
| atlanta-ip | 0 | 7200 | - (7200) | 7200 | 5.8k | - (7.3k) | 9.2k | 1.8m | - (2.4m) | 2.7m |
| bab5 | 0 | 7200 | - (7200) | 7200 | 15.3k | - (20.6k) | 36.9k | 710.2k | - (1.0m) | 1.5m |
| beasleyC3 | 0 | 7200 | - (7200) | 7200 | 1.0m | - (1.2m) | 1.5m | 21.9m | - (27.0m) | 34.5m |
| bell5 | 10 | 1 | 1 (1) | 1 | 1.1k | 1.1k (1.1k) | 1.1k | 1.3k | 1.4k (1.4k) | 1.6k |
| biella1 | 10 | 158 | 214 (214) | 269 | 1.9k | 3.0k (3.0k) | 4.9k | 250.9k | 341.5k (341.5k) | 434.8k |
| bienst2 | 10 | 237 | 301 (301) | 391 | 86.0k | 110.2k (110.2k) | 161.3k | 2.2m | 2.9m (2.9m) | 3.9m |
| binkar10_1 | 10 | 290 | 470 (470) | 797 | 90.5k | 140.5k (140.5k) | 204.9k | 675.9k | 1.0m (1.0m) | 1.5m |
| blend2 | 10 | 1 | 2 (2) | 2 | 183 | 221 (221) | 313 | 820 | 975 (975) | 1.1k |
| bley_xl1 | 10 | 1124 | 2010 (2010) | 3012 | 1 | 1 (1) | 1 | 2.6k | 14.4k (14.4k) | 97.8k |
| bnatt350 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| cap6000 | 10 | 8 | 8 (8) | 8 | 1.8k | 1.8k (1.8k) | 1.9k | 2.6k | 2.7k (2.7k) | 2.8k |
| core2536-691 | 10 | 10 | 11 (11) | 12 | 1 | 1 (1) | 1 | 9.1k | 11.5k (11.5k) | 15.5k |
| cov1075 | 0 | 7200 | - (7200) | 7200 | 1.1m | - (1.2m) | 1.4m | 37.8m | - (41.0m) | 44.4m |
| csched010 | 8 | 4327 | 5206 (5605) | 7200 | 503.2k | 616.8k (673.5k) | 947.6k | 25.9m | 32.0m (35.1m) | 48.9m |
| danoint | 1 | 7061 | 7061 (7186) | 7200 | 1.1m | 1.2m (1.2m) | 1.2m | 50.1m | 50.1m (53.8m) | 55.8m |
| dcmulti | 10 | 2 | 2 (2) | 3 | 6 | 9 (9) | 10 | 1.5k | 1.6k (1.6k) | 1.7k |
| dfn-gwin-UUM | 10 | 250 | 265 (265) | 295 | 67.3k | 67.3k (67.3k) | 67.3k | 1.0m | 1.0m (1.0m) | 1.0m |
| disctom | 10 | 0 | 1 (1) | 2 | 0 | 1 (1) | 1 | 0 | 358 (358) | 1.1k |
| dsbmip | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 974 | 1.2k (1.2k) | 1.3k |
| egout | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 37 | 39 (39) | 42 |
| eil33-2 | 10 | 89 | 100 (100) | 125 | 438 | 492 (492) | 518 | 22.5k | 24.3k (24.3k) | 26.3k |
| eilB101 | 10 | 89 | 110 (110) | 126 | 2.5k | 3.7k (3.7k) | 5.6k | 173.7k | 241.1k (241.1k) | 314.8k |
| enigma | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| enlight13 | 10 | 2225 | 2414 (2414) | 2630 | 924.0k | 924.0k (924.0k) | 924.0k | 4.2m | 4.2m (4.2m) | 4.2m |
| enlight14 | 0 | 7200 | - (7200) | 7200 | 3.8m | - (4.2m) | 4.5m | 18.1m | - (19.2m) | 20.2m |
| ex9 | 10 | 2 | 2 (2) | 3 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| fast0507 | 10 | 118 | 127 (127) | 140 | 462 | 585 (585) | 644 | 52.2k | 59.6k (59.6k) | 66.7k |
| fiber | 10 | 4 | 5 (5) | 5 | 1 | 12 (12) | 51 | 735 | 1.1k (1.1k) | 1.6k |
| fixnet6 | 10 | 5 | 5 (5) | 5 | 7 | 9 (9) | 12 | 1.4k | 1.7k (1.7k) | 2.2k |

Table 8: Detailed results for the ratio scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| flugpl | 10 | 0 | 0 (0) | 0 | 229 | 229 (229) | 229 | 187 | 187 (187) | 187 |
| gen | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 139 | 143 (143) | 146 |
| gesa2 | 10 | 4 | 4 (4) | 4 | 2 | 3 (3) | 3 | 1.2k | 1.3k (1.3k) | 1.4k |
| gesa2-o | 10 | 4 | 4 (4) | 4 | 2 | 4 (4) | 5 | 1.1k | 1.4k (1.4k) | 1.5k |
| gesa2_o | 10 | 4 | 4 (4) | 5 | 4 | 5 (5) | 5 | 1.4k | 1.6k (1.6k) | 1.7k |
| gesa3 | 10 | 4 | 4 (4) | 4 | 18 | 20 (20) | 22 | 1.6k | 1.7k (1.7k) | 1.7k |
| gesa3_o | 10 | 4 | 4 (4) | 5 | 7 | 9 (9) | 11 | 1.4k | 1.9k (1.9k) | 2.3k |
| glass4 | 10 | 17 | 36 (36) | 80 | 5.5k | 15.4k (15.4k) | 35.9k | 41.5k | 102.8k (102.8k) | 242.0k |
| gmu-35-40 | 0 | 7200 | - (7200) | 7200 | 8.9m | - (10.6m) | 11.8m | 24.2m | - (29.9m) | 37.1m |
| gt2 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 91 | 91 (91) | 91 |
| harp2 | 0 | 7200 | - (7200) | 7200 | 5.1m | - (5.8m) | 6.5m | 5.6m | - (6.8m) | 8.2m |
| iis-100-0-cov | 10 | 2119 | 2217 (2217) | 2318 | 79.0k | 82.9k (82.9k) | 86.9k | 3.2m | 3.4m (3.4m) | 3.6m |
| iis-bupa-cov | 0 | 7200 | - (7200) | 7200 | 129.9k | - (133.5k) | 137.8k | 7.8m | - (8.0m) | 8.2m |
| iis-pima-cov | 10 | 770 | 804 (804) | 863 | 6.1k | 6.4k (6.4k) | 6.9k | 375.9k | 397.2k (397.2k) | 422.0k |
| khb05250 | 10 | 1 | 1 (1) | 1 | 2 | 7 (7) | 9 | 323 | 554 (554) | 837 |
| l152lav | 10 | 1 | 2 (2) | 2 | 15 | 17 (17) | 19 | 1.5k | 1.8k (1.8k) | 2.0k |
| lectsched-4-obj | 10 | 1 | 1 (1) | 1 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| lseu | 10 | 1 | 1 (1) | 2 | 827 | 857 (857) | 910 | 3.1k | 3.2k (3.2k) | 3.2k |
| m100n500k4r1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 166 | 203 (203) | 260 |
| macrophage | 0 | 7200 | - (7200) | 7200 | 961.8k | - (1.1m) | 1.2m | 14.1m | - (17.8m) | 21.8m |
| manna81 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 3.1k | 3.2k (3.2k) | 3.2k |
| map18 | 10 | 406 | 477 (477) | 552 | 219 | 258 (258) | 289 | 83.6k | 90.9k (90.9k) | 98.9k |
| map20 | 10 | 384 | 445 (445) | 630 | 279 | 300 (300) | 329 | 83.9k | 89.7k (89.7k) | 95.5k |
| markshare1 | 0 | 7200 | - (7200) | 7200 | 22.6m | - (23.5m) | 23.9m | 46.1m | - (48.0m) | 48.8m |
| markshare2 | 0 | 7200 | - (7200) | 7200 | 23.9m | - (24.8m) | 25.6m | 57.2m | - (59.4m) | 61.5m |
| mas74 | 10 | 1518 | 1566 (1566) | 1672 | 2.9m | 2.9m (2.9m) | 2.9m | 11.3m | 11.3m (11.3m) | 11.3m |
| mas76 | 10 | 160 | 165 (165) | 178 | 403.2k | 403.2k (403.2k) | 403.2k | 1.3m | 1.3m (1.3m) | 1.3m |
| mcsched | 10 | 174 | 266 (266) | 409 | 10.2k | 18.3k (18.3k) | 29.5k | 378.0k | 657.4k (657.4k) | 1.0m |
| mik-250-1-100-1 | 10 | 2025 | 2166 (2166) | 2290 | 901.6k | 971.8k (971.8k) | 1.1m | 3.0m | 3.3m (3.3m) | 3.5m |
| mine-166-5 | 10 | 28 | 31 (31) | 33 | 186 | 340 (340) | 572 | 3.9k | 4.7k (4.7k) | 5.7k |
| mine-90-10 | 10 | 327 | 1129 (1129) | 2565 | 32.6k | 112.7k (112.7k) | 203.9k | 209.1k | 643.3k (643.3k) | 1.1m |
| misc03 | 10 | 1 | 2 (2) | 2 | 77 | 128 (128) | 225 | 2.2k | 2.9k (2.9k) | 3.9k |
| misc06 | 10 | 1 | 1 (1) | 2 | 3 | 7 (7) | 10 | 1.3k | 1.8k (1.8k) | 2.3k |
| misc07 | 10 | 45 | 56 (56) | 69 | 24.2k | 30.6k (30.6k) | 35.5k | 177.9k | 236.3k (236.3k) | 273.8k |
| mitre | 10 | 44 | 49 (49) | 58 | 1 | 1 (1) | 1 | 1.3k | 1.4k (1.4k) | 1.4k |

