Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

YUJI SHINANO, STEFAN HEINZ, STEFAN VIGERSKE, MICHAEL WINKLER

FiberSCIP – A shared memory parallelization of SCIP

*Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin

ZIB-Report 13-55 (September 2013)

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

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

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

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

FiberSCIP – A shared memory parallelization of SCIP

Yuji Shinano, Stefan Heinz, Stefan Vigerske, Michael Winkler

Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany

shinano@zib.de, heinz@zib.de, vigerske@zib.de, michael.winkler@zib.de

April 28, 2015

Abstract

Recently, parallel computing environments have become significantly popular. In order to obtain the benefit of using parallel computing environments, we have to deploy our programs for these effectively. This paper focuses on a parallelization of SCIP (Solving Constraint Integer Programs), which is a MIP solver and constraint integer programming framework available in source code. There is a parallel extension of SCIP named ParaSCIP, which parallelizes SCIP on massively parallel distributed memory computing environments. This paper describes FiberSCIP, which is yet another parallel extension of SCIP to utilize multi-threaded parallel computation on shared memory computing environments, and has the following contributions: First, the basic concept of having two parallel extensions and the relationship between them and the parallelization framework provided by UG (Ubiquity Generator) is presented, including an implementation of deterministic parallelization. Second, the difficulties to achieve a good performance that utilizes all resources on an actual computing environment and the difficulties of performance evaluation of the parallel solvers are discussed. Third, a way to evaluate the performance of new algorithms and parameter settings of the parallel extensions is presented. Finally, current performance of FiberSCIP for solving mixed-integer linear programs (MIPs) and mixed-integer non-linear programs (MINLPs) in parallel is demonstrated.

1 Introduction

The paradigm of constraint integer programming (CIP) (formal definition of CIP will be presented in Section 2) developed by Achterberg (2007) combines modeling and solving techniques from the fields of constraint programming (CP), mixed-integer programming (MIP), and satisfiability testing (SAT). The paradigm allows us to address a wide range of optimization problems. SCIP is an implementation of the idea of CIP as a branch-cut-and-price framework and is now continuously extended by a group of researchers centered at Zuse Institute Berlin (ZIB), Technical University Darmstadt, and Friedrich-Alexander-University Erlangen-Nürnberg. SCIP 2.1.0 can handle not only MIP, but also

pseudo-Boolean optimization problems (Berthold et al., 2009), mixed-integer quadratically constraint programming problems (MIQCP) (Berthold et al., 2011, 2012), mixed-integer nonlinear programming problems (MINLP) (Vigerske, 2013), etc.

SCIP solves CIPs by a branch-and-bound algorithm, which enumerate all feasible solutions implicitly by splitting the feasible region recursively, thereby generating a *branch-and-bound tree*. Each node of the tree represents a *subproblem* that can be processed independently. Therefore, SCIP is supposed to be a perfect candidate for the parallelization of tree search. However, it involves a mathematically supercharged tree search algorithm, employing sophisticated algorithms to keep the enumeration effort small as follows: At each subproblem, domain propagation is performed to exclude further values from the variables domains and a relaxation may be solved to obtain a local dual bound. The relaxation may be strengthened by adding further valid constraints (e.g., linear inequalities), which cut off the optimal solution of the relaxation. In case a subproblem is found to be infeasible, conflict analysis is performed to learn additional valid constraints (conflict clauses). Primal heuristics are used as supplementary methods to improve the primal bound, which is used to prune the branch-and-bound tree. These procedures make it difficult to parallelize the branch-and-bound algorithm efficiently.

Since parallel computing environments have become significantly popular nowadays, commercial MIP solvers such as CPLEX, Gurobi, and Xpress work in parallel on multi-core desktop computers. The development of FiberSCIP, a parallel version of SCIP on shared memory multi-core desktop computers, started in 2010. Unfortunately, parallelization was not in the initial design scope of SCIP, though SCIP was designed to be used as a thread-safe library. In order to parallelize the tree search of SCIP internally, one would have to redesign SCIP (which consists of more than 400,000 non-empty lines of C code) very carefully in order to keep its plugin-based design structure. On top of that, all developers of SCIP would have to take this parallelism into account. Therefore, we decided to parallelize the tree search for a single problem from "outside", that is, such that each thread uses its own SCIP instance to work on its subproblem. Another reason for this choice is given by the large differences in performance between "simple" and commercial state-of-the-art single- and multi-threaded MIP solvers, as is shown in Appendix A (the MIPLIB2010 benchmarks conducted by H. Mittelmann). These benchmarks show that the development of a new and state-of-the-art parallel MIP solver from scratch would require large resources, so that a parallelization of an existing state-of-the-art solver (like SCIP) from "outside" is a rational decision. Further, such a parallelization can immediately benefit from improvements on the used single-thread MIP solver.

There already exist a number of parallel state space search or tree search software frameworks (Goux et al., 2001; Ralphs et al., 2004; Eckstein et al., 2009; van Nieuwpoort et al., 2010; Sun et al., 2011; Bendjoudi et al., 2012)[1]. Especially recent ones are very scalable from a parallel tree search point of view. However, the examples used to show their scalability can often be considered as "easy" for state-of-the-art algorithms, thus

[1]We cited here only some of those which are still accessible. Many other frameworks have disappeared because the platform or used libraries are not available anymore.

not allowing for conclusions about their performance when combined with sophisticated solver implementations. For example, Sun et al. (2011) presented results for scaling up to 4096 processors on the basis of a Traveling Salesman Problem instance with only 25 cities. Further, it can be hard to integrate state-of-the-art solving algorithms into a scalable parallel tree search framework, because the algorithms already exploit very sophisticated tree search methods with shared knowledge and are therefore hard to adjust to an external tree search framework in general.

From a problem solving point of view, parallel implementations of SAT solvers, see Martins et al. (2012) for a survey, have been solving harder and larger instances than what was possible by state-of-the-art sequential solvers. Regarding MIP solving, Xu et al. (2009) summarized many software packages for solving MIP in parallel. These solvers are categorized in two types, those which are basically build from scratch (Eckstein, 1994; Linderoth, 1998; Bixby et al., 1999; Chen and Ferris, 2000; Chen et al., 2001; Eckstein et al., 2001; Ralphs et al., 2003, 2011) and those which embed an already existing powerful sequential/multi-threaded MIP solver (Mitra et al., 1997; Shinano et al., 2003; Nwana et al., 2004; Shinano and Fujie, 2007; Shinano et al., 2008; Bussieck et al., 2009; Shinano et al., 2012). Only a very few members of the latter category obtained competitive problem solving results when compared to state-of-the-art sequential programs at their time. One exceptional case is Bixby et al. (1999), which solved two open instances to exploit parallelism and a new branching variable selection rule. To the best of our knowledge, only GAMS/CPLEX/Condor (Bussieck et al., 2009) and ParaSCIP (Shinano et al., 2012) solved successfully several open instances from the popular MIPLIB2003 and MIPLIB2010 instance libraries (Achterberg et al., 2006; Koch et al., 2011) by using massively parallelized MIP solvers.

Our approach to build FiberSCIP is to parallelize tree search from "outside" of SCIP by dynamically splitting up the search tree and maintaining each subtree in a separate SCIP instance. SCIP's sophisticated tree search is performed in each SCIP solver, thereby directly benefiting from the state-of-the-art solving techniques of SCIP. Only the distribution of open subproblems to SCIP solvers is controlled from "outside", thereby trying to balance the workload effectively. An obvious disadvantage is that parallelization from "outside" leads to an inevitable overhead compared to an "internal" parallelization, as more memory is required and decisions such as when to interrupt the exploitation of a subtree are deferred. ParaSCIP, which parallelizes SCIP for massively parallel distributed memory computing environment also from "outside", has been run by using up to 7,000 cores on the supercomputer HLRN II[2] (Shinano et al., 2012; Koch et al., 2012) and up to 80,000 cores on Titan at Oak Ridge National Laboratory. Since ParaSCIP targets on solving very hard instances, where the overhead could be neglected, FiberSCIP is intended to run on a normal desktop computer with multi-core processor(s). Therefore, implementing an "outside" parallelization with low overhead on desktop computers is challenging, but also brings several advantages. That is, FiberSCIP virtually gives a development environment for ParaSCIP, because both solvers are

[2] `https://www.hlrn.de/`

implemented in the same software framework, named *Ubiquity Generator (UG) framework*[3]. Thus, algorithmic improvements done for FiberSCIP can be used directly in ParaSCIP and FiberSCIP can be used to gain some insight into the performance of ParaSCIP. Further, since parallelization is done from "outside", effects of different parallelization strategies can be investigated more clearly and SCIP developers can concentrate on improving algorithmic or mathematical technique within SCIP without caring about parallelization aspects.

Even if the parallelization feature is completely separated, the evaluation of the performance of parallel MIP or MINLP solvers on actual computers is difficult. This paper will illustrate these difficulties and investigate their source. The following sections are organized as follows. First, we formally define the problem classes CIP, MIP, and MINLP, and introduce SCIP briefly. Next, we explain main concepts of the UG framework, its instantiation in form of FiberSCIP, and the implementation of deterministic parallelism. After that, extensive computational experiments investigate various aspects of the performance of FiberSCIP. We finish with some concluding remarks.

2 SCIP

SCIP is the implementation of the idea of CIP. CIP is formally defined as follows:

Definition 1 (constraint integer program)**.** A *constraint integer program* (CIP) is a tuple $(\mathfrak{C}, I, c)$ that encodes the task of solving

$$\min\{\langle c, x\rangle : \mathfrak{C}(x),\ x \in \mathbb{R}^n,\ x_I \in \mathbb{Z}^{|I|}\},$$

where $c \in \mathbb{R}^n$ is the objective function vector, $\mathfrak{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ specifies the constraints $\mathcal{C}_j : \mathbb{R}^n \to \{0, 1\}$, $j \in [m]$, and $I \subseteq [n]$ specifies the set of variables that have to take integral values. Further, a CIP has to fulfill the condition

$$\forall \hat{x}_I \in \mathbb{Z}^{|I|}\ \exists (A', b') \in \mathbb{R}^{k\times n} \times \mathbb{R}^k :\ \{x \in \mathbb{R}^n : \mathfrak{C}(x), x_I = \hat{x}_I\} = \{x \in \mathbb{R}^n : A'x \leq b'\}, \quad (1)$$

where $C := [n] \setminus I$ and $k \in \mathbb{N}$.

Condition (1) states that the problem becomes a linear program when all integer variables are fixed. Thus, if the discrete variables are bounded, a CIP can be solved, in principle, by enumerating all values of the integral variables and solving the corresponding LPs.

A CIP where all constraints are linear is a mixed-integer linear program (MIP):

Definition 2 (mixed-integer linear program)**.** A *mixed-integer linear program* (MIP) is given by a tuple (A, b, c, I) with matrix $A \in \mathbb{R}^{m\times n}$, vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, and a subset $I \subseteq [n]$. The task is to solve

$$\min\{\langle c, x\rangle : Ax \leq b,\ x_I \in \mathbb{Z}^{|I|}\}. \quad (2)$$

[3] `http://ug.zib.de/`

Extending the definition of MIP towards nonlinear objective and constraint functions leads to the class of mixed-integer nonlinear programs (MINLPs):

Definition 3 (mixed-integer nonlinear program)**.** A *mixed-integer nonlinear program* (MINLP) is given by a tuple (c, g, I) with vector $c \in \mathbb{R}^n$, a function $g : \mathbb{R}^n \to \mathbb{R}^m$, and a subset $I \subseteq [n]$. The task is to solve

$$\min\{\langle c, x\rangle : g(x) \leq 0,\ x_I \in \mathbb{Z}^{|I|}\}.$$

Unlike MIP, MINLP is in general not a special case of CIP, since the nonlinear constraint $g(x) \leq 0$ may forbid a linear representation of the MINLP after fixing the integer variables, i.e., (1) would be violated (unless $I = [n]$). However, the main purpose of condition (1) is to ensure that the problem that remains after fixing all integer variables in the CIP can be solved efficiently. For practical applications, a spatial branch-and-bound algorithm that can solve the remaining nonlinear program within a finite number of steps up to a given precision is sufficient.

SCIP solves MIPs and MINLPs by a branch-and-bound algorithm that utilizes an LP relaxation for bounding. For a MIP, the LP relaxation is readily given by dropping the integrality restrictions from (2). For a MINLP, the LP relaxation is constructed by computing for each function $g_j(x)$, $\forall j \in \{1, \ldots, m\}$, linear functions that underestimate $g_j(x)$ for all x within the current variable bounds, see also Vigerske (2013) for details. The "convexification gap" between the underestimator of $g_j(x)$ and $g_j(x)$ itself depends on the width of the domain of the variables involved in $g_j(x)$. Thus, to close this gap, branching on any variable that is involved in $g_j(x)$ may be applied.

3 ParaSCIP and FiberSCIP

In our attempt to parallelize a solver from the "outside", an actual MIP/MINLP solver is abstracted and will be denoted as *base solver* in this paper. Generally, parallel tree search based solvers have three running phases, see also Ralphs (2006) and Xu et al. (2009): the *ramp-up phase* is the time period from the beginning until sufficiently many branch-and-bound nodes are available to keep all processing units busy the first time, the *primary phase* is the period between the first and last time all processing units were busy, and the *ramp-down phase* is the time period from the last time all processing units were busy until problem solving finished[4].

ParaSCIP and FiberSCIP have been developed by using a single software framework: *Ubiquity Generator(UG) framework*, which is written in C++. One of the goals of UG

[4]Note that this definition of ramp-down phase is different from the definition in Ralphs (2006), in which the ramp-down phase is loosely defined to start when the number of available tasks (subproblems) first falls below the number of available processors. In our system, we sometimes observed this situation during the period we consider the "primary phase". However, when our load balancing mechanism detected this situation, it often quickly recovered from this imbalance of workloads among SOLVERs, so that enough tasks (subproblems) were generated to keep all SOLVERs busy again. Therefore, we use the different definition of the ramp-down phase.

framework is to clarify the necessary interfaces that the underlying base SOLVER needs to provide for the parallelizations. At least the following three mechanisms need to be provided: First, a *ramp-up mechanism*, which provides algorithmic choices on how to keep all base solvers busy during this starting phase. Second, a *dynamic load balancing mechanism*, which ensures that work load is distributed to all base solvers in a dynamic way. Third, a *check-pointing and restarting mechanism*, which is a mechanism that can store the processing status of a problem to a file and is able to resume operations from that file in case of interruptions due to unforeseen events, e.g., hardware faults or limits on super computing resources. Such a mechanism is especially important for solving really hard instances, as it is not known in advance how long the algorithm will be running. UG is composed of a collection of base C++ classes, which define interfaces to actual MIP/MINLP solvers and to subproblems and solutions that are represented in a solver depending way. Additionally, there are base classes that define message passing based communications. Further, implementations of ramp-up, dynamic load balancing, and check-pointing and restarting mechanisms are available as a generic functionality. Note, that the branch-and-bound tree is maintained by the base solvers, while UG only extracts and manages only a very small number of subproblems (typically represented by variable bound changes) from the base solvers for load balancing.

Therefore, the basic concept of UG is to abstract from an actual MIP/MINLP solver and parallelization library and to provide a framework that can be used, in principle, to parallelize any powerful state-of-the-art MIP/MINLP solver on any computational environment (shared or distributed memory, multithreading or massively parallel). For a particular MIP/MINLP solver, only the interface to UG in form of specializations of base classes, as provided by UG, needs to be implemented. Similarly, for a particular parallelization library (e.g., MPI), a specialization of an abstract UG class is necessary.

3.1 Basic concept of having two parallel extensions

The message passing functions used in UG are limited to the least necessary and are wrapped within the base class. Therefore, adding support for an additional parallelization library should be easy. The most used libraries for implementing distributed parallel programs are MPI (Message Passing Interface) implementations. The virtual functions in the base class provided by UG can be mapped straightforward onto corresponding MPI functions. Pthreads is a popular library that is used to make multi-threaded programs and the UG specialization for Pthreads uses a simple message queue implementation, which has been developed as a part of UG code. From the UG framework point of view, a particular instantiated parallel solver is referred as ug [a specific solver name, a specific parallelization library name]. Here, the specific parallelization library is used to realize the message passing based communications. Using this notation, ParaSCIP is ug [SCIP, MPI] and FiberSCIP is ug [SCIP, Pthreads]. As ParaSCIP already existed, the realization of FiberSCIP only required to add code to interface with Pthreads as parallelization library. Note that UG framework is not restricted to SCIP, which can be replaced by any other thread-safe solver.

One of the motivations to have both FiberSCIP and ParaSCIP is to have a systematic

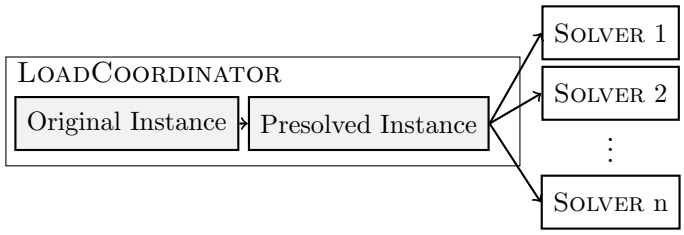

Figure 1: Initialization step

development environment for a large scale parallel branch-and-bound based solver based on SCIP. While FiberSCIP has completely the same generic parallelization mechanism as ParaSCIP, FiberSCIP (as a single process program) is much easier to debug than the MPI based ParaSCIP. For example, moving the communication point, see Section 3.4, was only possible due to the availability of FiberSCIP, since it leads to large debugging efforts. Further, also non-deterministic behavior leads to difficulties for debugging, as it may prohibit consistent reproducibility of a faulty program flow. Therefore, a deterministic parallel execution capability was introduced to UG when implementing FiberSCIP.

3.2 Parallelization mechanism provided by UG

UG provides a common parallelization mechanism that works in both ParaSCIP and FiberSCIP. The latter has two types of threads. One is the LoadCoordinator which makes all decisions about the dynamic load balancing, the other is the Solver which solves subproblems. In the beginning, the LoadCoordinator reads the instance data for a MIP/MINLP model which we refer to as the *original instance*. This instance is presolved directly inside the LoadCoordinator by use of the base solver. When using SCIP, different techniques try to fix or tighten variables, tighten constraints and detect and remove redundancies, for details see Achterberg (2007). The resulting, typically smaller, instance will be called the *presolved instance*. The presolved instance is extracted from the base solver environment, is broadcasted to all available Solver threads, and is embedded into the (local) base solver environment of each Solver. This is the only time when the complete instance is transferred. Later, only the differences between a subproblem and the presolved problem will be communicated. Figure 1 illustrates this initialization procedure. At the end of this initialization, all Solvers are instantiated with the presolved instance.

After the initialization step, the LoadCoordinator creates the root node of the branch-and-bound tree. Each node transferred through the system, we call such a node ParaNode, acts as the root of a subtree. The information that has to be sent consists only of bound changes of variables between the presolved instance and the subproblem which gets transferred. For the initial root node there is no difference between the presolved instance and the subproblem, thus it contains only administrative information such as a ParaNode identifier.

All nodes which are transferred to Solvers are kept in the LoadCoordinator with their solver statuses until the corresponding solving process terminates. The Solver which

receives a new branch-and-bound node instantiates the corresponding subproblem using the presolved instance (which was distributed in the initialization step) and the received bound changes. After that, the SOLVER starts working on the subproblem.

UG has two kinds of ramp-up mechanisms.

Normal ramp-up Active SOLVERs, which are the ones that already received a branch-and-bound node, are solving this subproblem by alternately solving nodes and transferring half of the child nodes back to the LOADCOORDINATOR. The LOADCOORDINATOR has a *node pool* to keep unassigned nodes, from which it assigns nodes to idle SOLVERs as long as an idle SOLVER exists. Even if none exists, the LOADCOORDINATOR still keeps collecting nodes from SOLVERs until it has p "good" (promising to have a large subtree underneath) unassigned nodes in its node pool. Here, p is a run-time parameter of UG, which in case of FiberSCIP is set to half of the number of SOLVER threads as default. When the LOADCOORDINATORs node pool accumulated p "good" nodes, it sends a message to quit sending nodes to all SOLVERs.

Racing ramp-up In this mechanism, the LOADCOORDINATOR sends the root branch-and-bound node to all SOLVERs and all SOLVERs start solving the root node of the presolved instance immediately. In order to generate different search trees, each SOLVER uses a different variation of parameter settings, branching variable selections, and permutations of variables and constraints. As shown in Koch et al. (2011), the latter can have a considerable impact on the performance of a solver. Due to these variations, we can expect that many SOLVERs independently generate different search trees. However, an incumbent solution found by one of the SOLVERs is broadcasted to all other SOLVERs and is used to prune parts of the SOLVERs search trees. Under certain criteria, involving the duality gap and the number of open nodes of each SOLVER, a particular SOLVER is chosen as "winner" of the *racing stage*. All open nodes of the winner SOLVER are then collected by the LOADCOORDINATOR and a termination message is send to all other SOLVERs. Next, the collected nodes are redistributed to all idle SOLVERs. If the winner SOLVER did not provide enough nodes, UG changes its strategy to normal ramp-up. If a SOLVER solves the complete problem during racing stage, the LOADCOORDINATOR terminates all SOLVERs and the solving process is finished.

Racing ramp-up has been designed to run on large scale distributed memory computing environments. In such an environment, normal ramp-up can take much time, i.e., many SOLVERs stay idle until the ramp-up finishes. We note, that racing ramp-up has already been considered in Mitra et al. (1997) and Nwana et al. (2004). In order to use it with state-of-the-art MIP solvers on large scale distributed computing environments, our extended implementation can switch to the second stage seamlessly without waiting for all SOLVERs to terminate the first stage and can switch to normal ramp-up adaptively. Although the racing stage in racing ramp-up, which can be considered as a "learning" or "tuning" process, is a

sequential process, all SOLVERs become active immediately after the LOADCOORDINATOR finished presolving. There are many ways when and how to choose the winner and how to distribute the branch-and-bound nodes of the winner. The racing ramp-up configuration of FiberSCIP will be explained in detail in the beginning of Section 4.4.

Load balancing for MIP solving highly depends on the *primal* and *dual bounds* (*upper* and *lower bounds* on the optimal value, since the presolved instance is always a minimization problem), which are updated during the solving process. The primal bound is given by the value of the best solution that has been found so far during the solving process. If one of the SOLVERs finds an improved solution, this solution is sent to the LOADCOORDINATOR, which distributes the updated primal bound or the new solution to all other SOLVERs, which will then immediately apply bounding, hence, prune all nodes in its search tree that cannot contain any better solution anymore, which is proven by the node dual bound.

Periodically, each SOLVER notifies the LOADCOORDINATOR about its number of unexplored nodes and the dual bound of its subtree; we call this information the *solver status*. The dual bound is a proven lower bound on the objective function value of the best solution in that subtree. It is derived from the linear programming relaxations at the individual nodes. At the same time, the SOLVER is notified about the lowest dual bound value of all nodes in the node pool of the LOADCOORDINATOR, which we will refer to as BESTDUALBOUND. Note that this does not include nodes that are currently processed by any SOLVER.

If a SOLVER is idle and the LOADCOORDINATOR has unprocessed nodes available in the node pool, the LOADCOORDINATOR sends one of these nodes to the idle SOLVER. To handle situations in which several solvers become idle at the same time, the LOADCOORDINATOR should always have a sufficient amount of unprocessed nodes left in its node pool. This ensures that the SOLVERs are kept busy throughout the computation, thus minimizing idle time. In order to keep at least p "good" nodes in the LOADCOORDINATOR, we introduce the *collecting mode*, similar to the one introduced in Shinano et al. (2008). We call a node *good*, if the dual bound value of its subtree (NODEDUALBOUND) is close to the dual bound value of the complete search tree (GLOBALDUALBOUND).

If the LOADCOORDINATOR is not in collecting mode and it detects that less than p good nodes with

$$\frac{\text{NODEDUALBOUND} - \text{GLOBALDUALBOUND}}{\max\{|\text{GLOBALDUALBOUND}|, 1.0\}} < \text{THRESHOLD} \tag{3}$$

are available in the node pool, the LOADCOORDINATOR switches to collecting mode and requests selected SOLVERs that have nodes which satisfy (3) to switch also into collecting mode. The selection is done in ascending order of the minimum lower bound of open nodes in the SOLVERs. The number of selected SOLVERs increases dynamically depending on the time that the node pool in the LOADCOORDINATOR has been empty. If the LOADCOORDINATOR is in collecting mode and the number of nodes in its pool that satisfy (3) is larger than $m_p \cdot p$, it requests all collecting mode SOLVERs to stop the collecting mode. Note that m_p is a runtime parameter and is set to 1.5 as a default value.

If a Solver receives the message to switch into collecting mode, it changes the search strategy to either "best estimate value order" or "best bound order"(see Achterberg (2007)) depending on the specification of UG run-time parameters. It will then alternately solve nodes and transfer them to the LoadCoordinator. This is done until the Solver receives the message to stop the collecting mode. If a node of the branch-and-bound tree is selected to be sent to the LoadCoordinator, the corresponding Solver collects the bound changes of that node w.r.t. the presolved instance, transfers the differing bounds to the LoadCoordinator, and prunes the node from the subproblem's branch-and-bound tree.

The most crucial issue for the load balancing mechanism is to avoid solving useless subproblems. Consider the situation that a Solver is solving a subproblem N for which the dual bound is already quite large. An improvement in the primal bound will then cause all nodes of this subproblem to be pruned. The Solver can detect this situation locally using the best dual bound value of all nodes in the node pool of the LoadCoordinator (BestDualBound). In this situation, the Solver requests another node from the LoadCoordinator while still continuing to solve the current node. After the LoadCoordinator sent a new node to the Solver and restored the solving node N in its node pool, the Solver stops the solution process and restarts with the new node. The solution of node N is "delayed". Note that in case that there is no node available in the LoadCoordinator, the Solver keeps continuing to solve node N. Further, note that the Solver takes a coordinating role here. Reasons for this exceptional behavior is that, first, the Solver can detect such a situation earlier than the LoadCoordinator, which has only delayed information, and, second, that it simplifies the process in cases where many Solvers get into a similar situation.

The termination phase starts when the LoadCoordinator detects that the node pool is empty and all Solvers are idle. In this phase, the LoadCoordinator collects statistical information from all Solvers and outputs the optimal solution and the statistics.

3.3 Difference between ParaSCIP and FiberSCIP

As already mentioned, the fundamental difference between ParaSCIP and FiberSCIP is the parallelization library. However, also the initialization of the Solvers with the presolved instance is different. While for FiberSCIP a pointer to the SCIP environment can be moved between threads easily and the problem copying functionality of SCIP can be used, for ParaSCIP the presolved instance needs to be either transferred from the LoadCoordinator in serialized form or the original problem needs to be read from file and presolved on the Solver side. Therefore, extensions of SCIP by user-defined constraint handlers can be used straightforwardly in FiberSCIP, while for ParaSCIP some additional effort may be required.

3.4 The point of communication in ug [SCIP, *]

We define a *communication point* as a point in the base solver's algorithm, in which the LoadCoordinator can communicate with the Solver by sending a message, in other

words, the Solver can receive (and process) a message from the LoadCoordinator. Originally, the communication point was limited to the time when a branch-and-bound node was selected for processing. That means, at that point in time, the decision has to be made whether this node is solved or passed to the LoadCoordinator. Note that this is the only time during branch-and-bound node processing where the LoadCoordinator can communicate with a Solver by sending a message, e.g. to interrupt the current solving process. However, it also means that no communication between the LoadCoordinator and a Solver can happen during the processing of a branch-and-bound node, so that vivid response to a request message from the LoadCoordinator may not be possible. This delay can sometimes lead to useless computations. For example, when the LoadCoordinator has decided a winner Solver during racing ramp-up, it sends a request message to interrupt the computation in all Solvers other than the winner. As the Solvers delay reception of these messages, they first finish processing of their current branch-and-bound node before discarding their current subproblems.

To allow for more frequent communication, we added additional communication points. These are after every variable bound change, every solve of the LP relaxation, every modification of the LP, or finding of a new incumbent. This leads to very different communication patterns between the LoadCoordinator and Solvers compared to previous implementation. Now, expensive node solving may be interrupted earlier due to pruning.

3.5 Implementation of deterministic parallelism

When a user of UG has a brilliant idea for improving performance, e.g., by transferring more information inside a ParaNode, FiberSCIP is a suitable development environment due to its easier debugging possibilities. However, non-deterministic behavior can still make testing and debugging difficult and inefficient. Even though technology like deterministic multiprocessor replay, which can be used to debug non-deterministic programs, is getting more practical (Montesinos et al., 2009), it fundamentally still needs too much resources to apply it to general MIP/MINLP solvers. Therefore, adding the possibility to execute FiberSCIP deterministically is an essential feature for debugging.

Another motivation for implementing a deterministic parallelism in UG is the hope that computational experiments in deterministic mode can be used to predict performance improvements in non-deterministic execution mode. Note, that in the development of SCIP, even small improvements in geometric mean running time for a set of benchmark instances are, in general, crucial to accept a new feature. If repeated runs of FiberSCIP with the same parameter settings and the same computing environment but non-deterministic behavior produce computational results with more than, say, 10% variation in geometric mean, we have to evaluate the effectiveness of the new feature by averaging the value of several repeated runs. If a deterministic execution could be used to predict the average running time of several non-deterministic runs, the amount of necessary computational experiments could be reduced. However, we note that also with deterministic behavior, several repeated runs of a solver, each using a different random seed or permutation of

variables and constraints, on a set of instances may be necessary to get a clear picture of the performance improvement that can be expected by a new feature (Koch et al., 2011).

Deterministic parallelism is also one of the most important features of commercial optimization solvers, because a user of a solver expects that repeated executions on the same computer generate the same results. Therefore, commercial optimization solvers such as Cplex, Gurobi, and Xpress have a deterministic option, which is also enabled by default. For all of these solvers, deterministic parallelism is implemented based on their shared memory computing environments. However, since UG is intended to run also on distributed memory computing environments, a new mechanism to implement deterministic parallelism had to be introduced. As UG has an inherent non-deterministic behavior, its deterministic option realizes a *weak* deterministic parallelism (Olszewski et al., 2009), that is, the program is still executed in a non-deterministic way, but it ensures that repeated runs with the same parameter settings and on the same computing environment generate always the same search tree and give the same optimal solution.

In UG, all communications are done between the LoadCoordinator and the Solvers. Therefore, all messages are serialized in the LoadCoordinator. A basic idea to implement deterministic parallelism is to ensure that messages between LoadCoordinator and Solvers are in the same order in repeated runs. One tool needed to realize this deterministic behavior is a *deterministic timer*, which is called deterministic clock in Olszewski et al. (2009), and which is a counter to indicate progress of computation in a Solver. Ideally, intervals of the counter are distributed almost uniformly with respected to wall-clock time. However, since it is difficult to have such an ideal deterministic timer with SCIP, a counter for the number of communication point calls is used as deterministic time. Although the intervals of the calls are not uniform, it is called frequently. Another tool is a *token* that is circulated among Solvers via the LoadCoordinator in a predefined order, Solvers can execute until reaching specific points of time in deterministic time. If the Solver reached such a point, it has to wait until it receives the token before it can send messages and hands over the token to the next Solver via the LoadCoordinator. If an idle Solver receives the token, but has no messages to send, it passes it on immediately.

This token circulation mechanism is similar to the one of Liu et al. (2011). We note the following points: First, the token is not sent to the next Solver directly, but it is sent as an ordinary message via the LoadCoordinator, thus it does not require a shared memory architecture. Second, the general parts of token passing and deterministic timer are implemented in the UG framework, while the actual deterministic timing mechanism is specialized for each solver.

4 Computational results

FiberSCIP is intended to run on a shared memory computing environment with small scale parallelization and only a few hours of computing time. Therefore, all computational experiments conducted in this paper have a two hours time limit. The objective of the computational experiments is not only to analyze the performance of FiberSCIP, but also

to point out difficulties of performance evaluation.

At first, we analyze the fundamental overhead of FiberSCIP compared to SCIP and give baseline sequential computational results for comparison. Second, in order to clarify the effects of using several SOLVER threads, we present computational results when omitting parallel tree search. Finally, we evaluate our parallel tree search computational results for solving MIPs and MINLPs with four and eight SOLVER threads. For MIP, we also present results when using deterministic parallelism.

For all our experiments we present the following numbers:

- The "geometric mean (solved: n)" is computed w.r.t. the n instances that are solved in all runs within one experiment. It is given for the number of search nodes and for the running time.
- The "geometric mean (all)" of time is computed w.r.t. all instances, where aborts and timeouts are accounted with the time limit of 7200 seconds. Note, that we omit this measure for the number of search nodes, since it is not clear how aborted instances should be accounted.
- The "solved/timeout/abort[5] rows shows the number of solved instances, the number of instances that hit the time limit, and the number of aborted instances.
- The "speedup (solved: n)" is computed w.r.t. the n instances which are solved in all runs within one experiment.

The detailed computational results are given in the Appendix.

4.1 Setup for computational experiments in MIP solving

Computational experiments were conducted on 40 PowerEdgeTM 2950 computers, each equipped with two Quad-Core Xeon E5420 CPUs at 2.5 GHz and 16 GB RAM.

The 87 instances of the benchmark set of MIPLIB2010 (Koch et al., 2011) are used for the computational experiments. Table 8 in Appendix shows for each instance the total number of variables, the number of constraints, and the number of variables of certain type in the original and presolved problem. For the latter, the table also shows the product of the number of variables and number of constraints, which will serve as a rough and simple estimate for the certain memory usage.

We used SCIP version 2.1.1 in default settings, except that we set the time limit in the Chvátal-Gomory separator to infinity to prohibit a non-deterministic behavior of this SCIP version[6]. In order to standardize memory usage of each SOLVER thread, we assume 2GB of RAM for each SOLVER thread and therefore set the SCIP option `limits/memory` to 1600 (the memory usage of the LP solver cannot be measured). As a consequence, SCIP switches

[5] Aborts are mainly due to running out of memory, e.g., for instances with a high memory estimate by #vars*#conss, or when huge search trees were generated for some SOLVER's parameter settings.

[6] SCIP 3.0 has this option set to infinity by default.

Table 1: Comparing the sequential executions of SCIP and FiberSCIP. Instancewise results are shown in Table 10.

Summary	SCIP		FiberSCIP			
			Presolve Once		Presolve Twice	
	Nodes	Time	Nodes	Time	Nodes	Time
geom. mean (solved: 56)	8711	623.5	8718	623.7	9204	669.5
geom. mean (all)	–	1466.2	–	1467.9	–	1554.7
solved/timeout/abort	58/22/7		59/20/8		58/21/8	
speedup (solved: 56)	reference		1.000		0.931	

the search strategy to depth first search when the memory usage of SCIP (excluding the LP solver) reaches 80% of 1.6GB.

4.2 Fundamental overhead of FiberSCIP

Even if FiberSCIP runs with only one Solver thread, the original instance is presolved twice in default settings, once in the LoadCoordinator and once in the Solver. In order to clarify the communication overhead between the LoadCoordinator and the Solver, we also conducted computational experiments where presolving in the LoadCoordinator was prohibited.

Table 1 summarizes the computational results of sequential runs executed by SCIP, FiberSCIP without presolving in the LoadCoordinator, and FiberSCIP with double presolving (in the LoadCoordinator and in the Solver). Figure 2 visualizes the speedup of the one thread FiberSCIP runs w.r.t. SCIP for all solved instances (56 in total). We see, that FiberSCIP with one presolve usually takes the same computing time as SCIP. The number of nodes processed by SCIP and that of FiberSCIP with one presolve is expected to be completely the same. However, Table 10 shows differences for only three instances (`timtab1`, `triptim1` and `zib54-UUE`). As FiberSCIP installs additional plugins into SCIP (e.g., event handler), its memory usage is slightly different, which seems to cause this difference. That is, due to different amounts of allocated memory, computations that get close to the specified memory limit may change the node selection strategy to depth first search at different points of the search. FiberSCIP (with single presolving) took basically the same computing time in geometric mean, which indicates that there is almost no communication overhead between the Solver and the LoadCoordinator. Note that FiberSCIP solves one more instance (`zib54-UUE`), shortly before hitting the time limit (Table 10). Analyzing this issue more carefully reveals that FiberSCIP changes to depth first search earlier due to reaching the memory limit (see Section 4.1), which is resulting in a better performance here.

