Primal Heuristics for Branch-and-Price Algorithms

Primalheuristiken für Branch-and-Price-Algorithmen
Master-Thesis von Christian Puchert
Januar 2011

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Arbeitsgruppe Optimierung

Primal Heuristics for Branch-and-Price Algorithms
Primalheuristiken für Branch-and-Price-Algorithmen

Vorgelegte Master-Thesis von Christian Puchert

1. Gutachten: Prof. Dr. Marco Lübbecke
2. Gutachten: Prof. Dr. Stefan Ulbrich

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 26. Januar 2011

(Christian Puchert)

Contents

1 Introduction

Many problems in industry and economy are modeled as *Mixed Integer Programs (MIPs)*. There exist efficient algorithms to solve Linear Programs like the Simplex method or the Ellipsoid method, the latter being a polynomial-time algorithm. However, no algorithm is known that solves a MIP within polynomial time; the problem is $\mathcal{NP}$-complete [GJ79]. Many MIP solvers – such as the SCIP framework [ABH$^+$, Ach07, Ach09], CPLEX [CPL09] or Gurobi [Gur10] – are based on the *branch-and-bound* algorithm which is basically a *divide-and-conquer* approach. An important variant of this algorithm is *branch-and-price*; it uses a reformulation of the original MIP, the *Dantzig-Wolfe Decomposition* [DW60], and thus exploits special structures in MIPs.

An important aspect of both of these algorithms are *primal heuristics*. As both branch-and-bound and branch-and-price need exponential solving time in the worst case, it is crucial to find good feasible solutions as soon as possible in the solving process. Primal heuristics are methods which search for such solutions and thus may help accelerate the solving process a lot. This thesis deals with primal heuristics in the context of branch-and-price algorithms. The problem reformulation provides some additional information about the problem structure and allows for specifically tailored heuristics which are potentially superior to standard heuristics used in a branch-and-bound algorithm. We developed such sophisticated heuristics and also modified standard heuristics such as rounding or diving methods such that they run in the branch-and-price context, too. The heuristics were implemented into the branch-and-price solver GCG [Gam10, GL10] which itself is an extension of the SCIP framework. Furthermore, we investigated in how far these primal heuristics lead to an improvement of the overall performance of the branch-and-price process.

The thesis is organized as follows: The remainder of this chapter deals with the mathematical background; we will give basic definitions of Mixed Integer Programming and present a brief outline of the branch-and-bound algorithm; furthermore, we will give a short description of the Dantzig-Wolfe decomposition. Chapter 2 gives an overview of the software used and the test instances on which the heuristics were tested. In Chapters 3 and 4, the heuristics are treated. While Chapter 3 deals with heuristics specifically tailored for branch-and-price, Chapter 4 focuses on standard heuristics of the SCIP framework which have been modified for the use with GCG. For each heuristic, we give the general mathematical idea behind it, discuss implementational details and analyze the individual impact it has on the performance of GCG. Chapter 5 is dedicated to some general results; we will make a general comparison between the heuristics from the chapters before and discuss their impact on further problem instances. Chapter 6 concludes this thesis. We will also give an outlook on possibilites for the further development of GCG and primal heuristics. Finally, there are two appendices, one giving an overview of the notation used in this thesis and the other containing the tables with the performed computations.

1.1 Mixed Integer Programming

Let us first give some basic definitions.

Definition 1.1. *Let $m, n \in \mathbb{N}$, $I \subseteq \{1, \ldots, n\}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. The problem*

$$\begin{array}{lrcll} \min & c^T x & & & \\ s.\ t. & Ax & \geq & b & \\ & x & \in & \mathbb{R}^n & \\ & x_i & \in & \mathbb{Z} & \textit{for all } i \in I \end{array} \tag{1.1}$$

is called a Mixed Integer (Linear) Program*, or shortly* MIP.

If $I = \{1, \ldots, n\}$, we speak of an Integer Linear Program (ILP)*, and in the case $I = \emptyset$ of a* Linear Program (LP).

Definition 1.2. *We call $x \mapsto c^T x$ the* objective function*. The constraints $Ax \geq b$ are called* linear constraints*, while the constraints $x_i \in \mathbb{Z}$, $i \in I$, are called* integrality constraints.

A special case of an ILP is a Binary Program:

Definition 1.3. *An ILP subject to $x \in \{0, 1\}^n$ is called a* Binary Program (BP).

Definition 1.4. *Let a MIP as stated in Definition 1.1 be given and $\bar{x} \in \mathbb{R}^n$. Then we call $\bar{x}$*

- LP feasible *if it satisfies the linear constraints,*
- integer feasible *if it satisfies the integrality constraints,*
- feasible *if it is LP feasible and integer feasible.*

Algorithms for MIPs often make use of the LP relaxation.

Definition 1.5. *The problem (1.1) without the integrality constraints is called the* LP relaxation *of (1.1).*

A feasible solution $\bar{x}$ obviously yields an *upper bound* of the MIP. On the other hand, if $\tilde{x}$ is the optimum of the LP relaxation and $\hat{x}$ the optimum of the MIP, we have $c^T\tilde{x} < c^T\hat{x}$ since the set of LP feasible solutions is a superset of the set of feasible solutions; hence, the solution of the LP relaxation yields a *lower bound*.

One of the most important techniques to solve MIPs is *branch-and-bound* which is basically a *divide-and-conquer* approach; the core idea is to divide the MIP into subproblems which are easier to solve. The procedure can be illustrated as a tree, the so-called *branch-and-bound tree*, with the root node being the original MIP. Starting at the root, we compute a lower bound for the current problem, e. g. by solving its LP relaxation; furthermore, an upper bound might be obtained if we find a feasible solution. If the current problem is not solved to optimality, we divide it by creating two child nodes of the current node, representing its subproblems (*branching*). We then repeat the process for another node in the tree, memorizing the best known feasible solution $\bar{x}$, the *incumbent*. If we have computed lower bounds for the childs of a node, their minimum may yield new lower bounds for it and its parent nodes (*bounding*). There are three ways to find out that a subproblem needs not to be explored any further, and its node can be *pruned*:

- the lower bound is greater or equal than the currently best known upper bound (the objective value of the incumbent);
- the subproblem is solved to optimality;

- the subproblem is infeasible.

The algorithm stops until there is no node left to be explored.

By comparing the objective value of the incumbent $\bar{x}$ to the best known lower bound, one can judge its quality:

Definition 1.6 ([Ber06]). *Let $\bar{x}$ be the incumbent and $\underline{z}$ be the currently best lower bound MIP. The* primal-dual gap *of $\bar{x}$ is defined as*

$$\gamma_{PD}(\bar{x}) = \begin{cases} 0 & \text{if } c^T\bar{x} = \underline{z} = 0 \\ \infty & \text{if } c^T\bar{x} > \underline{z} = 0 \vee c^T\bar{x} \cdot \underline{z} < 0 \\ 100 \cdot \frac{c^T\bar{x} - \underline{z}}{|\underline{z}|} & \text{else} \end{cases}.$$

A low gap indicates that $c^T\bar{x}$ is near to the optimum and possibly even the optimum itself, and if the gap is zero, optimality of $\bar{x}$ is proved.

1.2 The Dantzig-Wolfe Decomposition

Some MIPs have a special structure which can be exploited. This is where the *Dantzig-Wolfe Decomposition* [DW60] comes into play. Originally developed to solve LPs, it can also be used to effectively solve MIPs. It decomposes the MIP into two problems, a *master problem* and a *subproblem*; the solution of the MIP then works iteratively on these two problems.

In our presentation, we basically follow the notation from [Gam10].

First, we need to understand the geometrical meaning of the linear constraints: They define a polyhedron. In fact, every polyhedron can be defined by linear inequalities.

Definition 1.7. *Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. The set*

$$P(A,b) = \{x \in \mathbb{R}^n : Ax \geq b\}$$

is called a polyhedron. *A bounded polyhedron is called a* polytope.