Table 8: Detailed results for the ratio scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|--------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| mod008 | 10 | 5 | 5 (5) | 5 | 12 | 13 (13) | 15 | 580 | 601 (601) | 607 |
| mod010 | 10 | 1 | 1 (1) | 1 | 2 | 3 (3) | 4 | 905 | 1.0k (1.0k) | 1.3k |
| mod011 | 10 | 61 | 72 (72) | 91 | 549 | 784 (784) | 997 | 47.6k | 62.4k (62.4k) | 75.8k |
| modglob | 10 | 1 | 1 (1) | 2 | 10 | 21 (21) | 27 | 694 | 769 (769) | 1.0k |
| momentum1 | 0 | 7200 | - (7200) | 7200 | 20.2k | - (23.1k) | 26.8k | 1.2m | - (1.6m) | 2.1m |
| momentum2 | 2 | 5734 | 6457 (7051) | 7200 | 16.0k | 19.8k (24.7k) | 30.6k | 469.3k | 516.5k (723.6k) | 1.1m |
| msc98-ip | 10 | 969 | 1760 (1760) | 4134 | 2.8k | 6.3k (6.3k) | 18.1k | 743.4k | 1.6m (1.6m) | 4.1m |
| mspp16 | 0 | 7200 | - (7200) | 7200 | 0 | - (0) | 0 | 0 | - (0) | 0 |
| mzzv11 | 10 | 631 | 724 (724) | 778 | 12 | 206 (206) | 966 | 48.6k | 58.3k (58.3k) | 76.3k |
| mzzv42z | 10 | 497 | 557 (557) | 628 | 2 | 12 (12) | 20 | 32.9k | 40.4k (40.4k) | 46.5k |
| n3div36 | 0 | 7200 | - (7200) | 7200 | 99.5k | - (144.6k) | 181.8k | 777.6k | - (1.2m) | 1.6m |
| n3seq24 | 0 | 7200 | - (7200) | 7200 | 77.7k | - (110.4k) | 184.5k | 1.3m | - (1.9m) | 2.4m |
| n4-3 | 10 | 868 | 1260 (1260) | 1780 | 30.4k | 46.0k (46.0k) | 64.3k | 1.2m | 1.8m (1.8m) | 2.3m |
| neos-1109824 | 10 | 131 | 205 (205) | 292 | 4.7k | 8.5k (8.5k) | 15.0k | 54.2k | 86.0k (86.0k) | 141.0k |
| neos-1337307 | 0 | 7200 | - (7200) | 7200 | 233.7k | - (265.2k) | 305.7k | 3.1m | - (3.6m) | 4.4m |
| neos-1396125 | 10 | 276 | 484 (484) | 911 | 38.4k | 71.0k (71.0k) | 141.8k | 1.1m | 1.7m (1.7m) | 3.1m |
| neos-1601936 | 10 | 7 | 7 (7) | 8 | 1 | 1 (1) | 1 | 10.4k | 11.0k (11.0k) | 11.4k |
| neos-476283 | 10 | 318 | 380 (380) | 569 | 75 | 129 (129) | 201 | 8.0k | 11.3k (11.3k) | 16.4k |
| neos-686190 | 10 | 101 | 161 (161) | 189 | 1.9k | 2.2k (2.2k) | 2.6k | 36.1k | 41.2k (41.2k) | 45.1k |
| neos-849702 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| neos-916792 | 0 | 7200 | - (7200) | 7200 | 142.1k | - (180.5k) | 206.6k | 1.7m | - (2.2m) | 2.5m |
| neos-934278 | 0 | 7200 | - (7200) | 7200 | 7.3k | - (9.5k) | 13.9k | 5.9m | - (6.6m) | 7.8m |
| neos13 | 10 | 241 | 262 (262) | 298 | 10 | 12 (12) | 12 | 3.0k | 3.0k (3.0k) | 3.0k |
| neos18 | 10 | 74 | 97 (97) | 120 | 6.1k | 8.4k (8.4k) | 11.6k | 166.7k | 228.9k (228.9k) | 299.4k |
| net12 | 10 | 1608 | 2077 (2077) | 2447 | 2.1k | 3.1k (3.1k) | 4.0k | 1.5m | 2.1m (2.1m) | 2.6m |
| netdiversion | 10 | 1060 | 1413 (1413) | 2078 | 5 | 12 (12) | 49 | 30.0k | 50.1k (50.1k) | 138.1k |
| newdano | 4 | 4309 | 5757 (6623) | 7200 | 1.1m | 1.2m (1.5m) | 2.0m | 52.7m | 66.2m (72.2m) | 86.7m |
| noswot | 10 | 327 | 880 (880) | 3601 | 436.9k | 1.1m (1.1m) | 4.3m | 1.4m | 2.8m (2.8m) | 8.6m |
| ns1208400 | 10 | 315 | 456 (456) | 650 | 1 | 210 (210) | 702 | 142.1k | 337.6k (337.6k) | 646.9k |
| ns1688347 | 10 | 58 | 82 (82) | 115 | 6 | 105 (105) | 322 | 5.8k | 24.3k (24.3k) | 44.4k |
| ns1758913 | 7 | 2224 | 4119 (5043) | 7200 | 1 | 1 (1) | 1 | 29.3k | 50.0k (55.8k) | 88.0k |
| ns1766074 | 10 | 5197 | 5611 (5611) | 6066 | 895.4k | 907.9k (907.9k) | 942.4k | 3.5m | 3.6m (3.6m) | 3.8m |
| ns1830653 | 10 | 367 | 484 (484) | 791 | 14.9k | 19.0k (19.0k) | 33.8k | 933.0k | 1.3m (1.3m) | 2.3m |
| nsrand-ipx | 10 | 808 | 3768 (3768) | 7097 | 59.9k | 464.5k (464.5k) | 830.8k | 508.6k | 3.1m (3.1m) | 6.0m |
| nw04 | 10 | 1989 | 2283 (2283) | 3039 | 7 | 10 (10) | 12 | 1.3k | 1.6k (1.6k) | 1.9k |

Table 8: Detailed results for the ratio scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| opm2-z7-s2 | 10 | 705 | 883 (883) | 1092 | 537 | 976 (976) | 1.3k | 35.3k | 68.7k (68.7k) | 102.0k |
| opt1217 | 5 | 2 | 2 (3601) | 7200 | 1 | 1 (1.6m) | 3.6m | 747 | 798 (8.7m) | 22.6m |
| p0033 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 57 | 57 (57) | 57 |
| p0201 | 10 | 2 | 3 (3) | 3 | 4 | 10 (10) | 33 | 1.1k | 1.4k (1.4k) | 2.5k |
| p0282 | 10 | 1 | 1 (1) | 2 | 2 | 3 (3) | 3 | 459 | 649 (649) | 890 |
| p0548 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 331 | 331 (331) | 331 |
| p2756 | 10 | 4 | 4 (4) | 4 | 10 | 10 (10) | 10 | 538 | 547 (547) | 555 |
| pg5_34 | 10 | 1243 | 1304 (1304) | 1365 | 107.6k | 107.6k (107.6k) | 107.6k | 2.0m | 2.0m (2.0m) | 2.0m |
| pigeon-10 | 0 | 7200 | - (7200) | 7200 | 832.7k | - (1.0m) | 1.2m | 4.3m | - (5.8m) | 7.1m |
| pk1 | 10 | 137 | 139 (139) | 143 | 369.8k | 369.8k (369.8k) | 369.8k | 2.1m | 2.1m (2.1m) | 2.1m |
| pp08a | 10 | 2 | 2 (2) | 3 | 111 | 240 (240) | 402 | 2.1k | 3.3k (3.3k) | 5.7k |
| pp08aCUTS | 10 | 2 | 2 (2) | 3 | 75 | 155 (155) | 255 | 1.8k | 2.5k (2.5k) | 3.7k |
| protfold | 0 | 7200 | - (7200) | 7200 | 21.6k | - (35.7k) | 43.9k | 15.7m | - (20.8m) | 25.3m |
| pw-myciel4 | 7 | 3923 | 5559 (6051) | 7200 | 447.2k | 583.7k (607.0k) | 780.5k | 10.8m | 15.5m (16.9m) | 21.2m |
| qiu | 10 | 79 | 106 (106) | 134 | 10.2k | 12.7k (12.7k) | 15.7k | 315.4k | 416.9k (416.9k) | 522.3k |
| qnet1 | 10 | 9 | 10 (10) | 12 | 3 | 4 (4) | 6 | 3.7k | 4.0k (4.0k) | 4.5k |
| qnet1_o | 10 | 5 | 5 (5) | 6 | 3 | 4 (4) | 6 | 2.5k | 2.8k (2.8k) | 3.3k |
| rail507 | 10 | 113 | 140 (140) | 170 | 446 | 596 (596) | 678 | 49.1k | 60.4k (60.4k) | 67.9k |
| ran16x16 | 10 | 405 | 444 (444) | 522 | 245.1k | 286.9k (286.9k) | 332.6k | 2.1m | 2.4m (2.4m) | 2.9m |
| rd-rplusc-21 | 1 | 4462 | 4462 (6926) | 7200 | 38.4k | 135.5k (69.2k) | 135.5k | 248.3k | 584.3k (750.0k) | 2.0m |
| reblock67 | 10 | 174 | 245 (245) | 304 | 42.7k | 48.7k (48.7k) | 55.4k | 451.0k | 517.6k (517.6k) | 571.4k |
| rentacar | 10 | 5 | 6 (6) | 6 | 2 | 2 (2) | 2 | 2.8k | 3.2k (3.2k) | 3.8k |
| rgn | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 2 | 327 | 392 (392) | 556 |
| rmatr100-p10 | 10 | 106 | 112 (112) | 124 | 789 | 799 (799) | 805 | 64.1k | 64.2k (64.2k) | 64.3k |
| rmatr100-p5 | 10 | 128 | 134 (134) | 141 | 383 | 383 (383) | 383 | 66.8k | 66.8k (66.8k) | 66.8k |
| rmine6 | 4 | 3898 | 4981 (6312) | 7200 | 398.0k | 519.3k (731.9k) | 944.1k | 2.5m | 3.5m (5.1m) | 6.8m |
| rocII-4-11 | 10 | 631 | 1019 (1019) | 1566 | 13.1k | 27.6k (27.6k) | 50.1k | 162.4k | 303.4k (303.4k) | 523.0k |
| rococoC10-001000 | 2 | 3864 | 4825 (6725) | 7200 | 121.8k | 354.5k (343.0k) | 531.1k | 10.2m | 12.1m (23.3m) | 37.9m |
| roll3000 | 7 | 2335 | 4709 (5456) | 7200 | 255.7k | 558.7k (639.5k) | 868.2k | 3.6m | 6.8m (8.0m) | 12.5m |
| rout | 10 | 30 | 81 (81) | 160 | 7.9k | 32.6k (32.6k) | 66.1k | 167.6k | 484.9k (484.9k) | 948.0k |
| satellites1-25 | 10 | 915 | 1636 (1636) | 1928 | 1.1k | 3.4k (3.4k) | 7.2k | 124.0k | 855.6k (855.6k) | 1.6m |
| set1ch | 10 | 2 | 2 (2) | 3 | 3 | 5 (5) | 7 | 987 | 1.1k (1.1k) | 1.5k |
| seymour | 0 | 7200 | - (7200) | 7200 | 107.9k | - (112.7k) | 130.3k | 7.4m | - (8.0m) | 8.6m |
| sp97ar | 0 | 7200 | - (7200) | 7200 | 130.1k | - (153.3k) | 174.3k | 4.5m | - (4.9m) | 5.2m |
| sp98ic | 10 | 1287 | 2309 (2309) | 3184 | 61.1k | 117.2k (117.2k) | 168.3k | 941.4k | 1.9m (1.9m) | 2.8m |