Figure 2 also shows that FiberSCIP with double presolving works different to SCIP, one can observe both speedups and slowdowns depending on the instance. These differences result from the additional presolving in the LoadCoordinator and therefore the Solver gets a changed instance. This leads to a 7% increase in the geometric mean of the computing

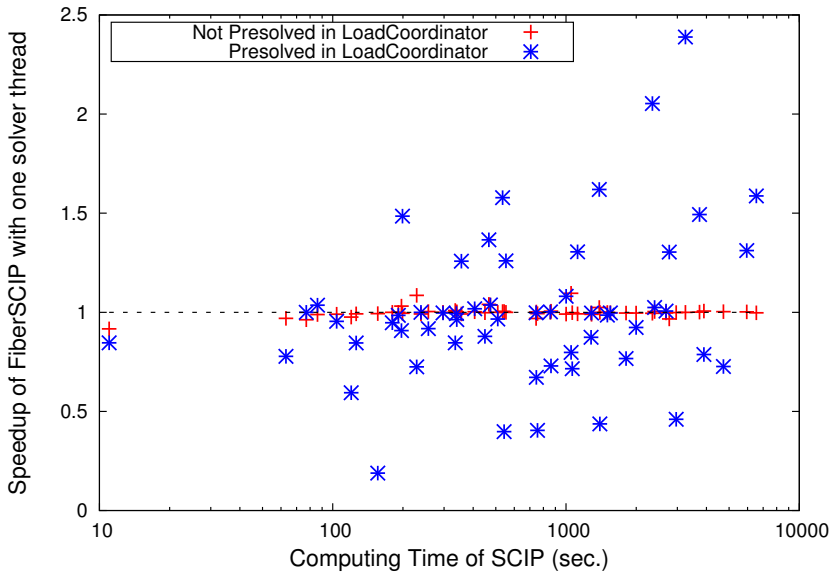

Figure 2: Computing time of FiberSCIP relative to SCIP.

time when using FiberSCIP with double presolve compared to SCIP using one thread. But to avoid repeating the same preprocessing steps multiple times when using multiple SOLVER threads and also to keep the memory consumption as small as possible, double presolving was decided to be the default setting in FiberSCIP. Using it with one SOLVER thread will serve us as the baseline for comparisons with the runs that use multiple SOLVER threads.

4.3 Omitting parallel tree search

We conducted two computational experiments running FiberSCIP in racing stage throughout the whole computation, meaning that the ramp-up phase is never left and thus the parallel tree search is omitted. All computational experiments were conducted in non-deterministic mode. First, we were disabling communication of the incumbents, whereas it was enabled in the second experiment. The SCIP parameter settings for each racing SOLVER is a combination of the emphasis settings "off", "fast", "default", and "aggressive" for the different groups of components like primal heuristics, presolvers, and separators. Exactly one SOLVER ran with SCIP's default settings.

4.3.1 Racing without communication of incumbents

In this subsection, we are evaluating the possible speedups when omitting communication of incumbents. Independent sequential SOLVER threads ran in parallel with different parameter settings. Theoretically, an increase in the number of threads should result in a better overall performance w.r.t. the running time and the number of solved instances. The choice of the winner is basically deterministic, though several highly competitive SOLVERs could introduce non-determinism.

Table 2: Comparing the sequential execution of FiberSCIP with different numbers of Solver threads in racing stage without communication of incumbents. Detailed results are shown in Table 11 in the Appendix.

	One thread		Two threads		Four threads		Six threads		Eight threads	
Summary	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
geom. mean (solved: 51)	9817	573.9	8148	529.7	6935	507.5	6386	569.4	5905	672.0
geom. mean (all)		– 1554.6		– 1460.1		– 1434.4		– 1564.4		– 1723.6
solved/timeout/abort	58/21/8		60/20/7		60/21/6		59/23/5		56/27/4	
speedups (solved: 51)	reference		1.084		1.131		1.008		0.854	

Table 2 summarizes the results for racing without communication of incumbents when using one (which gives the baseline), two, four, six, and eight threads. From an algorithmic point of view, a higher number of independent Solver threads should lead to a monotonous decrease in minimal computing time and an increase in solved instances, but the results for using six threads show nearly the same overall performance in speed, i.e., a factor of only 1.008, and the number of solved instances compared to one thread. For eight threads we even encounter less solved instances and a slowdown by a factor of 0.854. Even though the Solver threads do not interact logically with each other, their actual parallel execution in a single machine increases the number of context switches by the operating system and can lead to more frequent major or minor page faults, cache misses, and collisions in memory access. We note that hard disk access for swapping (major page faults) has not been observed. In the following, we aim to clarify these effects further.

In order to investigate the change in performance of the winner solver, we selected the results of the 12 instances where the number of nodes processed is the same in all runs with one, two, four, six, and eight threads (marked by '(i)' in Table 11). We assumed that an increase in memory access latency due to using more threads on the same machine is mostly responsible for the slow down effect. The latency is related to the fundamental memory usage of the solver. As an indicator of the memory usage, we use the product of number of variables and number of constraints of the presolved instance, see Table 8, as a matter of convenience, because it is easy to obtain from SCIP and does not change during the solving process (as the actual memory usage does and cannot be obtained accurately from SCIP). Figure 3 shows the number of minor page faults for each number of Solver threads, left, and the number of context switches for each number of Solver threads, right, in relation to the memory usage indicator for the 12 instances. While these numbers do not seem to be related to the memory usage indicator, they usually increase with the number of threads. Note that for better visibility we connected data points belonging to the same number of threads.

The left picture in Figure 4 illustrates the speedups for the 12 selected instances. Except for instance `ex9` at the far left side of the graph, only slowdowns can be observed when adding more Solver threads. The reason is the increase in the number of minor page faults and context switches.

The right picture in Figure 4 illustrates the speedups for the other 39 instances that

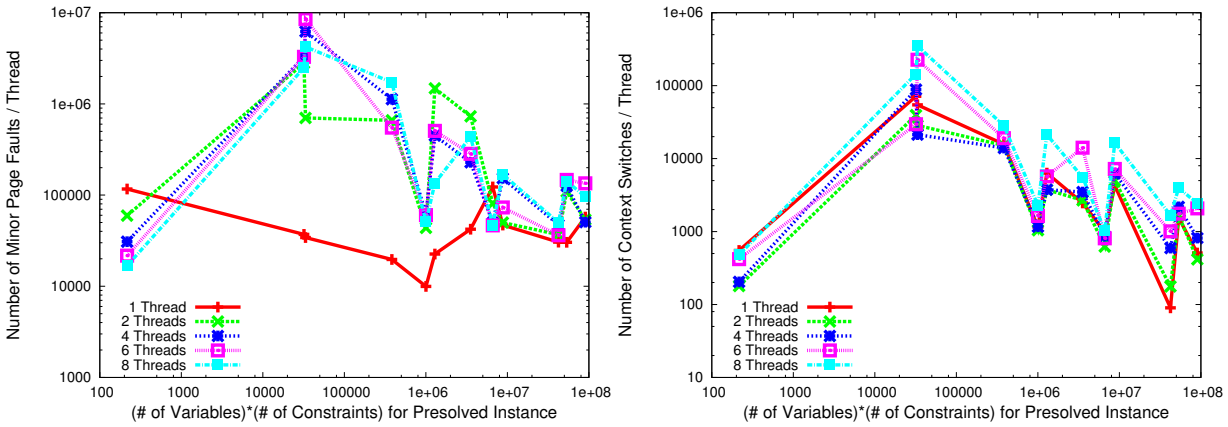

Figure 3: Minor Page faults (left) and Context switches (right) of FiberSCIP runs.

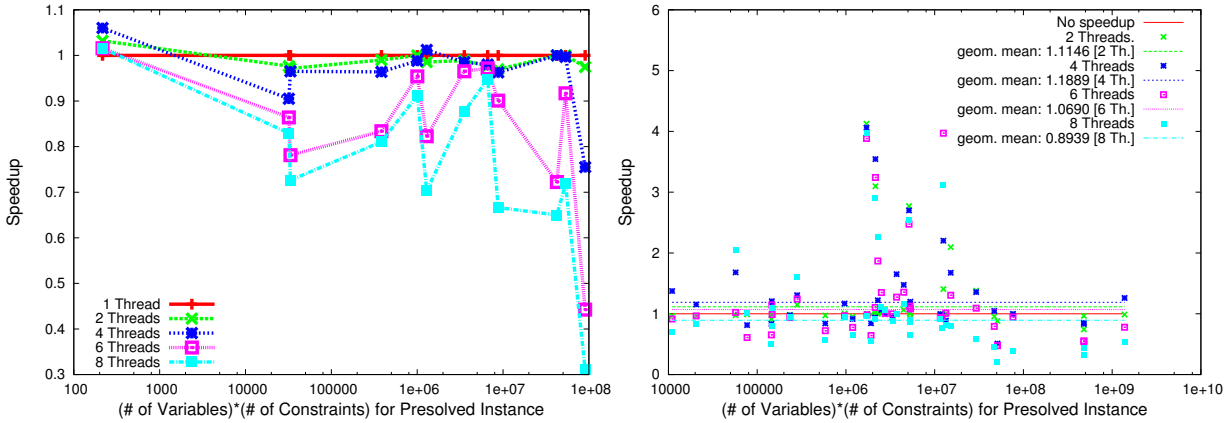

Figure 4: Computing time ratio of FiberSCIP for different numbers of SOLVER threads w.r.t. one thread for, first, the twelve instances in which the winner solver did not change (left) and, second, the 39 instances in which the winner solver changed (right).

were solved in all runs. For these 39 instances, the winner SOLVER changed when more SOLVER threads were added. If the performance of each SOLVER thread would remain the same when changing the amount of solver threads, a speedup should be observed for each instance. For each run, also the geometric mean of the speedups is shown in the figure. We see that in the case of four SOLVER threads, the best speedup can be observed, while more than four SOLVER threads lead to a slow down.

4.3.2 Racing with communication of incumbents

In contrast to the last subsection, we now enabled the communication of incumbents between SOLVER threads and LOADCOORDINATOR. We expect an overall performance improvement in running time. As communication of incumbents can lead to a non-deterministic behavior, five repeated runs were conducted for each number of threads.

Table 3: Comparing FiberSCIP in racing stage with and without communication of incumbents. Detailed results are shown in Tables 12, 13, 14, and 15 in the Appendix.

(a) Four Solver Threads

	Without comm.		Average			One thread		Average	
Summary	Nodes	Time	Nodes	Time	Summary	Nodes	Time	Nodes	Time
geom. mean (solved: 57)	7110	620.7	6725.4	591.69	geom. mean (solved: 56)	9173	670.3	6116.8	582.89
geom. mean (all)	–	1434.4	–	1379.54	geom. mean (all)	–	1554.6	–	1379.54
solved/timeout/abort	60/21/6		62.2/19.4/5.4		solved/timeout/abort	58/21/8		62.2/19.4/5.4	
speedups (solved: 57)	reference		1.049		speedups (solved: 56)	reference		1.150	

(b) Eight Solver Threads

	Without comm.		Average			One thread		Average	
Summary	Nodes	Time	Nodes	Time	Summary	Nodes	Time	Nodes	Time
geom. mean (solved: 52)	5128	706.1	4470.2	605.81	geom. mean (solved: 49)	9226	555.2	4781.2	552.68
geom. mean (all)	–	1723.6	–	1551.98	geom. mean (all)	–	1554.6	–	1551.98
solved/timeout/abort	56/27/4		57.2/25.4/4.4		solved/timeout/abort	58/21/8		57.2/25.4/4.4	
speedup (solved: 52)	reference		1.166		speedups (solved: 49)	reference		1.005	

The left-hand side of Table 3 shows the results of only racing with communication of incumbents for four and eight Solver threads, respectively. In both tables, the results of only racing without communication of incumbents for the same number of Solver threads are set as a baseline to clarify the effect of communication. The runs with four Solver threads always solved more instances and lead to an improvement of the geometric mean over all solved instances. The runs with eight Solver threads did not always solve more instances, but the geometric mean running time decreased always. The speedups w.r.t. solved instances are always larger when compared to the results with four Solver threads. However, the mean speedups are calculated w.r.t. 57 instances in case of four Solver threads and only 52 instances in case of eight Solver threads. As long as an instance is solved in the time limit, more Solver threads reduce the solving time, since communication of a newly found incumbent in one Solver may allow to prune branch-and-bound nodes in the search trees of other Solvers.

The right-hand side of Table 3 summarizes the same results for four and eight Solver threads, respectively, except that the baseline result is replaced by the result for one Solver thread (FiberSCIP with double presolving, see Table 1). Communicating of incumbents lead in both cases to a decrease of the geometric mean running time and for four Solver threads also to an increase in solved instances. As expected, the performance increases when comparing the results against the runs without communication. For eight threads, we can even overcome the slowdown due to context switches and page faults.

Tables 16 and 17 in Appendix show the variations in the number of enumerated nodes and computing times of repeated runs for each instance that was solved. The numbers show a high variation for repeated runs with communication of incumbents. The maximal deviations for nodes is nearly 190%, while it is around 93% for time. This indicates, that doing several repeated runs of a non-deterministic algorithm is crucial in order to measure

its performance.

4.3.3 Overall results on racing

Our results illustrate some of the difficulties in doing a parallel implementation which uses all computing resources efficiently. Note that, without communication of incumbents, the speedups are purely due to the different parameter settings in the different solvers, thus, essentially, due to running different algorithms in parallel and not due to parallelizing a single algorithm. Tables 2 and 3 show that for four Solver threads, the number of solved instances increases by 4.2 and the speedup is 1.150 on average when communicating the incumbents, while they were only 2 and 1.131, respectively, when disabling the communication. For eight Solver threads, the number of solved instances decreases by 0.8 and the speedup is 1.005 in average when communication is enabled, while they were 2 and 0.854, respectively, when using no communication. As expected, only if the gain of parallelization exceeds the decline in a Solver's performance due to the addition of more threads, parallelization can be beneficial.

4.4 Parallel tree search for MIP solving

We conducted computational experiments using normal and racing ramp-ups for four and eight Solver threads. Therefore, for the same number of Solver threads, two types of settings were tested. Since the non-deterministic mode is the default in FiberSCIP, for each setting five repeated runs were conducted. For comparison purpose, additional deterministic runs were performed using four and eight Solver threads (only one run for each setting).

Racing ramp-up is a distinguished feature of UG. Many variations are possible by configuring how the parameter sets are chosen in each Solver and how the racing stage is terminated. The configuration of FiberSCIP as used in this paper is as follows. The SCIP parameter settings for each racing Solver are the same as in Section 4.3. Termination of the racing stage is decided based on the performance of all Solvers. When the LoadCoordinator recognizes that

1. at least for half of the Solvers it received the first Solver status message[7],
2. a feasible solution has been found,
3. the candidate has more than 300 open nodes, and
4. the elapsed computation time has exceeded 36 seconds (0.5% of our 2 hours time-limit),

a Solver with best dual bound value is chosen as candidate for the winner Solver. Conditions 1 and 4 are essential, but if Condition 2 is not satisfied, the racing stage is continued until a solution is found, the problem is proven to be infeasible, or the time limit

[7]It is not unusual that a Solver sends a first status message late when some of the emphasis settings has been set to "aggressive".

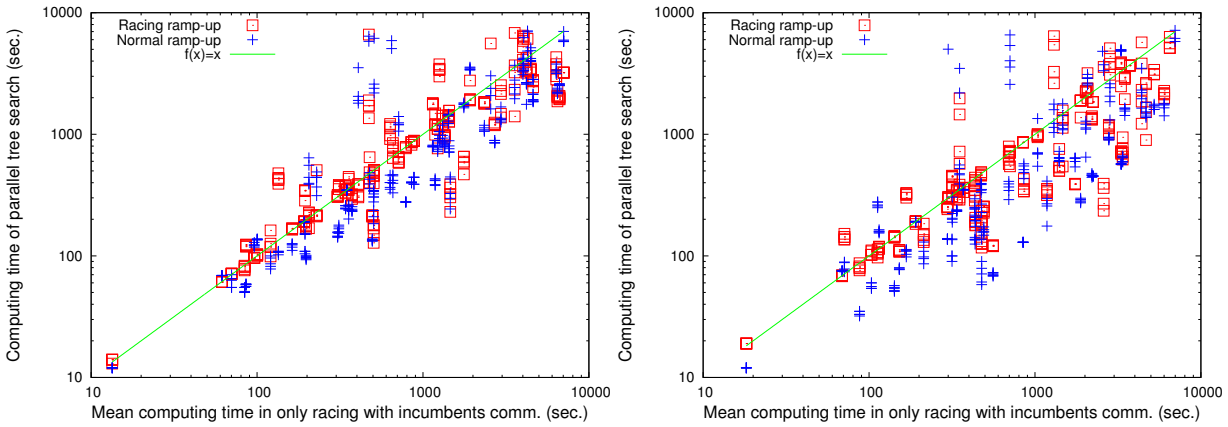

Figure 5: Relationship between only racing with communication of incumbents and racing followed by parallel tree search for four SOLVER threads (left) and eight SOLVER threads (right).

is reached. If only Condition 3 is not satisfied, the racing stage is terminated when the elapsed time of computation exceeds 1800 seconds (25% of our 2 hours time-limit). The parameters of this termination criteria can be specified via UG options. Note that, in case of runs in deterministic mode, the time limits for the above settings apply to the deterministic timer, though this time is not accurately emulating the walk-clock time.

Tables 18–28 in the Appendix show all computational results for MIP solving for all settings. For non-deterministic runs, we conducted five repeated runs and observed higher variation than in the racing with communication of incumbents experiments from Section 4.3, even when all SOLVER settings were the same.

Figure 5 shows the computing times for all parameter settings with respect to the mean computing time when doing only racing with communication of incumbents. The computing time for racing only is considered as an execution without parallel tree search, because the winner's search tree is generated by a single SOLVER thread. Figure 5 shows that the effectiveness of racing to choose a good parameter setting is not strong enough to overcome the advantages of a parallel tree search in our current implementation. Further, it shows that parallel tree search should be used after the winner SOLVER has been chosen, since the combination of racing and parallel tree search leads to better performance in many cases, especially when computing time increases. This trend is even better visible when the number of threads increases. Note that the plot also includes results for settings that are considered disadvantageous for FiberSCIP.

Table 4 summarizes the speedups and improvements in number of solved instances for each setting, including those for racing only from Section 4.3. The speedups for n SOLVER threads are calculated w.r.t. the sequential run with double presolve as baseline as follows:

$$\text{Speedup}(n) = \frac{\text{Geometric mean of computing times of FiberSCIP with } n \text{ SOLVER threads}}{\text{Geometric mean of computing times of FiberSCIP with 1 SOLVER thread}}$$

Since the number of solved instances changes among several runs with the same setting, we

Table 4: Speedups - Nondeterministic and deterministic runs. 87 instances in total, 58 solved by a sequential FiberSCIP run (double presolve). Detailed results are shown in Tables 18–28 in the Appendix.

	Non-deterministic						Deterministic		
	# solved				Speedup		# solved	Speedup	
Settings	Share	Min	Max	Average	Share	Average	Single	Share	Value
Four th. only racing with comm.	60	61	63	62.2	56	1.150	48	46	0.52
Four th. racing ramp-up	62	64	66	64.4	53	1.261	50	46	0.47
Four th. normal ramp-up	59	64	65	64.6	52	1.668	49	47	0.64
Eight th. only racing with comm.	53	56	59	57.8	49	1.005	47	45	0.35
Eight th. racing ramp-up	60	64	67	65.4	51	1.402	52	47	0.37
Eight th. normal ramp-up	58	62	64	63.0	50	1.898	48	47	0.61

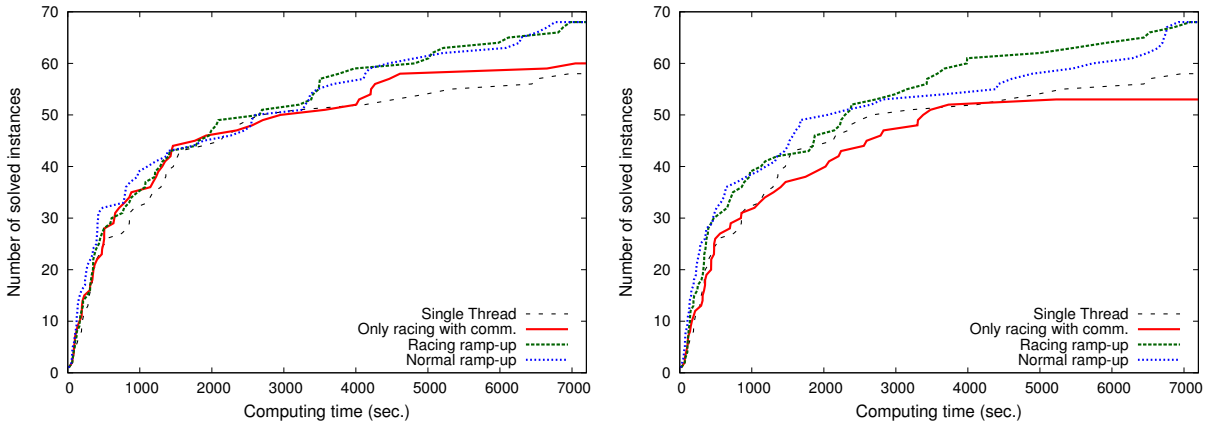

Figure 6: Profiles of number of solved instances w.r.t. solving time for four SOLVER threads (left) and eight SOLVER threads (right).

print the number of instances only solved by all multi-threaded runs and the number of instances solved by all multi-threaded runs together with the sequential run in the "Share" columns. The speedups are calculated w.r.t. the instances solved by the sequential and the multi-threaded runs. For each run, we show the minimum, maximum, and average number of instances solved by a multi-thread run. Since most of these numbers are higher than the number of instances solved by the sequential run (58 many), most of the multi-threaded runs solved more instances than the sequential run. Especially for the racing ramp-up settings, the number of instances solved by all multi-threaded runs of one setting are usually higher than the number of instances solved by the sequential run. However, for almost all settings, some instances that could be solved by the sequential run were not solved by at least one of the multi-threaded runs.

Whether normal or racing ramp-up is preferable is not clear. The normal ramp-up shows overall best average speedups but racing ramp-up dominates in the number of solved instances especially when eight SOLVER threads are used. Also the increase in the average number of additionally solved instances when increasing the number of SOLVER threads is larger for racing ramp-up (6.4 for four threads to 7.4 for eight threads) than for normal

ramp-up (6.6 for four threads to 5.0 for eight threads). A reason might be, that with only a few SOLVER threads, the effects of tuning w.r.t. a different parameter setting for each thread might not be as strong as when having more threads available. In order to clarify the performance differences, we generated profiles of the number of solved instances w.r.t. computing time, see Figure 6, where in case of five repeated runs average values were used. If an instance could not be solved, but a timeout or abort occurred, the solving time was accounted with 7200s. These cases are responsible for "jumps" in the graphs near the time limit, e.g., for the runs with eight SOLVER thread and normal ramp-up. They indicate that the corresponding configuration makes the solver unstable. For the runs with four SOLVER threads, racing ramp-up and normal ramp-up seem competitive. In contrast, for eight SOLVER threads, racing ramp-up even dominates normal ramp-up. These trends supplement the conclusions we have drawn from Table 4 and are the reason why racing ramp-up is the default setting of FiberSCIP.

We also performed deterministic runs to investigate if we can estimate the average results for non-deterministic runs for a specific setting. Such a possibility would be very useful, e.g., to conduct parameter tunings. Table 4 shows that, obviously, all deterministic runs have worse performance than that of a sequential run, especially when the number of threads increases. This is clearly because SOLVERs have to wait for the token and this waiting time increases when more SOLVER threads are added. Figures 7 and 8 show the relations between the deterministic and non-deterministic runs w.r.t. average number of enumerated nodes and average computing time for all instances solved by the deterministic runs. For the number of nodes, a good correlation can be observed, indicating the possibility to estimate the number of enumerated nodes in non-deterministic runs by those from deterministic ones. For the solving time, Figure 8 show no clear pattern for the changes in running times, that is, the accuracy is currently not sufficient to estimate the computing times of non-deterministic runs from that of deterministic runs. Figure 9 shows the correlation of relative computing time when doing racing ramp-up w.r.t. that of normal ramp-up between deterministic and non-deterministic runs. For non-deterministic runs, the average computing time of five repeated runs is used. If there were a good correlation, it would indicate that the relative racing ramp-up computing time w.r.t. that of normal ramp-up in non-deterministic runs is preserved when switching to deterministic runs. However, no correlations between them could be identified when looking at both pictures in Figure 9. Thus, deterministic runs cannot be used to evaluate a parameter setting for non-deterministic runs.

4.5 Sources of overhead

In Koch et al. (2012), general sources of overhead (reason for a loss of efficiency) are grouped into four categories: *Communication overhead*, *Idle time in ramp-up/ramp-down*, *Idle time due to latency/contention/starvation*, *Performance of redundant work*. It is not easy to identify and to measure the source of overhead clearly. FiberSCIP measures idle time as accurately as possible. Tables 19, 21, 23, 25, 27, and 29 in Appendix C show the fraction of solving time that the SOLVERs are idle, that is, do not spend on processing a subproblem. They do not include idle time due to latency/contention/starvation, which emerges as a

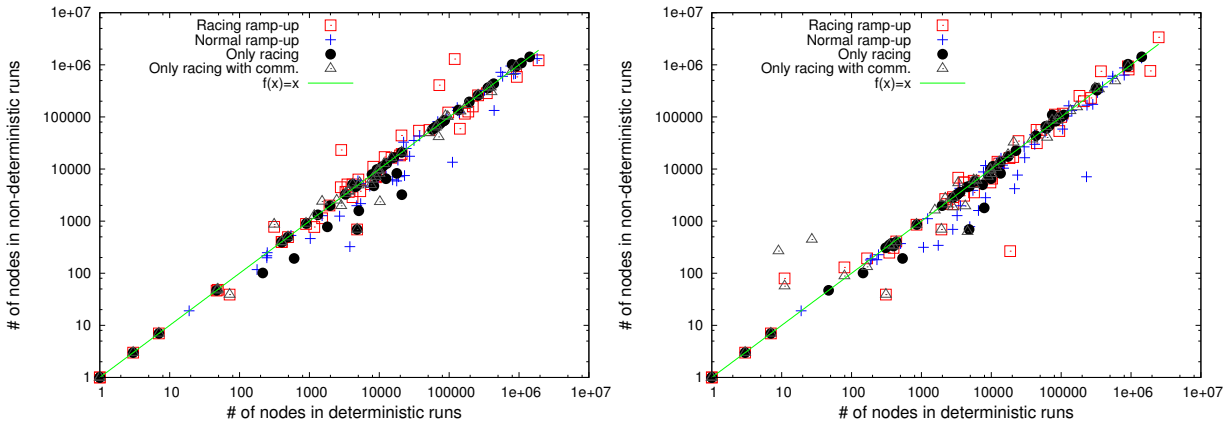

Figure 7: Relationship of average number of nodes between deterministic and non-deterministic runs for four SOLVER threads (left) and eight SOLVER threads (right).

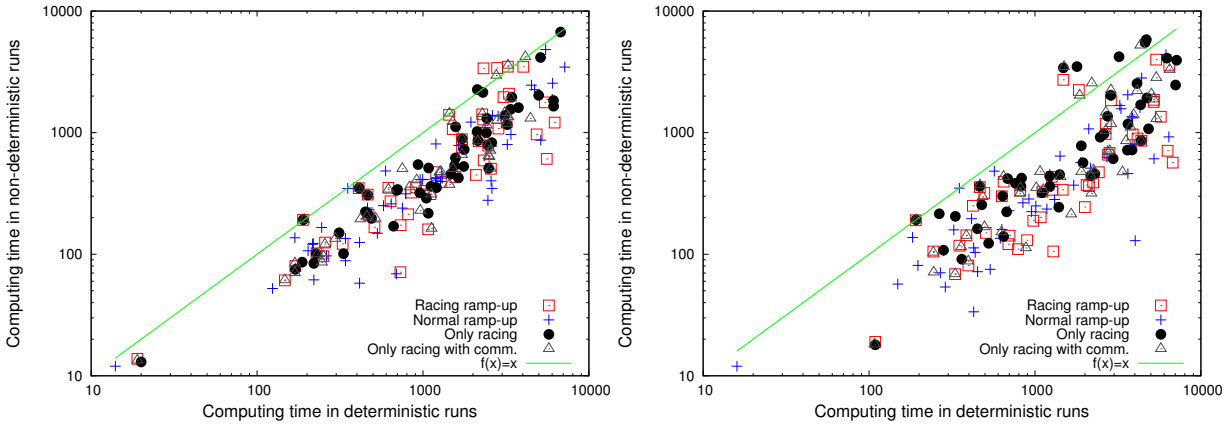

Figure 8: Relationship of average computing time between deterministic and non-deterministic runs for four SOLVER threads (left) and eight SOLVER threads (right).

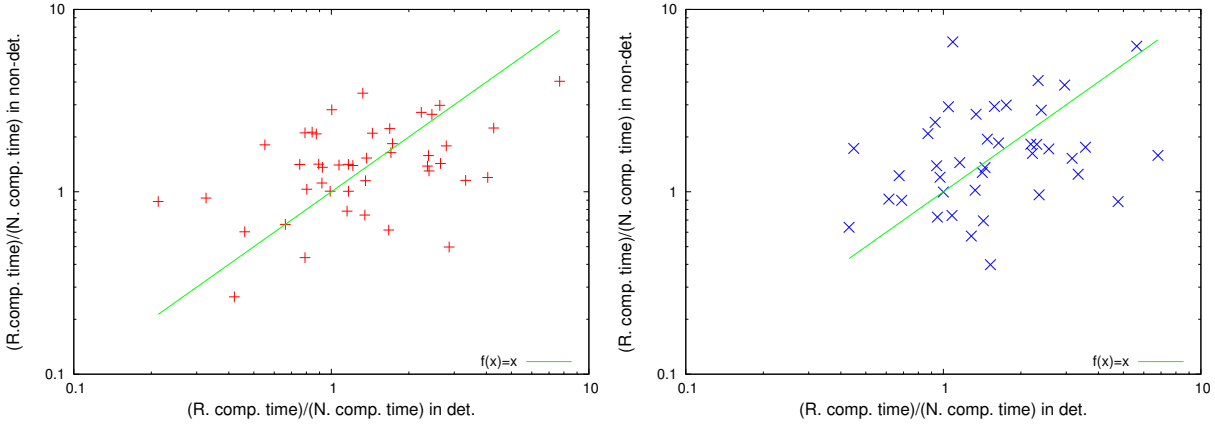

Figure 9: Relationship of relative computing time of racing ramp-up w.r.t. that of normal ramp-up between deterministic and non-deterministic runs for four SOLVER threads (left) and eight SOLVER threads (right).

Table 5: Comparing SCIP, ParaSCIP with seven SOLVER processes (distributed on seven computing nodes or sharing one node), and FiberSCIP with seven SOLVER threads in racing stage without communication of incumbents. Detailed results are shown in Table 32 in the Appendix.

	SCIP		ParaSCIP (Distributed)		ParaSCIP (Share)		FiberSCIP	
Summary	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
geom. mean (solved: 62)	9794	409.2	5421	271.2	5409	531.3	5335	695.6
geom. mean (all)	–	866.9	–	604.1	–	1056.2	–	1361.5
solved/timeout/abort	68/18/1		71/16/0		65/16/6		62/18/7	
speedups (solved: 62)	reference		1.51		0.77		0.59	

Table 6: Speedups - ParaSCIP and FiberSCIP nondeterministic runs on the same cluster node. 87 instances in total, 68 solved by SCIP. Detailed results are shown in Tables 33 and 34 in the Appendix.

	# solved				Speedup	
Settings	Share	Min	Max	Average	Share	Average
ParaSCIP with seven processes	63	69	70	69.4	63	1.25
FiberSCIP with seven threads	63	68	70	68.8	63	0.89

decline in performance of each SOLVER thread when more SOLVERs are added. This type of overhead is analyzed in the following section.

When the deterministic mode feature of FiberSCIP is used, the main source of overhead is clearly the time to wait for the token, which amounts to averagely 69.11 % of computing time with normal ramp-up and 60.88 % of that with racing ramp-up by using four SOLVER threads, and 81.53 % and 73.79 % respectively by using eight SOLVER threads. Therefore, the performance in deterministic mode cannot be compared with that in non-deterministic mode. From computing resource usage point of view, racing ramp-up is superior: For all computations, idle time with racing ramp-up is 0.65 % by using four SOLVER threads and 1.35 % by using eight SOLVER threads in average, while it is 10.72 % and 15.29 % respectively in average with normal ramp-up. In spite of this fact, the obvious performance differences between racing and normal ramp-ups cannot be observed from the computational results. The difficulty to identify the sources of overhead are caused by many trade offs that have to be taken into account. On the one hand, to learn additional information, racing ramp-up uses computing resources which would otherwise be idle in normal ramp-up. On the other hand, racing delays parallel tree search, as the racing stage can be considered to be a sequential tree search. Whether to do the additional presolve in the LOADCOORDINATOR is one of other trade offs. Further tuning of parameters, thereby taking these trade offs into account, is still necessary to minimize the sources of overhead.

4.6 Comparison of ParaSCIP and FiberSCIP

Tables 32, 33, 34 in Appendix C show a comparison between ParaSCIP and FiberSCIP. These tables have been added after the first review of this paper. Since the cluster used for the original experiments was not available anymore, the computational experiments for this comparison were performed on a cluster of Intel Xeon X5672 CPUs with 3.20 GHz (eight cores per node) and 48 GB RAM, running Kubuntu 14.04. We used SCIP Optimization Suite 3.1.1 and OpenMPI 1.6.5.

Table 5 shows the performance of ParaSCIP with seven Solver processes when doing racing without communication of incumbents and without presolve in the LoadCoordinator (to allow for a comparison with SCIP). We conducted two runs: 1) Each Solver process runs on a different cluster node, with an additional node for the LoadCoordinator. 2) All Solver processes and the LoadCoordinator run on the same cluster node. Additionally, we run SCIP and FiberSCIP with seven Solver threads on the same machine. In case 1), each Solver achieves the same performance as if SCIP was run with the corresponding settings. Therefore, three more instances were solved and a speedup of 1.51 is achieved when using seven Solvers on separate machines, compared to a run of SCIP in default settings. However, if all seven processes are run on the machine (case 2), three less instances are solved, when compared to SCIP, and a speedup of 0.77 on the solved instances can be observed. The comparison between ParaSCIP in the case 2) and FiberSCIP shows that one Solver thread in FiberSCIP takes about 1.3 times longer than that of ParaSCIP. Hence, FiberSCIP is slower than SCIP and solves six less instances compared to SCIP. Therefore, in this experiment (i.e., only racing without exchange of incumbents), running Solvers in separate processes gives better performance than as threads.

Table 6 shows computational results for ParaSCIP and FiberSCIP in default settings, that is parallel tree search with racing ramp-up and one presolve in the LoadCoordinator, when using seven Solvers on the same cluster node. Also here, the performance of FiberSCIP is worse than that of ParaSCIP and the performance decline is even worse than when doing racing only (Table 32), as there is now additional communication between the LoadCoordinator and Solvers. These results show that FiberSCIP could achieve better performance when the Pthreads parallelization part of code is improved. However, in spite of the fact that each Solver slows down about 2.5 times compared to SCIP and 1.3 times compared to ParaSCIP, FiberSCIP can solve almost the same number of instances as ParaSCIP and more instances than SCIP in three out of the five runs.

4.7 Computational results for MINLP solving

As the experiments for parallel MINLP solving required several fixes in SCIP and were done much later than the ones for MIP solving, we used SCIP 3.0 for the results on MINLP. Further, since we could not install SCIP with all required plugins for MINLP solving (in particular Ipopt) on the cluster where we conducted our MIP computations, we used a different computing environment with 8 cores divided into two Quad-Core Intel Xeon 5460 CPUs at 3.16 GHz and 48 GB RAM. We performed our test on the MINLPLib (Bussieck

Table 7: Speedups - Nondeterministic runs (MINLP). 86 instances in total, 27 solved by a sequential FiberSCIP run (double presolve). Detailed results are shown in Tables 30 and 31 in the Appendix.

	# solved				Speedup	
Settings	Share	Min	Max	Average	Share	Average
Four th. racing ramp-up (default)	28	28	29	28.4	26	1.780
Eight th. racing ramp-up (default)	26	27	30	29.0	23	2.779

et al., 2003) instance library. Even though this collection is not as well-balanced as the benchmark set of MIPLIB2010, it is commonly used for testing and comparing MINLP solvers. From MINLPLib (as of 02/17/2013), we took all instances except for those 12 instances that use the functions sin, cos, or erf, which cannot be handled by SCIP so far. Since the MINLPLib includes many trivial instances, we first run all remaining 252 instances with SCIP in sequential mode and then removed those that were solved within 60 seconds (135 instances), as we are interested here in measuring possible gains from running FiberSCIP on instances that are not already solved easily in sequential mode. Further, we removed instances where SCIP in sequential mode aborted (6 instances), reported a wrong optimal solution (9 instances), or could not compute a finite dual bound (16 instances). Table 9 in the Appendix shows for the remaining set of 86 instances the number of variables, number of constraints, and number of variables of certain types for the original and presolved problems. For the latter, we also distinguish between linear and nonlinear constraints.

We ran SCIP in sequential mode and FiberSCIP using racing ramp-up with four and eight Solver threads. The computing times are detailed in Tables 30 and 31 and summarized in Table 7. Out of the 86 selected instances from MINLPLib, SCIP solved 27 instances in sequential mode and hit the time limit on the remaining ones. With FiberSCIP, instance `netmod_dol1` was not solved anymore, but up to two additional instances could be solved when using 4 threads and up to three additional instances could be solved when using 8 threads. Unfortunately, we also noted aborts or wrong solutions on up to 10 (4 threads) or 12 (8 threads) instances. In some cases, aborts were due to running out of memory. Note, that when SCIP solves a MINLP, it generates many cutting planes during the branch-and-bound search. Storing these cuts not only for one but for 4 or 8 threads can cost a lot of memory.

Regarding speedup on the number of instances that were solved by SCIP and both runs of FiberSCIP, we observe average factors of 1.78 when using 4 threads and 2.779 when using 8 threads, which are larger speedups than observed for MIP. However, one should not assume that these numbers can be carried over to MINLP in general, since this set of solved instances is heavily dominated by the `fo*`, `no*`, and `o*` instances, which all have there origin in the same application.