The representation in Definition 1.7 is also called the *outer representation* of a polyhedron. Our goal is now to find a reformulation of a given MIP. From now on, we write MIPs in the form

$$\begin{array}{lrcl} \min & c^T x & & \\ \text{s. t.} & Ax & \geq & b \\ & x & \in & X, \end{array} \tag{1.2}$$

where

$$X = \{x \in \mathbb{R}^n : Dx \geq d, x_i \in \mathbb{Z} \text{ for all } i \in I\},$$

$D \in \mathbb{R}^{l\times n}$, $d \in \mathbb{R}^l$, $l \in \mathbb{N}$. Typically, $Ax \geq b$ are some “complicated” constraints, whereas the set X often bears some particular structure. For example, imagine that the constraints that describe X are *set packing/partitioning/covering* or *knapsack* constraints, and that optimizing over X can happen through special combinatorial algorithms.

In particular, the set X often has a (bordered) block diagonal structure, i. e.

$$X = X_1 \times \cdots \times X_K$$

with $X_k = \{x^k \in \mathbb{R}^{n_k} : D^k x^k \geq d^k, x_i^k \in \mathbb{Z} \text{ for all } i \in I_k\}$, $k = 1, \ldots, K$, $I_k \subseteq \{1, \ldots n_k\}$, $\sum_{k=1}^K n_k = n$.

The idea of the Dantzig-Wolfe decomposition is to split MIP into two new problems, the above mentioned master problem and subproblem. While the master problem uses another representation of X, the subproblem has $x \in X$ as its only constraints and can therefore be solved more easily than the original MIP. There are two basic variants of the Dantzig-Wolfe decomposition: *convexification* and *discretization*.

1.2.1 Convexification

The convexification approach uses the fact that there is another way to define a polyhedron, the *inner representation*:

Theorem 1.8 (Minkowski-Weyl)**.** *A set $X \subseteq \mathbb{R}^n$ is a polyhedron if and only if there exist finite sets $\{x^p\}_{p \in P}, \{x^r\}_{r \in R} \subset \mathbb{R}^n$ such that*

$$X = \text{conv}(\{x^p\}_{p \in P}) + \text{cone}(\{x^r\}_{r \in R}), \tag{1.3}$$

where P and R are some index sets.

Proof. See [NW88]. □

Remark 1.9. *A polyhedron is a polytope if and only if $R = \emptyset$ in its inner representation.*

Definition 1.10. *Let a polyhedron with a representation as in (1.3) be given. We will call $\{x^p\}_{p \in P}$ the* extreme points *and $\{x^r\}_{r \in R}$* the extreme rays *of the polyhedron; furthermore, the extreme points and rays together are called the* generators *and denoted by $\{x^g\}_{g \in G}$, $G = P \cup R$.*

Since $\text{conv}(X_k)$ is a polyhedron for every $k \in 1, \ldots, K$, it follows from Theorem 1.8 that every $x^k \in \text{conv}(X_k)$ – and in particular, every $x^k \in X_k$ – can be written in the form

$$x^k = \sum_{p \in P_k} \lambda_p x^{kp} + \sum_{r \in R_k} \lambda_r x^{kr}, \quad \sum_{p \in P_k} \lambda_p = 1, \quad \lambda_k \in \mathbb{R}_+^{|G_k|}.$$

Replacing the components $x^1, \ldots, x^k$ of x by this representation yields a new formulation for MIP (1.2) which we call the *master problem*:

$$\begin{aligned}
\min \quad & \sum_{k=1}^K \sum_{p \in P_k} c_p \lambda_{kp} + \sum_{k=1}^K \sum_{r \in R_k} c_r \lambda_{kr} \\
\text{s. t.} \quad & \sum_{k=1}^K \sum_{p \in P_k} a^p \lambda_{kp} + \sum_{k=1}^K \sum_{r \in R_k} a^r \lambda_{kr} \geq b \\
& \sum_{p \in P_k} \lambda_{kp} = 1 && \text{for all } k \\
& \lambda_k \in \mathbb{R}_+^{|G_k|} && \text{for all } k \\
& \sum_{p \in P_k} \lambda_{kp} x^{kp} + \sum_{r \in R_k} \lambda_{kr} x^{kr} = x^k && \text{for all } k \\
& x_i^k \in \mathbb{Z} && \text{for all } i \in I_k.
\end{aligned} \tag{1.4}$$

We will call the λ variables *master variables* and the x variables from MIP (1.2) *original variables*. We will also refer to MIP (1.2) as the *original MIP*. The number of master variables may be much higher than the number of original variables; the LP relaxation of the master problem is therefore solved by *column generation* [DL05]. The core of this algorithm is that we regard a *restricted master problem* which contains only a subset of the master variables, at the beginning none. For each block $k \in 1, \dots, K$, we can then solve a *pricing problem* (subproblem)

$$\begin{aligned} \min \quad & (c^k)^T x^k - (\pi^k)^T A^k x^k - \pi_0 \\ \text{s. t.} \quad & x^k \in X_k, \end{aligned} \tag{1.5}$$

in order to decide whether an optimum for the restricted master problem is also optimal for the master problem, too; if this is not the case, the pricing problems yield new master variables to be added to the master problem. As mentioned above, the pricing problems are often easier to solve than the original problem. Besides, the fact that in case of a block diagonal matrix D, we have several small pricing problems instead of one large can lead to an acceleration of the solution time.

Remember that in the branch-and-bound algorithm, we used the LP relaxation to compute lower bounds. The LP relaxation of the Dantzig-Wolfe decomposition also yields a lower bound on the original MIP, and a potentially better one. This is what *branch-and-price*, a variant of branch-and-bound, makes use of: it computes lower bounds by solving the master LP relaxation, and it uses column generation to do so [BJN+98].

1.2.2 Discretization

In the case of an ILP (i. e. $I = \{1, \dots, n\}$), we may also use the discretization approach which is quite similar to convexification. As proved in [NW88], we can write every $x^k \in X_k$ as

$$x^k = \sum_{p \in P_k} \lambda_{kp} x^{kp} + \sum_{r \in R_k} \lambda_{kr} x^{kr}, \quad \sum_{p \in P_k} \lambda_{kp} = 1, \quad \lambda_k \in \mathbb{Z}_+^{|G_k|},$$

where P_k and R_k are usually not the same sets as in convexification. The master problem then reads

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} \sum_{p \in P_k} c_p \lambda_{kp} + \sum_{k=1}^{K} \sum_{r \in R_k} c_r \lambda_{kr} \\ \text{s. t.} \quad & \sum_{k=1}^{K} \sum_{p \in P_k} a^p \lambda_{kp} + \sum_{k=1}^{K} \sum_{r \in R_k} a^r \lambda_{kr} \geq b \\ & \sum_{p \in P_k} \lambda_{kp} = 1 \qquad \text{for all } k \\ & \lambda_k \in \mathbb{Z}_+^{|G_k|} \qquad \text{for all } k, \end{aligned} \tag{1.6}$$

the main difference being that we now have integrality restrictions on the master variables and thus do not need to impose integrality on the original variables anymore. This allows in particular primal heuristics to be run on the master variables and not only on the original ones.

1.2.3 Aggregation of identical blocks

The discretization approach has another advantage over convexification. Suppose that X has a bordered block diagonal structure, and that some (or even all) the X_k are identical. Observe that in discretization, we no longer need the relation between original and master variables to enforce integrality. Hence, we may aggregate identical blocks in the following way: Suppose we have L classes of identical blocks, and that for each class $l \in 1, \ldots, L$, K_l blocks are identical (hence $\sum_{l=1}^{L} K_l = K$). Then we may for each class choose a representative P_{k_l}, sum up the λ_{kp} variables to $\nu_{lp} := \sum_{k=1}^{K_l} \lambda_{kp}$ and add the K_l convexity constraints; the master problem becomes

$$\begin{aligned}
\min \quad & \sum_{l=1}^{L} \sum_{p \in P_{k_l}} c_p \nu_{lp} + \sum_{l=1}^{L} \sum_{r \in R_{k_l}} c_r \nu_{lr} \\
\text{s. t.} \quad & \sum_{l=1}^{L} \sum_{p \in P_{k_l}} a^p \nu_{lp} + \sum_{l=1}^{L} \sum_{r \in R_{k_l}} a^r \nu_{lr} \geq b \\
& \sum_{p \in P_{k_l}} \nu_{lp} = K_l \qquad \text{for all } l \\
& \nu_l \in \mathbb{Z}_+^{|G_{k_l}|} \qquad \text{for all } l.
\end{aligned} \tag{1.7}$$

Obviously, this leads to a reduction of the master variables and hence to a smaller master problem. Also, for each class l of identical pricing problems, we need to solve only one instead of K_l pricing problems. This is nice since it further reduces computational effort.