Table 8: Detailed results for the ratio scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|---------------|---------|------------------|-------------|------|-----------------|-----------------|--------|----------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| sp98ir | 10 | 60 | 69 (69) | 75 | 1.5k | 1.8k (1.8k) | 2.1k | 32.2k | 41.8k (41.8k) | 46.7k |
| stein27 | 10 | 2 | 2 (2) | 2 | 3.8k | 4.1k (4.1k) | 4.4k | 13.4k | 14.2k (14.2k) | 14.7k |
| stein45 | 10 | 31 | 32 (32) | 34 | 48.3k | 50.3k (50.3k) | 53.0k | 254.3k | 262.4k (262.4k) | 271.7k |
| swath | 0 | 7200 | - (7200) | 7200 | 104.6k | - (160.7k) | 214.4k | 1.5m | - (2.0m) | 2.6m |
| tanglegram1 | 6 | 695 | 2543 (4406) | 7200 | 21 | 573 (1.2k) | 3.1k | 118.3k | 500.2k (864.5k) | 1.6m |
| tanglegram2 | 10 | 7 | 10 (10) | 17 | 3 | 4 (4) | 7 | 6.0k | 6.8k (6.8k) | 9.1k |
| timtab1 | 10 | 856 | 1235 (1235) | 1607 | 592.7k | 803.6k (803.6k) | 993.3k | 4.6m | 6.3m (6.3m) | 7.9m |
| timtab2 | 0 | 7200 | - (7200) | 7200 | 3.2m | - (3.5m) | 3.8m | 31.6m | - (34.8m) | 39.2m |
| tr12-30 | 10 | 3100 | 3707 (3707) | 4222 | 714.9k | 846.1k (846.1k) | 969.6k | 3.3m | 4.0m (4.0m) | 4.6m |
| triptim1 | 10 | 169 | 188 (188) | 232 | 1 | 1 (1) | 1 | 30.7k | 33.8k (33.8k) | 35.9k |
| unitcal_7 | 10 | 1465 | 2796 (2796) | 4669 | 10.3k | 27.4k (27.4k) | 58.0k | 195.9k | 455.2k (455.2k) | 987.2k |
| vpm1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 59 | 60 (60) | 60 |
| vpm2 | 10 | 2 | 2 (2) | 2 | 179 | 190 (190) | 205 | 1.7k | 1.8k (1.8k) | 2.0k |
| vpphard | 0 | 7200 | - (7200) | 7200 | 838 | - (3.5k) | 7.2k | 1.2m | - (1.6m) | 1.7m |
| zib54-UUE | 3 | 5441 | 5899 (6810) | 7200 | 437.3k | 487.7k (608.2k) | 745.6k | 23.7m | 25.3m (29.3m) | 32.4m |
| Total | 1249 | 849097 (3304297) | | | 138.5m (996.1m) | | | 1.6b (7.1b) | | |
| Arit. mean | | 680 (2078) | | | 110.9k (626.5k) | | | 1.3m (4.5m) | | |
| Sh. geo. mean | | 92 (245) | | | 1.4k (4.4k) | | | 36.2k (108.8k) | | |

Table 8: Detailed results for the `ratio` scoring function on the benchmark test set.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|---------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| 10teams | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 1.0k | 1.2k (1.2k) | 1.6k |
| 30n20b8 | 10 | 479 | 526 (526) | 605 | 7 | 14 (14) | 36 | 32.6k | 37.5k (37.5k) | 41.7k |
| alc1s1 | 0 | 7200 | - (7200) | 7200 | 308.8k | - (420.7k) | 510.0k | 11.5m | - (13.8m) | 15.4m |
| acc-tight5 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| afflow30a | 10 | 31 | 35 (35) | 38 | 1.7k | 2.2k (2.2k) | 3.3k | 32.9k | 41.2k (41.2k) | 52.5k |
| afflow40b | 10 | 1533 | 2437 (2437) | 3328 | 79.0k | 133.3k (133.3k) | 212.9k | 2.0m | 3.2m (3.2m) | 4.9m |
| air04 | 10 | 28 | 31 (31) | 35 | 6 | 7 (7) | 8 | 11.6k | 13.0k (13.0k) | 14.6k |
| app1-2 | 10 | 601 | 754 (754) | 1061 | 29 | 645 (645) | 3.1k | 8.8k | 27.5k (27.5k) | 105.4k |
| arki001 | 3 | 2992 | 5105 (6571) | 7200 | 747.9k | 1.3m (1.5m) | 1.9m | 4.4m | 6.3m (10.4m) | 15.5m |

Table 9: Detailed results for the `svts` scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| ash608gpia-3col | 10 | 79 | 88 (88) | 103 | 5 | 5 (5) | 9 | 8.0k | 10.3k (10.3k) | 19.1k |
| atlanta-ip | 0 | 7200 | - (7200) | 7200 | 4.2k | - (7.0k) | 8.5k | 1.6m | - (2.3m) | 2.8m |
| bab5 | 0 | 7200 | - (7200) | 7200 | 11.6k | - (17.6k) | 24.6k | 658.3k | - (990.6k) | 1.3m |
| beasleyC3 | 0 | 7200 | - (7200) | 7200 | 763.6k | - (966.1k) | 1.3m | 25.9m | - (29.9m) | 33.1m |
| bell5 | 10 | 1 | 1 (1) | 1 | 1.1k | 1.1k (1.1k) | 1.1k | 1.3k | 1.4k (1.4k) | 1.6k |
| biella1 | 10 | 150 | 215 (215) | 281 | 1.5k | 2.8k (2.8k) | 4.6k | 230.7k | 354.3k (354.3k) | 492.2k |
| bienst2 | 10 | 232 | 320 (320) | 436 | 84.8k | 123.1k (123.1k) | 174.1k | 2.2m | 3.3m (3.3m) | 4.5m |
| binkar10_1 | 10 | 248 | 437 (437) | 729 | 76.9k | 133.1k (133.1k) | 216.8k | 576.7k | 1.0m (1.0m) | 1.7m |
| blend2 | 10 | 1 | 2 (2) | 2 | 166 | 225 (225) | 346 | 796 | 993 (993) | 1.3k |
| bley_xl1 | 10 | 1102 | 1916 (1916) | 2738 | 1 | 1 (1) | 1 | 2.6k | 14.4k (14.4k) | 97.8k |
| bnatt350 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| cap6000 | 10 | 8 | 8 (8) | 9 | 1.8k | 1.9k (1.9k) | 1.9k | 2.6k | 2.7k (2.7k) | 2.8k |
| core2536-691 | 10 | 10 | 11 (11) | 13 | 1 | 1 (1) | 1 | 9.1k | 11.5k (11.5k) | 15.5k |
| cov1075 | 0 | 7200 | - (7200) | 7200 | 1.1m | - (1.2m) | 1.3m | 36.5m | - (40.5m) | 44.1m |
| csched010 | 9 | 4332 | 5550 (5715) | 7200 | 483.4k | 616.0k (632.5k) | 780.7k | 27.1m | 34.7m (35.7m) | 44.1m |
| danoint | 1 | 7182 | 7182 (7198) | 7200 | 1.1m | 1.2m (1.2m) | 1.2m | 50.4m | 50.4m (54.1m) | 58.0m |
| dcmulti | 10 | 2 | 2 (2) | 2 | 6 | 9 (9) | 10 | 1.5k | 1.6k (1.6k) | 1.7k |
| dfn-gwin-UUM | 10 | 245 | 269 (269) | 304 | 60.2k | 60.2k (60.2k) | 60.2k | 999.8k | 999.8k (999.8k) | 999.8k |
| disctom | 10 | 0 | 1 (1) | 2 | 0 | 1 (1) | 1 | 0 | 358 (358) | 1.1k |
| dsbmip | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 974 | 1.2k (1.2k) | 1.3k |
| egout | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 37 | 39 (39) | 42 |
| eil33-2 | 10 | 98 | 114 (114) | 142 | 444 | 482 (482) | 532 | 23.7k | 26.4k (26.4k) | 28.9k |
| eilB101 | 10 | 96 | 118 (118) | 135 | 2.0k | 3.4k (3.4k) | 5.7k | 203.7k | 284.9k (284.9k) | 360.5k |
| enigma | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| enlight13 | 0 | 7200 | - (7200) | 7200 | 3.8m | - (4.0m) | 4.3m | 18.7m | - (19.9m) | 20.7m |
| enlight14 | 0 | 7200 | - (7200) | 7200 | 4.0m | - (4.3m) | 4.6m | 18.8m | - (19.6m) | 20.6m |
| ex9 | 10 | 2 | 2 (2) | 3 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| fast0507 | 10 | 113 | 128 (128) | 143 | 504 | 602 (602) | 760 | 56.5k | 62.5k (62.5k) | 69.3k |
| fiber | 10 | 4 | 5 (5) | 5 | 1 | 13 (13) | 57 | 735 | 1.1k (1.1k) | 1.7k |
| fixnet6 | 10 | 5 | 5 (5) | 5 | 7 | 10 (10) | 13 | 1.4k | 1.7k (1.7k) | 2.2k |
| flugpl | 10 | 0 | 0 (0) | 0 | 240 | 240 (240) | 240 | 197 | 197 (197) | 197 |
| gen | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 139 | 143 (143) | 146 |
| gesa2 | 10 | 4 | 4 (4) | 4 | 2 | 3 (3) | 3 | 1.2k | 1.3k (1.3k) | 1.4k |
| gesa2-o | 10 | 4 | 4 (4) | 4 | 2 | 4 (4) | 5 | 1.1k | 1.4k (1.4k) | 1.5k |
| gesa2_o | 10 | 4 | 4 (4) | 4 | 4 | 5 (5) | 5 | 1.4k | 1.6k (1.6k) | 1.7k |