5 Concluding remarks

This paper introduced FiberSCIP, a parallel extension of SCIP, realized via the UG framework. FiberSCIP has been designed to preserve as much of the flexibility of the SCIP framework as possible, that is, extensions to SCIP, e.g., in order to solve further classes of optimization problems, are readily available for parallelization in FiberSCIP. By linking the same code to ParaSCIP, these additional classes can also be handled on large scale distributed memory computing environments.

We have shown quite good performance improvements when using FiberSCIP, especially when considering that parallelization is realized from "outside" of SCIP, i.e., a developer or user of SCIP does have to take care of this parallelization. As for any non-deterministic parallel solver, difficulties emerge regarding evaluation of solver performance and parameter tuning. This paper discussed in detail the performance and behavior of FiberSCIP on challenging MIP and MINLP problems and described a possibility to evaluate parallelization.

There are still many open questions about how to parallelize state-of-the-art solving techniques in a most efficient form. Already racing ramp-up offers a variety of choices, like which parameter settings to use in each Solver thread, when to terminate the racing stage, and how to choose a winner Solver. How to measure the performance of each configuration of parameter settings for SCIP and UG in non-deterministic runs is another open question. To evaluate the performance of FiberSCIP, we conducted several (non-deterministic) runs for each setting, which required a lot of computing resources. Therefore, an open point for future development is the improvement of the deterministic mode in FiberSCIP, so that it can be used to estimate the average behavior of non-deterministic runs.

Acknowledgments

This research is partially supported by the DFG Research Center Matheon *Mathematics for key technologies* in Berlin. The work of Yuji Shinano was supported by a Google Research Grant. We would like to thank the HPC department at ZIB for access to the Alibaba cluster and Prof. Katsuki Fujisawa for access to his computers. The authors would like to thank the anonymous reviewers for his/her valuable comments and suggestions to improve the quality of the paper.

References

Achterberg, Tobias. 2007. Constraint integer programming. Ph.D. thesis, Technische Universität Berlin. urn:nbn:de:0297-zib-11129.

Achterberg, Tobias, Thorsten Koch, Alexander Martin. 2006. MIPLIB 2003. *Operations Research Letters* **34** 1–12. doi:10.1016/j.orl.2005.07.009.

Bendjoudi, Ahcene, Nouredine Melab, El-Ghazali Talbi. 2012. An adaptive hierarchical

masterworker (AHMW) framework for grids – application to B&B algorithms. *Journal of Parallel and Distributed Computing* **72** 120 – 131. doi:10.1016/j.jpdc.2011.10.002.

Berthold, Timo, Ambros M. Gleixner, Stefan Heinz, Stefan Vigerske. 2012. Analyzing the computational impact of MIQCP solver components. *Numerical Algebra, Control and Optimization* **2** 739–748. doi:10.3934/naco.2012.2.739.

Berthold, Timo, Stefan Heinz, Marc E. Pfetsch. 2009. Nonlinear pseudo-boolean optimization: relaxation or propagation? Oliver Kullmann, ed., *Theory and Applications of Satisfiability Testing – SAT 2009*. No. 5584 in Lecture Notes in Computer Science, Springer, 441–446. doi:10.1007/978-3-642-02777-2_40.

Berthold, Timo, Stefan Heinz, Stefan Vigerske. 2011. Extending a CIP framework to solve MIQCPs. Jon Lee, Sven Leyffer, eds., *Mixed Integer Nonlinear Programming*, *The IMA Volumes in Mathematics and its Applications*, vol. 154. Springer, 427–444. doi:10.1007/978-1-4614-1927-3_15.

Bixby, Robert E., William Cook, Alan Cox, Eva K. Lee. 1999. Computational experience with parallel mixed integer programming in a distributed environment. *Annals of Operations Research* 19–43doi:10.1023/A:1018960631213.

Bussieck, Michael R., Arne S. Drud, Alex Meeraus. 2003. MINLPLib - a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing* **15** 114–119. doi:10.1287/ijoc.15.1.114.15159. `http://www.gamsworld.org/minlp/minlplib.htm`.

Bussieck, Michael R., Michael C. Ferris, Alexander Meeraus. 2009. Grid-enabled optimization with GAMS. *INFORMS Journal on Computing* **21** 349–362. doi:10.1287/ijoc.1090.0340.

Chen, Qun, Michael C. Ferris. 2000. FATCOP: A fault tolerant Condor-PVM mixed integer programming solver. *SIAM Journal on Optimization* **11** 1019–1036. doi:10.1137/S1052623499353911.

Chen, Qun, Michael C. Ferris, Jeff Linderoth. 2001. FATCOP 2.0: Advanced features in an opportunistic mixed integer programming solver. *Annals of Operations Research* **103** 17–32. doi:10.1023/A:1012982400848.

Eckstein, Jonathan. 1994. Parallel branch-and-bound algorithms for general mixed integer programming on the CM-5. *SIAM Journal on Optimization* **4** 794–814. doi:10.1137/0804046.

Eckstein, Jonathan, William E. Hart, Cynthia A. Phillips. 2001. PICO: An object-oriented framework for parallel branch-and-bound. *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*. Elsevier Scientific Series on Studies in Computational Mathematics, 219–265. doi:10.1016/S1570-579X(01)80014-8.

Eckstein, Jonathan, Cynthia A. Phillips, William E. Hart. 2009. PEBBL 1.0 user guide. RUTCOR Research Report RRR 19-2006, Rutgers Center for Operations Research, Rutgers University. URL `http://rutcor.rutgers.edu/pub/rrr/reports2006/19_2006.ps`.

Goux, Jean-Pierre, Sanjeev Kulkarni, Michael Yoder, Jeff Linderoth. 2001. Master−worker: An enabling framework for applications on the computational grid. *Cluster Computing* **4** 63–70. doi:10.1023/A:1011416310759.

Koch, Thorsten, Tobias Achterberg, Erling Andersen, Oliver Bastert, Timo Berthold, Robert E. Bixby, Emilie Danna, Gerald Gamrath, Ambros M. Gleixner, Stefan Heinz, Andrea Lodi, Hans Mittelmann, Ted K. Ralphs, Domenico Salvagnin, Daniel E. Steffy, Kati Wolter. 2011. MIPLIB 2010 – Mixed Integer Programming Library version 5. *Mathematical Programming Computation* **3** 103–163. doi:10.1007/s12532-011-0025-9.

Koch, Thorsten, Ted Ralphs, Yuji Shinano. 2012. Could we use a million cores to solve an integer program? *Mathematical Methods of Operations Research* **76** 67–93. doi:10.1007/s00186-012-0390-9. URL `http://dx.doi.org/10.1007/s00186-012-0390-9`.

Linderoth, Jeff. 1998. Topics in parallel integer optimization. Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Liu, Tongping, Charlie Curtsinger, Emery D. Berger. 2011. Dthreads: efficient deterministic multithreading. *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*. SOSP '11, ACM, New York, NY, USA, 327–336. doi:10.1145/2043556.2043587.

Martins, Ruben, Vasco Manquinho, Inês Lynce. 2012. An overview of parallel SAT solving. *Constraints* **17** 304–347. doi:10.1007/s10601-012-9121-3.

Mitra, Gautam, Ilan Hai, Mozafar T. Hajian. 1997. A distributed processing algorithm for solving integer programs using a cluster of workstations. *Parallel Computing* **23** 733–753. doi:10.1016/S0167-8191(97)00016-1.

Mittelmann, Hans. 2011. Mixed integer linear programming benchmark (serial codes). `http://plato.asu.edu/ftp/milpf.html`.

Montesinos, Pablo, Matthew Hicks, Samuel T. King, Josep Torrellas. 2009. Capo: A software-hardware interface for practical deterministic multiprocessor replay. *ACM SIGPLAN Notices* **44** 73–84. doi:10.1145/1508244.1508254.

Nwana, Vincent, Ken Darby-Dowman, Gautam Mitra. 2004. A twostage parallel branch and bound algorithm for mixed integer programs. *IMA Journal of Management Mathematics* **15** 227–242. doi:10.1093/imaman/15.3.227.

Olszewski, Marek, Jason Ansel, Saman Amarasinghe. 2009. Kendo: efficient deterministic multithreading in software. *ACM SIGPLAN Notices* **44** 97–108. doi:10.1145/1508284.1508256.

Ralphs, Ted K. 2006. Parallel branch and cut. E. Talbi, ed., *Parallel Combinatorial Optimization*. Wiley, New York, 53–101.

Ralphs, Ted K., Menal Güzelsoy, Ashutosh Mahajan. 2011. SYMPHONY 5.4 user's manual. Tech. rep., COR@L Laboratory, Lehigh University. URL `http://www.coin-or.org/SYMPHONY/doc/SYMPHONY-5.4.0-Manual.pdf`.

Ralphs, Ted K., Laszlo Ladányi, Matthew J. Saltzman. 2003. Parallel branch, cut, and price for large-scale discrete optimization. *Mathematical Programming Series B* **98** 253–280. doi:10.1007/s10107-003-0404-8.

Ralphs, Ted K., Laszlo Ladányi, Matthew J. Saltzman. 2004. A library hierarchy for implementing scalable parallel search algorithms. *The Journal of Supercomputing* **28** 215–234. doi:10.1023/B:SUPE.0000020179.55383.ad.

Shinano, Yuji, Tobias Achterberg, Timo Berthold, Stefan Heinz, Thorsten Koch. 2012. ParaSCIP – a parallel extension of SCIP. Christian Bischof, Heinz-Gerd Hegering, Wolfgang E. Nagel, Gabriel Wittum, eds., *Competence in High Performance Computing 2010*. Springer, 135–148. doi:10.1007/978-3-642-24025-6_12.

Shinano, Yuji, Tobias Achterberg, Tetsuya Fujie. 2008. A dynamic load balancing mechanism for new ParaLEX. *Proceedings of ICPADS 2008*. 455–462. doi:10.1109/ICPADS.2008.75.

Shinano, Yuji, Tetsuya Fujie. 2007. Paralex: A parallel extension for the CPLEX mixed integer optimizer. Franck Cappello, Thomas Herault, Jack Dongarra, eds., *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, *Lecture Notes in Computer Science*, vol. 4757. Springer Berlin Heidelberg, 97–106. doi:10.1007/978-3-540-75416-9_19.

Shinano, Yuji, Tetsuya Fujie, Yuusuke Kounoike. 2003. Effectiveness of parallelizing the ILOG-CPLEX mixed integer optimizer in the PUBB2 framework. Harald Kosch, Lszl Bszrmnyi, Hermann Hellwagner, eds., *Euro-Par 2003 Parallel Processing*, *Lecture Notes in Computer Science*, vol. 2790. Springer Berlin Heidelberg, 451–460. doi:10.1007/978-3-540-45209-6_67.

Sun, Yanhua, Gengbin Zheng, Pritish Jetley, Laxmikant V. Kalé. 2011. ParSSSE: An adaptive parallel state space search engine. *Parallel Processing Letters* **21** 319–338. doi:10.1142/S0129626411000242.

van Nieuwpoort, Rob V., Gosia Wrzesińska, Ceriel J.H. Jacobs, Henri E. Bal. 2010. Satin: A high-level and efficient grid programming model. *ACM Transactions on Programming Languages and Systems* **32** 1–39. doi:10.1145/1709093.1709096.

Vigerske, Stefan. 2013. Decomposition of multistage stochastic programs and a constraint integer programming approach to mixed-integer nonlinear programming. Ph.D. thesis, Humboldt-Universität zu Berlin. urn:nbn:de:kobv:11-100208240.

Xu, Yan, Ted K. Ralphs, Laszlo Ladányi, Matthew J. Saltzman. 2009. Computational experience with a software framework for parallel integer programming. *INFORMS Journal on Computing* **21** 383–397. doi:10.1287/ijoc.1090.0347.

Appendix A Mittelmann Benchmarks

The following table shows the results from H. Mittelmann's runs of the benchmark set of the Mixed Integer Linear Programming Library (MIPLIB2010, see also Table 8) (Mittelmann, 2011). For the numbers in row 1*, variables and constraints in the input data have been permuted and the average over five runs is reported.

Date	# threads	measure	GLPK	LPSOLVE	CBC	SCIP/FiberSCIP	CPLEX	GUROBI	XPRESS
14 Aug 2011	1	scaled	17.20	15.05	8.25	4.64	1.39	1	1.01
		# solved	3	5	41	54	76	76	72
	4	scaled	-	-	8.98	8.08	1.17	1	1.21
		# solved	-	-	52	64	80	83	78
	12	scaled	-	-	9.55	9.92	1.20	1	1.15
		# solved	-	-	55	65	84	86	80
11 April 2015	1	scaled	-	-	17.4	6.66	1	1.06	1.12
		# solved	-	-	48	68	84	84	84
	1*	scaled	-	-	18.0	7.61	1.05	1	1.63
		# solved	-	-	36	60	81	80	80
	4	scaled	-	-	17.4	12.7	1.05	1	1.03
		# solved	-	-	50	67	86	86	86
	12	scaled	-	-	14.0	15.0	1	1	1.14
		# solved	-	-	68	68	87	86	86

Appendix B Benchmark set

The 87 instances of the benchmark set of MIPLIB2010 (Koch et al., 2011) were used for the computational experiments in MIP solving. For MINLP, we used instances from the MINLPLib (Bussieck et al., 2003) instance library, see also Section 4.7. Tables 8 and 9 show the following properties of the MIP and MINLP instances, respectively:

Vars. The number of variables.
Cons. The number of constraints.
NZs The number of non-zero elements in constraint matrix.
Bin The number of binary variables.
Int The number of integer variables.
Cont The number of continuous variables.
Impl The number of implied integer variables.
LinCons The number of linear constraints.
NLCons The number of non-linear constraints.

Table 8: Benchmark set for MIP (MIPLIB2010 benchmark set)

	Original instance						Presolved instance						
Name	Vars.	Cons.	NZs	Bin	Int	Cont	Vars.	Cons.	Bin	Int	Impl	Cont	Vars.×Cons.
30n20b8	18380	576	109706	11036	7344	0	8028	403	7971	2	55	0	3235284
acc-tight5	1339	3052	16134	1339	0	0	998	2291	998	0	0	0	2286418
aflow40b	2728	1442	6783	1364	0	1364	1878	1144	940	0	0	938	2148432
air04	8904	823	72965	8904	0	0	7412	605	7412	0	0	0	4484260
app1-2	26871	53467	199175	13300	0	13571	26265	52555	13000	0	0	13265	1380357075
ash608gpia-3col	3651	24748	74244	3651	0	0	3651	24748	3651	0	0	0	90354948
bab5	21600	4964	155520	21600	0	0	21366	4819	21366	0	0	0	102962754
beasleyC3	2500	1750	5000	1250	0	1250	1704	1153	852	0	0	852	1964712
biella1	7328	1203	71489	6110	0	1218	7311	1202	6110	0	1197	4	8787822
bienst2	505	576	2184	35	0	470	449	520	35	0	0	414	233480
binkar10_1	2298	1026	4496	170	0	2128	1443	825	170	0	0	1273	1190475
bley_xl1	5831	175620	869391	5831	0	0	829	7988	829	0	0	0	6622052
bnatt350	3150	4923	19061	3150	0	0	1757	6587	1757	0	0	0	11573359
core2536-691	15293	2539	177739	15284	0	9	15269	1920	15269	0	0	0	29316480
cov1075	120	637	14280	120	0	0	120	637	120	0	0	0	76440
csched010	1758	351	6376	1457	0	301	1654	295	1457	0	1	196	487930
danoint	521	664	3232	56	0	465	513	656	56	0	0	457	336528
dfn-gwin-UUM	938	158	2632		90	848	936	156	0	90	0	846	146016
eil33-2	4516	32	44243	4516	0	0	4516	32	4516	0	0	0	144512
eilB101	2818	100	24120	2818	0	0	2818	100	2818	0	0	0	281800
enlight13	338	169	962	169	169	0	338	169	173	165	0	0	57122
enlight14	392	196	1120	196	196	0	392	196	200	192	0	0	76832
ex9	10404	40962	517112	10404	0	0	12	18	12	0	0	0	216
glass4	322	396	1815	302	0	20	317	392	298	0	0	19	124264
gmu-35-40	1205	424	4843	1200	0	5	560	325	555	0	0	5	182000
iis-100-0-cov	100	3831	22986	100	0	0	100	3831	100	0	0	0	383100
iis-bupa-cov	345	4803	38392	345	0	0	341	4803	341	0	0	0	1637823
iis-pima-cov	768	7201	71941	768	0	0	736	7201	736	0	0	0	5299936
lectsched-4-obj	7901	14163	82428	7665	236	0	2606	4801	2500	106	0	0	12511406
m100n500k4r1	500	100	2000	500	0	0	500	100	500	0	0	0	50000
macrophage	2260	3164	9492	2260	0	0	2260	3164	2260	0	0	0	7150640
map18	164547	328818	549920	146	0	164401	15398	31183	118	0	0	15280	480155834
map20	164547	328818	549920	146	0	164401	15398	31183	118	0	0	15280	480155834
mcsched	1747	2107	8088	1731	14	2	1495	1853	1495	0	0	0	2770235
mik-250-1-100-1	251	151	5351	100	150	1	201	159	107	93	1	0	31959
mine-166-5	830	8429	19412	830	0	0	746	7135	746	0	0	0	5322710
mine-90-10	900	6270	15407	900	0	0	800	4656	800	0	0	0	3724800
msc98-ip	21143	15850	92918	20237	53	853	12737	14986	11927	0	0	810	190876682
mspp16	29280	561657	27678735	29280	0	0	12737	14986	11927	0	0	810	190876682
mzzv11	10240	9499	134603	9989	251	0	6892	6816	6657	184	51	0	46975872
n3div36	22120	4484	340740	22120	0	0	19764	4454	19764	0	0	0	88028856
n3seq24	119856	6044	3232340	119856	0	0	119856	5950	119856	0	0	0	713143200
n4-3	3596	1236	14036		174	3422	3360	996	0	174	0	3186	3346560
neos-1109824	1520	28979	89528	1520	0	0	1520	9979	1520	0	0	0	15168080
neos-1337307	2840	5687	30799	2840	0	0	2840	5687	2840	0	0	0	16151080
neos-1396125	1161	1494	5511	129	0	1032	1158	1491	129	0	3	1026	1726578
neos13	1827	20852	253842	1815	0	12	1827	17320	1815	0	0	12	31643640
neos-1601936	4446	3131	72500	3906	0	540	3920	3105	3570	0	350	0	12171600
neos18	3312	11402	24614	3312	0	0	761	3300	761	0	0	0	2511300
neos-476283	11915	10015	3945693	5588	0	6327	11825	9583	5544	0	0	6281	113318975
neos-686190	3660	3664	18085	3600	60	0	3660	3658	3600	60	0	0	13388280
neos-849702	1737	1041	19308	1737	0	0	1713	996	1713	0	0	0	1706148
neos-916792	1474	1909	134442	717	0	757	1361	1408	708	0	0	653	1916288
neos-934278	23123	11495	125577	19955	0	3168	8121	8123	7354	0	767	0	65966883
net12	14115	14021	80384	1603	0	12512	12546	12787	1120	0	0	11426	160425702
netdiversion	129180	119589	615282	129180	0	0	128968	99483	128968	0	0	0	12830123544
newdano	505	576	2184	56	0	449	449	520	56	0	0	393	233480
noswot	128	182	735	75	25	28	120	171	75	20	0	25	20520
ns1208400	2883	4289	81746	2880	0	3	2596	1981	2596	0	0	0	5142676
ns1688347	2685	4191	66908	2685	0	0	1360	2599	1360	0	0	0	3534640
ns1758913	17956	624166	1283444	17822	0	134	17624	588479	17622	0	0	2	10371353896
ns1766074	100	182	666		90	10	100	110	90	0	0	10	11000
ns1830653	1629	2932	100933	1458	0	171	648	1489	514	0	134	0	964872
opm2-z7-s2	2023	31798	79762	2023	0	0	1900	26839	1900	0	0	0	50994100
pg5_34	2600	225	7700	100	0	2500	2600	225	100	0	0	2500	585000
pigeon-10	490	931	8150	400	0	90	400	525	370	0	0	30	210000
pw-myciel4	1059	8164	17779	1058	1	0	1036	4180	1035	1	0	0	4330480
qiu	840	1192	3432	48	0	792	840	1192	48	0	0	792	1001280
rail507	63019	509	468878	63009	0	10	62997	473	62997	0	0	0	29797581
ran16x16	512	288	1024	256	0	256	512	288	256	0	0	256	147456
reblock67	670	2523	7495	670	0	0	606	2130	606	0	0	0	1290780
rmatr100-p10	7359	7260	21877	100	0	7259	7359	7260	100	0	0	7259	53426340
rmatr100-p5	8784	8685	26152	100	0	8684	8784	8685	100	0	0	8684	76289040
rmine6	1096	7078	18084	1096	0	0	1084	7066	1084	0	0	0	7659544
rocII-4-11	9234	21738	243106	9086	0	148	1403	3734	1321	0	0	82	5238802
rococoC10-001000	3117	1293	11751	2993	124	0	2442	576	2442	0	0	0	1406592
roll3000	1166	2295	29386	246	492	428	862	1439	624	114	11	113	1240418
satellites1-25	9013	5996	59023	8509	0	504	7081	4203	7049	0	0	32	29761443
sp98ic	10894	825	316317	10894	0	0	10894	797	10894	0	0	0	8682518
sp98ir	1680	1531	71704	871	809	0	1557	1372	869	688	0	0	2136204
tanglegram1	34759	68342	205026	34759	0	0	34759	68342	34759	0	0	0	2375499578
tanglegram2	4714	8980	26940	4714	0	0	4714	8980	4714	0	0	0	42331720
timtab1	397	171	829	64	107	226	201	166	54	92	0	55	33366
triptim1	30055	15706	515436	20451	9597	7	21938	15679	12446	9492	0	0	343965902
unitcal_7	25755	48939	127595	2856	0	22899	20292	38651	2503	0	0	17789	784306092
vpphard	51471	47280	372305	51471	0	0	28966	24896	28966	0	0	0	721137536
zib54-UUE	5150	1809	15288	81	0	5069	5069	1761	80	0	0	4989	8926509

Table 9: Benchmark set for MINLP (MINLPLib selected instances)

Name	Original instance					Presolved instance						
	Vars.	Cons.	Bin	Int	Cont	Vars.	LinCons	NLCons	Bin	Int	Impl	Cont
4stufen	150	98	48	0	102	125	21	55	48	0	4	73
beuster	158	114	52	0	106	164	108	62	51	0	7	106
cecil_13	840	898	180	0	660	659	736	174	96	0	0	563
chp_partload	2248	2516	45	0	2203	1244	469	995	42	0	0	1202
csched2a	233	138	140	0	93	291	109	87	140	0	0	151
csched2	401	138	308	0	93	457	109	85	308	0	0	149
eg_all_s	8	28	0	7	1	46546	0	46566	2	5	0	46539
eg_disc2_s	8	28	0	3	5	46558	0	46578	0	3	0	46555
eg_disc_s	8	28	0	4	4	46552	0	46572	0	4	0	46548
eg_int_s	8	28	0	3	5	46546	0	46566	2	1	0	46543
enpro48pb	154	215	92	0	62	154	213	2	92	0	0	62
ex1233	53	65	12	0	41	83	46	41	10	0	0	73
fac2	67	34	12	0	55	67	27	4	12	0	0	55
feedtray	98	92	7	0	91	251	19	224	7	0	0	244
fo7	114	211	42	0	72	82	137	14	42	0	0	40
fo8_ar2_1	144	347	0	56	88	101	204	16	1	54	0	46
fo8_ar25_1	144	347	0	56	88	101	204	16	1	54	0	46
fo8_ar3_1	144	347	0	56	88	101	204	16	1	54	0	46
fo8_ar4_1	144	347	0	56	88	101	204	16	1	54	0	46
fo8_ar5_1	144	347	0	56	88	101	204	16	1	54	0	46
fo8	146	273	56	0	90	102	173	16	56	0	0	46
fo9_ar2_1	180	435	0	72	108	123	248	18	1	70	0	52
fo9_ar25_1	180	435	0	72	108	123	248	18	1	70	0	52
fo9_ar3_1	180	435	0	72	108	123	248	18	1	70	0	52
fo9_ar4_1	180	435	0	72	108	123	248	18	1	70	0	52
fo9_ar5_1	180	435	0	72	108	123	248	18	1	70	0	52
fo9	182	343	72	0	110	124	213	18	72	0	0	52
fuzzy	897	1057	120	0	777	819	694	55	110	0	10	699
ghg_1veh	30	38	12	0	18	77	49	55	4	0	0	73
ghg_3veh	96	119	36	0	60	359	357	218	35	0	8	316
hda	723	719	13	0	710	464	230	233	7	0	0	457
lop97ic	1754	92	831	831	92	5220	11509	0	708	704	3808	0
lop97icx	987	88	68	831	88	486	1135	0	68	67	351	0
mbtd	210	70	200	0	10	81310	360	81000	81200	0	0	110
netmod_dol1	1999	3138	462	0	1537	1993	3131	1	462	0	1530	1
no7_ar3_1	112	269	0	42	70	87	182	14	1	40	0	46
no7_ar4_1	112	269	0	42	70	87	182	14	1	40	0	46
no7_ar5_1	112	269	0	42	70	87	182	14	1	40	0	46
nous1	51	44	2	0	49	47	11	29	2	0	0	45
nuclear10a	13010	3339	10920	0	2090	23826	32761	3026	10920	0	0	12906
nuclear10b	23826	24971	10920	0	12906	23826	21945	3026	10920	0	0	12906
nuclear14a	992	633	600	0	392	1568	1801	560	600	0	0	968
nuclear14b	1568	1785	600	0	968	1568	1225	560	600	0	0	968
nuclear25a	1058	659	650	0	408	1683	1951	583	650	0	0	1033
nuclear25b	1683	1909	650	0	1033	1683	1326	583	650	0	0	1033
nuclear49a	3341	1431	2450	0	891	5742	7351	1283	2450	0	0	3292
nuclear49b	5742	6233	2450	0	3292	5742	4950	1283	2450	0	0	3292
nvs09	11	1	0	10	1	40	0	30	0	10	9	21
o7_2	114	211	42	0	72	90	153	14	42	0	0	48
o7_ar2_1	112	269	0	42	70	89	188	14	1	40	0	48
o7_ar25_1	112	269	0	42	70	89	188	14	1	40	0	48
o7_ar3_1	112	269	0	42	70	89	188	14	1	40	0	48
o7_ar4_1	112	269	0	42	70	89	188	14	1	40	0	48
o7_ar5_1	112	269	0	42	70	89	188	14	1	40	0	48
o7	114	211	42	0	72	90	153	14	42	0	0	48
o8_ar4_1	144	347	0	56	88	117	252	16	1	54	0	62
o9_ar4_1	180	435	0	72	108	137	290	18	1	70	0	66
oil2	937	927	2	0	935	1065	119	952	2	0	0	1063
oil	1535	1546	19	0	1516	1409	410	1143	19	0	0	1390
pb302035	601	51	600	0	1	1180	1790	0	600	0	580	0
pb302055	601	51	600	0	1	1152	1751	0	585	0	567	0
pb302075	601	51	600	0	1	1074	1640	0	544	0	530	0
pb302095	601	51	600	0	1	931	1436	0	469	0	462	0
pb351535	526	51	525	0	1	1035	1580	0	525	0	510	0
pb351555	526	51	525	0	1	1035	1580	0	525	0	510	0
pb351575	526	51	525	0	1	1035	1580	0	525	0	510	0
pb351595	526	51	525	0	1	1013	1547	0	514	0	499	0
product2	2842	3125	128	0	2714	480	338	128	128	0	0	352
qap	226	31	225	0	1	436	663	0	225	0	211	0
qapw	451	256	225	0	226	675	930	0	225	0	450	0
saa_2	4407	6205	400	0	4007	24115	2170	23743	366	0	0	23749
space25a	383	201	240	0	143	308	101	25	240	0	24	44
space960	5537	6497	0	960	4577	2657	2657	960	0	960	0	1697
stockcycle	481	98	432	0	49	481	97	1	432	0	48	1
super1	1308	1659	44	0	1264	928	592	667	31	0	3	894
super2	1308	1659	44	0	1264	929	592	667	31	0	3	895
super3	1308	1659	44	0	1264	943	603	672	37	0	3	903
super3t	1056	1343	44	0	1012	675	486	449	37	0	3	635
synheat	58	65	12	0	46	75	46	33	10	0	0	65
tln12	168	72	12	156	0	180	85	11	24	144	12	0
tln5	35	30	5	30	0	35	20	5	5	30	0	0
tln6	48	36	6	42	0	48	24	6	6	42	0	0
tls4	105	64	85	4	16	124	100	13	85	0	39	0
waste	2484	1991	400	0	2084	1238	516	2140	400	0	0	838
waterx	70	54	14	0	56	96	36	44	14	0	0	82
waterz	195	137	126	0	69	207	120	29	126	0	0	81

Appendix C All computational results

All tables with numeric values except for Tables 16 and 17 evaluate the number of nodes and the running times for different settings over a whole testset. The 'Nodes' column shows the number of processed nodes, the 'Time' column shows computing time in seconds. In the 'Time' column, a number is preceded by '>' if the execution hit the time limit of 2 hours and it is preceded by '!' if execution has aborted. A '−' in the number of nodes and computing time columns indicates that the result was lost for some reason[8]. This case is considered like an 'abort'. In the bottom of the table, geometric means of the number of processed nodes and computing times are presented w.r.t. the n instances that were solved by the various FiberSCIP runs and the baseline run. The number n is shown in "(solved: n)". The speedup is calculated by using the results for the n instances. The tables also show geometric means of solving time w.r.t. all instances in the "geom. mean (all)" row, where aborted instances are accounted with the time limit of 7200s. In the following row, in order to see the stability of parallel runs, geometric means of the number of processed nodes and computing times are presented w.r.t. the n' instances that were solved by the settings of all FiberSCIP runs, disregarding the baseline run. The number of solved instances, timeouts, and aborts are shown below and the speedups calculated by the results of n instances are presented at the bottom of the table. The average values of the geometric means, the computing time, and the speedups in the number of solved instances between the baseline result and the evaluated setting are added in the left most column of the summary.

Tables 16 and 17 show the variation of five repeated runs of FiberSCIP with racing only and communication of the incumbents enabled and the same number of threads on the same computer for all instances that are solved within 7200s.

Tables 19, 21, 23, 25, 27, and 29 show the fraction of solving time that the SOLVERs are idle, that is, do not spend on processing a subproblem. If no data on the idle time is available, e.g., when the solver aborted, no bar is printed. Otherwise, the length of each bar indicates the amount of idle time, averaged over all SOLVERs threads, in relation to the (wall-clock) solving time. In Tables 27 and 29, the maximal length (corresponding to the complete solving time) is indicated by dots. The different shades in the bar indicate the various sources for idling. For runs in non-deterministic mode, these are

▨ waiting for a first subproblem (ramp-up phase)

▤ waiting for receiving the next subproblem (primary phase)

▧ waiting for other SOLVERs to finish (ramp-down phase)

For runs in deterministic mode, these are

▦ waiting for the notification ID

[8] The main reason for loosing a result is that solving an LP took too long, the job was killed by the resource management system of the cluster, and the result files were not moved from the local disk of the worker node to the network file system.

 waiting for acknowledgment of completion

 waiting for the token (see Section 3.5)

In FiberSCIP, all SOLVERs are executed totally asynchronously. However, in order to control the number of nodes collected by the LOADCOORDINATOR and to measure statistics correctly, it is necessary to synchronize between the LOADCOORDINATOR and a SOLVER. This synchronization causes the waiting time for the notification ID and that for the acknowledgment of completion.