If we refer to the master problem in further chapters, we will always call the master variables λ no matter whether the master variables are aggregated or not, since this will simplify presentation.

1.3 Primal Heuristics

When we introduced branch-and-bound in Section 1.1, we mentioned that feasible solutions yield upper bounds for MIPs. These solutions can be obtained if the solution of an LP relaxation is integer feasible. Obviously, this is not always the case; it may very well happen that the relaxations do not yield feasible solutions at all. Since, however, they are needed to solve a MIP to optimality, one often needs *primal heuristics*. These are algorithms devised to search for feasible solutions.

Not only are heuristics sometimes necessary to solve a MIP; they may also lower the computational effort drastically. As we mentioned earlier, a node can be pruned if its lower bound is greater or equal than the best known upper bound. Now imagine that a primal heuristic finds a feasible, near-to-optimal solution early in the solving process. It is then likely that nodes are pruned which otherwise needed to be explored, and one saves the time it would need to solve subproblems.

It is therefore desirable to use heuristics during a branch-and-bound or a branch-and-price process. One should, on the other hand, also keep in mind that they need computational time themselves; if they are not successful, they might even increase the overall olving time. How often a heuristic is called during a solving process should therefore depend upon

- how much computational time it needs, and
- how often it finds feasible solutions when it is called.

Often, primal heuristics make use of an LP feasible solution $\tilde{x}$. Some heuristics try to round this solution (*Rounding Heuristics*) or search the Branch-and-Bound tree at deeper nodes (*Diving Heuristics*). Other heuristics try to solve the MIP with additional constraints – which we call *sub-MIP* – in the hope that this sub-MIP might be easier to solve. It is not guaranteed at all that this is the case; besides, sub-MIPs may be infeasible.

In a branch-and-price context, there is additional information available which heuristics may utilize; in particular, there are extreme points and rays available which heuristics can use to construct feasible solutions. Moreover, one has the master formulation available on which one can search for feasible solutions. The Dantzig-Wolfe decomposition therefore allows for special heuristics which might be superior to heuristics used in a standard branch-and-bound algorithm; these heuristics are discussed in Chapter 3. Apart from that, standard heuristics are also useful in branch-and-price algorithms as we will see in Chapter 4.

When judging the quality of a solution that a heuristic has found, we can compare it to the incumbent in the following way:

Definition 1.11 ([Ber06]). *Let $\bar{x}$ be a solution found by a heuristic and $\hat{x}$ be the best known feasible solution.The* primal gap *of $\bar{x}$ is then defined as*

$$\gamma_P(\bar{x}) = \begin{cases} 0 & \text{if } c^T\bar{x} = c^T\hat{x} = 0 \\ \infty & \text{if } c^T\bar{x} > c^T\hat{x} = 0 \\ 100 \cdot \frac{c^T\bar{x} - c^T\hat{x}}{|c^T\hat{x}|} & \text{else} \end{cases}.$$

2 The Computational Environment

In this chapter, we will shortly describe the software used and the problems on which we ran our primal heuristics. All heuristics were implemented into the branch-and-price solver GCG which itself is an extension of the MIP solving framework SCIP. The software is written in C and was compiled with `gcc 4.4.3` under `Ubuntu 10.04`.

The computations were performed on a 3 Ghz AMD Phenom II X4 945 QuadCore CPU with 512 KB cache and 8 GB RAM. For each computation, we list the overall time and number of nodes the solver needed to reach the optimum or the time limit, as well as the primal bound and the gap at this time; the time limit was set to half an hour, with the exceptions being the RAP and MIPLIB instances where we set the limit to one hour. Since our interest lies in finding feasible solutions, we also list the time needed for finding the first feasible solution as well as its objective function value and its gap at the time it was found. The obtained values are summarized by using the *(shifted) geometric mean*; for data $a_1, \ldots, a_n \in \mathbb{R}_+$ and a *shift* $s \in \mathbb{R}_+$, it is defined as

$$\mu = \sqrt[n]{\prod_{i=1}^{n} \max(a_i + s, 1)} - s.$$

Results for the individual heuristics can be found in Chapters 3 and 4 where these heuristics are discussed, while general results are discussed in Chapter 5.

2.1 SCIP

SCIP – ***S**olving **C**onstraint **I**nteger **P**rograms* – is a non-commercial Constraint Integer Programming solving framework developed by Achterberg et al. [ABH$^+$, Ach07, Ach09] at the *Zuse Institute Berlin*; in particular, it serves as a MIP solver. As other solvers such as CPLEX [CPL09] and Gurobi [Gur10], SCIP uses the branch-and-bound technique to solve MIPs.

Its main characteristic is its plugin-based architecture; thus, the whole MIP-solving process happens through plugins. These are e. g. *constraint handlers* to manage particular constraints such as knapsack constraints, or *cut separators* to compute cutting planes; other plugins like *branching rules* and *node selectors* help to manage the branch-and-bound tree. Furthermore, SCIP comes with a bunch of *primal heuristics*.

This architecture enables the user to easily implement his own plugins into SCIP. This way, it can be used as a framework for branch-and-price. For this purpose, the implementation of own *relaxators* and *variable pricers* is also supported.

Currently, SCIP is one of the fastest non-commercial MIP solvers (see [Mit]).

2.2 GCG

In 2010, Gamrath [Gam10, GL10] wrote an extension to SCIP called GCG (***G**eneric **C**olumn **G**eneration*); it is basically an implementation of the Dantzig-Wolfe decomposition and column generation and thus extends SCIP to a branch-and-price *solver*. While branch-and-price solvers are typically problem specific, the goal of GCG is to provide a generic solver which automatically recognizes and exploits special problem structures. Currently, this is not implemented yet; as additional input, GCG needs information about the block structure of the problem in order to be able to perform a decomposition.

Technically, GCG works with two problem instances: the *original instance* which represents the original MIP and the *extended instance* for the master problem. The latter is controlled by a relaxator plugin which replaces the default SCIP LP relaxator. In its initialization, the relaxator creates the master problem which initially contains no variables; they are added by a variable pricer plugin which control the pricing problems. Besides, special branching rules and node selectors make sure that the branch-and-bound trees in the original and the extended instance are built in the same way.

Before this thesis, no primal heuristics were implemented into GCG; only the default SCIP heuristics were used, and only on the extended instance. In order to run them on the original instance as well, they need to by adjusted such that they obtain an LP feasible solution $\tilde{x}$ via the GCG relaxator as they normally make use of the standard LP relaxation. The relaxator is able to translate an LP feasible master solution $\tilde{\lambda}$ into the original variables. Heuristics which are implemented as GCG plugins can access both the original and the extended instance and therefore have all information available the Dantzig-Wolfe decomposition and column generation yield about the problem.

2.3 The Test Instances

We tested the primal heuristics on four different classes of MIPs: Bin Packing, Vertex Coloring, Capacitated p-Median and Resource Allocation problems. For each class of problems, we will only state the MIP formulation as well as the master problem. The models presented here are taken from [Gam10]; we also use the same test instances.

Moreover, we considered nine decomposable instances taken from MIPLIB [AKM06].

2.3.1 Bin Packing

Given n bins with capacity C and a set of items I with nonnegative weights w_i, $i \in I$, assign each item to a bin such that the capacity of each bin is not exceeded and that we need as few bins as possible. The MIP is then formulated as

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} y_j \\
\text{s. t.} \quad & \sum_{j=1}^{n} x_{i,j} \geq 1 && \text{for all } i \in I \\
& \sum_{i \in I} w_i x_{i,j} \leq C y_j && \text{for all } 1 \leq j \leq n \\
& x_{i,j} \in \{0,1\} && \text{for all } i \in I,\ 1 \leq j \leq n \\
& y_j \in \{0,1\} && \text{for all } 1 \leq j \leq n,
\end{aligned} \tag{2.1}$$

where $x_{i,j} = 1$ if and only if item i is assigned to bin j and $y_j = 1$ if and only if bin j is used.