Table 9: Detailed results for the svts scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| gesa3 | 10 | 4 | 4 (4) | 4 | 16 | 20 (20) | 20 | 1.6k | 1.7k (1.7k) | 1.7k |
| gesa3_o | 10 | 4 | 4 (4) | 5 | 7 | 9 (9) | 11 | 1.4k | 1.9k (1.9k) | 2.3k |
| glass4 | 10 | 15 | 30 (30) | 53 | 5.5k | 11.8k (11.8k) | 22.5k | 37.6k | 74.4k (74.4k) | 147.2k |
| gmu-35-40 | 0 | 7200 | - (7200) | 7200 | 9.4m | - (10.6m) | 12.3m | 25.3m | - (31.3m) | 34.3m |
| gt2 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 91 | 91 (91) | 91 |
| harp2 | 0 | 7200 | - (7200) | 7200 | 4.6m | - (5.3m) | 6.0m | 5.4m | - (6.0m) | 6.6m |
| iis-100-0-cov | 10 | 2081 | 2229 (2229) | 2377 | 77.9k | 83.1k (83.1k) | 87.4k | 3.2m | 3.4m (3.4m) | 3.5m |
| iis-bupa-cov | 0 | 7200 | - (7200) | 7200 | 128.4k | - (132.7k) | 137.1k | 7.8m | - (7.9m) | 8.1m |
| iis-pima-cov | 10 | 722 | 802 (802) | 916 | 5.9k | 6.4k (6.4k) | 7.1k | 373.6k | 399.9k (399.9k) | 436.0k |
| khb05250 | 10 | 1 | 1 (1) | 1 | 2 | 7 (7) | 9 | 323 | 554 (554) | 837 |
| l152lav | 10 | 1 | 2 (2) | 2 | 15 | 16 (16) | 17 | 1.5k | 1.8k (1.8k) | 2.0k |
| lectsched-4-obj | 10 | 1 | 1 (1) | 1 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| lseu | 10 | 2 | 2 (2) | 2 | 1.1k | 1.1k (1.1k) | 1.2k | 3.7k | 3.9k (3.9k) | 4.0k |
| m100n500k4r1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 166 | 203 (203) | 260 |
| macrophage | 0 | 7200 | - (7200) | 7200 | 907.9k | - (1.1m) | 1.2m | 15.4m | - (17.3m) | 19.0m |
| manna81 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 3.1k | 3.2k (3.2k) | 3.2k |
| map18 | 10 | 411 | 484 (484) | 595 | 223 | 259 (259) | 287 | 84.5k | 92.3k (92.3k) | 103.0k |
| map20 | 10 | 379 | 424 (424) | 511 | 265 | 295 (295) | 327 | 84.5k | 90.1k (90.1k) | 95.9k |
| markshare1 | 0 | 7200 | - (7200) | 7200 | 22.6m | - (23.1m) | 23.4m | 46.1m | - (47.2m) | 47.9m |
| markshare2 | 0 | 7200 | - (7200) | 7200 | 23.7m | - (24.2m) | 24.6m | 56.7m | - (58.1m) | 58.9m |
| mas74 | 10 | 1599 | 1665 (1665) | 1835 | 3.0m | 3.0m (3.0m) | 3.0m | 11.6m | 11.6m (11.6m) | 11.6m |
| mas76 | 10 | 193 | 196 (196) | 205 | 475.5k | 475.5k (475.5k) | 475.5k | 1.6m | 1.6m (1.6m) | 1.6m |
| mcsched | 10 | 182 | 261 (261) | 348 | 11.5k | 17.3k (17.3k) | 25.3k | 414.7k | 642.2k (642.2k) | 961.7k |
| mik-250-1-100-1 | 10 | 1887 | 2374 (2374) | 2948 | 870.0k | 1.1m (1.1m) | 1.4m | 2.7m | 3.4m (3.4m) | 4.1m |
| mine-166-5 | 10 | 29 | 32 (32) | 37 | 237 | 402 (402) | 801 | 4.0k | 4.9k (4.9k) | 5.9k |
| mine-90-10 | 10 | 386 | 1387 (1387) | 2572 | 36.4k | 138.0k (138.0k) | 303.1k | 241.2k | 773.0k (773.0k) | 1.4m |
| misc03 | 10 | 2 | 2 (2) | 2 | 63 | 121 (121) | 207 | 2.0k | 2.7k (2.7k) | 3.7k |
| misc06 | 10 | 1 | 1 (1) | 2 | 3 | 7 (7) | 10 | 1.3k | 1.8k (1.8k) | 2.3k |
| misc07 | 10 | 40 | 48 (48) | 55 | 16.9k | 23.5k (23.5k) | 27.9k | 129.5k | 186.8k (186.8k) | 227.8k |
| mitre | 10 | 43 | 49 (49) | 61 | 1 | 1 (1) | 1 | 1.3k | 1.4k (1.4k) | 1.4k |
| mod008 | 10 | 5 | 5 (5) | 5 | 11 | 12 (12) | 12 | 544 | 594 (594) | 607 |
| mod010 | 10 | 1 | 1 (1) | 1 | 2 | 3 (3) | 4 | 905 | 1.0k (1.0k) | 1.3k |
| mod011 | 10 | 63 | 72 (72) | 83 | 611 | 852 (852) | 1.3k | 51.1k | 67.0k (67.0k) | 90.2k |
| modglob | 10 | 1 | 1 (1) | 2 | 10 | 20 (20) | 27 | 675 | 767 (767) | 1.0k |
| momentum1 | 0 | 7200 | - (7200) | 7200 | 13.9k | - (22.6k) | 26.6k | 1.1m | - (1.5m) | 1.9m |

Table 9: Detailed results for the svts scoring function on the benchmark test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|--------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| momentum2 | 2 | 5089 | 5299 (6820) | 7200 | 15.4k | 15.4k (22.5k) | 25.9k | 469.9k | 650.1k (783.4k) | 1.1m |
| msc98-ip | 10 | 677 | 1503 (1503) | 2348 | 2.2k | 4.8k (4.8k) | 12.1k | 542.1k | 1.3m (1.3m) | 2.3m |
| mspp16 | 0 | 7200 | - (7200) | 7200 | 0 | - (0) | 0 | 0 | - (0) | 0 |
| mzzv11 | 10 | 666 | 730 (730) | 809 | 12 | 213 (213) | 991 | 48.5k | 58.3k (58.3k) | 78.4k |
| mzzv42z | 10 | 502 | 535 (535) | 600 | 2 | 12 (12) | 20 | 32.9k | 40.6k (40.6k) | 46.5k |
| n3div36 | 0 | 7200 | - (7200) | 7200 | 105.3k | - (149.2k) | 180.9k | 877.1k | - (1.3m) | 1.7m |
| n3seq24 | 0 | 7200 | - (7200) | 7200 | 79.3k | - (109.1k) | 135.4k | 1.1m | - (1.9m) | 2.8m |
| n4-3 | 10 | 811 | 1246 (1246) | 1934 | 28.8k | 44.1k (44.1k) | 65.6k | 1.2m | 1.7m (1.7m) | 2.4m |
| neos-1109824 | 10 | 176 | 367 (367) | 619 | 7.9k | 19.7k (19.7k) | 34.6k | 92.4k | 178.8k (178.8k) | 303.9k |
| neos-1337307 | 0 | 7200 | - (7200) | 7200 | 239.3k | - (258.8k) | 280.5k | 3.6m | - (3.8m) | 4.0m |
| neos-1396125 | 10 | 312 | 396 (396) | 597 | 33.5k | 56.9k (56.9k) | 103.0k | 999.3k | 1.5m (1.5m) | 2.0m |
| neos-1601936 | 10 | 7 | 7 (7) | 8 | 1 | 1 (1) | 1 | 10.4k | 11.0k (11.0k) | 11.4k |
| neos-476283 | 10 | 357 | 432 (432) | 580 | 91 | 123 (123) | 168 | 8.0k | 11.3k (11.3k) | 16.3k |
| neos-686190 | 10 | 117 | 161 (161) | 261 | 1.5k | 2.1k (2.1k) | 5.0k | 35.5k | 42.8k (42.8k) | 79.9k |
| neos-849702 | 10 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 | 0 | 0 (0) | 0 |
| neos-916792 | 0 | 7200 | - (7200) | 7200 | 172.2k | - (203.4k) | 229.1k | 2.0m | - (2.4m) | 2.8m |
| neos-934278 | 0 | 7200 | - (7200) | 7200 | 6.1k | - (9.3k) | 13.0k | 5.5m | - (6.5m) | 7.9m |
| neos13 | 10 | 255 | 276 (276) | 309 | 10 | 12 (12) | 12 | 3.0k | 3.0k (3.0k) | 3.0k |
| neos18 | 10 | 64 | 86 (86) | 145 | 4.1k | 6.1k (6.1k) | 8.8k | 128.7k | 196.2k (196.2k) | 281.8k |
| net12 | 10 | 1061 | 2178 (2178) | 2975 | 1.8k | 3.1k (3.1k) | 4.7k | 1.0m | 2.2m (2.2m) | 3.2m |
| netdiversion | 10 | 896 | 1395 (1395) | 2319 | 5 | 14 (14) | 64 | 30.5k | 53.7k (53.7k) | 153.2k |
| newdano | 5 | 4277 | 5616 (6408) | 7200 | 1.1m | 1.4m (1.4m) | 1.8m | 52.3m | 67.7m (67.8m) | 81.2m |
| noswot | 10 | 332 | 817 (817) | 2766 | 413.9k | 922.0k (922.0k) | 2.8m | 1.4m | 2.7m (2.7m) | 7.2m |
| ns1208400 | 10 | 288 | 466 (466) | 653 | 1 | 232 (232) | 830 | 142.1k | 353.5k (353.5k) | 676.5k |
| ns1688347 | 10 | 58 | 84 (84) | 116 | 6 | 85 (85) | 328 | 5.8k | 23.5k (23.5k) | 36.8k |
| ns1758913 | 6 | 2203 | 3830 (5178) | 7200 | 1 | 1 (1) | 1 | 29.3k | 44.1k (53.6k) | 75.1k |
| ns1766074 | 10 | 5159 | 5666 (5666) | 6568 | 895.4k | 907.9k (907.9k) | 942.4k | 3.5m | 3.6m (3.6m) | 3.8m |
| ns1830653 | 10 | 409 | 448 (448) | 473 | 15.0k | 17.1k (17.1k) | 19.5k | 1.0m | 1.2m (1.2m) | 1.2m |
| nsrand-ipx | 5 | 805 | 3972 (5586) | 7200 | 67.5k | 497.6k (764.0k) | 1.3m | 615.8k | 3.0m (4.5m) | 7.3m |
| nw04 | 10 | 2024 | 2529 (2529) | 3112 | 7 | 10 (10) | 12 | 1.3k | 1.6k (1.6k) | 1.9k |
| opm2-z7-s2 | 10 | 590 | 787 (787) | 1075 | 533 | 768 (768) | 1.0k | 36.0k | 53.4k (53.4k) | 61.5k |
| opt1217 | 5 | 2 | 3 (3601) | 7200 | 1 | 1 (1.6m) | 3.7m | 747 | 798 (8.2m) | 22.7m |
| p0033 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 57 | 57 (57) | 57 |
| p0201 | 10 | 2 | 3 (3) | 4 | 4 | 10 (10) | 33 | 1.1k | 1.5k (1.5k) | 2.5k |
| p0282 | 10 | 1 | 2 (2) | 2 | 2 | 3 (3) | 3 | 459 | 649 (649) | 890 |