Table 10: Sequential executions of SCIP and FiberSCIP

Name	SCIP		FiberSCIP			
			Presolve Once		Presolve Twice	
	Nodes	Time	Nodes	Time	Nodes	Time
30n20b8	4691	> 7201	8753	> 7202	4639	> 7219
acc-tight5	1649	340	1649	339	1808	353
aflow40b	286358	2967	286358	2971	668667	6444
air04	144	126	144	127	215	149
app1-2	76	1997	76	2004	392	2163
ash608gpia-3col	7	77	7	80	7	77
bab5	17595	> 7202	17615	> 7202	12846	> 7201
beasleyC3	1151024	> 7203	1193762	> 7203	992738	> 7203
biella1	3270	1549	3270	1552	3270	1554
bienst2	117922	553	117922	553	86468	439
binkar10_1	114820	257	114820	256	135435	280
bley_xl1	17	467	17	449	1	342
bnatt350	6685	1121	6685	1130	4134	859
core2536-691	281	1064	281	1066	810	1487
cov1075	965740	> 7201	953142	> 7201	976932	> 7201
csched010	651293	> 7200	646222	> 7201	684362	> 7200
danoint	881262	> 7200	879302	> 7201	870942	> 7201
dfn-gwin-UUM	87889	239	87889	241	87613	239
eil33-2	10571	180	10571	180	10571	190
eilB101	9239	1287	9239	1300	9239	1292
enlight13	1116961	1388	1116961	1355	745476	857
enlight14	172555	229	172555	211	259155	316
ex9	1	191	1	191	1	194
glass4	3780000	! 5244	3735142	! 5150	2990992	! 4138
gmu-35-40	2080000	! 1905	2058222	! 1849	1970052	! 1785
iis-100-0-cov	106874	2685	106874	2694	106874	2669
iis-bupa-cov	111227	> 7200	113790	> 7201	113309	> 7202
iis-pima-cov	13011	1504	13011	1505	12469	1526
lectsched-4-obj	2772	156	2772	157	21026	826
m100n500k4r1	5025592	> 7201	5072925	> 7200	5050160	> 7201
macrophage	528171	> 7200	528062	> 7200	541612	> 7201
map18	293	862	293	857	261	1182
map20	353	1001	353	1010	526	926
mcsched	16113	339	16113	339	16113	341
mik-250-1-100-1	1421995	474	1421995	457	1421995	457
mine-166-5	6949	104	6949	105	6170	109
mine-90-10	92852	543	92852	541	189285	1364
msc98-ip	626	> 7203	624	> 7217	485	> 7212
mspp16	1	! 144	1	! 143	1	! 144
mzzv11	2621	755	2621	760	8361	1868
n3div36	40000	! 3101	37642	! 2857	37993	! 2825
n3seq24	542	> 7228	1	! 71	1	! 72
n4-3	48686	1281	48686	1285	57354	1466
neos-1109824	24162	356	24162	359	20142	283
neos-1337307	238371	> 7200	238721	> 7200	239423	> 7201
neos-1396125	65896	6547	65896	6563	40087	4125
neos13	17151	> 7201	17220	> 7201	17847	> 7201
neos-1601936	19862	> 7200	19874	> 7200	7771	5311
neos18	9133	63	9133	65	6827	81
neos-476283	1	! 483	1	! 163	1	! 273
neos-686190	9894	199	9894	200	4739	134
neos-849702	50303	1396	50303	1396	126716	3193
neos-916792	57471	449	57471	450	66213	511
neos-934278	2526	> 7200	2567	> 7201	2755	> 7206
net12	7260	5950	7260	5938	3678	4536
netdiversion	33	> 7278	33	> 7511	26	> 7295
newdano	3062795	> 7200	3052375	> 7201	2842106	> 7201
noswot	463569	197	463569	191	575618	217
ns1208400	12202	3251	12202	3252	4817	1361
ns1688347	4995	510	4995	510	4538	528
ns1758913	1	! 3043	1	! 2853	1	! 3132
ns1766074	946987	1051	946987	959	945109	1318
ns1830653	42213	746	42213	745	88276	1111
opm2-z7-s2	4401	2345	4401	2357	1978	1142
pg5_34	318742	2396	318742	2387	318742	2339
pigeon-10	3757759	> 7200	3535086	> 7200	2955734	> 7200
pw-myciel4	494676	> 7200	482684	> 7200	488828	> 7201
qiu	11012	86	11012	87	9811	83
rail507	1472	2772	1472	2868	1319	2126
ran16x16	371304	335	371304	332	441109	396
reblock67	140595	535	140595	533	83043	339
rmatr100-p10	901	297	901	297	901	298
rmatr100-p5	397	857	397	857	397	855
rmine6	530871	4728	530871	4712	687364	6508
rocII-4-11	13591	406	13591	405	13194	399
rococoC10-001000	365708	2679	365708	2687	1075742	> 7201
roll3000	977873	> 7200	978756	> 7201	1322672	> 7201
satellites1-25	9374	3902	9374	3879	16711	4956
sp98ic	40000	! 5253	41967	! 5110	40930	! 5053
sp98ir	4807	120	4807	123	8984	202
tanglegram1	27	1808	27	1815	27	2358
tanglegram2	3	11	3	12	3	13
timtab1	1039843	744	1085175	768	1014862	746
triptim1	30	4535	12	5122	39	> 7207
unitcal_7	17499	3734	17499	3731	12785	2501
vpphard	2319	> 7203	2318	> 7203	3693	> 7206
zib54-UUE	736111	> 7200	553363	6934	565989	6971
geom. mean (solved; 56)	8711	623.5	8718	623.7	9204	669.5
geom. mean (all)	–	1466.2	–	1467.9	–	1554.7
solved/timeout/abort	58/22/7		59/20/8		58/21/8	

Table 11: Racing only runs without communication of incumbents

Name	One thread Nodes	One thread Time	2 threads Nodes	2 threads Time	4 threads Nodes	4 threads Time	6 threads Nodes	6 threads Time	8 threads Nodes	8 threads Time
30n20b8	4639	> 7219	2539	> 7212	2568	> 7200	2038	> 7207	5277	> 7207
acc-tight5	1808	353	1808	353	777	289	748	189	308	156
aflow40b	668667	6444	238236	2080	194236	1818	192294	1987	178690	2220
air04	215	149	179	140	101	101	101	110	101	129
app1-2	392	2163	392	2192	47	1716	47	2776	47	3991
ash608gpia-3col (i)	7	77	7	79	7	102	7	174	7	248
bab5	12846	> 7201	12206	> 7210	10802	> 7201	9131	> 7201	2211	> 7203
beasleyC3	992738	> 7203	901232	> 7200	808112	> 7200	509292	> 7200	479393	> 7200
biella1 (i)	3270	1554	3270	1603	3270	1614	3270	1725	3270	2332
bienst2	86468	439	86468	447	86468	450	86468	468	84244	460
binkar10_1	135435	280	135435	294	135435	306	67149	361	67149	432
bley_xl1 (i)	1	342	1	350	1	349	1	352	1	361
bnatt350	4134	859	4134	866	5058	864	4134	919	2831	945
core2536-691	810	1487	192	1078	192	1096	192	1361	192	2566
cov1075	976932	> 7201	926202	> 7200	912832	> 7200	873782	> 7200	799632	> 7200
csched010	684362	> 7200	676752	> 7200	654132	> 7200	509701	> 7200	527202	> 7200
danoint	870942	> 7201	833042	> 7200	794262	> 7200	768882	> 7200	764862	> 7200
dfn-gwin-UUM	87613	239	87613	242	63562	198	43321	209	22548	218
eil33-2	10571	190	10571	208	10571	228	11423	291	10571	380
eilB101	9239	1292	10512	1124	9770	989	9770	1039	7381	805
enlight13	745476	857	745476	878	367322	510	109200	839	109200	418
enlight14	259155	316	259155	319	259155	388	259155	518	100054	311
ex9 (i)	1	194	1	188	1	183	1	191	1	191
glass4	2990992	! 4138	3954522	! 6086	1082160	1655	637482	! 1510	1082160	2281
gmu-35-40	1970052	! 1785	1354062	! 2376	2792362	! 3117	1881932	! 3486	1796808	! 3988
iis-100-0-cov (i)	106874	2669	106874	2696	106874	2769	106874	3201	106874	3293
iis-bupa-cov	113309	> 7202	112222	> 7200	107022	> 7200	91622	> 7200	75290	> 7200
iis-pima-cov	12469	1526	12469	1493	6523	1318	6523	1394	6523	2368
lectsched-4-obj	21026	826	4591	587	3213	375	1796	208	1796	265
m100n500k4r1	5050160	> 7201	5006321	> 7200	5027960	> 7200	4662679	> 7200	1	> 7200
macrophage	541612	> 7201	385422	> 7200	305392	> 7200	145872	> 7200	114520	> 7200
map18	261	1182	261	1589	493	1418	493	2129	347	3595
map20	526	926	526	958	399	1085	295	1707	333	2148
mcsched	16113	341	16113	342	16625	343	14860	339	13508	319
mik-250-1-100-1 (i)	1421995	457	1421995	468	1421995	505	1421995	529	1421995	551
mine-166-5	6170	109	6170	117	1583	91	4289	101	3582	126
mine-90-10	189285	1364	189285	1376	74459	826	74459	1071	74459	1374
msc98-ip	485	> 7212	393	> 7230	125	> 7209	102	> 7221	233	> 7208
mspp16	1	! 144	1	! 159	1	! 2541	1	! 2430	1	! 2566
mzzv11	8361	1868	4756	1954	4756	1784	4877	2355	4665	4148
n3div36	37993	! 2825	31316	! 3386	47030	! 4023	62009	> 7201	25456	> 7201
n3seq24	1	! 72	1	! 721	1	> 7225	6	> 7258	4	> 7312
n4-3	57354	1466	57354	1543	57354	1554	43434	1469	43434	1677
neos-1109824	20142	283	8268	135	8268	169	8268	217	8268	355
neos-1337307	239423	> 7201	182852	> 7200	176094	> 7200	165083	> 7200	85322	> 7200
neos-1396125	40087	4125	40087	4131	40087	4164	38909	4204	38909	4258
neos13	17847	> 7201	1	> 7202	1	> 7202	1947	> 7201	912	> 7201
neos-1601936	7771	5311	7771	5323	7771	5468	9458	5655	9458	6936
neos18	6827	81	6827	80	7855	76	5830	60	5830	73
neos-476283	1	! 273	372	1017	13	! 876	372	1667	372	2632
neos-686190	4739	134	4739	134	4739	151	3241	133	3241	164
neos-849702	126716	3193	20907	774	20907	786	17371	822	20907	803
neos-916792	66213	511	66213	518	66213	606	60176	796	66213	933
neos-934278	2755	> 7206	248	> 7218	161	> 7216	139	> 7207	45	> 7229
net12	3678	4536	3513	4152	3513	4162	3513	6633	2932	> 7201
netdiversion	26	> 7295	24	> 7313	16	> 7316	8	> 7718	1	> 7371
newdano	2842106	> 7201	2720074	> 7200	2874552	7199	2058508	6520	2058508	7024
noswot	575618	217	575618	225	434470	188	393317	225	324848	260
ns1208400	4817	1361	691	491	691	504	691	550	691	534
ns1688347 (i)	4538	528	4538	534	4538	536	4538	547	4538	602
ns1758913	1	! 3132	1	! 3903	1	! 4869	-	-	2	> 7881
ns1766074	945109	1318	907103	1367	934754	959	934754	1441	934754	1871
ns1830653	88276	1111	88276	1107	61053	950	61053	1149	60186	1178
opm2-z7-s2	1978	1142	1978	1297	1978	2238	1822	2405	1978	5401
pg5_34	318742	2339	318742	2407	318742	2782	318742	3239	253046	4173
pigeon-10	2955734	> 7200	2	> 7200	2	> 7200	2	! 613	1	! 3071
pw-myciel4	488828	> 7201	474846	> 7201	115605	4945	296556	4496	296556	4666
qiu (i)	9811	83	9811	83	9811	84	9811	87	9811	91
rail507	1319	2126	1319	3837	1319	4127	1319	4578	578	> 7202
ran16x16	441109	396	441109	401	356748	350	356748	401	356748	499
reblock67 (i)	83043	339	83043	344	83043	335	83043	412	83043	482
rmatr100-p10 (i)	901	298	901	298	901	299	901	325	901	415
rmatr100-p5	397	855	397	859	397	858	413	905	397	2171
rmine6	687364	6508	687364	6650	586282	> 7200	400132	> 7200	284372	> 7200
rocII-4-11	13194	399	13194	398	11288	342	11288	354	11288	443
rococoC10-001000	1075742	> 7201	1086082	> 7200	562119	4846	562119	5755	347586	! 4231
roll3000	1322672	> 7201	1310742	> 7200	1283482	> 7200	644372	> 7200	688092	> 7200
satellites1-25	16711	4956	16711	5064	3656	4244	3656	6115	11313	> 7203
sp98ic	40930	! 5053	85902	! 5845	83433	! 6251	87508	> 7201	36653	> 7201
sp98ir	8984	202	7236	199	8685	215	4999	183	4999	223
tanglegram1	27	2358	27	2516	27	3941	27	4546	22	> 7264
tanglegram2 (i)	3	13	3	13	3	13	3	18	3	20
timtab1 (i)	1014862	746	1014862	768	1014862	773	1014862	955	1014862	1027
triptim1	39	> 7207	162	6881	4	> 7346	8	> 7300	9	3183
unitcal_7	12785	2501	12785	2529	12785	3906	10723	> 7202	8171	> 7201
vpphard	3693	> 7206	5	> 7210	2	> 7226	4	> 7202	2	> 7215
zib54-UUE	565989	6971	565989	7090	537672	> 7200	492492	> 7200	383842	> 7200
geom. mean (solved: 51)	9817	573.9	8148	529.7	6935	507.5	6386	569.4	5905	672.0
geom. mean (all)	–	1554.6	–	1460.1	–	1434.4	–	1564.4	–	1723.6
geom. mean (solved: 51)			8148	529.7	6935	507.5	6386	569.4	5905	672.0
solved/timeout/abort	58/21/8		60/20/7		60/21/6		59/23/5		56/27/4	
speedup (solved: 51)			1.084		1.131		1.008		0.854	

Table 12: Racing only: Effect of communication of incumbents (Four SOLVER Threads)

Name	Without comm. Nodes	Time	Run - 1 Nodes	Time	Run - 2 Nodes	Time	Run - 3 Nodes	Time	Run - 4 Nodes	Time	Run - 5 Nodes	Time
30n20b8	2568	> 7200	902	> 7218	1022	> 7209	820	> 7204	516	> 7203	1174	> 7202
acc-tight5	777	289	39	62	39	62	39	61	39	61	39	61
aflow40b	194236	1818	181535	1875	181535	1856	181535	1841	175610	1639	198942	1611
air04	101	101	49	97	49	96	49	96	49	96	49	96
app1-2	47	1716	47	1930	47	1893	47	1953	47	1902	47	1923
ash608gpia-3col	7	102	7	101	7	100	7	101	7	99	7	100
bab5	10802	> 7201	16846	> 7201	15454	> 7202	12252	> 7201	15235	> 7201	10966	> 7201
beasleyC3	808112	> 7200	898942	> 7201	956912	> 7200	885932	> 7200	1099922	> 7200	801642	> 7200
biella1	3270	1614	2288	1159	2654	1340	1649	1408	1658	1062	1575	1309
bienst2	86468	450	94998	465	97769	499	104468	506	101544	531	104990	523
binkar10_1	135435	306	165250	395	136563	343	107798	259	121683	292	111086	267
bley_xl1	1	349	1	349	1	349	1	351	1	352	1	338
bnatt350	5058	864	5058	864	5058	863	5058	814	5058	864	5058	861
core2536-691	192	1096	901	1335	901	1337	810	1607	901	1340	810	1575
cov1075	912832	> 7200	944582	> 7200	1078532	> 7200	1077042	> 7200	950542	> 7200	1069092	> 7200
csched010	654132	> 7200	563541	> 7200	580012	> 7200	656582	> 7200	642848	> 7200	683582	> 7200
danoint	794262	> 7200	704140	6653	704140	6662	704140	6619	704140	6684	704140	6643
dfn-gwin-UUM	63562	198	63710	199	63909	201	63551	201	64307	201	61020	186
eil33-2	10571	228	9879	209	9401	201	8975	197	9759	212	9772	208
eilB101	9770	989	4872	692	4402	562	5689	758	4811	624	5049	659
enlight13	367322	510	367322	395	367322	490	367322	489	367322	496	367322	489
enlight14	259155	388	259155	305	259155	388	259155	556	259155	390	259155	398
ex9	1	183	1	191	1	197	1	192	1	191	1	191
glass4	1082160	1655	939959	1544	702114	1167	1050611	1650	2542544	3495	3421535	4949
gmu-35-40	2792362	! 3117	2405472	! 2458	2333392	! 2144	2375502	! 2429	1738622	! 2235	2518052	! 2415
iis-100-0-cov	106874	2769	98732	2731	94430	2598	104420	2863	91513	2490	104420	2855
iis-bupa-cov	107022	> 7200	106992	> 7200	111527	> 7200	106892	> 7200	106616	> 7200	106382	> 7200
iis-pima-cov	6523	1318	7124	1146	7124	1149	7124	1148	7124	1144	7124	1145
lectsched-4-obj	3213	375	4540	264	337	184	3506	237	2794	215	563	243
m100n500k4r1	5027960	> 7200	4959466	> 7200	4975506	> 7200	4915463	> 7200	4857441	> 7200	5004431	> 7200
macrophage	305392	> 7200	307152	> 7201	345402	> 7200	348672	> 7200	381712	> 7200	433012	> 7200
map18	493	1418	493	1427	493	1403	493	1428	493	1465	493	1410
map20	399	1085	399	1180	399	1097	399	1567	399	1184	399	1101
mcsched	16625	343	13704	309	20407	406	16861	344	15775	336	14646	312
mik-250-1-100-1	1421995	505	1421995	490	1421995	495	1421995	500	1421995	496	1421995	494
mine-166-5	1583	91	5138	85	3576	83	4167	87	1028	84	2888	82
mine-90-10	74459	826	291363	2503	65635	494	199943	1949	96075	873	199321	1483
msc98-ip	125	> 7209	45	> 7242	138	> 7209	209	> 7201	130	> 7242	138	> 7220
mspp16	1	! 2541	1	! 2544	1	! 2551	1	! 2536	1	! 2562	1	! 2575
mzzv11	4756	1784	4048	1147	6993	1612	4461	1264	4806	1305	4152	1227
n3div36	47030	! 4023	40083	! 4683	56160	! 5047	50186	! 4698	52637	! 4788	35660	! 4707
n3seq24	1	> 7225	152	> 7209	193	> 7210	285	> 7242	156	> 7225	1	! 674
n4-3	57354	1554	49902	1301	51789	1403	46044	1293	48271	1355	50043	1406
neos-1109824	8268	169	8164	171	8164	285	8164	171	8164	171	8164	175
neos-1337307	176094	> 7200	204864	> 7201	208705	> 7200	214352	> 7200	211707	> 7200	211819	> 7200
neos-1396125	40087	4164	39799	4152	42121	4796	41808	4444	44725	4925	43032	4750
neos13	1	> 7202	4882	3741	50	2952	46	3674	6882	4596	45	2895
neos-1601936	7771	5468	12605	> 7200	3967	3662	6206	6749	14135	> 7200	3191	3404
neos18	7855	76	6292	70	6292	70	6292	70	6292	71	6292	70
neos-476283	13	! 876	1	! 439	1	! 2840	12	! 869	31	! 855	70	! 1117
neos-686190	4739	151	3589	118	4306	129	3018	109	3679	123	3970	124
neos-849702	20907	786	20907	788	20907	786	20907	790	20907	788	20907	788
neos-916792	66213	606	66296	584	66044	581	68269	836	64479	594	69786	641
neos-934278	161	> 7216	712	> 7206	207	> 7205	171	> 7201	291	> 7222	209	> 7202
net12	3513	4162	3850	3314	3056	4018	3484	4383	4464	4095	4721	4216
netdiversion	16	> 7316	15	> 7315	14	> 7353	-	-	23	> 7317	16	> 7320
newdano	2874552	7199	2870908	6711	2719657	> 7200	1898642	4595	2394587	7152	2791732	6957
noswot	434470	188	301829	159	301829	158	301829	164	301829	170	301829	163
ns1208400	691	504	691	506	691	507	691	507	691	507	691	510
ns1688347	4538	536	3515	483	3752	476	4538	536	4310	505	2944	383
ns1758913	1	! 4869	1	! 7898	1	! 5011	1	! 4796	1	! 4709	-	-
ns1766074	934754	959	934754	1194	934754	1190	934754	1188	934754	1170	934754	1180
ns1830653	61053	950	38073	677	48232	810	43242	748	39359	655	37962	679
opm2-z7-s2	1978	2238	3233	2833	2321	2693	2156	4070	2485	2744	2285	2416
pg5_34	318742	2782	292798	2336	318401	2726	310197	2319	309644	2589	267308	1752
pigeon-10	2	> 7200	2	> 7200	2	> 7200	2	> 7200	2	> 7200	2	> 7200
pw-myciel4	115605	4945	400202	> 7200	411708	> 7200	255557	> 7200	419942	> 7200	396017	> 7200
qiu	9811	84	9543	87	9471	85	9467	86	9483	86	9471	85
rail507	1319	4127	1303	4485	1334	4519	1191	3921	1213	4174	1329	3920
ran16x16	356748	350	365028	380	392537	365	365185	371	377618	385	361003	367
reblock67	83043	335	107761	475	107282	438	91057	555	110107	445	107846	610
rmatr100-p10	901	299	901	299	901	309	901	307	901	306	901	308
rmatr100-p5	397	858	397	881	397	883	397	885	397	882	397	882
rmine6	586282	> 7200	607682	> 7200	563296	5305	561803	6544	578703	6873	607822	7126
rocII-4-11	11288	342	16131	413	14119	388	12484	348	10857	328	9250	306
rococoC10-001000	562119	4846	650113	4766	510849	4016	359182	3362	646531	5183	545121	4990
roll3000	1283482	> 7200	1277802	> 7200	912872	> 7200	1002162	> 7200	1236382	> 7200	1073372	> 7200
satellites1-25	3656	4244	11745	4256	11745	4124	11745	4289	11745	4309	11745	4354
sp98ic	83433	! 6251	109475	> 7201	101909	> 7201	106408	> 7200	115019	> 7200	105718	> 7201
sp98ir	8685	215	5720	142	4831	132	5312	130	5361	128	5902	140
tanglegram1	27	3941	27	4031	27	3914	27	3982	27	3991	27	5135
tanglegram2	3	13	3	14	3	13	3	13	3	13	3	14
timtab1	1014862	773	781984	715	758833	622	744540	609	708098	650	697892	573
triptim1	4	> 7346	3	> 7266	3	> 7233	3	> 7246	4	> 7226	3	> 7261
unitcal_7	12785	3906	15221	4499	15613	4424	12599	3796	12996	3933	16640	3570
vpphard	2	> 7226	2	> 7207	2	> 7201	2	> 7205	2	> 7203	2	> 7205
zib54-UUE	537672	> 7200	547411	7111	564644	7073	546678	7018	546544	7015	570034	7021
geom. mean (solved: 57): 591.69	7110	620.7	7047	591.0	6526	581.2	6715	604.8	6678	587.2	6661	594.3
geom. mean (all): 1379.54	–	1434.4	–	1386.5	–	1353.3	–	1398.4	–	1383.9	–	1375.7
geom. mean (solved: 60): 661.36			8132	661.5	7008	648.3	7186	675.7	7770	659.5	7134	661.9
solved/timeout/abort	60/21/6		61/21/5		62/20/5		63/18/6		62/20/5		63/18/6	
speedup (solved: 57): 1.049			1.05		1.07		1.03		1.06		1.04	

Table 13: Racing only: Effect of communication of incumbents (Eight Solver Threads)

Name	Without comm. Nodes	Time	Run - 1 Nodes	Time	Run - 2 Nodes	Time	Run - 3 Nodes	Time	Run - 4 Nodes	Time	Run - 5 Nodes	Time
30n20b8	5277	> 7207	6696	> 7204	17111	> 7201	7233	> 7204	6775	> 7216	7004	> 7217
acc-tight5	308	156	39	69	39	68	39	75	39	63	39	69
aflow40b	178690	2220	146859	1650	145062	1816	143459	1775	142275	1730	136915	1755
air04	101	129	41	119	44	120	59	119	94	99	44	118
app1-2	47	3991	254	3847	47	3905	47	2439	47	2478	47	3831
ash608gpia-3col	7	248	7	249	7	254	7	475	7	259	7	247
bab5	2211	> 7203	2478	> 7201	3685	> 7202	6790	> 7201	2926	> 7202	1883	> 7202
beasleyC3	479393	> 7200	834752	> 7201	537512	> 7200	556352	> 7200	505657	> 7200	768232	> 7200
biella1	3270	2332	1734	1148	1721	1741	1605	1144	2079	1426	2564	1048
bienst2	84244	460	91266	539	84210	454	87102	530	81682	431	81735	504
binkar10_1	67149	432	119952	371	142863	531	141806	489	129169	459	119952	536
bley_xl1	1	361	1	361	1	361	1	401	1	364	1	363
bnatt350	2831	945	2831	980	2831	693	2831	1533	2831	999	2831	984
core2536-691	192	2566	101	1318	325	2134	101	1298	87	1340	49	1258
cov1075	799632	> 7200	978382	> 7200	873462	> 7200	935292	> 7200	874972	> 7200	944552	> 7200
csched010	527202	> 7200	470959	> 7200	587666	> 7200	622132	> 7200	530962	> 7200	577262	> 7200
danoint	764862	> 7200	729792	> 7200	670262	> 7200	671842	> 7200	746122	> 7200	779562	> 7200
dfn-gwin-UUM	22548	218	20837	125	19914	107	42928	182	20837	123	58127	224
eil33-2	10571	380	10228	350	10508	366	8207	344	9958	359	8561	335
eilB101	7381	805	7897	881	8227	682	5314	702	5593	565	3626	647
enlight13	109200	418	138288	335	125558	201	188996	409	153848	377	167082	434
enlight14	100054	311	100054	298	100054	300	100054	309	100054	292	99882	! 304
ex9	1	191	1	193	1	191	1	189	1	190	1	191
glass4	1082160	2281	1323490	5001	507573	932	709838	1807	1748611	4002	450004	1229
gmu-35-40	1796808	! 3988	1574672	! 2505	2007802	! 2687	2073102	! 2772	1937782	! 2430	2108322	! 2776
iis-100-0-cov	106874	3293	99765	3265	104531	3438	97503	3306	93341	3249	93341	3275
iis-bupa-cov	75290	> 7200	70091	> 7200	78072	> 7199	78442	> 7200	101141	> 7200	77770	> 7200
iis-pima-cov	6523	2368	6547	2021	7155	2058	7589	2076	7321	2810	7321	2060
lectsched-4-obj	1796	265	1184	131	672	108	340	109	771	102	162	113
m100n500k4r1	1	> 7200	2	> 7200	1	> 7200	1	> 7200	2828190	> 7200	2	> 7200
macrophage	114520	> 7200	132312	> 7200	134282	> 7200	46432	> 7200	131312	> 7200	143382	> 7200
map18	347	3595	347	3421	339	3516	347	3571	347	3438	347	3455
map20	333	2148	333	2204	333	2262	333	2363	333	2172	333	1117
mcsched	13508	319	15356	348	12850	345	15074	355	12239	302	12840	340
mik-250-1-100-1	1421995	551	1421995	565	1432187	519	1421995	575	1421995	565	1421995	571
mine-166-5	3582	126	1792	97	2399	112	1848	100	1494	96	3028	112
mine-90-10	74459	1374	96792	1170	101568	1194	60457	719	98243	749	167794	2074
msc98-ip	233	> 7208	24	> 7222	24	> 7245	23	> 7215	24	> 7224	23	> 7212
mspp16	1	! 2566	1	! 2561	1	! 2413	1	! 2574	1	! 2569	1	! 2570
mzzv11	4665	4148	4374	2523	5502	2496	4131	2616	8043	3118	4931	3392
n3div36	25456	> 7201	22031	> 7201	24854	> 7201	25133	> 7201	17862	> 7201	20802	> 7202
n3seq24	4	> 7312	18	> 7222	5	> 7311	6	> 7246	1	! 2140	17	> 7213
n4-3	43434	1677	37108	1195	37831	1479	36245	1416	36261	1403	39036	1504
neos-1109824	8268	355	8164	355	8164	245	8164	360	8164	251	7810	370
neos-1337307	85322	> 7200	88012	> 7200	106106	> 7200	103066	> 7200	106682	> 7200	104462	> 7200
neos-1396125	38909	4258	35099	2834	28818	2812	31703	2699	34698	2802	33686	2792
neos13	912	> 7201	88	2085	63	1862	559	2093	284	2439	337	1880
neos-1601936	9458	6936	8825	> 7200	4556	5120	3711	3677	2123	3326	4969	5463
neos18	5830	73	6318	77	6318	78	5703	69	5450	54	6318	77
neos-476283	372	2632	495	2473	397	2598	443	2631	445	2493	441	2580
neos-686190	3241	164	2692	138	2704	145	2751	144	2767	140	2705	144
neos-849702	20907	803	20907	863	20907	859	17371	856	17371	803	17371	859
neos-916792	66213	933	67052	931	57690	1618	65350	971	67925	947	63971	1063
neos-934278	45	> 7229	41	> 7214	9	> 7227	54	> 7226	48	> 7202	49	> 7215
net12	2932	> 7201	992	> 7202	2001	6359	2412	6237	2119	6391	2418	6973
netdiversion	1	> 7371	2	> 7609	2	> 7562	1	> 7501	-	-	1	! 348
newdano	2058508	7024	2176391	5912	1976816	5849	2587985	> 7200	1659115	6580	1818678	5683
noswot	324848	260	345185	226	281390	185	362474	238	365152	196	362474	222
ns1208400	691	534	691	722	691	691	691	698	691	720	691	705
ns1688347	4538	602	2192	382	1449	311	1435	237	2185	357	2029	299
ns1758913	2	> 7881	-	-	-	-	-	-	-	-	-	-
ns1766074	934754	1871	934754	1833	934754	1884	934754	1916	934754	1896	934754	1893
ns1830653	60186	1178	48303	1034	33323	747	36032	820	37907	799	46111	882
opm2-z7-s2	1978	5401	1627	5261	1714	5706	1365	3799	1768	5599	1602	5762
pg5_34	253046	4173	259057	3129	250574	3126	290705	3606	260489	3148	304596	3885
pigeon-10	1	! 3071	1	> 7200	1	> 7200	1	> 7200	1	> 7200	1	! 3681
pw-myciel4	296556	4666	307406	> 7200	308496	> 7200	309270	> 7200	304966	> 7200	308893	> 7200
qiu	9811	91	9975	87	9692	87	9917	89	9713	87	9961	88
rail507	578	> 7202	373	> 7202	438	> 7205	662	> 7202	643	> 7204	653	> 7203
ran16x16	356748	499	418668	441	299623	346	302695	400	423285	452	316070	546
reblock67	83043	482	83902	489	83293	476	96807	439	81377	452	84623	529
rmatr100-p10	901	415	881	405	886	521	768	428	881	408	881	416
rmatr100-p5	397	2171	397	2121	403	2377	397	2279	397	2317	397	2098
rmine6	284372	> 7200	274192	> 7200	440912	> 7200	269093	> 7200	282942	> 7200	280202	> 7200
rocII-4-11	11288	443	10012	407	11029	418	12366	461	9107	486	10104	397
rococoC10-001000	347586	! 4231	364404	3613	351486	3628	480102	! 4297	596605	5416	633055	6006
roll3000	688092	> 7200	595095	> 7200	587822	> 7200	621274	> 7200	614672	> 7200	663042	> 7200
satellites1-25	11313	> 7203	3479	> 7201	11323	> 7201	3479	> 7202	11745	6990	9260	> 7204
sp98ic	36653	> 7201	38050	> 7201	43735	> 7201	73025	> 7201	45665	> 7201	42167	> 7201
sp98ir	4999	223	4939	164	5555	173	4920	171	4416	161	5105	169
tanglegram1	22	> 7264	22	> 7264	1	! 324	22	> 7231	22	> 7241	11	> 7381
tanglegram2	3	20	3	19	3	19	3	20	3	14	3	19
timtab1	1014862	1027	464378	460	491225	429	496261	495	498709	457	493857	530
triptim1	9	3183	9	3690	9	3746	9	3678	9	3783	9	3760
unitcal_7	8171	> 7201	8657	> 7201	6438	> 7201	6265	> 7201	7447	> 7201	6820	> 7201
vpphard	2	> 7215	2	> 7232	1	> 7213	1	> 7217	2	> 7208	2	> 7208
zib54-UUE	383842	> 7200	408932	> 7200	397992	> 7200	397412	> 7200	412702	> 7200	399582	> 7200
geom. mean (solved: 52): 605.81	5128	706.1	4708	615.4	4440	594.9	4322	610.5	4551	588.8	4330	619.5
geom. mean (all): 1551.98	–	1723.6	–	1557.3	–	1515.9	–	1551.8	–	1512.0	–	1622.8
geom. mean (solved: 53): 617.52			4368	629.7	4097	607.8	4158	624.8	4319	604.8	4126	632.6
solved/timeout/abort	56/27/4		56/28/3		58/25/4		56/27/4		59/23/5		57/24/6	
speedup (solved: 52): 1.166			1.15		1.19		1.16		1.20		1.14	

Table 14: Racing only with communication of incumbents: One vs Four Solver Threads

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	4639	> 7219	902	> 7218	1022	> 7209	820	> 7204	516	> 7203	1174	> 7202
acc-tight5	1808	353	39	62	39	62	39	61	39	61	39	61
aflow40b	668667	6444	181535	1875	181535	1856	181535	1841	175610	1639	198942	1611
air04	215	149	49	97	49	96	49	96	49	96	49	96
app1-2	392	2163	47	1930	47	1893	47	1953	47	1902	47	1923
ash608gpia-3col	7	77	7	101	7	100	7	101	7	99	7	100
bab5	12846	> 7201	16846	> 7201	15454	> 7202	12252	> 7201	15235	> 7201	10966	> 7201
beasleyC3	992738	> 7203	898942	> 7201	956912	> 7200	885932	> 7200	1099922	> 7200	801642	> 7200
biella1	3270	1554	2288	1159	2654	1340	1649	1408	1658	1062	1575	1309
bienst2	86468	439	94998	465	97769	499	104468	506	101544	531	104990	523
binkar10_1	135435	280	165250	395	136563	343	107798	259	121683	292	111086	267
bley_xl1	1	342	1	349	1	349	1	351	1	352	1	338
bnatt350	4134	859	5058	864	5058	863	5058	814	5058	864	5058	861
core2536-691	810	1487	901	1335	901	1337	810	1607	901	1340	810	1575
cov1075	976932	> 7201	944582	> 7200	1078532	> 7200	1077042	> 7200	950542	> 7200	1069092	> 7200
csched010	684362	> 7200	563541	> 7200	580012	> 7200	656582	> 7200	642848	> 7200	683582	> 7200
danoint	870942	> 7201	704140	6653	704140	6662	704140	6619	704140	6684	704140	6643
dfn-gwin-UUM	87613	239	63710	199	63909	201	63551	201	64307	201	61020	186
eil33-2	10571	190	9879	209	9401	201	8975	197	9759	212	9772	208
eilB101	9239	1292	4872	692	4402	562	5689	758	4811	624	5049	659
enlight13	745476	857	367322	395	367322	490	367322	489	367322	496	367322	489
enlight14	259155	316	259155	305	259155	388	259155	556	259155	390	259155	398
ex9	1	194	1	191	1	197	1	192	1	191	1	191
glass4	2990992	! 4138	939959	1544	702114	1167	1050611	1650	2542544	3495	3421535	4949
gmu-35-40	1970052	! 1785	2405472	! 2458	2333392	! 2144	2375502	! 2429	1738622	! 2235	2518052	! 2415
iis-100-0-cov	106874	2669	98732	2731	94430	2598	104420	2863	91513	2490	104420	2855
iis-bupa-cov	113309	> 7202	106992	> 7200	111527	> 7200	106892	> 7200	106616	> 7200	106382	> 7200
iis-pima-cov	12469	1526	7124	1146	7124	1149	7124	1148	7124	1144	7124	1145
lectsched-4-obj	21026	826	4540	264	337	184	3506	237	2794	215	563	243
m100n500k4r1	5050160	> 7201	4959466	> 7200	4975506	> 7200	4915463	> 7200	4857441	> 7200	5004431	> 7200
macrophage	541612	> 7201	307152	> 7201	345402	> 7200	348672	> 7200	381712	> 7200	433012	> 7200
map18	261	1182	493	1427	493	1403	493	1428	493	1465	493	1410
map20	526	926	399	1180	399	1097	399	1567	399	1184	399	1101
mcsched	16113	341	13704	309	20407	406	16861	344	15775	336	14646	312
mik-250-1-100-1	1421995	457	1421995	490	1421995	495	1421995	500	1421995	496	1421995	494
mine-166-5	6170	109	5138	85	3576	83	4167	87	1028	84	2888	82
mine-90-10	189285	1364	291363	2503	65635	494	199943	1949	96075	873	199321	1483
msc98-ip	485	> 7212	45	> 7242	138	> 7209	209	> 7201	130	> 7242	138	> 7220
mspp16	1	! 144	1	! 2544	1	! 2551	1	! 2536	1	! 2562	1	! 2575
mzzv11	8361	1868	4048	1147	6993	1612	4461	1264	4806	1305	4152	1227
n3div36	37993	! 2825	40083	! 4683	56160	! 5047	50186	! 4698	52637	! 4788	35660	! 4707
n3seq24	1	! 72	152	> 7209	193	> 7210	285	> 7242	156	> 7225	1	! 674
n4-3	57354	1466	49902	1301	51789	1403	46044	1293	48271	1355	50043	1406
neos-1109824	20142	283	8164	171	8164	285	8164	171	8164	171	8164	175
neos-1337307	239423	> 7201	204864	> 7201	208705	> 7200	214352	> 7200	211707	> 7200	211819	> 7200
neos-1396125	40087	4125	39799	4152	42121	4796	41808	4444	44725	4925	43032	4750
neos13	17847	> 7201	4882	3741	50	2952	46	3674	6882	4596	45	2895
neos-1601936	7771	5311	12605	> 7200	3967	3662	6206	6749	14135	> 7200	3191	3404
neos18	6827	81	6292	70	6292	70	6292	70	6292	71	6292	70
neos-476283	1	! 273	1	! 439	1	! 2840	12	! 869	31	! 855	70	! 1117
neos-686190	4739	134	3589	118	4306	129	3018	109	3679	123	3970	124
neos-849702	126716	3193	20907	788	20907	786	20907	790	20907	788	20907	788
neos-916792	66213	511	66296	584	66044	581	68269	836	64479	594	69786	641
neos-934278	2755	> 7206	712	> 7206	207	> 7205	171	> 7201	291	> 7222	209	> 7202
net12	3678	4536	3850	3314	3056	4018	3484	4383	4464	4095	4721	4216
netdiversion	26	> 7295	15	> 7315	14	> 7353	-	-	23	> 7317	16	> 7320
newdano	2842106	> 7201	2870908	6711	2719657	> 7200	1898642	4595	2394587	7152	2791732	6957
noswot	575618	217	301829	159	301829	158	301829	164	3018529	170	301829	163
ns1208400	4817	1361	691	506	691	507	691	507	691	507	691	510
ns1688347	4538	528	3515	483	3752	476	4538	536	4310	505	2944	383
ns1758913	1	! 3132	1	! 7898	1	! 5011	1	! 4796	1	! 4709	-	-
ns1766074	945109	1318	934754	1194	934754	1190	934754	1188	934754	1170	934754	1180
ns1830653	88276	1111	38073	677	48232	810	43242	748	39359	655	37962	679
opm2-z7-s2	1978	1142	3233	2833	2321	2693	2156	4070	2485	2744	2285	2416
pg5_34	318742	2339	292798	2336	318401	2726	310197	2319	309644	2589	267308	1752
pigeon-10	2955734	> 7200	2	> 7200	2	> 7200	2	> 7200	2	> 7200	2	> 7200
pw-myciel4	488828	> 7201	400202	> 7200	411708	> 7200	255557	> 7200	419942	> 7200	396017	> 7200
qiu	9811	83	9543	87	9471	85	9467	86	9483	86	9471	85
rail507	1319	2126	1303	4485	1334	4519	1191	3921	1213	4174	1329	3920
ran16x16	441109	396	365028	380	392537	365	365185	371	377618	385	361003	367
reblock67	83043	339	107761	475	107282	438	91057	555	110107	445	107846	610
rmatr100-p10	901	298	901	299	901	309	901	307	901	306	901	308
rmatr100-p5	397	855	397	881	397	883	397	885	397	882	397	882
rmine6	687364	6508	607682	> 7200	563296	5305	561803	6544	578703	6873	607822	7126
rocII-4-11	13194	399	16131	413	14119	388	12484	348	10857	328	9250	306
rococoC10-001000	1075742	> 7201	650113	4766	510849	4016	359182	3362	646531	5183	545121	4990
roll3000	1322672	> 7201	1277802	> 7200	912872	> 7200	1002162	> 7200	1236382	> 7200	1073372	> 7200
satellites1-25	16711	4956	11745	4256	11745	4124	11745	4289	11745	4309	11745	4354
sp98ic	40930	! 5053	109475	> 7201	101909	> 7201	106408	> 7200	115019	> 7200	105718	> 7201
sp98ir	8984	202	5720	142	4831	132	5312	130	5361	128	5902	140
tanglegram1	27	2358	27	4031	27	3914	27	3982	27	3991	27	5135
tanglegram2	3	13	3	14	3	13	3	13	3	13	3	14
timtab1	1014862	746	781984	715	758833	622	744540	609	708098	650	697892	573
triptim1	39	> 7207	3	> 7266	3	> 7233	3	> 7246	4	> 7226	3	> 7261
unitcal_7	12785	2501	15221	4499	15613	4424	12599	3796	12996	3933	16640	3570
vpphard	3693	> 7206	2	> 7207	2	> 7201	2	> 7205	2	> 7203	2	> 7205
zib54-UUE	565989	6971	547411	7111	564644	7073	546678	7018	546544	7015	570034	7021
geom. mean (solved: 56): 582.89	9173	670.3	6438	585.2	6014	579.8	6182	602.0	5987	571.8	5963	575.7
geom. mean (all): 1379.54	–	1554.6	–	1386.5	–	1353.3	–	1398.4	–	1383.9	–	1375.7
geom. mean (solved: 60): 661.36			8132	661.5	7008	648.3	7186	675.7	7770	659.5	7134	661.9
solved/timeout/abort	58/21/8		61/21/5		62/20/5		63/18/6		62/20/5		63/18/6	
speedup (solved: 56): 1.150			1.15		1.16		1.11		1.17		1.16	