The corresponding master problem is

$$\begin{aligned} \min \quad & \sum_{S \in \mathscr{S}} \lambda_S & & \\ \text{s. t.} \quad & \sum_{\substack{S \in \mathscr{S} \\ i \in S}} \lambda_S = 1 & & \text{for all } i \in I \\ & \lambda_S \in \{0,1\} & & \text{for all } S \in \mathscr{S}, \end{aligned} \tag{2.2}$$

where $\mathscr{S} = \{S \subseteq I : \sum_{i \in S} w_i \leq C\}$.

The test instances are taken from [SK] and called BINDATA1 throughout this thesis. Since this is quite a large test set with many similar instances, we only considered a smaller subset BINDATA1S.

2.3.2 Vertex Coloring

A similar problem class are vertex coloring problems. We are given an undirected graph $G = (V, E)$ and n colors and want to assign to each vertex $v \in V$ a color j such that no two adjacent vertices are assigned the same color and such that we need a minimum number of colors. The MIP is

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} y_j & & \\ \text{s. t.} \quad & \sum_{j=1}^{n} x_{v,j} \geq 1 & & \text{for all } v \in V \\ & x_{v,j} \leq y_j & & \text{for all } v \in V,\ 1 \leq j \leq n \\ & x_{u,j} + x_{v,j} \leq 1 & & \text{for all } (u,v) \in E,\ 1 \leq j \leq n \\ & x_{v,j} \in \{0,1\} & & \text{for all } v \in V,\ 1 \leq j \leq n \\ & y_j \in \{0,1\} & & \text{for all } 1 \leq j \leq n, \end{aligned} \tag{2.3}$$

where $x_{v,j} = 1$ if and only if color j is assigned to vertex v and $y_j = 1$ if and only if color j is used.

The master problem reads

$$\begin{aligned} \min \quad & \sum_{S \in \mathscr{S}} \lambda_S & & \\ \text{s. t.} \quad & \sum_{\substack{S \in \mathscr{S} \\ v \in S}} \lambda_S = 1 & & \text{for all } v \in V \\ & \lambda_S \in \{0,1\} & & \text{for all } S \in \mathscr{S}, \end{aligned} \tag{2.4}$$

where $\mathscr{S} = \{S \subseteq V : u \notin S \vee v \notin S \text{ for all } (u,v) \in E\}$.

The instances are taken from [JMT] and called COLORING. While some instances can be solved within a short time also without heuristics, there are other, harder instances where GCG does not find any solution at all. Therefore, we sometimes distinguish between the "easy" instances COLORINGEASY and the "hard" instances COLORINGHARD.

2.3.3 Capacitated p-Median

We are again given a graph with a set of nodes N and a subset $M \subseteq N$. At each node $i \in N$, a user is located, and at each node $j \in M$, a facility may be placed. Each user i has a demand $q_i \in \mathbb{Z}_+$ and each facility would have a capacity $Q_j \in \mathbb{Z}_+$ if placed at node j. We want to select p medians, i. e. locations for facilities $j \in M$, such that the sum of distances $d_{i,j} \in \mathbb{Z}_+$ from user $i \in N$ to the nearest facility $j \in M$ is minimal and such that for each facility, the total demand of users nearest to it does not exceed its capacity. The problem is modeled as

$$
\begin{aligned}
\min \quad & \sum_{i \in N} \sum_{j \in M} d_{i,j} x_{i,j} \\
\text{s. t.} \quad & \sum_{j \in M} x_{i,j} = 1 && \text{for all } i \in N \\
& \sum_{j \in M} y_j = p \\
& \sum_{i \in N} q_i x_{i,j} \leq Q_j y_j && \text{for all } j \in M \\
& x_{i,j} \in \{0,1\} && \text{for all } i \in N, j \in M \\
& y_j \in \{0,1\} && \text{for all } j \in M,
\end{aligned}
\tag{2.5}
$$

where $x_{i,j} = 1$ if and only if user i is assigned to facility j and y_j if and only if a facility is placed at node j.

The master problem is

$$
\begin{aligned}
\min \quad & \sum_{\substack{j \in M \\ S \in \mathscr{S}_j}} c_S^j \lambda_S^j \\
\text{s. t.} \quad & \sum_{j \in M} \sum_{\substack{S \in \mathscr{S}_j \\ i \in S}} \lambda_S^j = 1 && \text{for all } i \in N \\
& \sum_{S \in \mathscr{S}_j} \lambda_S^j \leq 1 && \text{for all } j \in M \\
& \sum_{j \in M} \sum_{S \in \mathscr{S}_j} \lambda_S^j = p \\
& \lambda_S^j \in \{0,1\} && \text{for all } S \in \mathscr{S}_j,\ j \in M,
\end{aligned}
\tag{2.6}
$$

where $\mathscr{S}_j = \{S \subseteq N : \sum_{i \in S} q_i \leq Q_j\}$ and $c_S^j = \sum_{i \in S} d_{i,j}$ for $S \in \mathscr{S}_j$.

We call the test set CPMP; in our computations, however, we will restrict ourselves to a subset CPMPS.

2.3.4 Resource Allocation

The fourth class of problems are resource allocation problems. We are given N *periods* and a set I of items with profits p_i and weights w_i, $i \in I$. For each period $1 \leq n \leq N$, we have a set

$I(n) \subseteq I$ of items that are alive in period n. We want to choose items such that the total amount of profit is maximized and such that in each period, the total weight of selected items that are alive does not exceed a capacity C. The model is

$$\begin{aligned} \max \quad & \sum_{i \in I} p_i x_i & & \\ \text{s. t.} \quad & \sum_{i \in I(n)} w_i x_i \leq C & & \text{for all } 1 \leq n \leq N \\ & x_i \in \{0,1\} & & \text{for all } i \in I. \end{aligned} \tag{2.7}$$

Although the problem does not have a bordered block diagonal structure, it is possible to transform it into such a structure and apply the Dantzig-Wolfe decomposition to it. This is done by splitting the set of periods into G groups. With $N(g)$ the periods of group g, $G(i)$ the set of groups in which item i is alive, $I(g)$ the set of items that are alive in at least one period of group g and $g(i)$ the first group i which item i is active, the master problem can then be formulated as

$$\begin{aligned} \max \quad & \sum_{i \in I} p_i x_i & & \\ \text{s. t.} \quad & \sum_{\substack{S \in \mathcal{S}_g \\ i \in S}} \lambda_S^g = \sum_{\substack{S \in \mathcal{S}_{g(i)} \\ i \in S}} \lambda_S^{g(i)} & & \text{for all } i \in I,\ g \in G(i) \setminus \{g(i)\} \\ & \sum_{S \in \mathcal{S}_g} \lambda_S^g = 1 & & \text{for all } 1 \leq g \leq G \\ & \lambda_S^g \in \{0,1\} & & \text{for all } 1 \leq g \leq G,\ S \in \mathcal{S}_g. \end{aligned} \tag{2.8}$$

A description of the problem can be found in [CFM10]; we will call our test set RAP. We sometimes distinguish between RAP32 and RAP64, where 32 and 64 periods, respectively, are grouped together. In our computations, we consider a smaller test set RAPS.

2.3.5 MIPLIB Instances

Apart from those instances, we also ran GCG on nine instances from the MIPLIB (***M**ixed **I**nteger **P**roblem **Lib**rary*) [AKM06]. These instances mostly contain not only binary, but also general integer and continous variables. The instances taken are:

Instance	Bin	Int	Cont
10teams	1800	0	225
aflow30a	421	0	421
misc07	259	0	1
noswot	75	25	28
nsrand-ipx	6620	0	1
p2756	2750	0	0
rout	300	15	241
set1ch	240	0	472
tr12-30	360	0	720

Table 2.1.: MIPLIB instances

3 Heuristics Using Decomposition

In this chapter, we discuss heuristics which make use of additional information the Dantzig-Wolfe decomposition provides us about an LP feasible solution $\tilde{x}$.