Table 9: Detailed results for the svts scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|----------|-------------|------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| p0548 | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 1 | 331 | 331 (331) | 331 |
| p2756 | 10 | 4 | 4 (4) | 5 | 10 | 10 (10) | 10 | 538 | 547 (547) | 555 |
| pg5_34 | 10 | 1243 | 1343 (1343) | 1652 | 109.5k | 109.5k (109.5k) | 109.5k | 2.1m | 2.1m (2.1m) | 2.1m |
| pigeon-10 | 0 | 7200 | - (7200) | 7200 | 854.8k | - (995.6k) | 1.2m | 3.7m | - (5.6m) | 7.1m |
| pk1 | 10 | 129 | 134 (134) | 142 | 344.3k | 344.3k (344.3k) | 344.3k | 2.0m | 2.0m (2.0m) | 2.0m |
| pp08a | 10 | 2 | 2 (2) | 3 | 117 | 240 (240) | 400 | 2.0k | 3.4k (3.4k) | 5.8k |
| pp08aCUTS | 10 | 2 | 2 (2) | 3 | 89 | 161 (161) | 283 | 1.8k | 2.5k (2.5k) | 3.8k |
| protfold | 0 | 7200 | - (7200) | 7200 | 18.9k | - (25.0k) | 30.6k | 15.3m | - (20.8m) | 24.4m |
| pw-myciel4 | 4 | 5756 | 6617 (6967) | 7200 | 480.0k | 637.1k (669.7k) | 809.1k | 15.7m | 18.0m (20.0m) | 23.0m |
| qiu | 10 | 82 | 105 (105) | 120 | 10.0k | 12.3k (12.3k) | 13.7k | 319.1k | 406.7k (406.7k) | 482.1k |
| qnet1 | 10 | 9 | 10 (10) | 12 | 3 | 4 (4) | 6 | 3.7k | 4.0k (4.0k) | 4.5k |
| qnet1_o | 10 | 5 | 5 (5) | 6 | 3 | 4 (4) | 6 | 2.5k | 2.8k (2.8k) | 3.3k |
| rail507 | 10 | 119 | 141 (141) | 184 | 568 | 601 (601) | 636 | 59.8k | 62.8k (62.8k) | 66.3k |
| ran16x16 | 10 | 399 | 442 (442) | 504 | 247.1k | 273.3k (273.3k) | 308.5k | 2.3m | 2.5m (2.5m) | 2.9m |
| rd-rplusc-21 | 0 | 7200 | - (7200) | 7200 | 48.9k | - (83.1k) | 172.9k | 320.6k | - (949.1k) | 2.3m |
| reblock67 | 10 | 162 | 232 (232) | 276 | 38.6k | 45.1k (45.1k) | 52.0k | 394.4k | 484.5k (484.5k) | 553.7k |
| rentacar | 10 | 5 | 6 (6) | 6 | 2 | 2 (2) | 2 | 2.8k | 3.2k (3.2k) | 3.8k |
| rgn | 10 | 1 | 1 (1) | 1 | 1 | 1 (1) | 2 | 327 | 392 (392) | 556 |
| rmatr100-p10 | 10 | 106 | 111 (111) | 126 | 739 | 745 (745) | 755 | 62.2k | 62.5k (62.5k) | 63.0k |
| rmatr100-p5 | 10 | 129 | 133 (133) | 139 | 381 | 382 (382) | 383 | 67.6k | 67.6k (67.6k) | 67.6k |
| rmine6 | 1 | 7074 | 7074 (7187) | 7200 | 818.0k | 909.7k (894.0k) | 942.4k | 5.6m | 5.6m (6.2m) | 6.8m |
| roclI-4-11 | 10 | 413 | 534 (534) | 807 | 5.1k | 8.4k (8.4k) | 15.2k | 83.9k | 119.6k (119.6k) | 194.5k |
| rococoC10-001000 | 2 | 2069 | 4622 (6684) | 7200 | 80.4k | 360.8k (296.9k) | 559.6k | 6.1m | 10.4m (23.8m) | 36.5m |
| roll3000 | 6 | 1499 | 3186 (4791) | 7200 | 157.0k | 356.4k (570.1k) | 931.1k | 2.4m | 4.6m (7.6m) | 13.8m |
| rout | 10 | 38 | 105 (105) | 164 | 10.7k | 41.7k (41.7k) | 67.1k | 206.5k | 607.4k (607.4k) | 891.0k |
| satellites1-25 | 10 | 951 | 1619 (1619) | 2244 | 1.1k | 3.5k (3.5k) | 6.6k | 180.8k | 802.0k (802.0k) | 1.8m |
| set1ch | 10 | 2 | 2 (2) | 3 | 3 | 5 (5) | 7 | 987 | 1.1k (1.1k) | 1.5k |
| seymour | 0 | 7200 | - (7200) | 7200 | 106.0k | - (111.0k) | 119.3k | 7.5m | - (7.9m) | 8.5m |
| sp97ar | 0 | 7200 | - (7200) | 7200 | 122.9k | - (147.3k) | 183.7k | 4.1m | - (4.8m) | 5.4m |
| sp98ic | 10 | 1614 | 3332 (3332) | 4676 | 76.0k | 159.7k (159.7k) | 222.7k | 1.4m | 2.9m (2.9m) | 4.1m |
| sp98ir | 10 | 57 | 70 (70) | 77 | 1.3k | 1.8k (1.8k) | 2.3k | 30.6k | 41.5k (41.5k) | 48.8k |
| stein27 | 10 | 2 | 2 (2) | 2 | 3.7k | 4.2k (4.2k) | 4.5k | 13.5k | 14.3k (14.3k) | 15.0k |
| stein45 | 10 | 32 | 33 (33) | 37 | 49.0k | 51.2k (51.2k) | 55.7k | 255.5k | 265.5k (265.5k) | 281.1k |
| swath | 0 | 7200 | - (7200) | 7200 | 105.3k | - (149.3k) | 193.5k | 1.5m | - (1.9m) | 2.5m |
| tanglegram1 | 5 | 827 | 1579 (4390) | 7200 | 21 | 168 (876) | 1.9k | 118.3k | 282.1k (899.9k) | 1.7m |

Table 9: Detailed results for the svts scoring function on the benchmark test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|---------------|---------|------------------|-------------|------|-----------------|-----------------|--------|----------------|-----------------|-------|
| | | min | avg | max | min | avg | max | min | avg | max |
| tanglegram2 | 10 | 7 | 10 (10) | 16 | 3 | 4 (4) | 7 | 6.0k | 6.9k (6.9k) | 9.1k |
| timtab1 | 10 | 778 | 1193 (1193) | 1579 | 495.8k | 790.9k (790.9k) | 1.0m | 4.1m | 6.3m (6.3m) | 8.5m |
| timtab2 | 0 | 7200 | - (7200) | 7200 | 3.0m | - (3.3m) | 3.6m | 30.1m | - (34.9m) | 38.3m |
| tr12-30 | 10 | 2856 | 3629 (3629) | 4583 | 634.3k | 792.2k (792.2k) | 977.1k | 3.2m | 4.0m (4.0m) | 4.7m |
| triptim1 | 10 | 160 | 185 (185) | 224 | 1 | 1 (1) | 1 | 30.7k | 33.8k (33.8k) | 35.9k |
| unitcal_7 | 9 | 1521 | 3101 (3511) | 7200 | 12.3k | 27.8k (34.3k) | 92.8k | 216.6k | 487.5k (558.4k) | 1.2m |
| vpm1 | 10 | 0 | 0 (0) | 0 | 1 | 1 (1) | 1 | 59 | 60 (60) | 60 |
| vpm2 | 10 | 2 | 2 (2) | 2 | 149 | 182 (182) | 195 | 1.6k | 1.8k (1.8k) | 1.9k |
| vpphard | 0 | 7200 | - (7200) | 7200 | 687 | - (6.5k) | 13.3k | 1.3m | - (1.5m) | 1.8m |
| zib54-UUE | 3 | 4985 | 5380 (6654) | 7200 | 447.1k | 466.3k (589.3k) | 708.8k | 22.0m | 23.4m (29.1m) | 32.8m |
| Total | 1226 | 778074 (3398874) | | | 126.9m (1.0b) | | | 1.6b (7.3b) | | |
| Arit. mean | | 635 (2138) | | | 103.5k (635.3k) | | | 1.3m (4.6m) | | |
| Sh. geo. mean | | 86 (249) | | | 1.3k (4.4k) | | | 33.5k (110.8k) | | |

Table 9: Detailed results for the *svts* scoring function on the benchmark test set.