Table 15: Racing only with communication of incumbents: One vs Eight SOLVER Threads

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	4639	> 7219	6696	> 7204	17111	> 7201	7233	> 7204	6775	> 7216	7004	> 7217
acc-tight5	1808	353	39	69	39	68	39	75	39	63	39	69
aflow40b	668667	6444	146859	1650	145062	1816	143459	1775	142275	1730	136915	1755
air04	215	149	41	119	44	120	59	119	94	99	44	118
app1-2	392	2163	254	3847	47	3905	47	2439	47	2478	47	3831
ash608gpia-3col	7	77	7	249	7	254	7	475	7	259	7	247
bab5	12846	> 7201	2478	> 7201	3685	> 7202	6790	> 7201	2926	> 7202	1883	> 7202
beasleyC3	992738	> 7203	834752	> 7201	537512	> 7200	556352	> 7200	505657	> 7200	768232	> 7200
biella1	3270	1554	1734	1148	1721	1741	1605	1144	2079	1426	2564	1048
bienst2	86468	439	91266	539	84210	454	87102	530	81682	431	81735	504
binkar10_1	135435	280	119952	371	142863	531	141806	489	129169	459	119952	536
bley_xl1	1	342	1	361	1	361	1	401	1	364	1	363
bnatt350	4134	859	2831	980	2831	693	2831	1533	2831	999	2831	984
core2536-691	810	1487	101	1318	325	2134	101	1298	87	1340	49	1258
cov1075	976932	> 7201	978382	> 7200	873462	> 7200	935292	> 7200	874972	> 7200	944552	> 7200
csched010	684362	> 7200	470959	> 7200	587666	> 7200	622132	> 7200	530962	> 7200	577262	> 7200
danoint	870942	> 7201	729792	> 7200	670262	> 7200	671842	> 7200	746122	> 7200	779562	> 7200
dfn-gwin-UUM	87613	239	20837	125	19914	107	42928	182	20837	123	58127	224
eil33-2	10571	190	10228	350	10508	366	8207	344	9958	359	8561	335
eilB101	9239	1292	7897	881	8227	682	5314	702	5593	565	3626	647
enlight13	745476	857	138288	335	125558	201	188996	409	153848	377	167082	434
enlight14	259155	316	100054	298	100054	300	100054	309	100054	292	99882	! 304
ex9	1	194	1	193	1	191	1	189	1	190	1	191
glass4	2990992	! 4138	1323490	5001	507573	932	709838	1807	1748611	4002	450004	1229
gmu-35-40	1970052	! 1785	1574672	! 2505	2007802	! 2687	2073102	! 2772	1937782	! 2430	2108322	! 2776
iis-100-0-cov	106874	2669	99765	3265	104531	3438	97503	3306	93341	3249	93341	3275
iis-bupa-cov	113309	> 7202	70091	> 7200	78072	> 7199	78442	> 7200	101141	> 7200	77770	> 7200
iis-pima-cov	12469	1526	6547	2021	7155	2058	7589	2076	7321	2810	7321	2060
lectsched-4-obj	21026	826	1184	131	672	108	340	109	771	102	162	113
m100n500k4r1	5050160	> 7201	2	> 7200	1	> 7200	1	> 7200	2828190	> 7200	2	> 7200
macrophage	541612	> 7201	132312	> 7200	134282	> 7200	46432	> 7200	131312	> 7200	143382	> 7200
map18	261	1182	347	3421	339	3516	347	3571	347	3438	347	3455
map20	526	926	333	2204	333	2262	333	2363	333	2172	333	1117
mcsched	16113	341	15356	348	12850	345	15074	355	12239	302	12840	340
mik-250-1-100-1	1421995	457	1421995	565	1432187	519	1421995	575	1421995	565	1421995	571
mine-166-5	6170	109	1792	97	2399	112	1848	100	1494	96	3028	112
mine-90-10	189285	1364	96792	1170	101568	1194	60457	719	98243	749	167794	2074
msc98-ip	485	> 7212	24	> 7222	24	> 7245	23	> 7215	24	> 7224	23	> 7212
mspp16	1	! 144	1	! 2561	1	! 2413	1	! 2574	1	! 2569	1	! 2570
mzzv11	8361	1868	4374	2523	5502	2496	4131	2616	8043	3118	4931	3392
n3div36	37993	! 2825	22031	> 7201	24854	> 7201	25133	> 7201	17862	> 7201	20802	> 7202
n3seq24	1	! 72	18	> 7222	5	> 7311	6	> 7246	1	! 2140	17	> 7213
n4-3	57354	1466	37108	1195	37831	1479	36245	1416	36261	1403	39036	1504
neos-1109824	20142	283	8164	355	8164	245	8164	360	8164	251	7810	370
neos-1337307	239423	> 7201	88012	> 7200	106106	> 7200	103066	> 7200	106682	> 7200	104462	> 7200
neos-1396125	40087	4125	35099	2834	28818	2812	31703	2699	34698	2802	33686	2792
neos13	17847	> 7201	88	2085	63	1862	559	2093	284	2439	337	1880
neos-1601936	7771	5311	8825	> 7200	4556	5120	3711	3677	2123	3326	4969	5463
neos18	6827	81	6318	77	6318	78	5703	69	5450	54	6318	77
neos-476283	1	! 273	495	2473	397	2598	443	2631	445	2493	441	2580
neos-686190	4739	134	2692	138	2704	145	2751	144	2767	140	2705	144
neos-849702	126716	3193	20907	863	20907	859	17371	856	17371	803	17371	859
neos-916792	66213	511	67052	931	57690	1618	65350	971	67925	947	63971	1063
neos-934278	2755	> 7206	41	> 7214	9	> 7227	54	> 7226	48	> 7202	49	> 7215
net12	3678	4536	992	> 7202	2001	6359	2412	6237	2119	6391	2418	6973
netdiversion	26	> 7295	2	> 7609	2	> 7562	1	> 7501	-	-	1	! 348
newdano	2842106	> 7201	2176391	5912	1976816	5849	2587985	> 7200	1659115	6580	1818678	5683
noswot	575618	217	345185	226	281390	185	362474	238	365152	196	362474	222
ns1208400	4817	1361	691	722	691	691	691	698	691	720	691	705
ns1688347	4538	528	2192	382	1449	311	1435	237	2185	357	2029	299
ns1758913	1	! 3132	-	-	-	-	-	-	-	-	-	-
ns1766074	945109	1318	934754	1833	934754	1884	934754	1916	934754	1896	934754	1893
ns1830653	88276	1111	48303	1034	33323	747	36032	820	37907	799	46111	882
opm2-z7-s2	1978	1142	1627	5261	1714	5706	1365	3799	1768	5599	1602	5762
pg5_34	318742	2339	259057	3129	250574	3126	290705	3606	260489	3148	304596	3885
pigeon-10	2955734	> 7200	1	> 7200	1	> 7200	1	> 7200	1	> 7200	1	! 3681
pw-myciel4	488828	> 7201	307406	> 7200	308496	> 7200	309270	> 7200	304966	> 7200	308893	> 7200
qiu	9811	83	9975	87	9692	87	9917	89	9713	87	9961	88
rail507	1319	2126	373	> 7202	438	> 7205	662	> 7202	643	> 7204	653	> 7203
ran16x16	441109	396	418668	441	299623	346	302695	400	423285	452	316070	546
reblock67	83043	339	83902	489	83293	476	96807	439	81377	452	84623	529
rmatr100-p10	901	298	881	405	886	521	768	428	881	408	881	416
rmatr100-p5	397	855	397	2121	403	2377	397	2279	397	2317	397	2098
rmine6	687364	6508	274192	> 7200	440912	> 7200	269093	> 7200	282942	> 7200	280202	> 7200
rocII-4-11	13194	399	10012	407	11029	418	12366	461	9107	486	10104	397
rococoC10-001000	1075742	> 7201	364404	3613	351486	3628	480102	! 4297	596605	5416	633055	6006
roll3000	1322672	> 7201	595095	> 7200	587822	> 7200	621274	> 7200	614672	> 7200	663042	> 7200
satellites1-25	16711	4956	3479	> 7201	11323	> 7201	3479	> 7202	11745	6990	9260	> 7204
sp98ic	40930	! 5053	38050	> 7201	43735	> 7201	73025	> 7201	45665	> 7201	42167	> 7201
sp98ir	8984	202	4939	164	5555	173	4920	171	4416	161	5105	169
tanglegram1	27	2358	22	> 7264	1	! 324	22	> 7231	22	> 7241	11	> 7381
tanglegram2	3	13	3	19	3	19	3	20	3	14	3	19
timtab1	1014862	746	464378	460	491225	429	496261	495	498709	457	493857	530
triptim1	39	> 7207	9	3690	9	3746	9	3678	9	3783	9	3760
unitcal_7	12785	2501	8657	> 7201	6438	> 7201	6265	> 7201	7447	> 7201	6820	> 7201
vpphard	3693	> 7206	2	> 7232	1	> 7213	1	> 7217	2	> 7208	2	> 7208
zib54-UUE	565989	6971	408932	> 7200	397992	> 7200	397412	> 7200	412702	> 7200	399582	> 7200
geom. mean (solved: 49): 552.68	9226	555.2	4993	552.6	4805	550.9	4628	558.7	4799	529.3	4681	571.9
geom. mean (all): 1551.98	–	1554.6	–	1557.3	–	1515.9	–	1551.8	–	1512.0	–	1622.8
geom. mean (solved: 53): 617.52			4368	629.7	4097	607.8	4158	624.8	4319	604.8	4126	632.6
solved/timeout/abort	58/21/8		56/28/3		58/25/4		56/27/4		59/23/5		57/24/6	
speedup (solved: 49): 1.005			1.00		1.01		0.99		1.05		0.97	

Table 16: Variation of solving time and number of nodes when doing only racing with communication of incumbents for the 63 instances that were solved at least once. The number of the cases an instance was solved is shown in parenthesis in the first column, if it is less than five. (Four SOLVER Threads)

	Number of enumerated nodes				Computing Time (sec.)			
Name	Mean	Std.Dev.	Min.	Max.	Mean	Std.Dev.	Min.	Max.
acc-tight5	39.0	0.00	39(100.00%)	39(100.00%)	61.4	0.49	61 (99.35%)	62(100.98%)
aflow40b	183831.4	7896.10	175610 (95.53%)	198942(108.22%)	1764.4	114.67	1611 (91.31%)	1875(106.27%)
air04	49.0	0.00	49(100.00%)	49(100.00%)	96.2	0.40	96 (99.79%)	97(100.83%)
app1-2	47.0	0.00	47(100.00%)	47(100.00%)	1920.2	21.22	1893 (98.58%)	1953(101.71%)
ash608gpia-3col	7.0	0.00	7(100.00%)	7(100.00%)	100.2	0.75	99 (98.80%)	101(100.80%)
biella1	1964.8	430.18	1575 (80.16%)	2654(135.08%)	1255.6	126.50	1062 (84.58%)	1408(112.14%)
bienst2	100753.8	3857.55	94998 (94.29%)	104990(104.20%)	504.8	22.96	465 (92.12%)	531(105.19%)
binkar10_1	128476.0	20946.90	107798 (83.91%)	165250(128.62%)	311.2	51.15	259 (83.23%)	395(126.93%)
bley_xl1	1.0	0.00	1(100.00%)	1(100.00%)	347.8	5.04	338 (97.18%)	352(101.21%)
bnatt350	5058.0	0.00	5058(100.00%)	5058(100.00%)	853.2	19.63	814 (95.41%)	864(101.27%)
core2536-691	864.6	44.58	810 (93.68%)	901(104.21%)	1438.8	124.69	1335 (92.79%)	1607(111.69%)
danoint	704140.0	0.00	704140(100.00%)	704140(100.00%)	6652.2	21.42	6619 (99.50%)	6684(100.48%)
dfn-gwin-UUM	63299.4	1167.43	61020 (96.40%)	64307(101.59%)	197.6	5.85	186 (94.13%)	201(101.72%)
eil33-2	9557.2	332.80	8975 (93.91%)	9879(103.37%)	205.4	5.54	197 (95.91%)	212(103.21%)
eilB101	4964.6	419.66	4402 (88.67%)	5689(114.59%)	659.0	65.61	562 (85.28%)	758(115.02%)
enlight13	367322.0	0.00	367322(100.00%)	367322(100.00%)	471.8	38.49	395 (83.72%)	496(105.13%)
enlight14	259155.0	0.00	259155(100.00%)	259155(100.00%)	407.4	81.65	305 (74.86%)	556(136.48%)
ex9	1.0	0.00	1(100.00%)	1(100.00%)	192.4	2.33	191 (99.27%)	197(102.39%)
glass4	1731352.6	1064311.33	702114 (40.55%)	3421535(197.62%)	2561.0	1441.00	1167 (45.57%)	4949(193.24%)
iis-100-0-cov	98703.0	5202.38	91513 (92.72%)	104420(105.79%)	2707.4	145.45	2490 (91.97%)	2863(105.75%)
iis-pima-cov	7124.0	0.00	7124(100.00%)	7124(100.00%)	1146.4	1.85	1144 (99.79%)	1149(100.23%)
lectsched-4-obj	2348.0	1647.73	337 (14.35%)	4540(193.36%)	228.6	27.22	184 (80.49%)	264(115.49%)
map18	493.0	0.00	493(100.00%)	493(100.00%)	1426.6	21.49	1403 (98.35%)	1465(102.69%)
map20	399.0	0.00	399(100.00%)	399(100.00%)	1225.8	174.60	1097 (89.49%)	1567(127.83%)
mcsched	16278.6	2320.79	13704 (84.18%)	20407(125.36%)	341.4	34.99	309 (90.51%)	406(118.92%)
mik-250-1-100-1	1421995.0	0.00	1421995(100.00%)	1421995(100.00%)	495.0	3.22	490 (98.99%)	500(101.01%)
mine-166-5	3359.4	1379.87	1028 (30.60%)	5138(152.94%)	84.2	1.72	82 (97.39%)	87(103.33%)
mine-90-10	170467.4	81044.62	65635 (38.50%)	291363(170.92%)	1460.4	721.83	494 (33.83%)	2503(171.39%)
mzzv11	4892.0	1083.26	4048 (82.75%)	6993(142.95%)	1311.0	159.25	1147 (87.49%)	1612(122.96%)
n4-3	49209.8	1935.42	46044 (93.57%)	51789(105.24%)	1351.6	48.18	1293 (95.66%)	1406(104.02%)
neos-1109824	8164.0	0.00	8164(100.00%)	8164(100.00%)	194.6	45.23	171 (87.87%)	285(146.45%)
neos-1396125	42297.0	1609.07	39799 (94.09%)	44725(105.74%)	4613.4	279.53	4152 (90.00%)	4925(106.75%)
neos-1601936 (3)	4454.7	1278.26	3191 (71.63%)	6206(139.31%)	4605.0	1519.69	3404 (73.92%)	6749(146.56%)
neos-686190	3712.4	428.31	3018 (81.30%)	4306(115.99%)	120.6	6.77	109 (90.38%)	129(106.97%)
neos-849702	20907.0	0.00	20907(100.00%)	20907(100.00%)	788.0	1.26	786 (99.75%)	790(100.25%)
neos-916792	66974.8	1851.16	64479 (96.27%)	69786(104.20%)	647.2	96.84	581 (89.77%)	836(129.17%)
neos13	2381.0	2927.68	45 (1.89%)	6882(289.04%)	3571.6	621.33	2895 (81.06%)	4596(128.68%)
neos18	6292.0	0.00	6292(100.00%)	6292(100.00%)	70.2	0.40	70 (99.72%)	71(101.14%)
net12	3915.0	613.01	3056 (78.06%)	4721(120.59%)	4005.2	366.92	3314 (82.74%)	4383(109.43%)
newdano (4)	2488967.2	385660.72	1898642 (76.28%)	2870908(115.35%)	6353.8	1027.37	4595 (72.32%)	7152(112.56%)
noswot	301829.0	0.00	301829(100.00%)	301829(100.00%)	162.8	4.26	158 (97.05%)	170(104.42%)
ns1208400	691.0	0.00	691(100.00%)	691(100.00%)	507.4	1.36	506 (99.72%)	510(100.51%)
ns1688347	3811.8	569.26	2944 (77.23%)	4538(119.05%)	476.6	51.26	383 (80.36%)	536(112.46%)
ns1766074	934754.0	0.00	934754(100.00%)	934754(100.00%)	1184.4	8.52	1170 (98.78%)	1194(100.81%)
ns1830653	41373.6	3927.34	37962 (91.75%)	48232(116.58%)	713.8	57.35	655 (91.76%)	810(113.48%)
opm2-z7-s2	2496.0	383.15	2156 (86.38%)	3233(129.53%)	2951.2	576.49	2416 (81.87%)	4070(137.91%)
pg5_34	299669.6	18197.10	267308 (89.20%)	318401(106.25%)	2344.4	333.80	1752 (74.73%)	2726(116.28%)
qiu	9487.0	28.51	9467 (99.79%)	9543(100.59%)	85.8	0.75	85 (99.07%)	87(101.40%)
rail507	1274.0	60.13	1191 (93.49%)	1334(104.71%)	4203.8	260.70	3920 (93.25%)	4519(107.50%)
ran16x16	372274.2	11566.69	361003 (96.97%)	392537(105.44%)	373.6	7.68	365 (97.70%)	385(103.05%)
reblock67	104810.6	6946.07	91057 (86.88%)	110107(105.05%)	504.6	67.10	438 (86.80%)	610(120.89%)
rmatr100-p10	901.0	0.00	901(100.00%)	901(100.00%)	305.8	3.54	299 (97.78%)	309(101.05%)
rmatr100-p5	397.0	0.00	397(100.00%)	397(100.00%)	882.6	1.36	881 (99.82%)	885(100.27%)
rmine6 (4)	577906.0	18495.68	561803 (97.21%)	607822(105.18%)	6462.0	699.14	5305 (82.10%)	7126(110.28%)
rocII-4-11	12568.2	2410.16	9250 (73.60%)	16131(128.35%)	356.6	39.04	306 (85.81%)	413(115.82%)
rococoC10-001000	542359.2	106786.75	359182 (66.23%)	650113(119.87%)	4463.4	678.23	3362 (75.32%)	5183(116.12%)
satellites1-25	11745.0	0.00	11745(100.00%)	11745(100.00%)	4266.4	77.96	4124 (96.66%)	4354(102.05%)
sp98ir	5425.2	369.93	4831 (89.05%)	5902(108.79%)	134.4	5.57	128 (95.24%)	142(105.65%)
tanglegram1	27.0	0.00	27(100.00%)	27(100.00%)	4210.6	463.73	3914 (92.96%)	5135(121.95%)
tanglegram2	3.0	0.00	3(100.00%)	3(100.00%)	13.4	0.49	13 (97.01%)	14(104.48%)
timtab1	738269.4	31348.99	697892 (94.53%)	781984(105.92%)	633.8	47.55	573 (90.41%)	715(112.81%)
unitcal_7	14613.8	1558.80	12599 (86.21%)	16640(113.86%)	4044.4	360.53	3570 (88.27%)	4499(111.24%)
zib54-UUE	555062.2	10172.13	546544 (98.47%)	570034(102.70%)	7047.6	38.24	7015 (99.54%)	7111(100.90%)

Table 17: Variation of solving time and number of nodes when doing only racing with communication of incumbents for the 59 instances that were solved at least once. The number of the cases an instance was solved is shown in parenthesis in the first column, if it is less than five. (Eight SOLVER Threads)

Name	Number of enumerated nodes				Computing Time (sec.)			
	Mean	Std.Dev.	Min.	Max.	Mean	Std.Dev.	Min.	Max.
acc-tight5	39.0	0.00	39(100.00%)	39(100.00%)	68.8	3.82	63 (91.57%)	75(109.01%)
aflow40b	142914.0	3372.54	136915 (95.80%)	146859(102.76%)	1745.2	55.30	1650 (94.55%)	1816(104.06%)
air04	56.4	19.83	41 (72.70%)	94(166.67%)	115.0	8.02	99 (86.09%)	120(104.35%)
app1-2	88.4	82.80	47 (53.17%)	254(287.33%)	3300.0	687.63	2439 (73.91%)	3905(118.33%)
ash608gpia-3col	7.0	0.00	7(100.00%)	7(100.00%)	296.8	89.20	247 (83.22%)	475(160.04%)
biella1	1940.6	349.66	1605 (82.71%)	2564(132.12%)	1301.4	253.49	1048 (80.53%)	1741(133.78%)
bienst2	85199.0	3627.03	81682 (95.87%)	91266(107.12%)	491.6	42.33	431 (87.67%)	539(109.64%)
binkar10_1	130748.4	10046.42	119952 (91.74%)	142863(109.27%)	477.2	60.17	371 (77.75%)	536(112.32%)
bley_xl1	1.0	0.00	1(100.00%)	1(100.00%)	370.0	15.54	361 (97.57%)	401(108.38%)
bnatt350	2831.0	0.00	2831(100.00%)	2831(100.00%)	1037.8	272.71	693 (66.78%)	1533(147.72%)
core2536-691	132.6	98.06	49 (36.95%)	325(245.10%)	1469.6	333.29	1258 (85.60%)	2134(145.21%)
dfn-gwin-UUM	32528.6	15465.71	19914 (61.22%)	58127(178.70%)	152.2	44.00	107 (70.30%)	224(147.17%)
eil33-2	9492.4	928.34	8207 (86.46%)	10508(110.70%)	350.8	10.91	335 (95.50%)	366(104.33%)
eilB101	6131.4	1717.20	3626 (59.14%)	8227(134.18%)	695.4	103.94	565 (81.25%)	881(126.69%)
enlight13	154754.4	22130.75	125558 (81.13%)	188996(122.13%)	351.2	82.08	201 (57.23%)	434(123.58%)
enlight14 (4)	100054.0	0.00	100054(100.00%)	100054(100.00%)	299.8	6.10	292 (97.41%)	309(103.09%)
ex9	1.0	0.00	1(100.00%)	1(100.00%)	190.8	1.33	189 (99.06%)	193(101.15%)
glass4	947903.2	506096.39	450004 (47.47%)	1748611(184.47%)	2594.2	1613.75	932 (35.93%)	5001(192.78%)
iis-100-0-cov	97696.2	4218.23	93341 (95.54%)	104531(107.00%)	3306.6	68.28	3249 (98.26%)	3438(103.97%)
iis-pima-cov	7186.6	348.75	6547 (91.10%)	7589(105.60%)	2205.0	303.04	2021 (91.66%)	2810(127.44%)
lectsched-4-obj	625.8	355.41	162 (25.89%)	1184(189.20%)	112.6	9.85	102 (90.59%)	131(116.34%)
map18	345.4	3.20	339 (98.15%)	347(100.46%)	3480.2	55.58	3421 (98.30%)	3571(102.61%)
map20	333.0	0.00	333(100.00%)	333(100.00%)	2023.6	457.93	1117 (55.20%)	2363(116.77%)
mcsched	13671.8	1282.41	12239 (89.52%)	15356(112.32%)	338.0	18.64	302 (89.35%)	355(105.03%)
mik-250-1-100-1	1424033.4	4076.80	1421995 (99.86%)	1432187(100.57%)	559.0	20.36	519 (92.84%)	575(102.86%)
mine-166-5	2112.2	543.22	1494 (70.73%)	3028(143.36%)	103.4	7.14	96 (92.84%)	112(108.32%)
mine-90-10	104970.8	34790.71	60457 (57.59%)	167794(159.85%)	1181.2	489.45	719 (60.87%)	2074(175.58%)
mzzv11	5396.2	1405.62	4131 (76.55%)	8043(149.05%)	2829.0	360.66	2496 (88.23%)	3392(119.90%)
n4-3	37296.2	1051.19	36245 (97.18%)	39036(104.66%)	1399.4	108.95	1195 (85.39%)	1504(107.47%)
neos-1109824	8093.2	141.60	7810 (96.50%)	8164(100.87%)	316.2	55.93	245 (77.48%)	370(117.01%)
neos-1396125	32800.8	2312.25	28818 (87.86%)	35099(107.01%)	2787.8	46.53	2699 (96.81%)	2834(101.66%)
neos-1601936 (4)	3839.8	1089.96	2123 (55.29%)	4969(129.41%)	4396.5	911.66	3326 (75.65%)	5463(124.26%)
neos-476283	444.2	31.05	397 (89.37%)	495(111.44%)	2555.0	61.35	2473 (96.79%)	2631(102.97%)
neos-686190	2723.8	29.54	2692 (98.83%)	2767(101.59%)	142.2	2.71	138 (97.05%)	145(101.97%)
neos-849702	18785.4	1732.28	17371 (92.47%)	20907(111.29%)	848.0	22.61	803 (94.69%)	863(101.77%)
neos-916792	64397.6	3621.31	57690 (89.58%)	67925(105.48%)	1106.0	260.05	931 (84.18%)	1618(146.29%)
neos13	266.2	181.17	63 (23.67%)	559(209.99%)	2071.8	207.97	1862 (89.87%)	2439(117.72%)
neos18	6021.4	371.97	5450 (90.51%)	6318(104.93%)	71.0	9.10	54 (76.06%)	78(109.86%)
net12 (4)	2237.5	182.35	2001 (89.43%)	2418(108.07%)	6490.0	284.72	6237 (96.10%)	6973(107.44%)
newdano (4)	1907750.0	191501.49	1659115 (86.97%)	2176391(114.08%)	6006.0	341.79	5683 (94.62%)	6580(109.56%)
noswot	343335.0	31777.95	281390 (81.96%)	365152(106.35%)	213.4	19.73	185 (86.69%)	238(111.53%)
ns1208400	691.0	0.00	691(100.00%)	691(100.00%)	707.2	12.12	691 (97.71%)	722(102.09%)
ns1688347	1858.0	344.66	1435 (77.23%)	2192(117.98%)	317.2	50.17	237 (74.72%)	382(120.43%)
ns1766074	934754.0	0.00	934754(100.00%)	934754(100.00%)	1884.4	27.75	1833 (97.27%)	1916(101.68%)
ns1830653	40335.2	5838.33	33323 (82.62%)	48303(119.75%)	856.4	98.78	747 (87.23%)	1034(120.74%)
opm2-z7-s2	1615.2	138.58	1365 (84.51%)	1768(109.46%)	5225.4	734.08	3799 (72.70%)	5762(110.27%)
pg5_34	273084.2	20811.52	250574 (91.76%)	304596(111.54%)	3378.8	312.23	3126 (92.52%)	3885(114.98%)
qiu	9851.6	123.41	9692 (98.38%)	9975(101.25%)	87.6	0.80	87 (99.32%)	89(101.60%)
ran16x16	352068.2	56553.42	299623 (85.10%)	423285(120.23%)	437.0	65.99	346 (79.18%)	546(124.94%)
reblock67	86000.4	5509.79	81377 (94.62%)	96807(112.57%)	477.0	31.36	439 (92.03%)	529(110.90%)
rmatr100-p10	859.4	45.74	768 (89.36%)	886(103.10%)	435.6	43.44	405 (92.98%)	521(119.61%)
rmatr100-p5	398.2	2.40	397 (99.70%)	403(101.21%)	2238.4	110.03	2098 (93.73%)	2377(106.19%)
rocII-4-11	10523.6	1104.03	9107 (86.54%)	12366(117.51%)	433.8	34.02	397 (91.52%)	486(112.03%)
rococoC10-001000 (4)	486387.5	129168.15	351486 (72.26%)	633055(130.15%)	4665.8	1065.87	3613 (77.44%)	6006(128.73%)
satellites1-25 (1)	11745.0	0.00	11745(100.00%)	11745(100.00%)	6990.0	0.00	6990(100.00%)	6990(100.00%)
sp98ir	4987.0	365.89	4416 (88.55%)	5555(111.39%)	167.6	4.45	161 (96.06%)	173(103.22%)
tanglegram2	3.0	0.00	3(100.00%)	3(100.00%)	18.2	2.14	14 (76.92%)	20(109.89%)
timtab1	488886.0	12503.65	464378 (94.99%)	498709(102.01%)	474.2	34.89	429 (90.47%)	530(111.77%)
triptim1	9.0	0.00	9(100.00%)	9(100.00%)	3731.4	40.64	3678 (98.57%)	3783(101.38%)

Table 18: Tree search after racing ramp-up (Four SOLVER Threads)

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	4639	> 7219	82016	> 7264	52395	4203	56947	5196	89225	> 7264	71511	6099
acc-tight5	1808	353	39	61	39	61	39	61	39	61	39	61
aflow40b	668667	6444	164816	655	122697	596	77394	468	63537	582	147765	653
air04	215	149	49	96	49	96	44	97	49	96	49	96
app1-2	392	2163	47	1940	47	1943	44	2769	47	1899	47	1905
ash608gpia-3col	7	77	7	103	7	103	7	101	7	102	7	100
bab5	12846	> 7201	147018	> 7313	131523	> 7312	96201	> 7313	75320	> 7313	107256	> 7312
beasleyC3	992738	> 7203	4068580	> 7260	3633095	> 7260	2583438	> 7260	3643541	> 7260	3831586	> 7260
biella1	3270	1554	19168	2988	22710	3427	24806	3464	20208	3374	29093	3762
bienst2	86468	439	127363	396	121624	457	120472	479	121156	455	116227	457
binkar10_1	135435	280	16391	354	17191	375	17485	375	17424	383	16424	370
bley_xl1	1	342	1	350	1	350	1	349	1	349	1	350
bnatt350	4134	859	5058	860	5058	861	5058	859	5058	820	5058	863
core2536-691	810	1487	810	1564	810	1562	462	1267	901	1273	901	1351
cov1075	976932	> 7201	1269446	> 7260	1204434	> 7260	1249083	> 7260	1220968	> 7260	1245627	> 7260
csched010	684362	> 7200	1379112	4178	1215300	3975	1179608	3894	1308398	4163	1096865	3682
danoint	870942	> 7201	733418	1988	703213	1950	718611	2021	700568	1940	754964	2072
dfn-gwin-UUM	87613	239	6489	174	10677	180	6442	167	6532	176	6384	169
eil33-2	10571	190	11172	227	7928	169	6398	213	7041	227	4759	219
eilB101	9239	1292	4069	846	5322	639	5752	736	5739	767	4750	804
enlight13	745476	857	4285351	1903	3517588	1712	10317209	6611	2641311	1356	3403198	1718
enlight14	259155	316	259155	303	259155	379	259155	387	259155	311	259155	391
ex9	1	194	1	192	1	193	1	190	1	191	1	192
glass4	2990992	! 4138	17278266	> 7260	17334282	> 7260	19570349	> 7260	13187482	5588	25731787	> 7260
gmu-35-40	1970052	! 1785	14913239	! 2286	32530467	! 3997	28594116	! 3518	17856936	! 2602	16319902	! 2477
iis-100-0-cov	106874	2669	55739	1247	55712	1207	56185	1207	57079	1198	55801	1187
iis-bupa-cov	113309	> 7202	149362	3728	149433	3728	149295	3736	143644	3817	143527	3822
iis-pima-cov	12469	1526	2882	1765	2882	1763	2882	1808	2882	1765	2882	1766
lectsched-4-obj	21026	826	3347	214	36875	507	5785	211	5919	217	3366	214
m100n500k4r1	5050160	> 7201	20703390	> 7260	20732526	> 7260	20519577	> 7260	20699432	> 7260	20537627	> 7260
macrophage	541612	> 7201	3334860	> 7261	4114547	> 7260	2395517	> 7260	3996963	> 7260	4238186	> 7260
map18	261	1182	493	1421	493	1386	493	1445	493	1419	493	1405
map20	526	926	399	1197	399	1176	399	1104	399	1112	399	766
mcsched	16113	341	16299	332	17329	374	19160	378	14068	327	15144	349
mik-250-1-100-1	1421995	457	295811	216	295253	211	269619	212	269993	211	295423	215
mine-166-5	6170	109	3361	76	4573	82	3239	78	3362	81	3745	83
mine-90-10	189285	1364	28948	230	281440	388	190705	323	101852	303	190755	364
msc98-ip	485	> 7212	103	> 7267	467	> 7264	5970	> 7265	4144	> 7266	4248	> 7265
mspp16	1	! 144	1	! 2545	1	! 2610	1	! 2538	1	! 2532	1	! 2538
mzzv11	8361	1868	3507	1198	5455	1344	7493	1425	7486	1427	3853	1186
n3div36	37993	! 2825	720314	> 7266	680770	> 7266	534695	> 7266	774418	> 7266	512355	> 7266
n3seq24	1	! 72	366	> 7308	1	! 774	204	> 7308	1	! 794	1	! 1431
n4-3	57354	1466	47663	969	36082	958	37746	934	49170	983	49819	995
neos-1109824	20142	283	5063	345	5079	285	5061	345	5063	341	5072	346
neos-1337307	239423	> 7201	846805	> 7261	823037	> 7261	829211	> 7261	818196	> 7261	887078	> 7261
neos-1396125	40087	4125	63396	2436	71912	2794	71777	2789	72474	2746	69883	2724
neos13	17847	> 7201	4772	3689	7	1404	709	6813	111	2326	89	3229
neos-1601936	7771	5311	61131	> 7262	75961	> 7262	49998	> 7262	94694	> 7262	49041	> 7262
neos18	6827	81	6292	72	6292	71	6292	71	6292	71	6292	71
neos-476283	1	! 273	411	1246	363	1230	327	! 1282	14	! 879	38	! 1070
neos-686190	4739	134	3859	126	3411	162	3741	118	3856	119	3372	99
neos-849702	126716	3193	20907	767	19445	785	18633	787	18747	789	19365	786
neos-916792	66213	511	1929977	> 7262	1971421	> 7262	2114777	> 7262	1963079	> 7262	2037323	> 7262
neos-934278	2755	> 7206	1215	3540	1239	> 7261	1280	3526	1263	3585	1422	> 7261
net12	3678	4536	4168	5858	2098	3614	4037	5219	5358	6490	2675	4103
netdiversion	26	> 7295	14	> 7355	15	> 7315	14	> 7434	14	> 7369	14	> 7359
newdano	2842106	> 7201	3207739	4307	3266717	2825	3745968	3776	3637999	3007	3330535	3592
noswot	575618	217	578076	165	605292	168	590844	168	592198	168	566218	165
ns1208400	4817	1361	691	508	691	510	691	506	691	494	691	504
ns1688347	4538	528	1752	399	12457	648	3451	487	2325	434	2378	399
ns1758913	1	! 3132	1	! 4774	-	-	-	-	1	! 4600	-	-
ns1766074	945109	1318	934754	1201	934754	992	934754	1189	934754	1191	934754	1486
ns1830653	88276	1111	54156	583	74846	654	53766	592	53695	587	60156	628
opm2-z7-s2	1978	1142	2900	1375	1518	2534	3165	1473	1051	2147	1360	2271
pg5_34	318742	2339	352765	1785	362969	1809	382798	1845	378131	1849	345258	1785
pigeon-10	2955734	> 7200	2	! 145	8550515	> 7260	8490984	> 7260	8080905	> 7260	7678656	> 7260
pw-myciel4	488828	> 7201	1599672	7166	1307118	6766	885110	4618	749182	3657	872659	3860
qiu	9811	83	6774	122	6888	123	7112	120	6930	124	7057	119
rail507	1319	2126	628	3299	845	3325	979	4211	773	3207	630	3376
ran16x16	441109	396	42832	294	45162	315	56901	316	56267	314	70935	321
reblock67	83043	339	118426	179	91280	128	165962	178	148683	168	108363	148
rmatr100-p10	901	298	901	308	805	317	901	308	901	306	901	307
rmatr100-p5	397	855	397	882	397	884	397	880	397	887	397	878
rmine6	687364	6508	886012	1866	1012940	2081	1011117	2016	1086584	2052	1046427	2290
rocII-4-11	13194	399	25494	431	16409	382	14252	356	17087	347	18075	447
rococoC10-001000	1075742	> 7201	1346999	3430	1321586	3381	1161137	3226	1289085	3384	1345795	3479
roll3000	1322672	> 7201	288676	972	805173	2265	358465	1135	282969	1012	302479	1024
satellites1-25	16711	4956	5175	> 7326	13426	6086	36324	> 7326	8498	> 7326	7575	> 7326
sp98ic	40930	! 5053	682537	4071	1050617	6495	1007465	6502	1231151	6868	1169778	6645
sp98ir	8984	202	2719	429	4120	392	4440	420	3959	438	3509	478
tanglegram1	27	2358	3115	> 7264	4160	> 7264	3253	> 7264	4397	> 7264	4194	> 7264
tanglegram2	3	13	3	14	3	13	3	14	3	14	3	14
timtab1	1014862	746	934481	914	1441921	1130	1626211	1216	1291078	1162	811181	980
triptim1	39	> 7207	31	> 7291	35	> 7356	30	> 7325	38	> 7323	39	> 7357
unitcal_7	12785	2501	8472	> 7360	2749	6333	3331	> 7355	3710	> 7355	13174	6113
vpphard	3693	> 7206	2987	> 7280	22982	> 7280	911	> 7280	4417	> 7281	1144	> 7282
zib54-UUE	565989	6971	749814	3217	743738	3209	749926	3214	742726	3215	749814	3218
geom. mean (solved: 53): 511.82	10514	645.1	6215	499.5	7065	518.7	6823	525.4	6439	506.7	6321	508.8
geom. mean (all): 1301.82	–	1554.6	–	1271.1	–	1295.5	–	1341.0	–	1290.7	–	1310.8
geom. mean (solved: 62): 671.85			11621	657.3	11878	676.5	12175	692.5	11229	655.4	11017	663.1
solved/timeout/abort	58/21/8		64/19/4		66/17/4		64/19/4		64/18/5		64/18/5	
speedup (solved: 53): 1.261			1.29		1.24		1.23		1.27		1.27	