Restricted Master Heuristics [JMS+10] restrict the master problem to a promising subset of the columns that are already known; they fix all master variables belonging to the other columns to zero and solve the resulting sub-MIP, in the hope that this will quickly lead to a feasible solution. We implemented two variants of these heuristics: *LP Restricted Master* and *Feasibility Restricted Master*.

Column Selection Heuristics [JMS+10] are a constructive approach to generate feasible solutions which exploit the special structure of the convexity constraints in the master problem; they start with $\bar{x} = 0$ and add columns x_p in the hope to reach feasibility. Here, we regard three variants: *Row Greedy Column Selection*, *Feasibility Greedy Column Selection* and *Relaxation Based Column Selection*. The latter also takes the master LP relaxation solution into account and can also be regarded as a rounding heuristic on the master variables.

The Extreme Points Crossover heuristic uses the fact that every LP feasible solution $\tilde{x}$ obtained from the master problem decomposes into integer feasible extreme points x_p, $p \in P$. It tries to obtain a feasible solution by performing a crossover on a promising selection of these points. It is a heuristic which solves a sub-MIP on the original variables.

In the last section, we look at a modification of the Feasibility Pump [FGL05] implemented in SCIP. It tries to round the variables of an LP feasible solution $\bar{x}$ by their reduced costs instead of their fractionality, in the hope to keep feasibility w. r. t. the master constraints.

3.1 Restricted Master Heuristics

The Idea

As already discussed in Section 1.2, the master problem in the Dantzig-Wolfe decomposition is solved by column generation. That is, the master problem is restricted to only a subset of variables, and during its solving process, variables improving the current solution are added dynamically. As the total number of master variables may be exponentially high compared to the number of original variables, this means that the master problem may soon reach a size where searching for feasible solutions would become very time-comsuming.

This is what *Restricted Master Heuristics* (see also [JMS+10]) try to overcome. Starting from the already known master variables (i. e. generators G), they concentrate only on a subset $G' \subset G$ by again restricting the master problem to the corresponding variables. This means that they regard a problem which is of considerably smaller size than the current master formulation, one for which there hopefully exist integer feasible solutions.

Technically, all master variables λ_g with $g \in G \setminus G'$ are fixed to zero, and we solve a sub-MIP of the current master problem which reads

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K} \sum_{p \in P_k} c_p \lambda_{kp} + \sum_{k=1}^{K} \sum_{r \in R_k} c_r \lambda_{kr} \\
\text{s. t.} \quad & \sum_{k=1}^{K} \sum_{p \in P_k} a^p \lambda_{kp} + \sum_{k=1}^{K} \sum_{r \in R_k} a^r \lambda_{kr} \geq b \\
& \sum_{p \in P_k} \lambda_{kp} = 1 && \text{for all } k \\
& \lambda_k \in \mathbb{Z}_+^{|G_k|} && \text{for all } k \\
& \lambda_{kg} = 0 && \text{for all } k \\
& && \text{and } g \in G_k \setminus G'_k
\end{aligned}
\tag{3.1}
$$

in the case of discretization (analogously for convexification). Note that this sub-MIP is just solved as a static MIP, i. e. no pricing is applied and no new columns are generated. It is the hope that restricting the master problem to the subset $G' = \bigcup_{k=1}^{k} G'_k$ – and thus making the master problem smaller – will lead quickly to an integer feasible solution.

Of course, such a restricted master problem may not always be feasible; it can happen that generators are selected which can not be combined to a feasible solution. One therefore needs a strategy to choose generators such that the restricted master problem is likely to be feasible. We implemented two approaches, *LP Restricted Master* and *Feasibility Restricted Master*.

Algorithm 1: Restricted Master Heuristics

Input: A set G of currently known generators; the master LP solution $\tilde{\lambda}$

1 select $G' \subset G$;
2 create MIP (3.1) by fixing master variables λ_j with $j \in G \setminus G'$ to zero ;
3 solve MIP (3.1) ;

LP Restricted Master

At each node in the branch-and-bound tree, we solve the master LP relaxation and obtain – in case of feasibility – a master LP solution. As this solution already satisfies the master constraints, there may be the chance that its generators may also be combined to a solution which is not only LP feasible but also integer feasible. Hence, LP Restricted Master chooses as generators G'_k all those which are part of the master LP solution, i. e. $g \in G'_k$ if and only if $\tilde{\lambda}_{kg} > 0$.

Feasibility Restricted Master

As we want to avoid infeasibility of the restricted master problem, it is desirable to choose generators that violate the master constraints as little as possible. The Feasibility Restricted Master approach therefore judges the generators x^{kg} by their total violations of the constraints $Ax \geq b$, and chooses those which cause as few infeasibility as possible: for each master variable, we compute the total infeasibility of the corresponding column, which we define as

$$\sum_{i=1}^{m} \max(0, b_i - a_i^g)$$

		LPResM	FeaResM ($\alpha = 0.25$)	FeaResM ($\alpha = 0.50$)
BINDATA1S	Solutions found	44	34	43
	Best solution	16	0	0
	Running time	0:00:04	0:00:10	0:00:38
	Running time (avg.)	1.00	1.03	1.18
COLEASY	Solutions found	23	2	16
	Best solution	3	0	2
	Running time	0:00:06	0:00:01	0:00:06
	Running time (avg.)	1.02	1.00	1.02
COLHARD	Solutions found	10	1	7
	Best solution	7	1	5
	Running time	0:00:40	0:00:37	0:01:07
	Running time (avg.)	1.48	1.24	1.87
CPMPS	Solutions found	0	0	26
	Best solution	0	0	5
	Running time	0:00:10	0:10:22	0:30:17
	Running time (avg.)	1.00	3.68	9.51
RAPS	Solutions found	39	37	39
	Best solution	7	0	0
	Running time	0:00:03	0:00:04	0:00:06
	Running time (avg.)	1.00	1.00	1.00

Table 3.1.: Summarized Results for Restricted Master Heuristics

with $g \in G'$. Then, we restrict the master problem to the master variables with the least infeasibilities. Naturally, it is here the question how many variables to choose. For Feasibility Restricted Master, we therefore introduced a *restriction rate* parameter $\alpha \in (0, 1)$ which indicates the percentage of master variables to be regarded in the restricted master problem. For the sake of feasibility, α should not be too small, but it should not be too large, either, in order to keep the computational effort little: the nearer α is to 1, the less would the size of the master problem be decreased – which actually was the goal of the Restricted Master Heuristics.

Computational Results

We ran GCG with the Restricted Master Heuristics on our four classes of test sets, once only with LP Restricted Master and once only with Feasibility Restricted Master. For the latter, we compared two different choices of α: $\alpha = 0.25$ and $\alpha = 0.5$.

It turned out that these heuristics were quite effective on each set of problems, except for the capacitated p-median instances.

LP Restricted Master found solutions on almost every bin packing and easy coloring instance, and the solutions for the bin packing instances usually were of good quality: in 16 out of the 44 instances where it was successful, it also found the optimum.

Additionally, LP Restricted Master also had success on the hard coloring instances: It found a solution in three instances (fpsol2.i.2, fpsol2.i.3 and queen9_9) where the GCG master heuristic were successless, and in four other instances (1-FullIns_4, fpsol2.i.1, queen8_8 and queen10_10), the solutions found were better.

On the RAPS instances, it was only three times unable to find a solution, while on the other hand, in seven instances the solutions found were the best known during the solving process.

Feasibility Restricted Master turned out not to be as good as LP Restricted Master. It was less successful on the bin packing and resource allocation and significantly less successful on the coloring instances; only rarely were the found solutions on the BINDATA1S and COLORING test sets better than with LP Restricted Master, and especially for $\alpha = 0.25$, it found less solutions. $\alpha = 0.5$ turned out to be the better choice; in the CpmpS instances, this was in fact the only variant of Restricted Master that found solutions. This was, however, at the cost of a much higher running time: the heuristic ran in total half an hour on the instances.

In general, LP Restricted Master is superior to Feasibility Restricted Master, and it is additionally the fastest variant (except for ColoringEasy, where it needed yet only slightly more time than Feasibility Restricted Master with $\alpha = 0.25$).

3.2 Column Selection Heuristics

Column Selection Heuristics [JMS+10] are a constructive approach to obtain feasible master solutions. Out of the already known generators, they try to build a feasible solution from scratch.