B.2 Tree test set results

Tables 10, 11 and 12 give the results on the tree test set for the scoring functions `product`, `ratio`, and `svts`, respectively.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|---------------|---------|----------|---------------|-------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| blp-ar98 | 0 | 43200 | - (43200) | 43200 | 1.5m | - (2.0m) | 2.3m | 13.5m | - (17.0m) | 20.2m |
| csched007 | 10 | 26734 | 31000 (31000) | 39186 | 4.8m | 5.7m (5.7m) | 7.0m | 101.7m | 124.5m (124.5m) | 152.4m |
| csched008 | 10 | 750 | 1440 (1440) | 3065 | 58.2k | 112.4k (112.4k) | 221.9k | 7.4m | 15.4m (15.4m) | 33.7m |
| enlight15 | 0 | 43200 | - (43200) | 43200 | 20.1m | - (21.9m) | 22.8m | 101.4m | - (111.2m) | 116.3m |
| enlight16 | 0 | 43200 | - (43200) | 43200 | 23.2m | - (24.1m) | 25.3m | 99.6m | - (103.6m) | 109.1m |
| glass4 | 10 | 15 | 28 (28) | 45 | 5.4k | 11.5k (11.5k) | 18.9k | 36.3k | 75.2k (75.2k) | 130.2k |
| gmu-35-40 | 0 | 43200 | - (43200) | 43200 | 64.2m | - (70.4m) | 78.4m | 157.1m | - (172.4m) | 186.6m |
| gmu-35-50 | 0 | 43200 | - (43200) | 43200 | 51.1m | - (58.6m) | 72.5m | 165.5m | - (213.7m) | 246.4m |
| go19 | 0 | 43200 | - (43200) | 43200 | 9.9m | - (10.7m) | 11.3m | 333.2m | - (355.0m) | 377.6m |
| k16x240 | 10 | 19787 | 23835 (23835) | 29016 | 18.3m | 23.2m (23.2m) | 27.7m | 74.4m | 90.7m (90.7m) | 112.9m |
| leo1 | 0 | 43200 | - (43200) | 43200 | 3.6m | - (3.9m) | 4.4m | 44.8m | - (49.7m) | 52.8m |
| markshare_5_0 | 0 | 43200 | - (43200) | 43200 | 121.2m | - (123.8m) | 126.0m | 217.9m | - (222.6m) | 226.5m |
| maxgasflow | 0 | 43200 | - (43200) | 43200 | 1.4m | - (1.8m) | 2.3m | 8.3m | - (12.8m) | 18.8m |
| mc11 | 0 | 43200 | - (43200) | 43200 | 2.4m | - (3.2m) | 3.9m | 42.1m | - (49.9m) | 60.4m |
| momentum1 | 0 | 43200 | - (43200) | 43200 | 143.2k | - (166.0k) | 188.9k | 4.1m | - (4.7m) | 5.8m |
| neos-1426635 | 0 | 43200 | - (43200) | 43200 | 35.4m | - (40.6m) | 44.1m | 244.4m | - (263.1m) | 283.0m |
| neos-1426662 | 0 | 43200 | - (43200) | 43200 | 18.6m | - (21.2m) | 23.4m | 125.3m | - (137.8m) | 154.4m |
| neos-1436709 | 0 | 43200 | - (43200) | 43200 | 14.3m | - (17.1m) | 18.6m | 181.6m | - (214.3m) | 275.9m |
| neos-1440460 | 0 | 43200 | - (43200) | 43200 | 20.0m | - (21.1m) | 22.5m | 275.8m | - (298.6m) | 340.2m |
| neos-1442119 | 0 | 43200 | - (43200) | 43200 | 16.8m | - (18.7m) | 20.4m | 186.4m | - (207.1m) | 227.1m |
| neos-1442657 | 0 | 43200 | - (43200) | 43200 | 15.4m | - (17.7m) | 20.0m | 216.4m | - (249.7m) | 297.3m |
| neos-1616732 | 10 | 8341 | 9182 (9182) | 9915 | 2.3m | 2.5m (2.5m) | 2.6m | 31.6m | 34.0m (34.0m) | 36.1m |
| neos-1620770 | 0 | 43200 | - (43200) | 43200 | 4.8m | - (6.2m) | 7.7m | 70.3m | - (94.7m) | 123.4m |
| neos-820146 | 0 | 43200 | - (43200) | 43200 | 8.8m | - (10.1m) | 11.1m | 341.1m | - (367.0m) | 386.1m |
| neos-820157 | 0 | 43200 | - (43200) | 43200 | 7.0m | - (7.5m) | 8.1m | 290.2m | - (320.5m) | 347.5m |
| neos-826650 | 0 | 43200 | - (43200) | 43200 | 1.5m | - (2.4m) | 3.5m | 137.8m | - (198.7m) | 230.6m |
| neos-826841 | 0 | 43200 | - (43200) | 43200 | 3.8m | - (4.3m) | 4.7m | 127.2m | - (141.8m) | 160.5m |
| neos-847302 | 0 | 43200 | - (43200) | 43200 | 2.6m | - (3.0m) | 3.4m | 407.3m | - (422.1m) | 434.1m |
| neos-911880 | 0 | 43200 | - (43200) | 43200 | 18.9m | - (24.3m) | 30.6m | 65.2m | - (103.5m) | 142.6m |
| neos-942830 | 10 | 1831 | 9184 (9184) | 17919 | 237.4k | 1.5m (1.5m) | 3.2m | 11.1m | 72.4m (72.4m) | 139.5m |
| neos15 | 1 | 34352 | 34352 (42315) | 43200 | 12.3m | 12.3m (16.5m) | 17.7m | 106.9m | 106.9m (134.1m) | 148.0m |
| neos16 | 0 | 43200 | - (43200) | 43200 | 7.3m | - (8.6m) | 9.7m | 63.3m | - (77.8m) | 89.9m |
| neos858960 | 10 | 3687 | 3997 (3997) | 4338 | 2.7m | 2.8m (2.8m) | 3.0m | 51.3m | 54.0m (54.0m) | 56.9m |
| nobel-eu-DBE | 0 | 43200 | - (43200) | 43200 | 7.3m | - (8.7m) | 10.6m | 48.2m | - (56.2m) | 65.1m |

Table 10: Detailed results for the product scoring function on the tree test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|--------------------|---------------|-------|---------------|-----------------|--------|----------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| noswot | 10 | 272 | 556 (556) | 868 | 367.0k | 679.8k (679.8k) | 1.0m | 1.2m | 2.1m (2.1m) | 3.2m |
| ns1456591 | 0 | 43200 | - (43200) | 43200 | 297.4k | - (513.6k) | 892.9k | 11.8m | - (23.1m) | 45.7m |
| ns1766074 | 10 | 5495 | 5747 (5747) | 6173 | 895.3k | 925.7k (925.7k) | 949.9k | 3.4m | 3.7m (3.7m) | 3.8m |
| ns2081729 | 0 | 43200 | - (43200) | 43200 | 8.8m | - (18.3m) | 31.1m | 34.6m | - (53.1m) | 75.7m |
| ns894788 | 10 | 4 | 4 (4) | 4 | 1 | 1 (1) | 1 | 303 | 387 (387) | 487 |
| p2m2p1m1p0n100 | 0 | 43200 | - (43200) | 43200 | 197.4m | - (198.4m) | 199.4m | 103.3m | - (103.8m) | 104.4m |
| p80x400b | 0 | 43200 | - (43200) | 43200 | 24.2m | - (27.3m) | 30.3m | 80.4m | - (86.5m) | 95.9m |
| pg | 10 | 23 | 24 (24) | 24 | 203 | 203 (203) | 203 | 14.3k | 14.3k (14.3k) | 14.3k |
| pigeon-10 | 0 | 43200 | - (43200) | 43200 | 3.7m | - (4.4m) | 5.2m | 13.8m | - (24.9m) | 39.8m |
| pigeon-11 | 0 | 43200 | - (43200) | 43200 | 3.0m | - (3.9m) | 4.7m | 12.6m | - (17.0m) | 24.9m |
| pigeon-12 | 0 | 43200 | - (43200) | 43200 | 3.1m | - (3.7m) | 4.0m | 12.3m | - (23.1m) | 38.6m |
| ran14x18 | 10 | 16956 | 23039 (23039) | 31792 | 13.1m | 18.4m (18.4m) | 25.3m | 102.2m | 139.1m (139.1m) | 189.5m |
| ran14x18-disj-8 | 0 | 17544 | - (25432) | 38252 | 5.7m | - (7.8m) | 12.5m | 80.9m | - (114.5m) | 177.4m |
| reblock166 | 4 | 14474 | 16858 (32663) | 43200 | 1.4m | 1.6m (3.5m) | 7.3m | 5.3m | 6.5m (13.4m) | 26.4m |
| timtab1 | 10 | 754 | 1142 (1142) | 1536 | 532.7k | 774.5k (774.5k) | 1.0m | 4.4m | 6.0m (6.0m) | 8.0m |
| umts | 10 | 1544 | 8726 (8726) | 24776 | 219.7k | 1.5m (1.5m) | 3.8m | 2.6m | 15.3m (15.3m) | 49.8m |
| wachplan | 10 | 12679 | 15334 (15334) | 18784 | 269.8k | 324.2k (324.2k) | 398.4k | 107.9m | 131.1m (131.1m) | 162.9m |
| wnq-n100-mw99-14 | 0 | 43200 | - (43200) | 43200 | 39.2k | - (42.4k) | 45.8k | 3.8m | - (4.0m) | 4.2m |
| Total | 155 | 1434174 (17024498) | | | 602.1m (8.7b) | | | 7.0b (57.5b) | | |
| Arit. mean | | 9253 (32739) | | | 3.9m (16.8m) | | | 45.3m (110.6m) | | |
| Sh. geo. mean | | 2136 (17459) | | | 361.9k (3.2m) | | | 5.7m (37.3m) | | |

Table 10: Detailed results for the product scoring function on the tree test set.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------|---------|----------|---------------|-------|-------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| blp-ar98 | 0 | 43200 | - (43200) | 43200 | 1.6m | - (1.9m) | 2.3m | 14.9m | - (17.2m) | 19.1m |
| csched007 | 10 | 20695 | 24218 (24218) | 28186 | 3.7m | 4.2m (4.2m) | 4.8m | 90.5m | 101.9m (101.9m) | 123.7m |
| csched008 | 10 | 824 | 1737 (1737) | 2992 | 63.6k | 115.6k (115.6k) | 179.2k | 8.6m | 19.7m (19.7m) | 35.7m |
| enlight15 | 0 | 43200 | - (43200) | 43200 | 25.6m | - (26.9m) | 28.0m | 106.2m | - (111.8m) | 117.5m |
| enlight16 | 0 | 43200 | - (43200) | 43200 | 23.8m | - (25.1m) | 26.7m | 106.6m | - (113.0m) | 120.5m |
| glass4 | 10 | 15 | 38 (38) | 101 | 5.5k | 15.4k (15.4k) | 35.9k | 41.5k | 102.8k (102.8k) | 242.0k |