Table 19: Tree search after racing ramp-up (Four SOLVER Threads): idle times

Name	Run - 1	Run - 2	Run - 3	Run - 4	Run - 5
30n20b8					
acc-tight5					
aflow40b					
air04					
app1-2					
ash608gpia-3col					
bab5					
beasleyC3					
biella1					
bienst2					
binkar10_1					
bley_xl1					
bnatt350					
core2536-691					
cov1075					
csched010					
danoint					
dfn-gwin-UUM					
eil33-2					
eilB101					
enlight13					
enlight14					
ex9					
glass4					
gmu-35-40	–	–	–	–	–
iis-100-0-cov					
iis-bupa-cov					
iis-pima-cov					
lectsched-4-obj					
m100n500k4r1					
macrophage					
map18					
map20					
mcsched					
mik-250-1-100-1					
mine-166-5					
mine-90-10					
msc98-ip					
mspp16	–	–	–	–	–
mzzv11					
n3div36					
n3seq24		–		–	–
n4-3					
neos-1109824					
neos-1337307					
neos-1396125					
neos-1601936					
neos-476283			–	–	–
neos-686190					
neos-849702					
neos-916792					
neos-934278					
neos13					
neos18					
net12					
netdiversion	–		–	–	–
newdano					
noswot					
ns1208400					
ns1688347					
ns1758913	–	–	–	–	–
ns1766074					
ns1830653					
opm2-z7-s2					
pg5_34					
pigeon-10	–				
pw-myciel4					
qiu					
rail507					
ran16x16					
reblock67					
rmatr100-p10					
rmatr100-p5					
rmine6					
rocII-4-11					
rococoC10-001000					
roll3000					
satellites1-25					
sp98ic					
sp98ir					
tanglegram1					
tanglegram2					
timtab1					
triptim1	–	–	–	–	–
unitcal_7					
vpphard					
zib54-UUE					

Table 20: Tree search after racing ramp-up (Eight SOLVER Threads)

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	4639	> 7219	29279	> 7264	31435	> 7264	40653	> 7264	51489	> 7264	34872	> 7264
acc-tight5	1808	353	39	69	39	68	39	69	39	68	39	69
aflow40b	668667	6444	50010	387	51392	388	63069	392	49686	390	53691	390
air04	215	149	94	119	121	118	44	118	94	114	44	119
app1-2	392	2163	254	2480	47	3778	47	3818	47	3908	254	3894
ash608gpia-3col	7	77	7	254	7	252	7	242	7	253	7	250
bab5	12846	> 7201	40699	> 7316	48101	> 7316	117415	> 7316	4296	> 7316	73236	> 7316
beasleyC3	992738	> 7203	4163497	> 7260	4537236	> 7260	4159752	> 7260	4223344	> 7260	4218954	> 7260
biella1	3270	1554	17315	2627	51138	5454	61487	6412	28280	3269	1	! 2697
bienst2	86468	439	96810	226	115405	255	111005	252	120284	254	112816	234
binkar10_1	135435	280	13152	207	11665	182	13596	188	18369	235	12339	194
bley_xl1	1	342	1	364	1	361	1	361	1	335	1	363
bnatt350	4134	859	2883	945	3057	988	2831	1005	2831	965	2831	976
core2536-691	810	1487	101	1319	101	1366	587	1570	85	544	92	1130
cov1075	976932	> 7201	1419783	7016	1412202	6983	1317749	6324	1447273	6917	1427398	7067
csched010	684362	> 7200	1228004	3796	1099838	3595	1028131	3075	1037169	3144	1065496	3561
danoint	870942	> 7201	708555	1092	908777	1282	857283	1198	698945	1081	1	! 1
dfn-gwin-UUM	87613	239	5926	112	7077	112	6670	108	6643	109	6388	109
eil33-2	10571	190	4575	385	9168	339	4160	355	4717	290	5001	302
eilB101	9239	1292	8065	793	7557	652	7543	661	2929	617	3642	570
enlight13	745476	857	1493063	571	10088388	> 7260	3445307	1456	3254196	1974	1795035	721
enlight14	259155	316	100054	302	100054	296	100054	297	100054	306	100054	297
ex9	1	194	1	191	1	192	1	183	1	194	1	191
glass4	2990992	! 4138	925791	392	835344	269	13164686	3581	730633	235	1254667	362
gmu-35-40	1970052	! 1785	57685343	! 5169	29127724	! 2821	65683670	! 5430	53379097	! 5267	63852223	! 5675
iis-100-0-cov	106874	2669	56623	724	56098	711	56588	691	56705	724	55055	702
iis-bupa-cov	113309	> 7202	133620	2271	134302	2268	135700	2073	133586	2274	135506	2228
iis-pima-cov	12469	1526	3061	1320	3260	1407	3056	1326	2949	1362	3056	1341
lectsched-4-obj	21026	826	819	107	55	104	162	113	130	97	161	106
m100n500k4r1	5050160	> 7201	23927950	> 7260	25687439	> 7260	20745571	> 7260	20372478	> 7260	40291034	> 7260
macrophage	541612	> 7201	2901767	> 7260	2837778	> 7260	2954520	> 7260	2634932	> 7260	2643068	> 7261
map18	261	1182	228	2917	341	1945	229	2849	217	2942	227	2946
map20	526	926	304	2246	306	2265	301	2262	313	2211	308	2218
mcsched	16113	341	13119	349	11443	291	14218	342	13421	351	11633	303
mik-250-1-100-1	1421995	457	201651	120	196231	121	203907	122	201536	120	200197	122
mine-166-5	6170	109	2172	102	2042	102	2801	111	1843	102	1800	109
mine-90-10	189285	1364	203796	345	210412	334	284973	343	247719	316	210008	356
msc98-ip	485	> 7212	126	> 7266	127	> 7266	312	> 7265	130	> 7266	290	> 7266
mspp16	1	! 144	1	! 2583	1	! 2567	1	! 2577	1	! 2567	1	! 2561
mzzv11	8361	1868	4073	3562	13493	4371	8675	5099	4202	3809	3675	3087
n3div36	37993	! 2825	375674	> 7266	351501	> 7269	79861	> 7266	329138	> 7266	349069	> 7266
n3seq24	1	! 72	6	> 7271	5	> 7313	5	> 7336	5	> 7288	5	> 7350
n4-3	57354	1466	39034	502	34888	541	32110	533	38771	757	27312	499
neos-1109824	20142	283	5004	457	5005	451	5057	448	7594	370	5013	444
neos-1337307	239423	> 7201	1271688	6639	1019426	> 7261	999690	> 7261	1003723	> 7261	915206	> 7261
neos-1396125	40087	4125	46434	1172	42493	1122	43682	999	47905	1197	44035	1137
neos13	17847	> 7201	49	2381	9	1990	109	2378	2334	> 7263	416	3185
neos-1601936	7771	5311	366	1222	72724	3818	12389	2323	6577	1859	53151	5699
neos18	6827	81	6333	137	5436	143	5035	143	5367	152	5524	136
neos-476283	1	! 273	170	! 3260	49	! 2498	41	! 2623	290	! 4413	310	! 3616
neos-686190	4739	134	2617	146	2734	143	2665	141	2578	144	2672	142
neos-849702	126716	3193	17371	857	17099	857	17077	855	17224	860	16206	863
neos-916792	66213	511	2350938	> 7262	2106718	> 7262	2230441	> 7262	2373289	> 7262	2246125	> 7262
neos-934278	2755	> 7206	58	> 7261	137	> 7261	9	> 7219	343	> 7261	10	> 7261
net12	3678	4536	4506	6330	4195	5131	4181	5142	3694	5595	3841	5131
netdiversion	26	> 7295	1	> 7556	1	> 7467	1	> 7512	1	> 7453	1	> 7358
newdano	2842106	> 7201	3521739	1988	4420636	2278	3666312	2126	4024671	2186	4190551	2146
noswot	575618	217	1164879	184	758217	152	675010	135	626020	130	849453	148
ns1208400	4817	1361	691	545	691	703	691	708	691	732	691	717
ns1688347	4538	528	1355	310	1658	310	2088	362	3494	449	1946	387
ns1758913	1	! 3132	-	-	-	-	-	-	-	-	-	-
ns1766074	945109	1318	934754	1908	934754	1888	934754	1887	934754	1371	934754	1893
ns1830653	88276	1111	56707	341	46901	338	62877	428	58534	355	57455	392
opm2-z7-s2	1978	1142	734	3102	681	3126	10609	> 7293	723	3130	721	3391
pg5_34	318742	2339	194960	759	119808	620	153676	655	177763	710	230330	944
pigeon-10	2955734	> 7200	9290583	> 7260	10476462	> 7260	8916280	> 7260	8885860	> 7260	12870820	> 7260
pw-myciel4	488828	> 7201	1003255	3013	1011506	2903	1038567	3273	1028942	3315	983796	3282
qiu	9811	83	6930	80	7004	79	7557	82	7143	76	7164	87
rail507	1319	2126	913	3912	1422	6470	1361	6189	1238	6430	1566	6644
ran16x16	441109	396	24746	180	40850	206	25188	178	30706	189	35222	190
reblock67	83043	339	112381	125	141633	151	104642	117	119474	127	100338	132
rmatr100-p10	901	298	881	410	881	398	835	440	881	306	800	421
rmatr100-p5	397	855	396	1848	436	1845	399	1844	375	1844	435	1844
rmine6	687364	6508	693739	1927	748724	1843	787625	2183	717891	1360	702162	2058
rocII-4-11	13194	399	15352	285	20052	379	15318	347	17397	373	13042	313
rococoC10-001000	1075742	> 7201	672561	1668	547314	1520	1074370	2581	1141007	2644	319131	897
roll3000	1322672	> 7201	136085	440	157371	456	392001	1192	252389	869	332340	1522
satellites1-25	16711	4956	7901	> 7333	6439	> 7326	8946	> 7326	7818	> 7326	4588	> 7326
sp98ic	40930	! 5053	863052	6259	1162802	> 7264	893500	7149	796554	5983	868740	6018
sp98ir	8984	202	3491	322	3605	301	3721	327	3527	329	3470	317
tanglegram1	27	2358	299	> 7264	224	> 7267	505	> 7264	744	> 7264	398	> 7264
tanglegram2	3	13	3	19	3	19	3	19	3	19	3	19
timtab1	1014862	746	786299	457	811442	486	710921	461	718671	488	791352	469
triptim1	39	> 7207	9	3669	9	3658	9	3690	9	3686	9	3677
unitcal_7	12785	2501	4311	> 7360	2938	4779	3639	5840	2491	> 7359	4225	> 7358
vpphard	3693	> 7206	91	> 7280	338	> 7280	190	> 7280	653	> 7280	835	> 7280
zib54-UUE	565989	6971	649007	2325	639751	2293	623506	1868	654120	2666	620298	2314
geom. mean (solved: 51): 463.70	10165	649.8	4501	453.5	4864	467.6	4739	469.8	4487	453.6	4653	474.0
geom. mean (all): 1198.24	–	1554.6	–	1139.0	–	1195.6	–	1248.1	–	1182.1	–	1226.4
geom. mean (solved: 60): 587.75			7980	563.9	8517	574.8	8914	614.8	8081	569.8	8274	591.6
solved/timeout/abort	58/21/8		67/16/4		65/18/4		66/17/4		65/18/4		64/17/6	
speedup (solved: 51): 1.402			1.43		1.39		1.38		1.43		1.37	

Table 21: Tree search after racing ramp-up (Eight SOLVER Threads): idle times

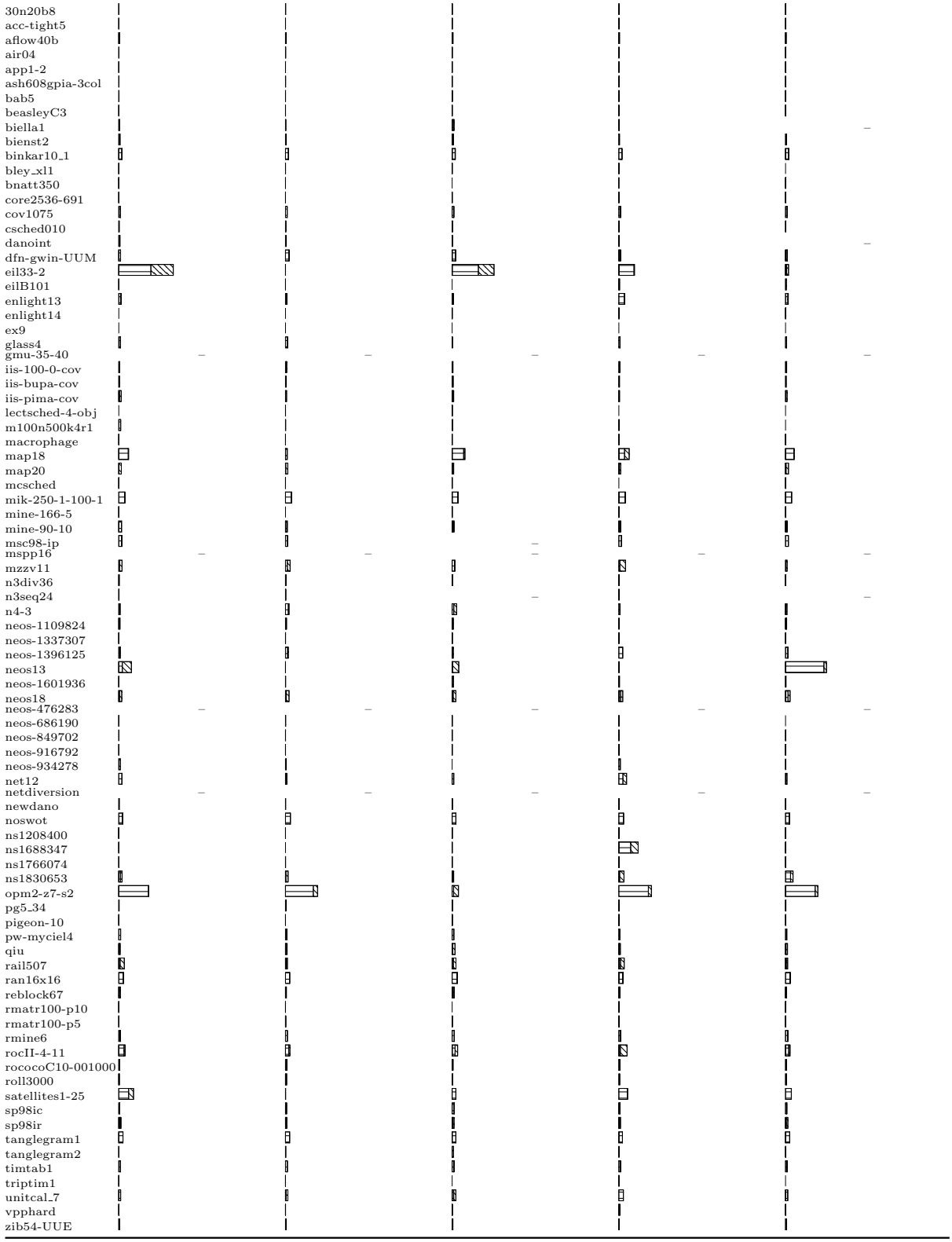

Table 22: Tree search after normal ramp-up (Four SOLVER Threads)

Name	One thread Nodes	Time	Run - 1 Nodes	Time	Run - 2 Nodes	Time	Run - 3 Nodes	Time	Run - 4 Nodes	Time	Run - 5 Nodes	Time
30n20b8	4639	> 7219	81963	5286	41885	> 7264	68158	5385	52136	6529	66997	6956
acc-tight5	1808	353	324	69	323	69	324	69	324	69	324	69
aflow40b	668667	6444	199766	1697	227115	1815	203343	1651	220194	1758	213336	1782
air04	215	149	117	123	129	129	117	123	111	121	113	118
app1-2	392	2163	1256	3515	1222	3371	1311	3451	1224	3412	1224	3570
ash608gpia-3col	7	77	19	138	19	136	19	137	19	137	19	135
bab5	12846	> 7201	64152	> 7313	51915	> 7313	54112	> 7313	21346	> 7313	77307	> 7308
beasleyC3	992738	> 7203	4195344	> 7260	4294521	> 7260	3941075	> 7260	4293587	> 7260	4166184	> 7260
biella1	3270	1554	3989	885	5224	923	5815	1051	6390	1057	4397	982
bienst2	86468	439	129234	343	115570	305	119105	305	121934	338	116683	278
binkar10_1	135435	280	99426	158	108890	179	106746	173	90778	153	105955	167
bley_xl1	1	342	1	347	1	348	1	347	1	347	1	347
bnatt350	4134	859	7212	404	7183	403	7038	399	7033	398	7196	404
core2536-691	810	1487	390	802	725	886	460	761	337	710	407	718
cov1075	976932	> 7201	1484506	> 7260	1392562	> 7260	1506849	> 7260	1445224	> 7260	1462785	> 7260
csched010	684362	> 7200	647186	4176	647907	4090	674938	4341	622854	4114	616394	3932
danoint	870942	> 7201	947533	2544	946838	2510	953678	2560	951045	2523	966768	2613
dfn-gwin-UUM	87613	239	73400	99	67125	95	68306	93	61762	101	72443	97
eil33-2	10571	190	4384	556	4917	388	5073	395	4779	442	4335	641
eilB101	9239	1292	5198	330	7350	395	3622	466	6240	418	3866	455
enlight13	745476	857	2054888	! 807	12753630	> 7260	10078148	6413	9306378	5812	12487312	> 7260
enlight14	259155	316	4074498	2050	3479374	1803	3857738	1900	13296118	> 7260	5671053	3531
ex9	1	194	1	190	1	190	1	190	1	190	1	190
glass4	2990992	! 4138	7494170	3031	6498497	2386	4991904	! 1836	5501925	2081	3529068	1620
gmu-35-40	1970052	! 1785	7591430	! 1726	9649466	! 2063	8754046	! 1847	8762998	! 1910	8343312	! 1824
iis-100-0-cov	106874	2669	80479	851	82497	891	80438	852	82551	890	80413	851
iis-bupa-cov	113309	> 7202	172833	3344	173358	3350	172829	3338	173098	3354	172724	3365
iis-pima-cov	12469	1526	17638	802	17499	789	17554	814	17553	798	17503	790
lectsched-4-obj	21026	826	37101	443	29303	369	33035	442	40165	490	24187	313
m100n500k4r1	5050160	> 7201	12812885	> 7260	11926306	> 7260	14651645	> 7260	14235241	> 7260	18058502	> 7260
macrophage	541612	> 7201	1726584	> 7260	1807982	> 7260	1966116	> 7260	1772585	> 7260	1833770	> 7260
map18	261	1182	194	1285	214	1227	216	1421	252	1464	210	1454
map20	526	926	216	773	185	746	187	885	185	721	216	803
mcsched	16113	341	22471	246	25540	259	22788	243	28191	247	24914	259
mik-250-1-100-1	1421995	457	680433	138	672858	136	671790	132	675404	135	678234	133
mine-166-5	6170	109	1724	51	1868	50	2378	55	2062	55	1868	50
mine-90-10	189285	1364	142375	297	112080	326	157639	428	110894	243	141964	446
msc98-ip	485	> 7212	1205	> 7265	585	> 7264	1198	> 7265	527	> 7265	1626	> 7265
mspp16	1	! 144	1	! 2543	1	! 2530	1	! 2533	1	! 2543	1	! 2539
mzzv11	8361	1868	6171	930	7110	980	7149	1186	4348	877	6219	859
n3div36	37993	! 2825	486338	> 7266	445521	> 7266	310499	! 5725	423125	! 7003	521435	> 7266
n3seq24	1	! 72	350	> 7309	138	> 7308	91	> 7308	354	> 7308	67	> 7309
n4-3	57354	1466	42573	787	41952	790	42741	803	43395	822	44618	837
neos-1109824	20142	283	7986	109	9959	153	5187	104	6880	150	7451	109
neos-1337307	239423	> 7201	324925	> 7261	598538	> 7261	528393	> 7261	517255	> 7261	522242	> 7261
neos-1396125	40087	4125	35295	1882	36492	1934	34652	1847	34977	1880	35474	1865
neos13	17847	> 7201	923	2790	1460	2151	364	2426	1291	1962	5585	3692
neos-1601936	7771	5311	9979	2127	13551	2823	13747	2825	13545	2825	9495	2145
neos18	6827	81	5208	63	5070	65	9468	70	4460	55	5346	55
neos-476283	1	! 273	1129	! 2357	796	! 1847	1038	2661	761	! 2062	772	! 1948
neos-686190	4739	134	9545	96	5678	82	4692	78	6766	86	9338	100
neos-849702	126716	3193	13501	277	13505	278	13412	278	13526	278	13478	277
neos-916792	66213	511	1449303	> 7262	980620	5898	1466680	> 7262	853944	5070	1498253	> 7262
neos-934278	2755	> 7206	3862	> 7261	4455	> 7261	1309	> 7261	2019	> 7261	3587	> 7261
net12	3678	4536	5939	3263	6287	4118	5821	3092	7449	4164	6544	3799
netdiversion	26	> 7295	16	> 7315	25	> 7315	25	> 7316	26	> 7315	50	> 7315
newdano	2842106	> 7201	3648675	3318	3250287	3381	4043488	3785	3283351	3182	3212696	3159
noswot	575618	217	719673	111	704807	123	690812	126	730983	125	740649	118
ns1208400	4817	1361	35267	2390	25811	2189	208125	> 7260	90499	6161	35968	2588
ns1688347	4538	528	4157	357	7339	427	5104	390	6182	461	10917	484
ns1758913	1	! 3132	-	-	-	-	-	-	-	-	-	-
ns1766074	945109	1318	661665	421	667310	416	677790	379	661708	431	668777	386
ns1830653	88276	1111	115923	1062	147070	1400	146625	1208	118807	1197	139496	1243
opm2-z7-s2	1978	1142	2689	1318	2346	1354	1833	1680	2410	1317	1563	1262
pg5_34	318742	2339	305029	1062	317551	1090	353814	1186	326189	1131	311572	1061
pigeon-10	2955734	> 7200	7263076	> 7260	4605693	! 4384	8127279	> 7260	7150034	> 7260	7192792	> 7260
pw-myciel4	488828	> 7201	1030009	4773	1020457	4858	1019259	4723	1011804	4486	1131200	> 7261
qiu	9811	83	13066	56	13238	58	13134	59	13116	58	13220	58
rail507	1319	2126	1365	1764	1347	2664	1192	2579	1212	2667	1238	2605
ran16x16	441109	396	407862	234	396327	237	398390	254	395543	232	396936	242
reblock67	83043	339	157768	345	153298	323	127601	184	200477	210	115787	236
rmatr100-p10	901	298	494	155	551	145	551	142	513	157	550	145
rmatr100-p5	397	855	240	443	257	388	240	443	259	388	246	443
rmine6	687364	6508	560402	2201	599044	2396	639800	2155	633707	2319	580808	2228
rocII-4-11	13194	399	16930	262	16023	256	12967	201	14862	242	13414	237
rococoC10-001000	1075742	> 7201	686793	3971	1015720	5149	977730	4872	689741	4048	758232	4203
roll3000	1322672	> 7201	1175475	6077	998800	4924	908010	4715	906029	4624	759170	3786
satellites1-25	16711	4956	22448	> 7326	50530	4838	51454	6964	20464	> 7326	42250	7082
sp98ic	40930	! 5053	1090082	> 7264	818385	> 7264	860923	> 7264	682807	> 7264	700612	> 7264
sp98ir	8984	202	5986	102	6776	109	5431	107	5864	109	5720	107
tanglegram1	27	2358	524	> 7264	463	> 7264	582	> 7264	436	> 7264	369	> 7264
tanglegram2	3	13	3	12	3	12	3	12	3	12	3	12
timtab1	1014862	746	1313918	401	1230278	401	1235504	353	1199520	366	1587546	461
triptim1	39	> 7207	269	> 7265	269	> 7265	244	> 7274	211	> 7266	218	> 7407
unitcal_7	12785	2501	9766	3676	11222	4228	8459	3349	11404	3450	27647	6237
vpphard	3693	> 7206	12118	> 7284	9024	> 7284	10409	> 7281	10969	> 7281	7768	> 7283
zib54-UUE	565989	6971	665451	5835	643949	5952	733288	7020	658769	5822	656183	5786
geom. mean (solved: 52): 411.03	9227	685.3	7872	402.7	8229	416.6	7875	410.9	7993	410.0	8024	414.9
geom. mean (all): 1178.30	–	1554.6	–	1161.4	–	1172.2	–	1185.5	–	1184.0	–	1188.4
geom. mean (solved: 59): 531.69			12182	522.4	12791	536.1	12045	531.6	12334	524.7	12666	534.3
solved/timeout/abort	58/21/8		64/18/5		65/17/5		65/17/5		65/17/5		64/19/4	
speedup (solved: 52): 1.668			1.70		1.65		1.67		1.67		1.65	

Table 23: Tree search after normal ramp-up (Four Solver Threads): idle times

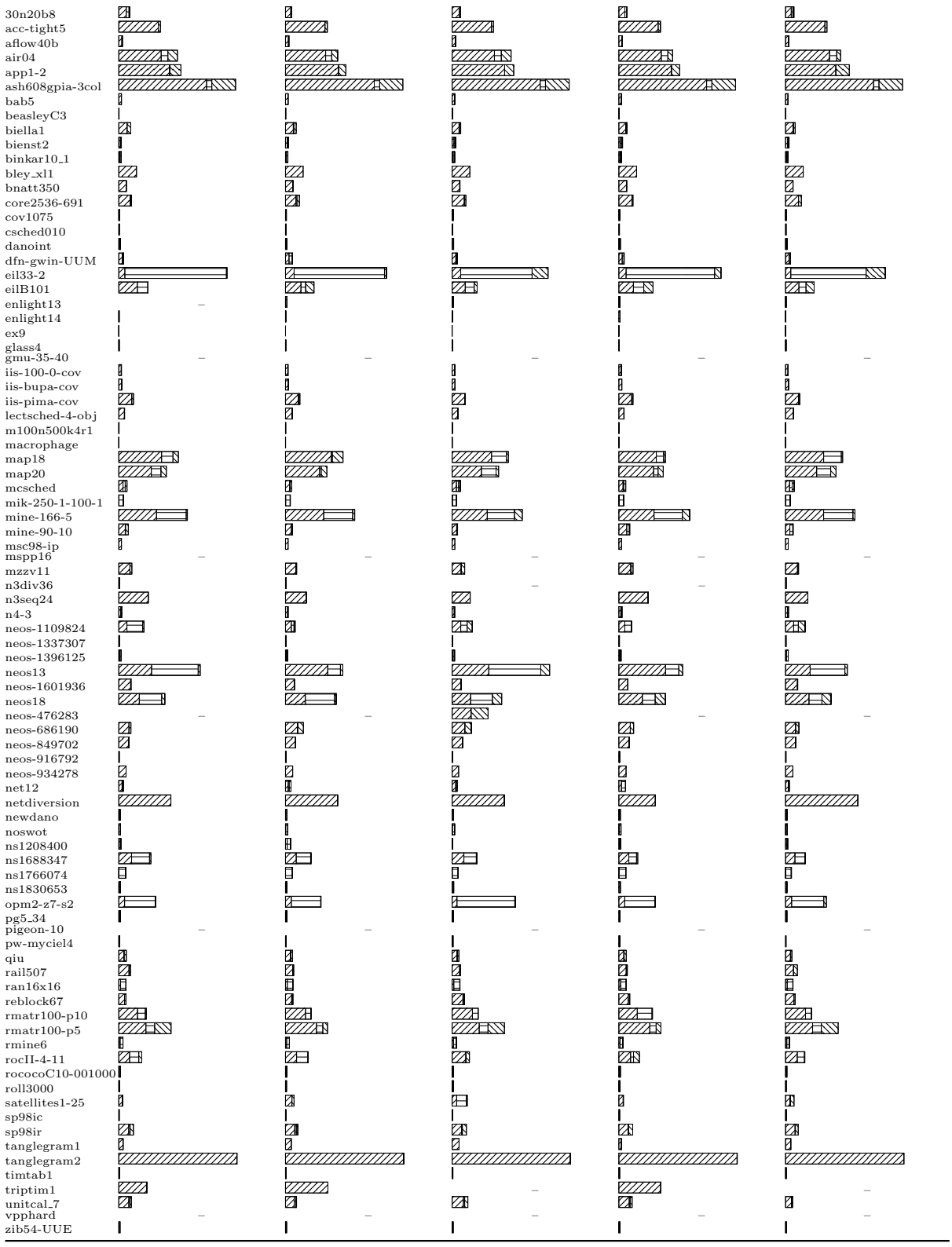

Table 24: Tree search after racing ramp-up (Eight SOLVER Threads)

Name	One thread Nodes	Time	Run - 1 Nodes	Time	Run - 2 Nodes	Time	Run - 3 Nodes	Time	Run - 4 Nodes	Time	Run - 5 Nodes	Time
30n20b8	4639	> 7219	50758	> 7264	86983	5450	166578	6715	65305	> 7264	162060	7199
acc-tight5	1808	353	348	75	341	75	330	76	347	75	349	75
aflow40b	668667	6444	180560	635	163980	636	189685	639	157344	539	180625	598
air04	215	149	232	158	182	164	162	157	166	154	206	160
app1-2	392	2163	534	4924	534	4837	544	4975	543	4831	538	4916
ash608gpia-3col	7	77	19	137	19	137	19	137	19	137	19	139
bab5	12846	> 7201	20310	> 7316	15370	> 7316	5532	> 7316	24007	> 7316	25943	> 7316
beasleyC3	992738	> 7203	4266670	> 7260	4328413	> 7260	4333290	> 7260	4178910	> 7260	4420647	> 7260
biella1	3270	1554	8934	1140	6342	1034	7637	1230	10017	1592	11014	1772
bienst2	86468	439	129503	148	128972	157	140029	158	137437	173	128241	164
binkar10_1	135435	280	58792	61	56644	71	53559	58	65854	90	57738	79
bley_xl1	1	342	1	353	1	349	1	347	1	350	1	350
bnatt350	4134	859	22276	546	22353	693	22015	688	44324	1348	22125	705
core2536-691	810	1487	246	1113	358	1430	205	1089	517	1783	249	1191
cov1075	976932	> 7201	1764076	6087	1748105	6158	1778973	6551	1777983	5969	1737144	6648
csched010	684362	> 7200	970006	3716	948493	3559	1032955	3602	912473	3424	1095223	3994
danoint	870942	> 7201	779000	1447	792158	1460	784396	1453	782157	1475	790119	1439
dfn-gwin-UUM	87613	239	62082	80	1	! 1	53107	77	52526	78	51764	78
eil33-2	10571	190	3456	464	4359	560	3890	524	3872	390	3890	474
eilB101	9239	1292	1949	369	1281	391	1853	345	1446	290	1467	410
enlight13	745476	857	7957824	3482	17353672	> 7260	14355830	> 7260	4740224	2192	13073027	> 7260
enlight14	259155	316	20407853	> 7260	9069390	> 7260	17426513	> 7260	19755862	> 7260	10815820	5010
ex9	1	194	1	190	1	190	1	193	1	190	1	191
glass4	2990992	! 4138	11075826	3336	17113644	> 7263	20968618	> 7260	19838625	> 7260	17091658	3732
gmu-35-40	1970052	! 1785	16949335	! 2501	18551379	! 2608	17154064	! 2362	15877660	! 2266	16520291	! 2317
iis-100-0-cov	106874	2669	77671	573	78201	575	74428	567	75795	563	76990	567
iis-bupa-cov	113309	> 7202	155435	2629	157047	2643	158469	2670	157055	2670	156627	2668
iis-pima-cov	12469	1526	7941	457	7374	463	7794	477	7529	442	7668	463
lectsched-4-obj	21026	826	15268	257	16402	262	18470	277	14434	249	17876	278
m100n500k4r1	5050160	> 7201	19080177	> 7260	23487435	> 7260	26538761	> 7260	28922081	> 7260	25832322	> 7260
macrophage	541612	> 7201	1842048	> 7260	1549242	> 7260	1845397	> 7260	1766582	> 7260	1800379	> 7260
map18	261	1182	186	1649	180	1552	180	1619	167	1427	185	1614
map20	526	926	157	1102	191	1283	176	649	192	1210	191	1129
mcsched	16113	341	29768	237	27926	235	29565	232	31750	234	30011	239
mik-250-1-100-1	1421995	457	535864	72	533951	70	567185	69	554630	68	564939	73
mine-166-5	6170	109	2165	54	1296	56	2161	60	2803	54	1429	60
mine-90-10	189285	1364	216914	412	155984	296	174236	268	199973	257	91622	176
msc98-ip	485	> 7212	511	> 7266	568	> 7266	329	> 7265	67	> 7265	107	> 7265
mspp16	1	! 144	1	! 4689	1	! 4711	1	! 4468	1	! 4679	1	! 4691
mzzv11	8361	1868	7662	1313	9844	2058	14643	2604	8749	1854	10964	2416
n3div36	37993	! 2825	35540	> 7267	329159	> 7266	391599	> 7266	357239	> 7266	68918	> 7266
n3seq24	1	! 72	328	> 7334	56	> 7311	262	> 7312	27	> 7314	51	> 7311
n4-3	57354	1466	44323	608	51743	724	47043	631	40777	645	43231	598
neos-1109824	20142	283	3618	90	2038	100	3558	100	4281	136	7469	138
neos-1337307	239423	> 7201	857799	> 7261	568652	> 7261	782860	> 7261	791665	> 7261	750376	> 7261
neos-1396125	40087	4125	52022	946	52324	942	50162	898	48977	901	52561	928
neos13	17847	> 7201	516	2631	2128	3107	711	2544	1579	2883	1439	2917
neos-1601936	7771	5311	11962	2201	9788	1961	6344	1640	23304	2312	27558	3487
neos18	6827	81	10503	78	11812	74	13963	76	14076	88	9011	89
neos-476283	1	! 273	634	! 4068	531	4818	691	! 3406	453	! 3145	485	! 4031
neos-686190	4739	134	755	55	605	51	1273	57	711	54	580	52
neos-849702	126716	3193	7039	129	7160	130	7029	129	7992	129	7185	130
neos-916792	66213	511	2418476	> 7262	2481609	> 7262	2272150	> 7262	1763322	> 7262	2048871	> 7262
neos-934278	2755	> 7206	7975	5810	486	> 7261	486	> 7261	147	> 7261	302	> 7261
net12	3678	4536	5516	> 7275	6183	> 7275	6781	> 7275	6724	> 7275	6973	> 7275
netdiversion	26	> 7295	35	> 7317	39	> 7350	41	> 7397	41	> 7327	33	> 7317
newdano	2842106	> 7201	3275726	1419	3209603	1779	3522105	1651	3469205	1740	3337785	1775
noswot	575618	217	849292	95	765034	86	814779	98	891314	109	1050260	130
ns1208400	4817	1361	68762	4029	208778	6570	49730	3572	101722	5376	51655	2574
ns1688347	4538	528	1419	189	2989	200	1824	223	5484	304	2387	204
ns1758913	1	! 3132	-	-	-	-	-	-	-	-	-	-
ns1766074	945109	1318	623409	298	633261	278	654862	273	631705	291	631187	283
ns1830653	88276	1111	107756	512	103066	432	97787	549	113045	572	96843	486
opm2-z7-s2	1978	1142	626	1675	672	1601	808	1722	507	1769	854	1601
pg5_34	318742	2339	326399	664	255659	591	307432	650	262733	598	317089	642
pigeon-10	2955734	> 7200	10094457	> 7260	3313541	> 7260	8919522	! 4713	10200522	> 7260	5155955	! 3065
pw-myciel4	488828	> 7201	1998937	6339	2187613	6920	1877371	6024	2939040	> 7261	2379164	6733
qiu	9811	83	11762	35	11840	32	12209	33	10780	35	11729	33
rail507	1319	2126	719	3416	1466	5200	1164	4467	1392	4997	836	3865
ran16x16	441109	396	336208	194	460150	209	380988	207	329889	195	381008	175
reblock67	83043	339	254753	352	157230	267	115359	137	167176	227	123175	157
rmatr100-p10	901	298	370	135	373	125	359	142	398	137	353	136
rmatr100-p5	397	855	226	455	225	453	223	453	238	451	223	453
rmine6	687364	6508	584026	1595	672698	1787	558257	1618	586858	1669	619667	1377
rocII-4-11	13194	399	18178	250	13344	216	18565	242	16025	295	14339	246
rococoC10-001000	1075742	> 7201	642320	1830	553704	1411	468019	1375	502728	1474	469472	1530
roll3000	1322672	> 7201	425597	1357	415860	1385	282730	1054	344322	1136	294706	1035
satellites1-25	16711	4956	8389	> 7326	15738	5793	9065	6149	25491	7170	22624	> 7326
sp98ic	40930	! 5053	211217	6120	258183	> 7264	320602	6881	251310	5786	258098	6476
sp98ir	8984	202	4207	113	3841	103	4329	98	4217	109	4744	111
tanglegram1	27	2358	343	> 7264	273	> 7264	235	> 7264	311	> 7264	334	> 7264
tanglegram2	3	13	3	12	3	12	3	12	3	12	3	12
timtab1	1014862	746	1240404	331	1455376	386	1379817	394	1390633	391	1316559	347
triptim1	39	> 7207	166	> 7266	165	> 7278	164	> 7278	167	> 7394	107	> 7268
unitcal_7	12785	2501	4159	> 7359	11406	> 7357	11338	> 7358	7113	> 7360	7880	> 7359
vpphard	3693	> 7206	5256	> 7281	4416	> 7284	4325	> 7281	4538	> 7281	5326	> 7281
zib54-UUE	565989	6971	571699	4406	661284	5040	600480	4594	603252	4606	580244	4391
geom. mean (solved: 50): 351.10	8811	665.8	6708	344.1	6762	353.8	6738	340.9	7282	366.4	6795	350.3
geom. mean (all): 1080.47	–	1554.6	–	1048.1	–	1137.2	–	1056.4	–	1091.6	–	1069.1
geom. mean (solved: 58): 456.45			11404	447.3	11728	459.3	11402	441.0	12385	470.9	11630	454.7
solved/timeout/abort	58/21/8		64/19/4		62/21/4		63/19/5		62/21/4		64/18/5	
speedup (solved: 50): 1.898			1.94		1.88		1.95		1.82		1.90	