The Idea

Consider the master problem (1.6) in which discretization was used to reformulate the original MIP. Here, the integrality constraints are enforced on the master variables rather than on the original variables. This means that any feasible solution $\bar{x}$ corresponds to an integer solution $\bar{\lambda}$ which satisfies the convexity constraints

$$\sum_{p \in P_k} \lambda_{kp} = 1 \text{ for all } k = 1, \dots, K$$

as well as the master constraints. The convexity constraints together with the integrality constraints in fact mean that $\lambda_{kp} \in \{0, 1\}$ for every $k = 1, \dots, K$, $p \in P_k$, and it is thus very easy to construct an integer solution satisfying these constraints: For every k, choose some $p \in P_k$ and set $\lambda_{kp} = 1$. Thinking in terms of the original variables, this means that one starts with $\bar{x} = 0$ and adds for every block k a generator x^{kp} to the solution. As in the paper [JMS+10], we call this method *Column Selection* since one adds generators – corresponding to columns in the master problem – to a partial solution in order to satisfy the convexity constraints.

Up to now, we ignored the fact that one may also have extreme rays in the master formulation. Of course, they may also be added to a partial solution. But since their choice is not limited by any convexity constraints, they may be selected infinitely often. In order to make sure that the algorithm terminates, one needs to introduce an additional abortion criterion.

Naturally, a feasible solution must also satisfy the master constraints, so one needs also take care of them when constructing a solution. This is achieved by *selection strategies*: [JMS+10] suggests to select the generators by their "pseudo cost" which may be the ratio of reduced cost per satisfied master constraints. We developed two similar strategies, *Row Greedy Column Selection* and *Feasibility Greedy Column Selection*, that work similarly, but focus on the master constraints only. Additionally, we implemented another suggestion from the above mentioned paper, *Relaxation Based Column Selection*, which does not start from scratch, but from a rounded down master LP relaxation solution.

Row Greedy Column Selection

If a constructed solution should satisfy the master constraints, this means that no constraint must be violated. A greedy strategy would be to judge each generator by how adding it would lower the number of currently violated master constraints, and add it to the working solution;

Algorithm 2: Outline of Column Selection Heuristics

Input: A starting point $\bar{\lambda} \in \mathbb{Z}_+^{|G|}$ and a corresponding original solution $\bar{x}$

repeat
    select a variable λ_p ;
    increase λ_p by 1 in $\bar{\lambda}$;
    update $\bar{x}$ accordingly ;
until *convexity constraints or an abortion criterion are satisfied* ;

this is what Row Greedy Column Selection does. It starts with $\bar{\lambda} = 0$ and selects master variables such that as many row violations as possible are eliminated. In the ideal case, it is at any time be possible to eliminate constraint violations, and in the end, no violated constraints remain so that the resulting solution is feasible.

For extreme rays, this gives rise to an easy abortion criterion: stop if no choice of a master variable would eliminate constraint violations.

Feasibility Greedy Column Selection

Quite similarly works Feasibility Greedy Column Selection which considers the total infeasibility instead of the number of row violations. The total row infeasibility of $\bar{\lambda}$ is

$$\sum_{i=1}^{m} \max(0, b_i - \sum_{k=1}^{K} \sum_{g \in G_k} a_k^g \bar{\lambda}_{kg}),$$

see also the Feasibility Restricted Master Heuristic from Section 3.1. We choose a master variable which reduces as most of this infeasibility as possible. In contrast to Row Greedy Column Selection, this variant also takes the amount of row violations into account and therefore allows for a finer distinction between possible choices of master variables.

The abortion criterion would also be similar to that above: abort if it is not possible to reduce the total row infeasibility.

Relaxation Based Column Selection

Apart from constructing a solution from scratch, one can also base column selection heuristics upon the LP relaxation solution $\tilde{\lambda}$. As $\tilde{\lambda}$ satisfies all linear constraints and in particular the master constraints, it is the hope that some integer feasible points are near it. We therefore use a column selection strategy that is actually a rounding strategy on $\tilde{\lambda}$, namely Relaxation Based Column Selection. As a starting point, we choose the rounded down master LP solution, $\bar{\lambda} = \lfloor \tilde{\lambda} \rfloor$ and keep a list $\Lambda = \{\lambda_{kp} : \tilde{\lambda}_{kp} \notin \mathbb{Z}\}$ of master variables that appeared fractionally in the master LP solution. Then, we proceeds as in Row Greedy Column Selection with the same abortion criterion, except that we first consider only variables from Λ for adding to the working solution. Only if for each block k, such an element was chosen, the other master variables variables are taken into account.

Algorithm 3: Relaxation Based Column Selection

Input: The current master LP solution $\tilde{\lambda}$

$\bar{\lambda} \leftarrow \lfloor \tilde{\lambda} \rfloor$;
$\bar{x} \leftarrow$ original solution corresponding to $\bar{\lambda}$;
$\Lambda \leftarrow \{\lambda_{kp} : \tilde{\lambda}_{kp} \notin \mathbb{Z}\}$;
repeat
    select $\lambda_{kp} \in \Lambda$;
    $\Lambda \leftarrow \Lambda \setminus \{\lambda_p\}$;
    increase λ_{kp} by 1 in $\bar{\lambda}$;
    update $\bar{x}$ accordingly ;
until $\Lambda = \emptyset$ **or** *convexity constraints or an abortion criterion are satisfied* ;
repeat
    select a variable λ_{kp} ;
    increase λ_{kp} by 1 in $\bar{\lambda}$;
    update $\bar{x}$ accordingly ;
until *convexity constraints or an abortion criterion are satisfied* ;

Applying the ideas to other Integer Programs

When we implemented these heuristics, we had the Dantzig-Wolfe decomposition and the column generation scenario in mind, but one could of course apply them to any Integer Program

$$\begin{array}{lrcrcl} \min & & & c^T x & & \\ \text{s. t.} & & & Ax & \geq & b \\ & l & \leq & x & \leq & u \\ & & & x & \in & \mathbb{Z}^n \end{array}$$

with $l, u \in \mathbb{Z}^n$. Here, one can start with $x = l$ and increase variables either until the upper bound is reached or no improvement in feasibility is possible. Dobson [Dob82] described such an algorithm; he chooses variables x_i with $i = \text{argmin}_{1 \leq k \leq n}(c_j / \sum_{j=1}^{m} a_{ij})$ and updates the entries a_{ij} and b_i after each step, aborting if he reaches $b = 0$.

Computational Results

We compared the three column selection strategies on the test sets BINDATA1S, COLORINGEASY, COLORINGHARD, CPMPS and RAPS.

It turned out that all strategies were totally successless on the p-median instances; they were not able to find any solution. This unfortunately also held for Relaxation Based Column Selecting on the bin packing and coloring instances. Only on the resource allocation problems the latter was successful, where it in almost any instance found solutions of good quality: in only three out of the 32 instances where it found solutions, the primal gap to the best known solutions was above 5 percent. The other two strategies found more solutions, which were, however, only of poor quality: the gap was always over 90 percent, except for the instance new3_5_64.

Another observation was that there was actually not much difference between the Row Greedy and the Feasibility Greedy strategy in our problem sets; it did not matter whether we chose master variables according to the number of violated rows or the total violation. Only at the resource allocation instances, slight differences could be observed: in new7_9_64, the Row

		Row Greedy	Feas. Greedy	Relaxation Based
BINDATA1S	Solutions found	40	40	6
	Best solution	0	0	6
	Running time	0:00:17	0:00:16	0:00:03
	Running time (avg.)	1.04	1.04	1.00
COLEASY	Solutions found	18	18	0
	Best solution	1	1	0
	Running time	0:00:14	0:00:10	0:00:03
	Running time (avg.)	1.09	1.05	1.02
COLHARD	Solutions found	12	12	0
	Best solution	9	9	0
	Running time	0:00:47	0:00:40	0:00:16
	Running time (avg.)	1.66	1.51	1.14
CPMPS	Solutions found	0	0	0
	Best solution	0	0	0
	Running time	0:05:24	0:04:50	0:00:59
	Running time (avg.)	2.37	2.23	1.21
RAPS	Solutions found	51	50	36
	Best solution	1	1	1
	Running time	0:00:00	0:00:00	0:00:01
	Running time (avg.)	1.00	1.00	1.00

Table 3.2.: Summarized Results for Column Selection Heuristics

Greedy strategy found one solution more. This may well be because of the master constraints themselves.