Table 11: Detailed results for the ratio scoring function on the tree test set (continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|----------------|---------|----------|---------------|-------|--------|-----------------|--------|---------------|----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| gmu-35-40 | 2 | 7147 | 15954 (37751) | 43200 | 11.1m | 23.6m (61.0m) | 75.4m | 26.0m | 54.0m (158.9m) | 217.1m |
| gmu-35-50 | 0 | 43200 | - (43200) | 43200 | 50.0m | - (62.7m) | 73.9m | 169.3m | - (200.3m) | 228.2m |
| go19 | 0 | 43200 | - (43200) | 43200 | 9.3m | - (9.8m) | 10.6m | 342.5m | - (355.9m) | 377.2m |
| k16x240 | 10 | 15703 | 20899 (20899) | 28244 | 16.2m | 20.6m (20.6m) | 28.1m | 58.4m | 86.2m (86.2m) | 118.0m |
| leo1 | 0 | 43200 | - (43200) | 43200 | 3.3m | - (3.8m) | 4.3m | 48.1m | - (51.2m) | 54.5m |
| markshare_5_0 | 0 | 43200 | - (43200) | 43200 | 126.2m | - (133.5m) | 137.7m | 231.0m | - (244.3m) | 252.1m |
| maxgasflow | 0 | 43200 | - (43200) | 43200 | 1.4m | - (1.6m) | 1.9m | 7.8m | - (15.0m) | 21.4m |
| mc11 | 0 | 43200 | - (43200) | 43200 | 2.6m | - (3.2m) | 4.0m | 29.6m | - (39.3m) | 50.2m |
| momentum1 | 0 | 43200 | - (43200) | 43200 | 125.9k | - (162.4k) | 187.3k | 3.8m | - (4.5m) | 5.1m |
| neos-1426635 | 0 | 43200 | - (43200) | 43200 | 38.3m | - (41.1m) | 44.6m | 245.9m | - (266.6m) | 292.7m |
| neos-1426662 | 0 | 43200 | - (43200) | 43200 | 20.1m | - (21.7m) | 23.3m | 126.7m | - (136.2m) | 149.1m |
| neos-1436709 | 0 | 43200 | - (43200) | 43200 | 11.9m | - (18.3m) | 21.1m | 154.4m | - (203.6m) | 376.6m |
| neos-1440460 | 0 | 43200 | - (43200) | 43200 | 19.7m | - (21.0m) | 23.6m | 259.9m | - (290.8m) | 397.0m |
| neos-1442119 | 0 | 43200 | - (43200) | 43200 | 18.0m | - (19.2m) | 20.7m | 188.3m | - (203.6m) | 230.9m |
| neos-1442657 | 0 | 43200 | - (43200) | 43200 | 16.1m | - (18.3m) | 19.6m | 210.2m | - (251.1m) | 321.3m |
| neos-1616732 | 10 | 7160 | 7773 (7773) | 8407 | 2.1m | 2.3m (2.3m) | 2.5m | 27.3m | 29.6m (29.6m) | 31.3m |
| neos-1620770 | 0 | 43200 | - (43200) | 43200 | 5.3m | - (6.3m) | 7.3m | 76.6m | - (94.7m) | 116.6m |
| neos-820146 | 0 | 43200 | - (43200) | 43200 | 8.6m | - (10.3m) | 12.0m | 334.8m | - (373.5m) | 431.3m |
| neos-820157 | 0 | 43200 | - (43200) | 43200 | 6.9m | - (7.7m) | 9.4m | 288.2m | - (317.3m) | 348.6m |
| neos-826650 | 0 | 43200 | - (43200) | 43200 | 1.8m | - (2.2m) | 3.1m | 151.4m | - (206.6m) | 235.5m |
| neos-826841 | 0 | 43200 | - (43200) | 43200 | 4.0m | - (4.5m) | 5.1m | 114.0m | - (137.8m) | 159.1m |
| neos-847302 | 0 | 43200 | - (43200) | 43200 | 2.8m | - (3.1m) | 3.3m | 383.0m | - (414.0m) | 435.1m |
| neos-911880 | 0 | 43200 | - (43200) | 43200 | 23.0m | - (27.7m) | 33.0m | 86.9m | - (118.9m) | 165.8m |
| neos-942830 | 10 | 1634 | 8689 (8689) | 13644 | 228.5k | 1.4m (1.4m) | 2.7m | 11.2m | 67.8m (67.8m) | 93.8m |
| neos15 | 1 | 31393 | 31393 (42019) | 43200 | 11.6m | 11.6m (16.6m) | 17.9m | 95.9m | 95.9m (135.6m) | 149.3m |
| neos16 | 0 | 43200 | - (43200) | 43200 | 7.0m | - (8.9m) | 11.5m | 60.9m | - (77.6m) | 97.4m |
| neos858960 | 10 | 3542 | 3676 (3676) | 3886 | 2.5m | 2.6m (2.6m) | 2.8m | 46.9m | 48.7m (48.7m) | 51.5m |
| nobel-eu-DBE | 0 | 43200 | - (43200) | 43200 | 6.4m | - (7.6m) | 8.7m | 56.7m | - (64.7m) | 70.8m |
| noswot | 10 | 344 | 900 (900) | 3519 | 436.9k | 1.1m (1.1m) | 4.3m | 1.4m | 2.8m (2.8m) | 8.6m |
| ns1456591 | 0 | 43200 | - (43200) | 43200 | 194.3k | - (472.5k) | 716.5k | 13.7m | - (31.3m) | 49.7m |
| ns1766074 | 10 | 5038 | 5463 (5463) | 5760 | 895.4k | 907.9k (907.9k) | 942.4k | 3.5m | 3.6m (3.6m) | 3.8m |
| ns2081729 | 0 | 43200 | - (43200) | 43200 | 10.4m | - (22.2m) | 35.9m | 33.5m | - (63.9m) | 97.9m |
| ns894788 | 10 | 4 | 4 (4) | 5 | 1 | 1 (1) | 1 | 303 | 387 (387) | 487 |
| p2m2p1m1p0n100 | 0 | 43200 | - (43200) | 43200 | 196.7m | - (199.1m) | 204.2m | 102.9m | - (104.2m) | 106.9m |
| p80x400b | 0 | 43200 | - (43200) | 43200 | 25.0m | - (27.2m) | 28.8m | 75.0m | - (84.6m) | 94.9m |

Table 11: Detailed results for the ratio scoring function on the tree test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|--------------------|---------------|-------|---------------|-----------------|--------|----------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| pg | 10 | 23 | 24 (24) | 26 | 197 | 197 (197) | 197 | 14.5k | 14.5k (14.5k) | 14.5k |
| pigeon-10 | 0 | 43200 | - (43200) | 43200 | 4.0m | - (4.7m) | 5.2m | 18.2m | - (25.2m) | 38.1m |
| pigeon-11 | 0 | 43200 | - (43200) | 43200 | 3.1m | - (4.2m) | 5.5m | 13.1m | - (18.3m) | 27.8m |
| pigeon-12 | 0 | 43200 | - (43200) | 43200 | 3.2m | - (3.9m) | 4.6m | 12.5m | - (19.7m) | 36.0m |
| ran14x18 | 10 | 13454 | 19209 (19209) | 26510 | 10.5m | 15.0m (15.0m) | 20.7m | 81.0m | 113.2m (113.2m) | 152.9m |
| ran14x18-disj-8 | 0 | 11086 | - (16055) | 21427 | 3.6m | - (5.0m) | 6.3m | 49.6m | - (74.0m) | 94.1m |
| reblock166 | 6 | 11729 | 19725 (29115) | 43200 | 1.5m | 2.0m (2.7m) | 6.1m | 5.7m | 8.5m (12.0m) | 24.9m |
| tintab1 | 10 | 855 | 1200 (1200) | 1547 | 592.7k | 803.6k (803.6k) | 993.3k | 4.6m | 6.3m (6.3m) | 7.9m |
| umts | 10 | 1296 | 6009 (6009) | 13658 | 186.5k | 1.1m (1.1m) | 2.4m | 2.5m | 10.6m (10.6m) | 27.2m |
| wachplan | 10 | 10904 | 15674 (15674) | 24922 | 247.9k | 327.1k (327.1k) | 458.9k | 92.2m | 133.2m (133.2m) | 217.3m |
| wnq-n100-mw99-14 | 0 | 43200 | - (43200) | 43200 | 31.4k | - (35.5k) | 37.6k | 3.3m | - (3.5m) | 3.7m |
| Total | 159 | 1336785 (16660531) | | | 575.3m (8.8b) | | | 6.5b (56.6b) | | |
| Arit. mean | | 8407 (32039) | | | 3.6m (17.0m) | | | 40.8m (108.9m) | | |
| Sh. geo. mean | | 2191 (17044) | | | 376.9k (3.1m) | | | 5.8m (36.9m) | | |

Table 11: Detailed results for the ratio scoring function on the tree test set.