Table 25: Tree search after normal ramp-up (Eight Solver Threads): idle times

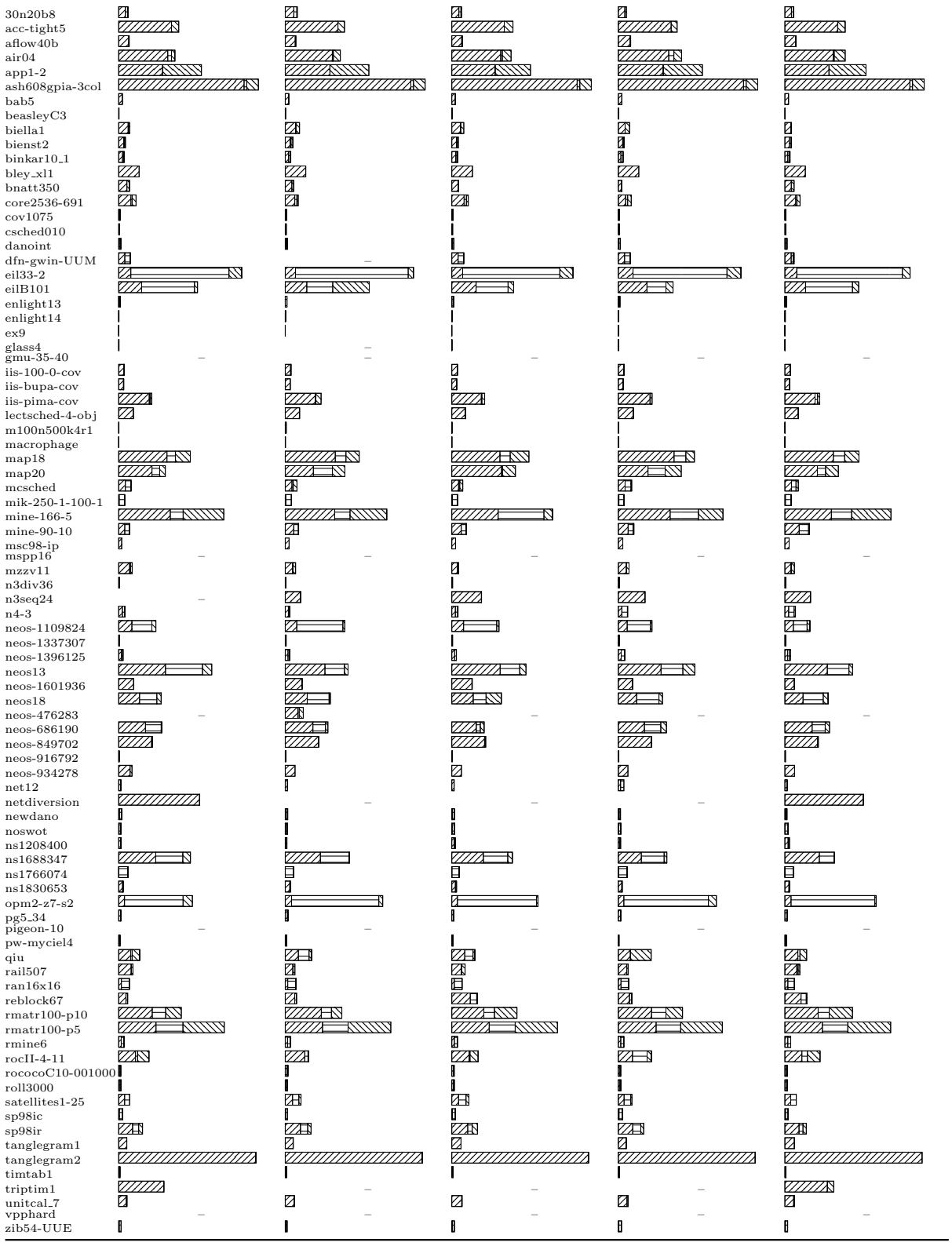

Table 26: Deterministic runs (Four SOLVER threads)

Name	One thread Nodes	One thread Time	Comm Nodes	Comm Time	Racing Nodes	Racing Time	Normal Nodes	Normal Time
30n20b8	4639	> 7219	447	> 7203	445	> 7239	18145	> 7265
acc-tight5	1808	353	72	147	72	147	3794	691
aflow40b	668667	6444	58258	! 2688	168035	2322	238996	! 3506
air04	215	149	49	251	49	251	178	218
app1-2	392	2163	47	3388	47	3299	2705	7157
ash608gpia-3col	7	77	7	231	7	228	19	169
bab5	12846	> 7201	4352	> 7201	18716	> 7312	11643	> 7313
beasleyC3	992738	> 7203	127919	! 3820	461521	! 3336	450397	! 3240
biella1	3270	1554	2854	2793	2854	2796	4386	2111
bienst2	86468	439	95433	1515	96858	2085	139404	786
binkar10_1	135435	280	144577	1121	11933	1046	92369	245
bley_xl1	1	342	1	411	1	410	1	352
bnatt350	4134	859	4134	2140	4134	2167	18865	2573
core2536-691	810	1487	313	1443	313	1436	1028	2601
cov1075	976932	> 7201	229954	> 7201	273275	> 7261	705549	> 7260
csched010	684362	> 7200	100830	! 5267	457088	! 4910	517559	> 7260
danoint	870942	> 7201	211546	> 7201	438911	> 7261	986745	6043
dfn-gwin-UUM	87613	239	64391	518	10272	732	69917	262
eil33-2	10571	190	9984	459	9984	470	7300	595
eilB101	9239	1292	7571	1719	7571	1718	5395	995
enlight13	745476	857	367322	1256	2195391	! 3160	5057077	! 6019
enlight14	259155	316	259155	916	259155	909	5077785	> 7261
ex9	1	194	1	192	1	189	1	191
glass4	2990992	! 4138	468754	! 3582	-	-	2099207	! 2202
gmu-35-40	1970052	! 1785	299818	! 1006	1175507	! 655	1074030	! 692
iis-100-0-cov	106874	2669	98454	> 7200	52225	6213	68698	5128
iis-bupa-cov	113309	> 7202	44866	> 7200	80058	> 7260	156623	> 7261
iis-pima-cov	12469	1526	8920	2424	4138	5436	27161	3224
lectsched-4-obj	21026	826	10199	963	8207	637	22274	961
m100n500k4r1	5050160	> 7201	429553	! 2215	1523014	! 1727	1595007	! 1444
macrophage	541612	> 7201	138396	> 7201	1	! 3784	352322	! 7108
map18	261	1182	493	2281	493	2276	246	2836
map20	526	926	399	1512	399	1512	246	1637
mcsched	16113	341	14948	619	14948	619	24303	578
mik-250-1-100-1	1421995	457	808088	! 685	345850	813	889615	341
mine-166-5	6170	109	3278	169	3278	170	4766	124
mine-90-10	189285	1364	178919	3117	217764	849	440563	2603
msc98-ip	485	> 7212	432	> 7209	439	> 7234	1	> 7266
mspp16	1	! 144	1	! 2541	1	! 2544	1	! 2536
mzzv11	8361	1868	5064	4441	8568	> 7427	15524	3392
n3div36	37993	! 2825	9129	! 2401	35485	! 2794	44137	! 2787
n3seq24	1	! 72	1	> 7208	1	> 7233	541	> 7319
n4-3	57354	1466	50613	3263	20777	4841	37813	1196
neos-1109824	20142	283	10058	420	3495	1020	23009	415
neos-1337307	239423	> 7201	107488	> 7201	212402	> 7262	255456	> 7263
neos-1396125	40087	4125	13223	> 7201	22407	> 7260	31323	6010
neos13	17847	> 7201	1519	3254	1519	3245	16912	> 7263
neos-1601936	7771	5311	5412	> 7201	4176	> 7262	7266	> 7263
neos18	6827	81	7855	169	5019	731	17765	220
neos-476283	1	! 273	51	1398	51	1397	146	> 7508
neos-686190	4739	134	3676	256	3676	257	12529	341
neos-849702	126716	3193	20907	2473	20907	2474	111640	2464
neos-916792	66213	511	67156	2487	566959	! 6253	549452	! 4712
neos-934278	2755	> 7206	1	> 7219	1	> 7208	2739	> 7262
net12	3678	4536	3770	> 7203	3703	> 7272	4570	> 7276
netdiversion	26	> 7295	-	-	-	-	161	> 7563
newdano	2842106	> 7201	448819	! 6098	1529148	> 7260	1711583	! 3618
noswot	575618	217	403652	1126	928475	510	547869	216
ns1208400	4817	1361	4817	2561	4817	2567	22369	> 7260
ns1688347	4538	528	3633	1283	2810	1201	9578	1307
ns1758913	1	! 3132	1	> 7898	1	> 7734	-	-
ns1766074	945109	1318	907103	3205	907103	3192	838665	1210
ns1830653	88276	1111	70621	2545	142230	5572	131170	1944
opm2-z7-s2	1978	1142	2461	2768	1938	3056	5420	2623
pg5_34	318742	2339	70272	! 2938	277995	> 7261	320562	2365
pigeon-10	2955734	> 7200	1	> 7200	1	> 7200	2055539	! 2209
pw-myciel4	488828	> 7201	114161	> 7200	407777	> 7261	625101	> 7261
qiu	9811	83	9525	246	8965	328	13385	415
rail507	1319	2126	1174	4137	1174	4023	1517	4507
ran16x16	441109	396	392938	1444	36995	1109	429457	463
reblock67	83043	339	88623	751	187620	1074	130467	643
rmatr100-p10	901	298	893	467	893	464	544	531
rmatr100-p5	397	855	403	1722	403	1715	249	1188
rmine6	687364	6508	188361	! 6328	732559	! 4223	582738	4707
rocII-4-11	13194	399	11711	829	19806	1281	18472	752
rococoC10-001000	1075742	> 7201	88488	! 3414	120082	2316	355951	! 3655
roll3000	1322672	> 7201	173583	! 5164	72191	2304	763075	5485
satellites1-25	16711	4956	5787	> 7200	5813	> 7201	9636	> 7326
sp98ic	40930	! 5053	19035	! 4746	60527	! 2498	61767	! 4565
sp98ir	8984	202	5239	296	5346	1563	4945	203
tanglegram1	27	2358	18	> 7322	22	> 7264	258	> 7265
tanglegram2	3	13	3	19	3	19	3	14
timtab1	1014862	746	778328	2513	1905393	2849	1815457	1275
triptim1	39	> 7207	11	> 7564	11	> 7567	224	> 7502
unitcal_7	12785	2501	15040	> 7208	2548	> 7356	28243	> 7359
vpphard	3693	> 7206	300	> 7204	300	> 7212	5122	> 7287
zib54-UUE	565989	6971	67815	! 2744	141304	! 2149	162715	! 1741
geom. mean (solved: 43)	5576	460.3	4003	880.4	3426	989.0	8355	793.3
geom. mean (all)	–	1554.6	–	2386.0	–	2457.1	–	2273.0
geom. mean (solved: 44)			3625	889.7	3113	996.8	9268	829.1
solved/timeout/abort	58/21/8		48/22/17		50/23/14		49/22/16	
speedup (solved: 43)			0.52		0.47		0.64	

Table 27: Deterministic runs (Four Solver Threads): idle times

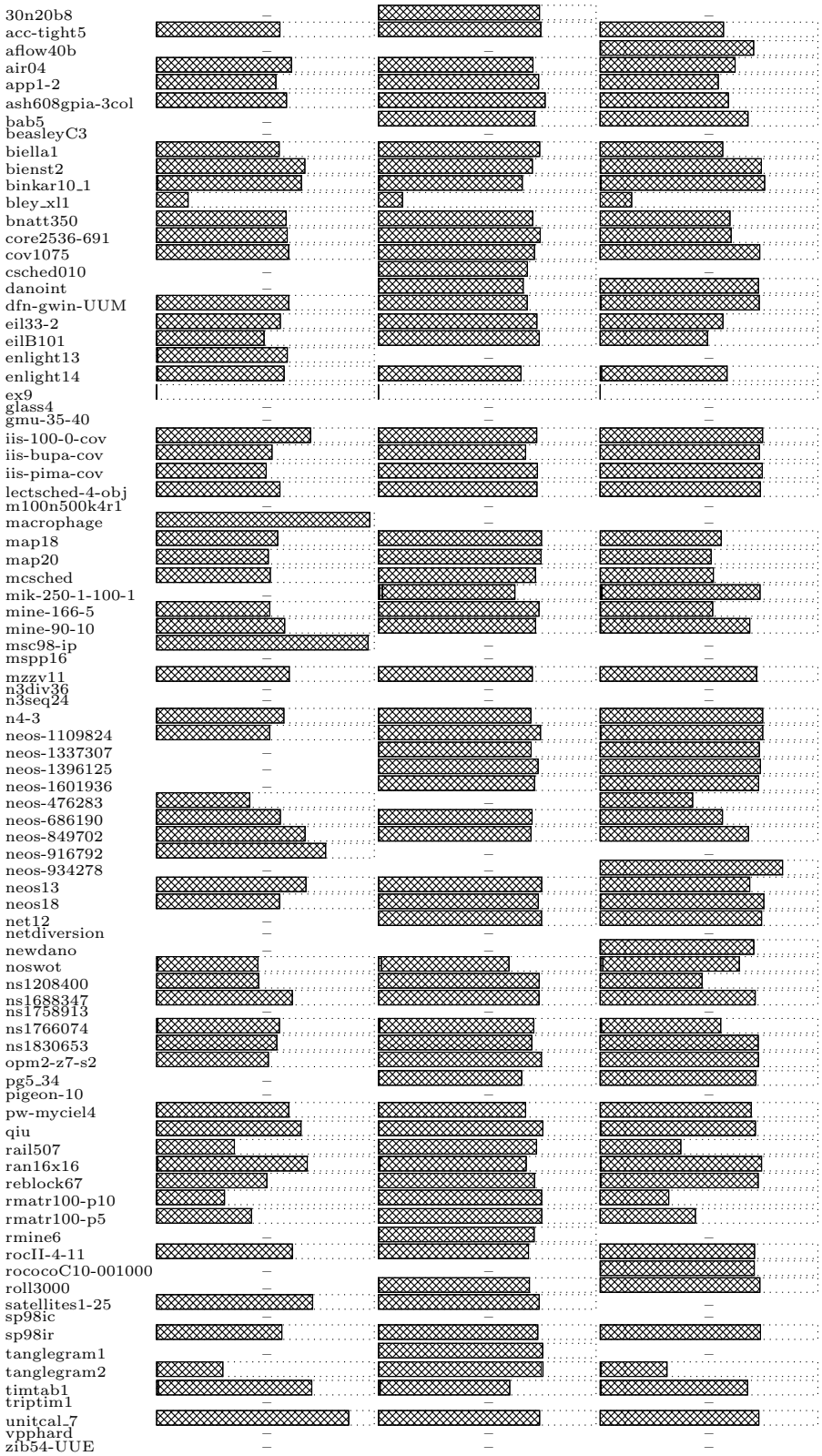

Table 28: Deterministic runs (Eight SOLVER threads)

Name	One thread Nodes	One thread Time	Comm Nodes	Comm Time	Racing Nodes	Racing Time	Normal Nodes	Normal Time
30n20b8	4639	> 7219	347	> 7206	346	> 7220	4582	> 7265
acc-tight5	1808	353	310	330	310	330	1741	537
aflow40b	668667	6444	73054	! 3664	94045	2244	283445	5214
air04	215	149	11	359	11	350	191	324
app1-2	392	2163	79	6475	79	6445	617	> 7285
ash608gpia-3col	7	77	7	484	7	422	19	183
bab5	12846	> 7201	1670	> 7216	10582	> 7312	10619	> 7317
beasleyC3	992738	> 7203	120315	! 6085	803204	! 4835	795977	! 5507
biella1	3270	1554	4242	5391	4242	5426	7727	3827
bienst2	86468	439	88341	2251	81525	1996	141730	631
binkar10_1	135435	280	135816	1901	12270	1082	105139	451
bley_xl1	1	342	1	466	1	465	1	350
bnatt350	4134	859	2831	2666	2831	2666	29574	3957
core2536-691	810	1487	167	2672	167	2670	1069	3874
cov1075	976932	> 7201	127550	> 7200	882921	> 7260	879769	> 7260
csched010	684362	> 7200	76215	! 7009	442503	> 7260	794212	> 7261
danoint	870942	> 7201	126984	> 7201	418710	> 7260	1016569	> 7260
dfn-gwin-UUM	87613	239	20880	628	10250	790	64020	337
eil33-2	10571	190	9727	817	9727	814	5458	569
eilB101	9239	1292	5598	2741	5598	2757	6543	1261
enlight13	745476	857	168971	1018	3349375	6610	3891279	! 6716
enlight14	259155	316	100054	636	100054	631	5435865	> 7260
ex9	1	194	1	193	1	191	1	191
glass4	2990992	! 4138	443722	! 5771	2486063	3884	-	-
gmu-35-40	1970052	! 1785	252817	! 2068	1978327	! 1284	1930314	! 1145
iis-100-0-cov	106874	2669	54637	> 7201	52111	6299	77800	1887
iis-bupa-cov	113309	> 7202	28012	> 7201	80329	> 7261	143979	> 7260
iis-pima-cov	12469	1526	8920	4126	4097	5731	23780	3631
lectsched-4-obj	21026	826	4580	881	18650	1289	30085	846
m100n500k4r1	5050160	> 7201	1	! 3404	2591489	! 2545	2734885	! 2010
macrophage	541612	> 7201	131945	> 7201	1	! 6360	464013	> 7261
map18	261	1182	347	1497	347	1482	185	3301
map20	526	926	399	1859	399	1838	231	2109
mcsched	16113	341	12961	1123	12961	1111	41810	1180
mik-250-1-100-1	1421995	457	682671	! 1073	211378	691	546627	269
mine-166-5	6170	109	2332	244	2332	244	3617	149
mine-90-10	189285	1364	92041	2880	267561	1276	234543	1314
msc98-ip	485	> 7212	414	> 7327	414	> 7322	823	> 7270
mspp16	1	! 144	1	! 2560	1	! 2559	1	! 4970
mzzv11	8361	1868	3359	5385	3359	5372	15010	3626
n3div36	37993	! 2825	9789	! 4086	53986	! 3811	72402	! 3652
n3seq24	1	! 72	1	> 7210	1	> 7206	588	> 7311
n4-3	57354	1466	38529	3935	24528	6759	41518	1417
neos-1109824	20142	283	10058	811	4012	1263	21572	427
neos-1337307	239423	> 7201	49208	> 7201	272893	> 7262	289948	> 7262
neos-1396125	40087	4125	3208	> 7200	15028	> 7260	58921	6408
neos13	17847	> 7201	9	5033	9	4999	3239	4395
neos-1601936	7771	5311	2447	> 7200	3192	> 7262	7662	> 7262
neos18	6827	81	6141	245	4155	702	14517	197
neos-476283	1	! 273	27	2199	27	2044	647	! 6049
neos-686190	4739	134	2194	386	2194	386	4924	288
neos-849702	126716	3193	17371	4367	20575	4371	230617	4017
neos-916792	66213	511	61166	3766	744050	> 7263	917823	! 6968
neos-934278	2755	> 7206	201	> 7219	201	> 7203	3013	> 7265
net12	3678	4536	1956	> 7200	3226	> 7275	3639	> 7275
netdiversion	26	> 7295	-	-	-	-	271	> 7489
newdano	2842106	> 7201	269610	> 7202	2348882	> 7261	2974156	! 5739
noswot	575618	217	347127	1652	925777	505	800662	436
ns1208400	4817	1361	1932	2823	1932	2813	28556	> 7261
ns1688347	4538	528	2824	2188	2867	2116	8300	954
ns1758913	1	! 3132	1	> 7687	1	> 7689	-	-
ns1766074	945109	1318	903761	5184	903761	5183	799620	920
ns1830653	88276	1111	63027	3541	44557	2046	121423	2158
opm2-z7-s2	1978	1142	1555	4314	1555	4330	2808	3244
pg5_34	318742	2339	55189	! 4576	250591	> 7260	329582	2717
pigeon-10	2955734	> 7200	1	> 7200	1	> 7200	1090801	! 921
pw-myciel4	488828	> 7201	51757	> 7200	467276	> 7261	631072	> 7262
qiu	9811	83	9757	396	9757	397	8263	427
rail507	1319	2126	1065	> 7202	1063	> 7202	1227	6155
ran16x16	441109	396	304484	2172	44900	977	391666	415
reblock67	83043	339	74015	1073	105347	896	128671	698
rmatr100-p10	901	298	851	647	851	647	504	618
rmatr100-p5	397	855	437	2867	437	2866	243	1228
rmine6	687364	6508	124995	> 7201	1180463	! 6113	546948	4294
rocII-4-11	13194	399	11728	1424	18210	1469	13759	1012
rococoC10-001000	1075742	> 7201	78646	! 5113	370821	5206	586323	! 4989
roll3000	1322672	> 7201	147918	> 7201	182877	4150	868439	> 7261
satellites1-25	16711	4956	626	> 7893	626	> 7891	6485	> 7327
sp98ic	40930	! 5053	18354	> 7201	99677	! 4637	112019	! 6401
sp98ir	8984	202	5662	496	5662	495	3537	282
tanglegram1	27	2358	18	> 7289	18	> 7220	288	> 7269
tanglegram2	3	13	3	108	3	109	3	16
timtab1	1014862	746	591562	3370	1887068	2424	1414433	1712
triptim1	39	> 7207	1	> 7265	1	> 7335	-	-
unitcal_7	12785	2501	4390	> 7232	3021	> 7367	19923	> 7359
vpphard	3693	> 7206	204	> 7221	204	! 7234	3491	> 7291
zib54-UUE	565989	6971	54628	! 3776	278505	! 6518	291722	! 3160
geom. mean (solved: 43)	5820	459.0	3354	1306.7	3171	1253.9	9175	832.9
geom. mean (all)	–	1554.6	–	2953.2	–	2805.4	–	2381.4
geom. mean (solved: 44)			3006	1322.2	2846	1267.9	9175	832.9
solved/timeout/abort	58/21/8		47/27/13		52/24/11		48/23/16	
speedup (solved: 43)			0.35		0.37		0.61	

Table 29: Deterministic runs (Eight SOLVER Threads): idle times

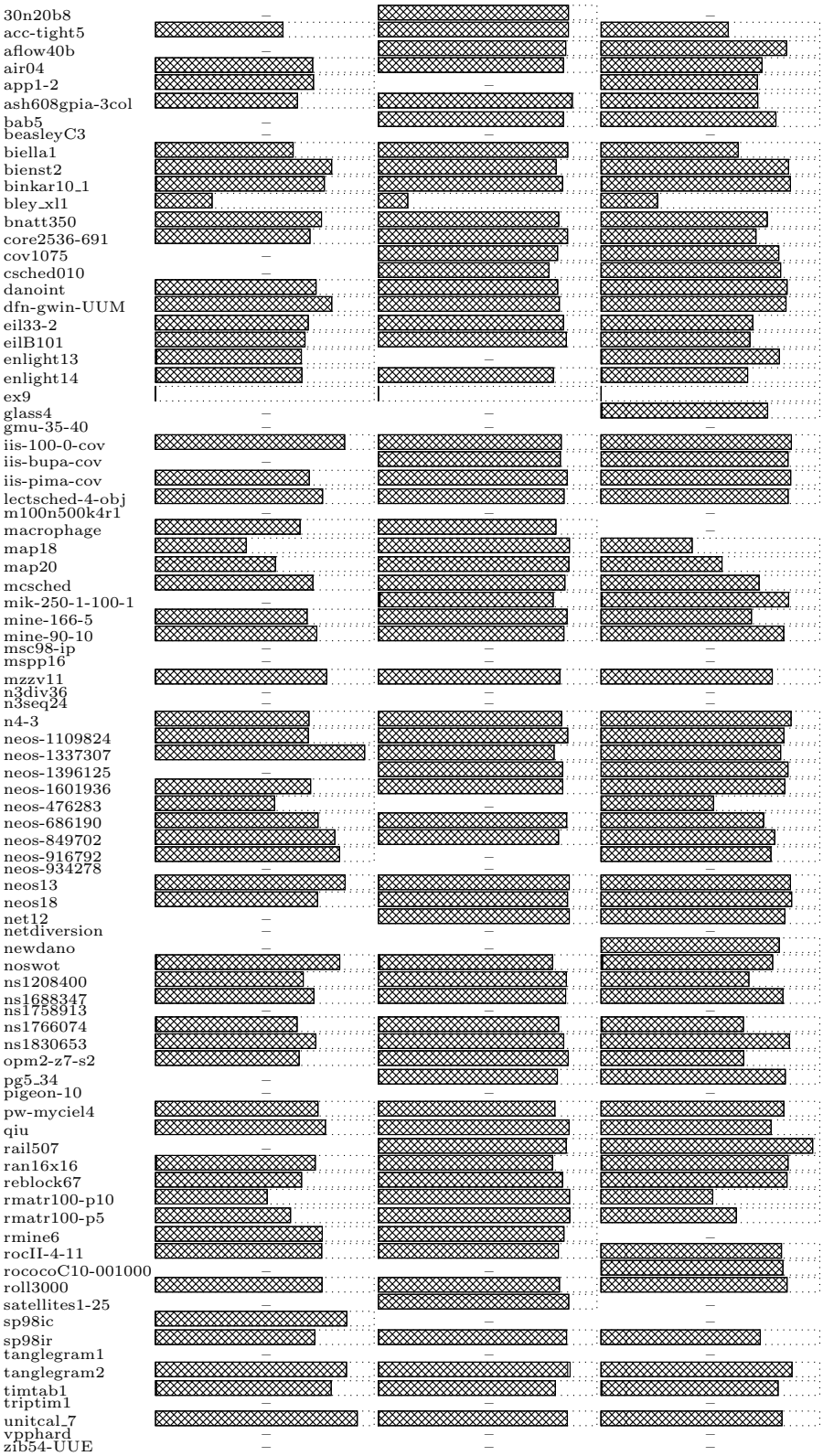

Table 30: 5 runs with four SOLVER threads on selected instances from MINLPLib

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
4stufen	2455190	> 7200	13911395	> 7200	16548286	> 7200	19034269	> 7652	15645529	> 7565	12304282	> 7567
beuster	2019633	> 7200	13014702	> 7200	18364128	> 7200	6791064	! 7200	26906183	> 7200	15487388	> 7200
cecil_13	1608177	> 7200	7542778	> 7200	7626310	> 7200	7469557	> 7200	7472804	> 7200	7531383	> 7200
chp_partload	8353	> 7200	1	! 1	1	! 1	6077	> 7200	6077	> 7200	6077	> 7200
csched2a	1235475	> 7200	7775333	> 7200	7764494	> 7200	605	> 7200	9439345	> 7200	7784271	> 7200
csched2	182072	> 7200	4147624	> 7200	4165828	> 7200	4171026	> 7200	4166937	> 7200	4166972	> 7200
eg_all_s	139	> 7200	290	> 7200	155	> 7200	381	> 7200	272	> 7200	183	> 7200
eg_disc2_s	133	> 7200	133	> 7200	133	> 7200	133	> 7200	133	> 7200	133	> 7200
eg_disc_s	1	> 7200	65	> 7200	65	> 7200	65	> 7200	65	> 7200	65	> 7200
eg_int_s	4	> 7200	4	> 7200	4	> 7200	4	> 7200	4	> 7200	4	> 7200
enpro48pb	279848	70	237	1	247	1	230	1	210	1	155	1
ex1233	7960984	> 7200	186408	! 7200	220588	! 7200	216410	! 7200	186358	! 7200	216244	! 7200
fac2	4565461	> 7200	147	! 7200	30861424	> 7200	30671920	> 7200	25970158	> 7200	30167018	> 7200
feedtray	274621	> 7200	1719088	> 7200	1632670	> 7200	1624430	> 7200	1715885	> 7200	1629798	> 7200
fo7	201538	132	224835	69	260981	75	261451	76	222269	68	263872	76
fo8_ar2_1	435919	259	1365702	262	1444653	278	1641153	266	1496715	286	1399248	265
fo8_ar25_1	269179	157	733697	138	730500	137	729980	138	729918	137	729903	138
fo8_ar3_1	202993	109	191768	46	182710	43	182371	44	182340	44	180710	44
fo8_ar4_1	140062	120	150836	53	136182	65	145590	72	126452	64	92835	52
fo8_ar5_1	182645	133	544302	120	475015	124	597211	135	501529	127	519301	117
fo8	226582	180	594971	167	598557	169	631827	176	620740	174	618466	173
fo9_ar2_1	3763883	2779	5811784	1148	8739087	1358	5839421	1136	6386396	1244	5573359	1065
fo9_ar25_1	10780884	> 7200	35779189	6243	35684117	6228	36104803	6364	35447904	6353	36794119	6759
fo9_ar3_1	121189	113	630474	191	471243	147	782951	227	508882	164	567036	177
fo9_ar4_1	559215	484	1462533	465	1929344	553	968852	346	1676248	511	1810449	525
fo9_ar5_1	2457271	1982	2850095	1045	2510947	959	2007240	791	2218384	860	2837909	1056
fo9	2357571	2030	2041971	557	5021402	1166	1861546	515	1849769	513	1921768	532
fuzzy	1585116	> 7200	10023015	> 7200	8462365	> 7200	6948795	> 7200	8738087	> 7200	8115071	> 7200
ghg_1veh	8935058	> 7200	18594	! 7200	28933	! 7200	280997	! 7200	14047	! 7200	13950	! 7200
ghg_3veh	287687	> 7200	2555840	> 7200	2406659	> 7200	2405426	> 7200	1968285	> 7200	3592804	> 7200
hda	354972	> 7200	1319819	> 7200	1898941	> 7200	1970578	> 7200	1680755	> 7200	1624566	> 7200
lop97ic	120167	> 7200	527440	> 7200	762745	> 7200	568072	> 7200	619989	> 7200	620526	> 7200
lop97icx	4155061	> 7200	20501488	> 7200	20323880	> 7200	20067847	> 7200	20430108	> 7200	20150643	> 7200
mbtd	238	> 7200	22	> 7200	1	> 7200	1	> 7200	32	> 7200	22	> 7200
netmod_dol1	60544	2431	831440	> 7200	829894	> 7200	797756	> 7200	830953	> 7200	829147	> 7200
no7_ar3_1	334433	215	1139459	257	1151387	259	1145184	258	1152217	258	1138864	257
no7_ar4_1	196757	146	912547	202	850199	201	920112	203	914816	202	856896	204
no7_ar5_1	100305	71	292324	78	306521	81	304009	80	308110	79	303954	79
nous1	3246625	> 7200	573133	! 7200	273645	! 7200	539698	! 7200	344206	! 7200	4611479	! 7200
nuclear10a	1	> 7200	53	> 7200	53	> 7200	53	> 7200	53	> 7200	53	> 7200
nuclear10b	1	> 7200	17	> 7200	16	> 7200	16	> 7200	17	> 7200	16	> 7200
nuclear14a	20547	> 7200	2747	> 7200	2747	> 7200	1882	> 7200	5078	> 7200	2747	> 7200
nuclear14b	27545	> 7200	97802	> 7200	97964	> 7200	97972	> 7200	98036	> 7200	98013	> 7200
nuclear25a	22200	> 7200	2824	> 7200	2464	> 7200	6569	> 7200	5117	> 7200	5367	> 7200
nuclear25b	19011	> 7200	84105	> 7200	84101	> 7200	84153	> 7200	84209	> 7200	84139	> 7200
nuclear49a	1423	> 7200	180	> 7200	178	> 7200	181	> 7200	180	> 7200	178	> 7200
nuclear49b	2135	> 7200	3426	> 7200	4843	> 7200	4565	> 7200	6102	> 7200	1886	> 7200
nvs09	18167003	> 7200	256235	! 7200	240053	! 7200	250168	! 7200	375048	! 7200	305545	! 7200
o7_2	1712445	1268	1440404	332	1443884	334	1444696	332	1446732	332	1441533	332
o7_ar2_1	136342	88	388692	84	386113	78	399248	85	256046	68	246555	64
o7_ar25_1	571227	429	1008091	221	1011031	222	1012629	222	1010864	222	1007948	222
o7_ar3_1	1072205	811	2178610	474	1886281	464	2165857	468	2159799	465	2180282	474
o7_ar4_1	1826075	1478	1286052	732	1214438	691	1243887	717	1238410	717	1256653	711
o7_ar5_1	692346	514	1651732	418	1623912	393	1804285	426	1744544	428	1645883	418
o7	2514799	1765	3670381	776	3666527	774	3665744	776	3675888	776	3673982	778
o8_ar4_1	7880052	> 7200	18318695	4426	18473306	4474	18387973	4439	18359604	4447	18322512	4427
o9_ar4_1	7430950	> 7200	10756715	7191	10776074	> 7200	10879109	> 7200	10297478	7062	10672331	> 7200
oil2	900257	> 7200	6905564	> 7200	4978548	> 7200	6457890	> 7200	5754460	> 7200	7804032	> 7200
oil	174359	> 7200	345769	> 7200	416704	> 7200	1002907	> 7200	999362	> 7200	757104	> 7200
pb302035	982598	> 7200	4039918	> 7200	4141148	> 7200	4191117	> 7200	4268354	> 7200	4069185	> 7200
pb302055	935753	> 7200	5431636	> 7200	4268845	> 7200	4185877	> 7200	4161476	> 7200	4058472	> 7200
pb302075	1003884	> 7200	4444158	> 7200	4441808	> 7200	4443191	> 7200	4444259	> 7200	4432590	> 7200
pb302095	1330925	> 7200	9015095	> 7200	9063748	> 7200	8986409	> 7200	9048688	> 7200	9067327	> 7200
pb351535	1084866	> 7200	4787791	> 7200	4788423	> 7200	5858564	> 7200	4790986	> 7200	4781073	> 7200
pb351555	1382010	> 7200	4644701	> 7200	4640221	> 7200	4630128	> 7200	4631447	> 7200	4637019	> 7200
pb351575	1140277	> 7200	4878882	> 7200	4876086	> 7200	4882624	> 7200	4517807	> 7200	4879988	> 7200
pb351595	1312069	> 7200	6486937	> 7200	5972223	> 7200	6642799	> 7200	6476313	> 7200	6440787	> 7200
product2	2053746	> 7200	20653254	> 7200	21339260	> 7200	21703194	> 7200	22408926	> 7200	21464437	> 7200
qap	6819196	> 7200	23288193	> 7200	24689088	> 7200	27289323	> 7200	26340681	> 7200	22864190	> 7200
qapw	1624641	> 6912	5967547	> 7200	6585605	> 7200	5962659	> 7200	7136379	> 7200	6138323	> 7200
saa_2	1066	> 7200	2	> 7200	926	! 7200	938	! 7200	2	> 7200	840	! 7200
space25a	417210	> 7200	9904663	> 7200	509172	> 7200	587471	> 7200	2103903	> 7200	547764	> 7200
space960	2535	> 7200	22204	> 7200	22172	> 7200	2640	> 7200	22213	> 7200	3380	> 7200
stockcycle	26114	179	59487	112	64710	117	66098	119	69454	122	78847	135
super1	57058	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super2	62412	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super3	89036	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super3t	84138	> 7200	357425	> 7200	481771	> 7200	389744	> 7200	391731	> 7200	338342	> 7200
synheat	19811144	> 7200	30010130	> 7200	40450259	> 7200	1751302	! 7200	31009463	> 7200	39701239	> 7200
tln12	241317	> 7200	463756	> 7200	484588	> 7200	134913	> 7200	474206	> 7200	455869	> 7200
tln5	955333	678	1198258	615	2236748	703	1864270	831	458880	201	1130696	222
tln6	478059	> 7200	1632987	> 7200	10550801	> 7200	11256910	> 7200	10950320	> 7200	1520317	> 7200
tls4	29220	72	13997	15	11880	11	6936	10	9819	10	10506	11
waste	3252200	> 7200	13225050	> 7200	13583630	> 7200	13486890	> 7200	13128913	> 7200	13320013	> 7200
waterx	806977	> 7200	3749303	> 7200	3750381	> 7200	3751809	> 7200	3748270	> 7200	3747995	> 7200
waterz	1029662	> 7200	7864470	> 7200	14197862	> 7200	14263701	> 7200	14433207	> 7200	12083292	> 7200
geom. mean (solved: 26): 176.15	391149	313.2	521868	179.8	551568	184.4	516141	180.2	480654	167.9	495175	168.4
geom. mean (all): 2328.20	–	2755.6	–	2342.6	–	2360.8	–	2344.5	–	2294.6	–	2298.6
geom. mean (solved: 28): 223.66			689169	228.9	725664	234.4	682455	229.5	638321	214.9	657036	215.9
solved/timeout/abort	27/59/0		29/48/9		28/49/9		28/48/10		29/50/7		28/50/8	
speedup (solved: 26): 1.780			1.74		1.70		1.74		1.86		1.86	