The solutions yielded by those two Column Selection Heuristics were in general not as good as in LP Restricted Master. But still, they turned out to be of use for the hard coloring instances: they found solutions in ten of them, and in nine cases, this was the best known solution. In particular, they found solutions on DSJC125.9, inithx.i.1 and fpsol2.i.*, and the solutions for fpsol2.i.1 and queen10_10 were better than those found by the GCG master heuristics.

All in all, of should only use one of the greedy strategies for the bin packing, coloring and capacitated p-median instances; for the resource allocation problems, Relaxation Based Column Selection is the better choice.

3.3 Extreme Points Crossover

Extreme Points Crossover is a heuristic which directly uses the fact that every LP feasible solution $\tilde{x}$ in a branch-and-price process is obtained from the master problem, and thus (integer feasible) extreme points and rays are known of which this solution is a convex/conical combination. Currently, the heuristic works with extreme points only, hence the name.

The Idea

Suppose we have an LP feasible solution $\tilde{x}$ which we obtained by solving the LP relaxation of the master problem. In case of a bordered block-diagonal structure, $\tilde{x}$ splits up into $\tilde{x} = \tilde{x}^1 \times \cdots \times \tilde{x}^K$ ($K = 1$ if there is no such structure). Then, in both the convexification and discretization scenario, we have points $\{x^p\}_{p \in P}$ such that for every $k = 1, \dots, K$,

$$\tilde{x}^k = \sum_{p \in P_k} \lambda_{kp} x^p.$$

For each k, the only relevant points $p \in P_k$ are those with $\lambda_{kp} > 0$, i. e. they are the points of which $\tilde{x}^k$ is actually a convex combination; from now on, we will call them *members of decomposition*. We denote the members of decomposition by V_k, $k = 1, \dots, K$, and write $V = V_1 \cup \dots \cup V_K$. Note that all members of decomposition are integer feasible (since they have been generated by the pricer) but possibly violate the master constraints.

If there exist variables which have identical values on several members of decomposition, it is likely that there are more integer solutions where these variables take the same values. Possibly, some of those solutions are even LP feasible, i. e. satisfy the master constraints. It seems hence promising to analyze the set V and to search for such solutions. Therefore, we choose a subset $V' = \{x^{p_1}, \dots, x^{p_\kappa}\} \subseteq V$, $\kappa \in \mathbb{N}$, such that its elements have as many identical variables as possible. Then, we fix all variables which are identical for all elements in V' and solve a sub-MIP. That is, with the set of identical variables being

$$I' = \{1 \leq i \leq n : x_i^{p_k} = x_i^{p_l} \text{ for all } 1 \leq k, l \leq \kappa\},$$

we solve the original MIP with the additional constraint

$$x_i = x_i^{p_1} \text{ for all } i \in I'.$$

This method is called a *crossover* on the points in V'. In fact, the heuristic may be viewed as a variant of the Crossover heuristic treated in [Ber06] and Section 4.3, the main difference being that it is conceptually not an improvement heuristic but a start heuristic.

Choosing members of decomposition and fixing variables

In order to be successful, we need to choose V' such that we are likely to find a new solution which is both feasible and of good quality. As we want to obtain a feasible solution, the points that we choose for crossover should not lie too "far away" from the polyhedron $P(A, b)$. We therefore need to choose points which violate the master constraints as little as possible, and prefer points x^p with the least total row violation

$$\sum_{i=1}^{m} \max(0, (b_i - A_{i.} x^p)).$$

Another question is how many points x^p we actually choose for crossover. The more points we choose, the less variables will get fixed and the bigger the resulting sub-MIP will be. On the other hand, choosing few points and thus potentially fixing more variables may lead to an infeasible sub-MIP. In our tests, we chose $\kappa = 2$.

As indicated above, we are interested in fixing variables with identical values for many members of decomposition. In order to achieve this, the points used for crossover should be "similar" in the sense they should differ only in a certain percentage of their variables. During the development of this heuristic, we often observed that in our test instances, the x^p consist of many zero variables and only a few non-zero variables. Since this means that the zero variables are already likely to be equal for many points, we directed our focus on the non-zero variables.

In order to select similar points, we proceed as follows: We choose a *similarity rate* $\alpha \in (0, 1)$ and select a reference point, say x^p. Then we check for each remaining point x^q in how many percent of the non-zero variables – the variables which are non-zero for either x^p or x^q – the two points differ and add x^q to the selection if this percentage is greater or equal than α. Thus, we have a good chance to fix a number of variables to a non-zero value. Note, however, that in the case $\kappa > 2$, this method does not guarantee that this is really the case; a method such that this condition holds pairwise for all chosen points would be computatioally too expensive.

Furthermore, α should not be chosen too high; otherwise, it may happen that we do not find a selection of proper size. We can overcome this by allowing the heuristic to reduce α if it is not successful in finding a selection. However, we allow this only a certain number of times during the whole solving process since we do not want α to be too low, either. That is to say, a low α means that less non-zero variables are fixed which we actually wanted to avoid. If α cannot be reduced anymore, there remain two possibilities: abort the heuristic or choose points randomly. The exact method is stated in Algorithm 4.

Algorithm 4: Selecting extreme points for crossover

Input: $V \subset \mathbb{R}^n$

Parameters: $\kappa \in \mathbb{N}$ (number of used solutions); $\alpha \in (0,1)$ (rate by which to decide similarity of solutions); maxReds $\in \mathbb{N}$ (maximum number of allowed reductions of α); reds $\in N$ (reductions until now); $T \in \mathscr{P}(V)$ (already used tuples)

Output: $V' \subseteq V$ points for crossover

1 $U \leftarrow V$;
2 **repeat**
3   select a point $x^p \in U$;
4     $V' \leftarrow \{x^p\}$, $U \leftarrow U \setminus \{x^p\}$;
5   **repeat**
6   select a point $x^q \in U$;
7         $N \leftarrow$ Variables $1 \leq i \leq n$ with $x_i^p \neq 0$ or $x_i^q \neq 0$;
8         $M \leftarrow$ Variables $i \in N$ with $x_i^p = x_i^q$;
9   **if** $|N| \neq 0$ **and** $|M|/|N| \geq \alpha$ **then**
10             $V' \leftarrow V' \cup \{x^q\}$, $U \leftarrow U \setminus \{x^q\}$;
11   **end**
12   **until** $|V'| = \kappa$ **or** $U = \emptyset$;
13   **if** $|V'| = \kappa$ **and** $V' \notin T$ **then**
14         $T = T \cup \{V'\}$;
15   **STOP.**
16   **end**
17 **until** $U = \emptyset$;
/* if we are here, then no appropriate V' could be found */
18 **if** reds $<$ maxReds **then**
19   reduce α ;
20   increment reds by 1 ;
21   **go to** Step 1 ;
22 **end**
23 **if** *randomization is allowed* **then**
24   select V' randomly;
25 **end**

As noted above, the members of decomposition typically consisted of many zero variables. This bears the danger of fixing too many variables and obtaining an infeasible sub-MIP, because of which it seems to make sense not to fix all zero variables. Suppose we have already chosen a set of points for crossover. Now, there are variables x_i which are zero for all points in V' but not for the whole set V – in particular, we have that $\bar{x}_i > 0$. These variables are hence more likely to be greater than zero in a feasible solution $\tilde{x}$ than the other zero variables. Therefore, we decided to leave them unfixed with a certain probability. Note that this is a deviation from the original concept of crossover.

There is also the possibility to run the heuristic several times on the same relaxation solution, each time choosing another tuple of extreme points; since this turned out to be of no use and solving a sub-MIP is computationally expensive, we decided to run the heuristic only once on each solution.

The whole heuristic is stated in Algorithm 5.