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|---------------|---------|----------|---------------|-------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| blp-ar98 | 0 | 43200 | - (43200) | 43200 | 1.4m | - (1.7m) | 1.9m | 13.9m | - (17.4m) | 20.3m |
| csched007 | 10 | 18507 | 25355 (25355) | 37366 | 3.0m | 4.3m (4.3m) | 5.9m | 77.9m | 105.6m (105.6m) | 143.6m |
| csched008 | 10 | 555 | 1641 (1641) | 4820 | 46.1k | 118.6k (118.6k) | 267.0k | 5.2m | 18.3m (18.3m) | 59.2m |
| enlight15 | 0 | 43200 | - (43200) | 43200 | 23.1m | - (24.8m) | 25.9m | 103.0m | - (111.6m) | 117.4m |
| enlight16 | 0 | 43200 | - (43200) | 43200 | 21.8m | - (25.2m) | 26.3m | 97.5m | - (113.6m) | 119.0m |
| glass4 | 10 | 17 | 29 (29) | 51 | 5.5k | 11.8k (11.8k) | 22.5k | 37.6k | 74.4k (74.4k) | 147.2k |
| gmu-35-40 | 1 | 39750 | 39750 (42855) | 43200 | 59.9m | 59.9m (71.2m) | 79.2m | 132.3m | 132.3m (190.1m) | 224.3m |
| gmu-35-50 | 0 | 43200 | - (43200) | 43200 | 49.0m | - (61.6m) | 75.1m | 178.0m | - (206.2m) | 249.4m |
| go19 | 0 | 43200 | - (43200) | 43200 | 9.6m | - (10.3m) | 11.0m | 327.1m | - (352.8m) | 370.0m |
| k16x240 | 10 | 16692 | 20462 (20462) | 24643 | 15.3m | 19.7m (19.7m) | 24.0m | 63.6m | 82.9m (82.9m) | 107.4m |
| leo1 | 0 | 43200 | - (43200) | 43200 | 3.1m | - (3.4m) | 3.9m | 50.2m | - (54.8m) | 59.0m |
| markshare_5_0 | 0 | 43200 | - (43200) | 43200 | 125.0m | - (128.0m) | 130.1m | 228.9m | - (234.2m) | 238.1m |
| maxgasflow | 0 | 43200 | - (43200) | 43200 | 1.3m | - (1.7m) | 2.1m | 8.2m | - (16.2m) | 20.7m |

Table 12: Detailed results for the svts scoring function on the tree test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|-----------------|---------|----------|---------------|-------|--------|-----------------|--------|---------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| mc11 | 0 | 43200 | - (43200) | 43200 | 2.3m | - (3.1m) | 3.5m | 35.8m | - (47.8m) | 57.0m |
| momentum1 | 0 | 43200 | - (43200) | 43200 | 128.9k | - (176.4k) | 197.3k | 4.4m | - (5.1m) | 5.8m |
| neos-1426635 | 0 | 43200 | - (43200) | 43200 | 36.5m | - (40.8m) | 44.2m | 208.8m | - (249.2m) | 276.9m |
| neos-1426662 | 0 | 43200 | - (43200) | 43200 | 17.7m | - (20.8m) | 22.9m | 119.7m | - (138.4m) | 153.3m |
| neos-1436709 | 0 | 43200 | - (43200) | 43200 | 16.5m | - (18.8m) | 20.8m | 163.8m | - (184.7m) | 212.8m |
| neos-1440460 | 0 | 43200 | - (43200) | 43200 | 18.4m | - (20.7m) | 21.9m | 246.9m | - (282.5m) | 332.5m |
| neos-1442119 | 0 | 43200 | - (43200) | 43200 | 18.6m | - (19.5m) | 20.4m | 192.1m | - (200.8m) | 215.1m |
| neos-1442657 | 0 | 43200 | - (43200) | 43200 | 15.4m | - (18.1m) | 19.5m | 211.3m | - (243.0m) | 275.0m |
| neos-1616732 | 10 | 6605 | 7487 (7487) | 8361 | 1.8m | 2.1m (2.1m) | 2.3m | 25.8m | 28.4m (28.4m) | 31.3m |
| neos-1620770 | 0 | 43200 | - (43200) | 43200 | 4.2m | - (5.9m) | 6.9m | 64.9m | - (90.7m) | 112.0m |
| neos-820146 | 0 | 43200 | - (43200) | 43200 | 9.3m | - (10.2m) | 10.9m | 355.7m | - (373.3m) | 393.6m |
| neos-820157 | 0 | 43200 | - (43200) | 43200 | 7.1m | - (7.7m) | 8.6m | 297.0m | - (319.2m) | 346.5m |
| neos-826650 | 0 | 43200 | - (43200) | 43200 | 1.8m | - (2.3m) | 3.3m | 162.4m | - (206.9m) | 234.4m |
| neos-826841 | 0 | 43200 | - (43200) | 43200 | 3.6m | - (4.3m) | 5.0m | 116.9m | - (142.3m) | 171.9m |
| neos-847302 | 0 | 43200 | - (43200) | 43200 | 2.6m | - (2.9m) | 3.4m | 396.7m | - (410.4m) | 424.2m |
| neos-911880 | 0 | 43200 | - (43200) | 43200 | 15.4m | - (26.3m) | 31.9m | 68.0m | - (104.3m) | 136.6m |
| neos-942830 | 10 | 2538 | 9145 (9145) | 18941 | 375.5k | 1.4m (1.4m) | 3.0m | 18.7m | 73.7m (73.7m) | 156.3m |
| neos15 | 1 | 29801 | 29801 (41860) | 43200 | 11.2m | 11.2m (16.2m) | 17.8m | 97.9m | 97.9m (139.3m) | 151.4m |
| neos16 | 0 | 43200 | - (43200) | 43200 | 8.1m | - (8.9m) | 10.3m | 72.2m | - (79.6m) | 89.7m |
| neos858960 | 10 | 3359 | 3665 (3665) | 3784 | 2.5m | 2.6m (2.6m) | 2.8m | 46.9m | 48.7m (48.7m) | 51.5m |
| nobel-eu-DBE | 0 | 43200 | - (43200) | 43200 | 7.5m | - (8.2m) | 8.9m | 41.5m | - (58.2m) | 69.4m |
| noswot | 10 | 331 | 799 (799) | 2625 | 413.9k | 922.0k (922.0k) | 2.8m | 1.4m | 2.7m (2.7m) | 7.2m |
| ns1456591 | 0 | 43200 | - (43200) | 43200 | 178.7k | - (463.2k) | 696.1k | 14.8m | - (31.5m) | 48.1m |
| ns1766074 | 10 | 5349 | 5607 (5607) | 6443 | 895.4k | 907.9k (907.9k) | 942.4k | 3.5m | 3.6m (3.6m) | 3.8m |
| ns2081729 | 0 | 43200 | - (43200) | 43200 | 11.0m | - (22.8m) | 32.6m | 38.5m | - (67.6m) | 97.2m |
| ns894788 | 10 | 4 | 4 (4) | 5 | 1 | 1 (1) | 1 | 303 | 387 (387) | 487 |
| p2m2p1m1p0n100 | 0 | 43200 | - (43200) | 43200 | 196.3m | - (197.8m) | 199.5m | 102.7m | - (103.5m) | 104.4m |
| p80x400b | 0 | 43200 | - (43200) | 43200 | 24.2m | - (26.4m) | 29.2m | 82.3m | - (89.2m) | 96.4m |
| pg | 10 | 23 | 24 (24) | 25 | 223 | 223 (223) | 223 | 14.7k | 14.7k (14.7k) | 14.7k |
| pigeon-10 | 0 | 43200 | - (43200) | 43200 | 3.4m | - (4.4m) | 5.0m | 15.5m | - (24.3m) | 41.6m |
| pigeon-11 | 0 | 43200 | - (43200) | 43200 | 3.0m | - (4.2m) | 5.9m | 12.6m | - (18.3m) | 28.4m |
| pigeon-12 | 0 | 43200 | - (43200) | 43200 | 3.1m | - (3.9m) | 4.4m | 11.5m | - (20.4m) | 40.1m |
| ran14x18 | 10 | 13372 | 19987 (19987) | 26400 | 10.1m | 14.8m (14.8m) | 19.8m | 85.9m | 122.7m (122.7m) | 158.6m |
| ran14x18-disj-8 | 0 | 14834 | - (20026) | 23964 | 4.4m | - (5.9m) | 7.2m | 70.4m | - (94.7m) | 118.8m |
| reblock166 | 6 | 10968 | 17803 (27962) | 43200 | 1.2m | 1.9m (2.8m) | 7.8m | 5.0m | 8.1m (12.0m) | 26.4m |

Table 12: Detailed results for the svts scoring function on the tree test set(continues in the next page).

| Instance | #solved | time (s) | | | nodes | | | LP iterations | | |
|------------------|---------|--------------------|---------------|-------|---------------|-----------------|--------|----------------|-----------------|--------|
| | | min | avg | max | min | avg | max | min | avg | max |
| timtab1 | 10 | 739 | 1168 (1168) | 1685 | 495.8k | 790.9k (790.9k) | 1.0m | 4.1m | 6.3m (6.3m) | 8.5m |
| umts | 10 | 1536 | 5581 (5581) | 9636 | 258.5k | 999.4k (999.4k) | 1.5m | 2.6m | 8.9m (8.9m) | 17.9m |
| wachplan | 10 | 11060 | 15542 (15542) | 25357 | 247.9k | 327.1k (327.1k) | 458.9k | 92.2m | 133.2m (133.2m) | 217.3m |
| wnq-n100-mw99-14 | 0 | 43200 | - (43200) | 43200 | 29.9k | - (33.5k) | 35.9k | 3.1m | - (3.4m) | 3.7m |
| Total | 158 | 1341327 (16747991) | | | 572.3m (8.8b) | | | 6.6b (56.7b) | | |
| Arit. mean | | 8489 (32208) | | | 3.6m (16.9m) | | | 42.0m (109.1m) | | |
| Sh. geo. mean | | 2140 (17096) | | | 364.2k (3.1m) | | | 5.7m (37.2m) | | |

Table 12: Detailed results for the `svts` scoring function on the tree test set.

References

- [1] T. Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007.
- [2] T. Achterberg. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, July 2009.
- [3] T. Achterberg and T. Berthold. Hybrid branching. In WJ. Hoes and J. N. Hooker, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5547 of *Lecture Notes in Computer Science*, pages 309–311. Springer, 2009.
- [4] T. Achterberg, T. Koch, and A. Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):361–372, 2006.
- [5] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer programming*, volume 271 of *Graduate Texts in Mathematics*. Springer, 2014.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [7] N. Jacobson. *Basic Algebra I: Second Edition*. Dover, 2009.
- [8] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D. E. Steffy, and K. Wolter. MIPLIB 2010. *Mathematical Programming Computation*, 3(2):103–163, 2011.
- [9] O. Kullmann. Fundamentals of branching heuristics. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 205–244. IOS Press, 2009.
- [10] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):pp. 497–520, 1960.
- [11] A. Lodi and A. Tramontani. *Performance Variability in Mixed-Integer Programming*, chapter 2, pages 1–12. 2013.
- [12] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.

- [13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.
- [14] Larry J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1 – 22, 1976.
- [15] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal of Computing*, 20(5):865–877, October 1991.