Table 31: 5 runs with eight SOLVER threads on selected instances from MINLPLib

Name	One thread Nodes	One thread Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
4stufen	2455190	> 7200	12033095	> 7200	15852047	> 7200	22286235	> 7656	23020391	> 7476	34203808	> 7467
beuster	2019633	> 7200	4135192	! 7200	223181	! 7200	56872033	> 7200	3026206	! 7200	67651660	> 7200
cecil_13	1608177	> 7200	10984760	> 7200	11183498	> 7200	11073979	> 7200	11139108	> 7200	11028381	> 7200
chp_partload	8353	> 7200	6079	> 7200	1	! 1	9045	> 7200	1	! 1	9057	> 7200
csched2a	1235475	> 7200	3091464	> 7200	3095075	> 7200	3149039	> 7200	3089886	> 7200	3163849	> 7200
csched2	182072	> 7200	822985	> 7200	1140405	> 7200	544494	> 7200	533560	> 7200	938737	> 7200
eg_all_s	139	> 7200	312	> 7200	574	> 7200	261	> 7200	62	> 7200	1068	> 7200
eg_disc2_s	133	> 7200	117	> 7200	116	> 7200	86	> 7200	117	> 7200	117	> 7200
eg_disc_s	1	> 7200	65	> 7200	65	> 7200	65	> 7200	65	> 7200	65	> 7200
eg_int_s	4	> 7200	4	> 7200	4	> 7200	4	> 7200	4	> 7200	4	> 7200
enpro48pb	279848	70	30	1	30	1	30	1	30	1	30	1
ex1233	7960984	> 7200	107083	! 7200	100841	! 7200	81258	! 7200	407826	! 7200	125777	! 7200
fac2	4565461	> 7200	4145	! 7200	97651387	> 7200	79494184	> 7200	235	! 7200	653786	! 7200
feedtray	274621	> 7200	3950717	> 7200	3752993	> 7200	3096609	> 7200	3701396	> 7200	3895314	> 7200
fo7	201538	132	246752	43	236999	43	224567	42	247206	43	248810	45
fo8_ar2_1	435919	259	1321109	160	1315191	158	1302143	148	1234877	146	1416874	152
fo8_ar25_1	269179	157	731066	108	608888	82	647065	92	771176	116	809433	95
fo8_ar3_1	202993	109	144347	26	151387	25	153571	28	201626	27	154877	29
fo8_ar4_1	140062	120	101315	41	149892	54	182673	58	96818	38	198120	57
fo8_ar5_1	182645	133	604771	87	646099	114	532854	84	578545	86	505181	78
fo8	226582	180	577162	97	576158	93	576919	93	580117	98	574081	95
fo9_ar2_1	3763883	2779	5192236	708	5152661	! 7200	4846507	729	5503414	761	4399944	598
fo9_ar25_1	10780884	> 7200	33304444	3427	35043974	3615	30292659	3173	34994196	3680	38235650	4366
fo9_ar3_1	121189	113	1062324	163	1138025	172	1377085	199	1000274	167	1369034	211
fo9_ar4_1	559215	484	1778287	341	1927962	361	2052871	378	1906236	375	2308804	409
fo9_ar5_1	2457271	1982	2019190	490	3114197	681	1804832	451	3194335	682	3343729	672
fo9	2357571	2030	1864856	306	1936322	318	1660398	281	1524991	269	2013510	321
fuzzy	1585116	> 7200	17123093	> 7200	16728957	> 7200	13741904	> 7200	9825960	> 7200	14736156	> 7200
ghg_1veh	8935058	> 7200	9927	! 7200	15769	! 7200	25385	! 7200	23082	! 7200	10304	! 7200
ghg_3veh	287687	> 7200	1339122	> 7200	1383414	> 7200	3278664	> 7200	1524479	> 7200	2056687	> 7200
hda	354972	> 7200	2332139	> 7200	2298413	> 7200	2451015	> 7200	2450766	> 7200	2254485	> 7200
lop97ic	120167	> 7200	646199	> 7200	650835	> 7200	573431	> 7200	588521	> 7200	557847	> 7200
lop97icx	4155061	> 7200	30538789	> 7200	31299204	> 7200	31691975	> 7200	30954987	> 7200	31078684	> 7200
mbtd	238	> 7200	1	> 7200	1	> 7200	1	> 7200	1	> 7200	1	> 7200
netmod_dol1	60544	2431	1450368	> 7200	1425750	> 7200	1431416	> 7200	1207269	6507	1438466	> 7200
no7_ar3_1	334433	215	1136073	164	1109829	152	1165448	161	1128204	155	1116733	157
no7_ar4_1	196757	146	537976	89	592435	93	549182	84	545558	82	626282	99
no7_ar5_1	100305	71	279909	55	280088	56	275427	59	256007	48	277131	56
nous1	3246625	> 7200	1394092	! 7200	523996	! 7200	764004	! 7200	1844853	! 7200	178850	! 7200
nuclear10a	1	> 7200	26	> 7200	26	> 7200	28	> 7200	27	> 7200	27	> 7200
nuclear10b	1	> 7200	14	> 7200	16	> 7200	15	> 7200	11	> 7200	15	> 7200
nuclear14a	20547	> 7200	7056	> 7200	11672	> 7200	11672	> 7200	6619	> 7200	7207	> 7200
nuclear14b	27545	> 7200	159323	> 7200	151675	> 7200	21760	> 7200	151426	> 7200	159545	> 7200
nuclear25a	22200	> 7200	6386	> 7200	6390	> 7200	6384	> 7200	6324	> 7200	6380	> 7200
nuclear25b	19011	> 7200	164895	> 7200	164956	> 7200	164964	> 7200	164932	> 7200	105484	> 7200
nuclear49a	1423	> 7200	956	> 7200	962	> 7200	946	> 7200	946	> 7200	884	> 7200
nuclear49b	2135	> 7200	54	> 7200	54	> 7200	60	> 7200	60	> 7200	54	> 7200
nvs09	18167003	> 7200	7996	! 7200	105279	! 7200	5663	! 7200	4238	! 7200	89704	! 7200
o7_2	1712445	1268	1662135	205	1612626	202	1623607	218	1627693	208	1625435	209
o7_ar2_1	136342	88	240563	44	317402	! 7200	318530	! 7200	248355	44	251244	46
o7_ar25_1	571227	429	1049534	150	1173570	! 7200	961521	139	1038808	156	1055023	156
o7_ar3_1	1072205	811	1957794	324	1878914	291	1913462	319	1907804	304	2054804	323
o7_ar4_1	1826075	1478	1381503	507	1159115	451	1197005	467	1320063	482	1141514	449
o7_ar5_1	692346	514	1761340	285	2206006	352	1471924	240	2576377	365	2493977	354
o7	2514799	1765	3499660	439	3456239	444	3442085	456	3525205	463	3501460	441
o8_ar4_1	7880052	> 7200	19067153	3419	19690506	3340	19457535	3478	20611322	3463	21533468	4255
o9_ar4_1	7430950	> 7200	12158990	4215	11407137	4076	11818736	4282	10397894	3961	12086431	4326
oil2	900257	> 7200	323124	532	25736	131	6728710	> 7200	7568222	> 7200	207369	745
oil	174359	> 7200	1178218	> 7200	1108114	> 7200	1036154	> 7200	1	! 1	1	! 1
pb302035	982598	> 7200	4920362	> 7200	5014100	> 7200	5023188	> 7200	4918437	> 7200	5116637	> 7200
pb302055	935753	> 7200	5028859	> 7200	5031086	> 7200	5027951	> 7200	5021697	> 7200	5009082	> 7200
pb302075	1003884	> 7200	5584437	> 7200	5587891	> 7200	5588603	> 7200	5576462	> 7200	5586124	> 7200
pb302095	1330925	> 7200	8614776	> 7200	7403973	> 7200	7267930	> 7200	7312612	> 7200	8663940	> 7200
pb351535	1084866	> 7200	6211008	> 7200	6234805	> 7200	6213995	> 7200	6551712	> 7200	6252899	> 7200
pb351555	1382010	> 7200	6616606	> 7200	6291594	> 7200	6505900	> 7200	6481530	> 7200	6289345	> 7200
pb351575	1140277	> 7200	6581971	> 7200	6291427	> 7200	6307275	> 7200	6302357	> 7200	6602648	> 7200
pb351595	1312069	> 7200	6871874	> 7200	6887632	> 7200	6882440	> 7200	6878978	> 7200	6886228	> 7200
product2	2053746	> 7200	34360899	> 7200	32558591	> 7200	33115708	> 7200	33870590	> 7200	32832781	> 7200
qap	6819196	> 7200	35277773	> 7200	35587971	> 7200	33834718	> 7200	34623296	> 7200	30006983	> 7200
qapw	1624641	> 6912	10146832	> 7200	10820443	> 7200	8658298	> 7200	12100861	> 7200	10830118	> 7200
saa_2	1066	> 7200	313	> 7200	564	> 7200	220	> 7200	1405	! 7200	98	> 7200
space25a	417210	> 7200	9085675	> 7200	6156412	> 7200	14361749	> 7200	8878438	> 7200	9490889	> 7200
space960	2535	> 7200	3350	> 7200	4294	> 7200	3350	> 7200	3350	> 7200	9404	> 7200
stockcycle	26114	179	82634	89	42952	64	79105	86	89020	93	61910	75
super1	57058	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super2	62412	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super3	89036	> 7200	1	! 1	1	! 1	1	! 1	1	! 1	1	! 1
super3t	84138	> 7200	626321	> 7200	490952	> 7200	554117	> 7200	583393	> 7200	548388	> 7200
synheat	19811144	> 7200	49225606	! 7200	80134106	> 7200	112570775	> 7200	105790967	> 7200	59013114	! 7200
tln12	241317	> 7200	504391	> 7200	792103	> 7200	796505	> 7200	553625	> 7200	449659	> 7200
tln5	955333	678	1398511	401	686373	188	427094	82	1349762	219	201371	48
tln6	478059	> 7200	3210473	> 7200	4230220	> 7200	4563693	> 7200	8056125	> 7200	8153379	> 7200
tls4	29220	72	7041	8	9442	8	9976	8	5854	8	8900	7
waste	3252200	> 7200	17161873	> 7200	17164966	> 7200	17332460	> 7200	17460938	> 7200	17238692	> 7200
waterx	806977	> 7200	6853080	> 7200	6051828	> 7200	6898720	> 7200	5984326	> 7200	6013572	> 7200
waterz	1029662	> 7200	16110109	> 7200	17062767	> 7200	19107092	> 7200	16703260	> 7200	14754598	> 7200
geom. mean (solved: 23):	106.89		365039	296.8	406184	110.4	403776	108.1	392831	102.7	416403	108.5	408799	104.7
geom. mean (all):	2039.29		–	2755.6	–	1958.0	–	2183.8	–	2098.0	–	2009.6	–	1947.0
geom. mean (solved: 26):	160.27				635911	165.4	633044	162.4	614945	154.9	649316	163.1	645803	160.8
solved/timeout/abort			27/59/0		30/46/10		27/47/12		28/50/8		30/44/12		30/46/10	
speedup (solved: 23):	2.779				2.69		2.75		2.89		2.73		2.84	

Table 32: SCIP, ParaSCIP with seven SOLVERs, each running on a different machine, ParaSCIP with seven SOLVERs, all running on the same machine, and FiberSCIP with seven SOLVERs

Name	SCIP Nodes	SCIP Time	ParaSCIP (Distributed) Nodes	ParaSCIP (Distributed) Time	ParaSCIP (Share) Nodes	ParaSCIP (Share) Time	FiberSCIP Nodes	FiberSCIP Time
30n20b8	14	181	3	63	3	132	4	208
acc-tight5	8900	695	270	74	270	125	270	234
aflow40b	217954	1561	146507	2090	146507	3405	106559	1942
air04	178	73	152	61	152	117	152	127
app1-2(i)	677	1169	677	1185	677	2399	677	2716
ash608gpia-3col(i)	10	22	10	22	10	48	10	74
bab5	27492	> 7201	29508	> 7200	10852	> 7202	7550	> 7200
beasleyC3	883797	> 7201	733632	> 7201	422172	> 7201	346532	> 7200
biella1	2576	536	1197	516	1197	963	1197	1209
bienst2(i)	85969	245	85969	246	85969	404	85969	617
binkar10_1	136487	175	58760	77	58760	136	58760	206
bley_xl1	20	440	1	291	1	555	1	679
bnatt350	7691	641	4818	422	4818	748	4818	1014
core2536-691(i)	156	227	156	228	156	452	156	532
cov1075	1621967	> 7201	1710242	> 7201	1	> 7203	631742	> 7200
csched010	983615	> 7201	1067696	7028	641652	> 7201	481362	> 7200
danoint(i)	1006690	4804	1006690	4799	891092	> 7202	574242	> 7200
dfn-gwin-UUM	58405	119	57278	118	57278	201	57278	354
eil33-2(i)	735	53	735	53	781	117	735	189
eilB101(i)	5008	320	5008	319	5008	570	5008	671
enlight13(i)	131423	76	131423	76	131423	160	131423	229
enlight14(i)	709364	400	709364	405	709364	819	709364	1232
ex9(i)	1	38	1	37	1	79	1	135
glass4	4701308	2090	20681292	> 7200	-	-	1	! 1999
gmu-35-40	18719417	> 7210	1	> 7202	-	-	1	! 1868
iis-100-0-cov	97505	1605	91650	1428	94285	2540	91650	2913
iis-bupa-cov	178316	6034	178316	6034	116240	> 7201	83242	> 7200
iis-pima-cov	19275	1363	6281	612	6281	1193	6281	1453
lectsched-4-obj	123437	1214	3322	178	3322	345	3322	471
m100n500k4r1	7484704	> 7200	4883702	4796	4483743	> 7201	3274242	> 7200
macrophage	1337678	> 7201	981612	> 7201	434792	> 7202	514262	> 7201
map18	383	436	631	431	631	1428	631	1410
map20	325	355	852	371	852	986	852	1248
mcsched	27301	266	13660	175	13660	293	13660	423
mik-250-1-100-1	699043	248	2424374	675	2424374	1113	680680	676
mine-166-5	2199	35	1960	19	1960	38	1960	41
mine-90-10	222131	1045	49309	206	49309	401	49309	595
msc98-ip	3814	> 7201	3797	> 7201	1	> 7205	1347	> 7203
mspp16	1	! 480	1	33	1	48	1	! 85
mzzv11	3271	354	2285	297	2523	657	2285	953
n3div36	251029	> 7203	254293	> 7200	-	-	1	! 2748
n3seq24	396	> 7205	1	> 7206	-	-	1	! 64
n4-3	32248	557	32680	564	32680	972	43114	1462
neos-1109824	16639	139	8337	86	8337	228	8337	263
neos-1337307	413108	> 7201	378238	> 7201	143705	> 7201	126806	> 7200
neos-1396125	70646	739	48174	577	48174	864	48174	1484
neos13(i)	4430	1502	4430	1438	4430	2497	4430	4490
neos-1601936	4230	7201	15542	> 7201	6318	> 7202	4427	> 7202
neos18	6778	31	5669	28	4965	50	5669	73
neos-476283	660	234	488	194	-	-	1	! 42
neos-686190	7776	90	4757	69	4288	140	4288	196
neos-849702	41469	833	3142	109	3142	185	3142	347
neos-916792	263990	1187	110312	433	110312	975	110312	1130
neos-934278	801	> 7200	1289	> 7203	1	> 7204	1	> 7203
net12	3970	2882	3135	2421	3135	5046	3135	5259
netdiversion	72	> 7202	27	3198	1	> 7236	1	> 7206
newdano	2522637	3941	1817385	3490	1817385	5669	1689135	> 7201
noswot	769339	143	586194	125	586194	194	586194	323
ns1208400	4004	2795	1426	1013	1426	1639	1426	2142
ns1688347	6077	576	4781	319	4781	556	4781	606
ns1758913	2	> 7204	1	> 7207	-	-	1	! 89
ns1766074(i)	944135	652	944135	663	944135	1446	944135	1834
ns1830653	58831	763	30124	333	30124	569	30124	874
opm2-z7-s2	6940	1200	2058	496	1668	1535	2058	1615
pg5_34	290435	1537	265066	1425	265066	2589	265066	2990
pigeon-10	18543130	> 7201	17078156	> 7202	1	> 7202	1	> 7200
pw-myciel4	870147	4247	749096	3370	749096	6284	718480	> 7200
qiu	12487	83	9430	57	9430	93	9862	110
rail507	799	236	738	207	738	576	738	553
ran16x16	328459	250	270696	224	270696	356	270696	720
reblock67	118660	253	63005	160	63005	271	68503	442
rmatr100-p10(i)	862	140	862	141	862	239	862	353
rmatr100-p5(i)	439	283	439	283	439	522	439	749
rmine6	1574759	4869	621120	2251	621120	5454	521375	4867
rocII-4-11	19884	296	15413	266	17857	504	15413	851
rococoC10-001000	321750	1762	202044	1113	202044	1931	202044	2479
roll3000	2090131	5494	267388	964	267388	1742	267388	2183
satellites1-25	10212	1465	8927	1538	8927	2896	10255	4544
sp98ic	44857	> 7201	37192	> 7201	8637	> 7201	9391	> 7201
sp98ir	6344	92	5890	84	5730	154	5890	273
tanglegram1	37	1148	37	1128	37	3252	39	2447
tanglegram2	5	15	3	11	3	19	3	28
timtab1(i)	627924	257	627924	264	627924	427	627924	710
triptim1	63	1751	1	331	1	980	1	1010
unitcal_7	95086	4302	25483	1777	25483	5918	25483	5671
vpphard	2691	> 7201	9746	> 7201	1	> 7205	437	> 7201
zib54-UUE	584005	4398	173678	1473	173678	2610	173678	3193
geom. mean (solved: 62)	9794	409.2	5421	271.2	5409	531.3	5335	695.6
geom. mean (all)	–	866.9	–	604.1	–	1056.2	–	1361.5
geom. mean (solved: 62)			5421	271.2	5409	531.3	5335	695.6
solved/timeout/abort	68/18/1		71/16/0		65/16/6		62/18/7	
speedup (solved: 62)			1.51		0.77		0.59	

Table 33: Tree search after racing ramp-up with ParaSCIP (Seven SOLVER Processes on a cluster node)

Name	SCIP Nodes	SCIP Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	14	181	5	256	6	334	8	303	6	333	6	332
acc-tight5	8900	695	50	71	50	70	50	70	50	71	50	70
aflow40b	217954	1561	122699	954	92142	831	114296	979	42437	408	53017	520
air04	178	73	88	99	87	102	90	104	83	101	108	100
app1-2	677	1169	773	6566	647	5135	726	5020	726	4972	647	4916
ash608gpia-3col	10	22	10	50	10	50	10	49	10	44	10	49
bab5	27492	> 7201	164	> 7205	164	> 7203	163	> 7203	161	> 7203	164	> 7204
beasleyC3	883797	> 7201	167	> 7202	167	> 7203	167	> 7203	167	> 7202	167	> 7203
biella1	2576	536	6481	669	5762	717	11725	794	2056	459	7123	636
bienst2	85969	245	161285	149	148827	145	158701	157	144375	136	148171	153
binkar10_1	136487	175	31735	34	51048	63	40611	38	16954	28	14950	27
bley_xl1	20	440	1	1214	1	1158	1	1175	1	1238	1	1235
bnatt350	7691	641	1477	442	1747	447	1588	447	1547	416	1567	455
core2536-691	156	227	59	286	59	320	59	335	35	326	51	266
cov1075	1621967	> 7201	1513407	2159	1667502	2312	1660088	2353	1716884	2332	1666806	2304
csched010	983615	> 7201	829762	3140	785616	3103	818766	3239	814202	3187	814685	3060
danoint	1006690	4804	1051712	1339	1073407	1348	1337347	1627	1067099	1392	1040870	1369
dfn-gwin-UUM	58405	119	29422	54	35973	60	29588	55	33378	61	35483	59
eil33-2	735	53	539	112	749	109	623	106	693	107	637	108
eilB101	5008	320	1918	273	1615	276	1183	213	2922	248	1538	245
enlight13	131423	76	85488	80	79030	79	92465	82	81515	77	85240	88
enlight14	709364	400	785848	931	785848	873	785848	904	785848	897	785848	883
ex9	1	38	1	53	1	52	1	52	1	54	1	54
glass4	4701308	2090	-	-	-	-	-	-	-	-	-	-
gmu-35-40	18719417	> 7210	-	-	-	-	-	-	-	-	-	-
iis-100-0-cov	97505	1605	70325	432	73430	531	70051	492	70604	494	70389	497
iis-bupa-cov	178316	6034	152705	2023	153166	2039	154028	1925	152722	2014	159095	1978
iis-pima-cov	19275	1363	6089	733	5225	604	6980	595	6870	615	19133	726
lectsched-4-obj	123437	1214	47770	478	21614	344	27989	361	27187	333	38328	463
m100n500k4r1	7484704	> 7200	1090922	278	1087283	276	24431582	5923	1087143	275	579755	163
macrophage	1337678	> 7201	219	> 7202	219	> 7202	219	> 7202	219	> 7202	219	> 7201
map18	383	436	265	804	317	904	323	839	315	895	295	892
map20	325	355	297	971	304	881	297	952	323	839	300	952
mcsched	27301	266	25228	207	15724	229	21699	228	14336	208	16489	203
mik-250-1-100-1	699043	248	2801931	253	2837580	249	2863666	250	2859730	252	2846811	251
mine-166-5	2199	35	1393	52	1647	52	2552	51	1190	50	1173	49
mine-90-10	222131	1045	194567	213	156884	184	113283	170	207409	234	223497	330
msc98-ip	3814	> 7201	55	> 7205	55	> 7204	55	> 7205	55	> 7204	101	> 7205
mspp16	1	! 480	-	-	-	-	-	-	-	-	-	-
mzzv11	3271	354	9442	1248	924	1158	2528	889	823	1129	5000	1114
n3div36	251029	> 7203	322595	> 7204	221500	> 7203	223032	> 7202	-	-	-	-
n3seq24	396	> 7205	-	-	-	-	-	-	-	-	-	-
n4-3	32248	557	71488	771	71768	684	93866	903	91655	893	76506	768
neos-1109824	16639	139	1830	107	1963	98	1669	90	1964	99	2279	110
neos-1337307	413108	> 7201	1025382	6414	973054	6852	507429	> 7202	622095	> 7202	985544	6385
neos-1396125	70646	739	62538	393	45849	352	77980	395	71134	424	37310	303
neos13	4430	1502	1091	2211	4532	3383	34009	7203	12683	6223	1128	2695
neos-1601936	4230	7201	179	> 7204	6941	3958	2160	2285	17991	6299	10055	4637
neos18	6778	31	3992	62	2701	64	4055	63	6058	77	4243	64
neos-476283	660	234	-	-	-	-	-	-	-	-	-	-
neos-686190	7776	90	872	86	899	84	14185	130	1025	86	758	80
neos-849702	41469	833	4788	221	4788	207	4788	224	4788	223	4788	222
neos-916792	263990	1187	182108	> 7202	1200303	> 7202	700673	> 7202	1303128	> 7203	873190	> 7203
neos-934278	801	> 7200	9	> 7204	6	> 7204	2070	5885	2410	5864	9	> 7204
net12	3970	2882	6205	1762	5674	1701	5578	1572	5815	1688	5575	1701
netdiversion	72	> 7202	1	> 7209	1	> 7207	1	> 7209	1	> 7207	1	> 7207
newdano	2522637	3941	2550262	1761	2439242	2645	2550472	1774	3219746	1611	2629258	1811
noswot	769339	143	1128103	94	1125789	96	1131871	94	1198274	116	1129018	95
ns1208400	4004	2795	670	1480	4199	3417	665	1469	598	1547	710	1589
ns1688347	6077	576	5303	449	3596	456	6225	312	8138	392	1118	269
ns1758913	2	> 7204	-	-	-	-	-	-	-	-	-	-
ns1766074	944135	652	947776	1398	947776	1401	947776	1435	947776	1413	947776	1439
ns1830653	58831	763	33192	285	27814	283	45133	279	35756	271	28962	231
opm2-z7-s2	6940	1200	1086	1036	1590	1286	2821	1367	3921	2127	1258	1073
pg5_34	290435	1537	167834	478	154939	486	190742	497	200380	566	189390	504
pigeon-10	18543130	> 7201	4530	> 7201	5095	> 7201	5219	> 7201	4498	> 7201	5223	> 7201
pw-myciel4	870147	4247	2503863	3978	2156896	3512	2096905	3215	2321417	3774	1933226	3001
qiu	12487	83	8711	102	10162	87	9315	111	8652	92	12764	100
rail507	799	236	682	471	744	475	688	501	756	509	682	458
ran16x16	328459	250	154306	95	155478	93	159675	94	135930	91	149329	93
reblock67	118660	253	176711	215	99867	113	138355	131	89363	116	91864	119
rmatr100-p10	862	140	349	158	386	158	356	157	403	158	408	159
rmatr100-p5	439	283	287	382	293	416	287	393	291	397	293	412
rmine6	1574759	4869	855236	1095	701762	898	661948	854	871641	906	645901	876
rocII-4-11	19884	296	106191	327	29187	206	69028	286	66529	257	71824	292
rococoC10-001000	321750	1762	933193	1707	434755	1472	1026186	1750	692233	1907	370526	1237
roll3000	2090131	5494	302558	765	220204	550	506977	968	188989	870	350578	997
satellites1-25	10212	1465	5473	1326	4313	1709	5340	1808	10618	2437	7756	1841
sp98ic	44857	> 7201	57523	> 7202	28491	> 7202	463	> 7202	73518	> 7204	11817	> 7202
sp98ir	6344	92	1064	93	1188	88	1017	86	1077	89	6556	116
tanglegram1	37	1148	73	3251	50	2156	73	3240	73	3184	43	2226
tanglegram2	5	15	3	17	3	16	3	17	3	17	3	17
timtab1	627924	257	1933989	278	1875732	260	2001428	278	1721324	243	1809283	256
triptim1	63	1751	1	588	1	587	1	480	1	558	1	579
unitcal_7	95086	4302	19081	2243	31212	2758	23783	2855	55979	> 7204	10515	2278
vpphard	2691	> 7201	115	> 7204	115	> 7205	115	> 7204	59	> 7204	59	> 7205
zib54-UUE	584005	4398	247547	1447	290761	1551	306530	1745	281414	1659	275684	1553
geom. mean (solved: 63): 354.04	12000	443.3	7004	360.2	6415	354.7	7477	355.8	6582	353.0	6653	346.5
geom. mean (all): 751.94	–	866.9	–	752.8	–	745.8	–	775.5	–	759.6	–	726.0
geom. mean (solved: 66): 378.06			8818	380.9	8113	375.7	9850	395.1	8323	374.2	8325	364.4
solved/timeout/abort	68/18/1		69/12/6		70/11/6		69/12/6		69/11/7		70/10/7	
speedup (solved: 63): 1.25			1.23		1.25		1.25		1.26		1.28	

Table 34: Tree search after racing ramp-up with FiberSCIP (Seven SOLVER Threads)

Name	SCIP Nodes	SCIP Time	Run - 1 Nodes	Run - 1 Time	Run - 2 Nodes	Run - 2 Time	Run - 3 Nodes	Run - 3 Time	Run - 4 Nodes	Run - 4 Time	Run - 5 Nodes	Run - 5 Time
30n20b8	14	181	6	333	6	327	6	303	6	326	6	305
acc-tight5	8900	695	50	119	50	126	50	72	75	127	50	126
aflow40b	217954	1561	194045	1338	81899	830	67826	908	32184	551	90283	956
air04	178	73	85	113	83	139	104	107	85	116	57	175
app1-2	677	1169	349	4650	400	6177	325	4283	409	6260	503	4289
ash608gpia-3col	10	22	10	60	10	75	10	70	10	57	10	73
bab5	27492	> 7201	34238	> 7202	133	> 7202	133	> 7201	133	> 7201	133	> 7203
beasleyC3	883797	> 7201	167	> 7202	157	> 7201	155	> 7201	155	> 7201	167	> 7202
biella1	2576	536	6486	1107	12794	1167	14358	1071	13141	1241	8443	761
bienst2	85969	245	168573	324	147688	293	154605	292	140593	284	146363	307
binkar10_1	136487	175	14778	45	18208	49	28802	93	42721	111	18853	45
bley_xl1	20	440	1	409	1	371	1	373	1	381	1	408
bnatt350	7691	641	1547	651	1555	529	1681	605	1553	602	1657	583
core2536-691	156	227	48	301	66	377	66	487	57	434	66	444
cov1075	1621967	> 7201	1659524	2989	1863821	3154	1660198	3130	1659070	3214	1676464	3025
csched010	983615	> 7201	786399	3802	796956	3918	789383	3991	784859	4520	807549	4031
danoint	1006690	4804	855714	2522	867073	1783	884722	2623	856875	2595	886004	2075
dfn-gwin-UUM	58405	119	36743	114	33146	110	33804	120	42379	125	112297	142
eil33-2	735	53	509	114	706	167	620	174	505	117	844	165
eilB101	5008	320	4309	387	2989	456	4596	371	4638	431	2232	389
enlight13	131423	76	75080	147	99963	131	106471	179	72794	154	76475	154
enlight14	709364	400	785848	1307	785848	1183	785848	1194	785848	1190	785848	1314
ex9	1	38	1	38	1	38	1	38	1	38	1	37
glass4	4701308	2090	20452320	3230	17528782	2835	20889708	2874	22208172	! 3218	1	! 3770
gmu-35-40	18719417	> 7210	1	! 1742	1	! 1860	1	! 1850	1	! 1915	1	! 2046
iis-100-0-cov	97505	1605	67812	606	73751	1012	71730	615	67460	854	69931	627
iis-bupa-cov	178316	6034	152695	2613	153024	2992	152748	2478	152648	2420	154584	2493
iis-pima-cov	19275	1363	5268	720	5164	762	5324	836	5304	785	5302	738
lectsched-4-obj	123437	1214	28049	512	29945	504	11862	414	20479	606	31166	488
m100n500k4r1	7484704	> 7200	529575	303	395435	244	5595737	7200	589093	347	7838806	2310
macrophage	1337678	> 7201	219	> 7200	219	> 7201	219	> 7200	219	> 7200	219	> 7203
map18	383	436	265	817	241	826	246	907	249	829	211	913
map20	325	355	315	912	307	900	315	911	312	828	301	886
mcsched	27301	266	20831	339	19555	389	23835	398	19528	339	19644	242
mik-250-1-100-1	699043	248	4039114	484	3719928	558	4061940	473	4097967	604	4063399	608
mine-166-5	2199	35	2323	50	1203	48	896	53	1254	74	1702	48
mine-90-10	222131	1045	73873	120	140956	198	148926	404	116706	258	209602	259
msc98-ip	3814	> 7201	1728	> 7204	55	> 7203	47	> 7204	4562	> 7203	55	> 7203
mspp16	1	! 480	1	! 499	1	! 494	1	! 497	1	! 491	1	! 494
mzzv11	3271	354	6064	1495	10134	1220	2745	1103	1126	1168	6400	1495
n3div36	251029	> 7203	196769	> 7202	196094	> 7201	218824	> 7201	193637	! 7202	181494	! 7202
n3seq24	396	> 7205	1	! 76	1	! 87	1	! 86	1	! 79	1	! 88
n4-3	32248	557	80159	1076	78894	1293	77949	1154	78337	1164	75333	1097
neos-1109824	16639	139	2488	145	1942	171	1327	171	7794	169	1746	172
neos-1337307	413108	> 7201	743901	7200	1002975	6957	748536	> 7200	726691	> 7200	1017020	7119
neos-1396125	70646	739	46988	433	45470	620	44767	632	44994	638	46499	609
neos13	4430	1502	23231	6735	1278	2300	19052	6720	2159	2379	12373	5241
neos-1601936	4230	7201	11624	> 5410	5182	3673	800	1442	828	1457	808	1340
neos18	6778	31	2189	102	1346	97	3815	124	1621	94	4120	114
neos-476283	660	234	1	! 31	1	! 33	1	! 34	1	! 32	1	! 33
neos-686190	7776	90	2946	174	678	116	659	141	965	143	949	152
neos-849702	41469	833	4788	425	4788	425	3905	418	2296	424	4788	418
neos-916792	263990	1187	122650	> 7201	86428	> 7201	116723	> 7201	165696	> 7201	84768	> 7201
neos-934278	801	> 7200	3	> 7204	3	> 7203	3	7204	3	7203	3	> 7203
net12	3970	2882	5003	2019	5624	2167	4999	2060	5567	1987	3849	1759
netdiversion	72	> 7202	1	> 7206	1	> 7207	1	> 7206	1	> 7206	1	> 7206
newdano	2522637	3941	2965091	3236	2922203	3881	2207151	4069	2564849	2870	1993532	4124
noswot	769339	143	1136704	157	1132747	181	1128260	183	1128969	181	1141144	170
ns1208400	4004	2795	674	2013	3728	3413	674	1887	7603	6276	2345	3095
ns1688347	6077	576	2981	397	1712	314	1907	388	2699	389	1353	386
ns1758913	2	> 7204	1	! 115	1	! 124	1	! 124	1	! 114	1	! 126
ns1766074	944135	652	947776	1980	947776	1759	947776	1711	947776	1705	947776	1730
ns1830653	58831	763	36300	487	27788	394	30591	436	30081	361	25976	360
opm2-z7-s2	6940	1200	1627	1261	3374	1593	1788	1297	2010	1331	1632	1450
pg5_34	290435	1537	400605	1068	385275	1115	399986	1142	379645	1169	360005	1166
pigeon-10	18543130	> 7201	4929	> 7200	4575	> 7200	1173	> 7201	5075	> 7200	4515	> 7200
pw-myciel4	870147	4247	2652697	5590	1996774	4655	2935022	7201	2103828	5510	1505715	4089
qiu	12487	83	8862	163	8940	187	9597	155	9174	174	9565	155
rail507	799	236	536	497	630	699	2660	727	558	593	558	618
ran16x16	328459	250	161399	210	151447	168	179353	151	309395	210	151234	175
reblock67	118660	253	102524	269	153311	217	127179	158	146304	244	181270	305
rmatr100-p10	862	140	351	286	347	181	360	286	335	282	361	273
rmatr100-p5	439	283	287	466	287	461	287	459	287	464	287	438
rmine6	1574759	4869	917442	1378	1023262	1477	853169	1558	617071	1029	921478	1220
rocII-4-11	19884	296	96679	313	61719	288	70933	440	32803	240	52083	431
rococoC10-001000	321750	1762	635459	2767	1417726	2858	644689	1849	463019	1425	833070	2008
roll3000	2090131	5494	797596	1857	895488	2056	1084543	2430	976542	1739	695568	1300
satellites1-25	10212	1465	19557	3409	14340	4717	2657	1975	9685	2761	23486	2357
sp98ic	44857	> 7201	2665	> 7202	91226	> 7202	7398	> 7202	101311	> 7201	290	> 7201
sp98ir	6344	92	2988	151	2740	141	921	136	850	161	1359	138
tanglegram1	37	1148	73	2806	73	2914	73	2854	73	2838	73	2840
tanglegram2	5	15	3	28	3	27	3	25	3	27	3	28
timtab1	627924	257	1817676	437	1924627	503	1781394	465	1783327	523	1709680	487
triptim1	63	1751	1	566	1	523	1	536	1	548	1	529
unitcal_7	95086	4302	79153	7203	7693	7204	30126	3546	15756	> 7203	47642	7203
vpphard	2691	> 7201	115	> 7203	115	> 7204	79	> 7204	115	> 7203	115	> 7203
zib54-UUE	584005	4398	299499	1893	278520	2170	269324	1864	264946	1903	295738	1957
geom. mean (solved: 63): 490.38	11035	436.0	6708	482.8	6503	493.4	6239	493.5	6154	492.5	6590	489.7
geom. mean (all): 963.56	–	866.9	–	949.4	–	954.8	–	979.9	–	961.5	–	972.2
geom. mean (solved: 66): 532.78			7904	531.2	7594	539.8	7083	532.5	6993	532.7	7467	527.7
solved/timeout/abort	68/18/1		69/13/5		70/12/5		68/14/5		68/13/6		69/12/6	
speedup (solved: 63): 0.89			0.90		0.88		0.88		0.89		0.89	