Algorithm 5: Extreme Points Crossover

Input: A relaxation solution $\tilde{x}$ with members of decomposition $V \subset \mathbb{R}^n$

Parameters: maxRuns (maximum number of runs on $\tilde{x}$; zeroFixProb (probability that a zero variable is fixed)

for $i \leftarrow 1$ **to** maxRuns **do**
  determine $V' \subset V$ according to Algorithm 4 ;
  fix all variables which are nonzero for each point in V' ;
  fix variables which are zero for all points in V' with probability zeroFixProb ;
  solve the resulting sub-MIP ;
  if *a feasible solution was found* **then**
    STOP.
  end
end

Computational Results and Comparison to RENS

We compared the Extreme Points Crossover heuristic to the RENS heuristic from Section 4.1, as these are both heuristics that search for a solution by solving a sub-MIP on the original variables. In particular, both heuristics fix certain variables to integer values; in case of Binary Programs, the only difference between these two heuristics is how the variables to be fixed are chosen and to which values they are fixed.

The first observation is that Extreme Points Crossover spent more computational time than RENS. This is not surprising since the ideas stated in the previous paragraphs need their time. In GCG, the extreme points are stored in the master variables' data, so they need to be fetched from there first. Then, one needs to compute the violations; finally, Algorithm 4 spends some time until the sub-MIP can be solved. This is a reason why we chose to let the parameter maxRuns be 1, as a higher number would have led to a unreasonably high effort.

In the bin packing instances, both heuristics often found solutions, while the solutions found by RENS were of usually better quality than those found by Extreme Points Crossover. In 28 out of the 54 instances, RENS found the best solution which is more than half of the test set. However, these instance were in general so easy to solve that even without any heuristics, the optimal solution could be found within less than then (or in the case of the N3C*W* instances:

		RENS	Extreme Points Crossover
BINDATA1S	Solutions found	28	27
	Best solution	28	6
	Running time	0:02:01	0:53:46
	Running time (avg.)	1.57	16.48
COLEASY	Solutions found	18	18
	Best solution	18	14
	Running time	0:01:09	0:05:29
	Running time (avg.)	1.59	3.86
COLHARD	Solutions found	1	1
	Best solution	1	1
	Running time	0:05:48	0:19:28
	Running time (avg.)	3.47	5.39
CPMPS	Solutions found	7	55
	Best solution	1	6
	Running time	0:00:19	0:11:21
	Running time (avg.)	1.01	4.75
RAPS	Solutions found	39	39
	Best solution	17	18
	Running time	0:00:05	0:01:07
	Running time (avg.)	1.00	1.52

Table 3.3.: Summarized Results for RENS and Extreme Points Crossover

mostly less than 30) seconds. Therefore, one does not need RENS there, and the effort spent by Extreme Points Crossover is far too high.

On the easy coloring instances, RENS outperformed Extreme Points Crossover. Both heuristics often found solutions of good quality, and there were instances where the one heuristic found solutions but the other did not. They could improve the total solution times for 3-FullIns_3, 4-FullIns_3, 5-FullIns_3, and they led to a significant improvement for the homer instance (20.25 and 40.27 seconds vs. 1715.51 seconds without any heuristics). Additionally, RENS could improve the solution times for miles250, miles500, miles750, the mulsol.i.* and the zeroin.i.* instances.

Both heuristics found only one solution on the hard instances: RENS on fpsol2.i.3 and Extreme Points Crossover on le450_25b, where it, however, found a better solution than the rounding and improvement heuristics from Chapter 4.

On the capacitated p-median instances, the picture turns: here, Extreme Points Crossover found by far more solutions than RENS. It found 55 solutions, whereas RENS was only seven times successful.

The RAPS instances are a test set where both heuristics are almost equally successful; there is no instance where the one heuristic finds a solution and the other does not. Both heuristics need only few running time compared to the other instances.

To sum up, both heuristics should not be used on the bin packing instances as they are too easy; however, RENS is a good choice for COLORING, and Extreme Points Crossover has its advantages on CPMP, and both are useful on RAP.

3.4 Feasibility Pump Heuristics

The Idea – The Standard Feasibility Pump

An iterative approach to obtain feasible solutions is the *Feasibility Pump* [FGL05, FS09]. Its basic idea is to construct two series of points, one with integer feasible points that are possibly LP infeasible and one with points that are LP feasible, but have potentially fractional values. The goal of this heuristic is to construct the series in such a fashion that they converge towards each other, and thus hopefully yield a point satisfying all constraints – a feasible solution.

As a starting point, there is already one point available: the LP relaxation solution $\tilde{x}$. By rounding each value to its nearest integer, one obtains the nearest integer solution $\bar{x}$, possibly LP infeasible. What one does next is generate another LP solution, namely again the one which least distance from $\bar{x}$. To do so, one needs to solve the LP relaxation with changed objective function values; instead of minimizing the objective function, one minimizes the L_1 distance to $\bar{x}$. As this expression is not linear, the distance function is replaced by a linearized expression which uses auxiliary variables. The LP to be solved then reads

$$\begin{array}{rrcll}
\min & \multicolumn{4}{l}{\sum\limits_{\substack{i\in I\\ \bar{x}_i=l_i}} (x_i - l_i) + \sum\limits_{\substack{i\in I\\ \bar{x}_i=u_i}} (u_i - x_i) + \sum\limits_{\substack{i\in I\\ l_i<\bar{x}_i<u_i}} (x_i^+ - x_i^-)} \\
\text{s. t.} & Ax & \geq & b & \\
& x & \in & X & \\
& x_i & = & \bar{x}_i + x_i^+ + x_i^- & \text{for all } i \in I \text{ with } l_i < \bar{x}_i < u_i \\
& x_i^+ & \geq & 0 & \text{for all } i \in I \text{ with } l_i < \bar{x}_i < u_i \\
& x_i^- & \geq & 0 & \text{for all } i \in I \text{ with } l_i < \bar{x}_i < u_i.
\end{array} \tag{3.2}$$

with l_i, u_i being the lower and upper bounds of the variables x_i, respectively. Having obtained a new LP solution, one can repeat the whole process again; since one always chooses a point with minimal distance, the two series will hopefully converge towards each other.

To improve the quality of the solutions, Achterberg and Berthold [AB07] suggested to take also the original objective function into account; the new objective function is then a convex combination of the old one and the distance function.

It might happen that the procedure gets caught in a *cycle*, i. e. that one gets a point $\bar{x}$ that has already been visited some steps before. In that case, one can flip the most fractional variables or do a random perturbation; for more details, see [FGL05, Ber06].

This heuristic was implemented into `SCIP` by Timo Berthold. We adapted it to `GCG` and developed a variant which rounds the variables in a slightly different way, using the Lagrangean Relaxation.

Using the Lagrangean Relaxation

If we have solved the master LP relaxation in the Dantzig-Wolfe decomposition, this yields us a solution $\tilde{x}$ which satisfies the master constraints. Unfortunately, rounding it will very likely destroy that. We therefore need to round it such that the master constraints are violated as little as possible.

Wedelin [Wed95] and Caprara, Fischetti and Toth [CFT99] developed heuristics for Binary Programs that make use of the *Lagrangean Relaxation*; it is defined as

$$\begin{array}{ll}
\min & c^T x - \pi^T (Ax - b) \\
\text{s. t.} & x \in X,
\end{array}$$

where $-\pi^T(Ax - b)$ is a *penalty term* that penalizes the violation of master constraints.

Disregarding the constant term $\pi^T b$, the objective coefficients in this minimization problem are the *reduced costs*

$$\bar{c} = c - \pi^T A.$$

We now want to round our LP relaxation solution $\tilde{x}$; the best possible rounding w. r. t. the Lagrange objective is

$$\bar{x}_i = \begin{cases} \lfloor \tilde{x} \rfloor & \text{if } \bar{c}_i > 0 \\ \lceil \tilde{x} \rceil & \text{if } \bar{c}_i < 0 \\ \text{any value in } \{\lfloor \tilde{x} \rfloor, \lceil \tilde{x} \rceil\} & \text{if } \bar{c}_i = 0. \end{cases}$$

This is to ensure that the rounded solution has at the same time a good objective value and at the same time the violation of master constraints is avoided. As dual solution π, we choose the optimal dual solution w. r. t. the master constraints.

The remaining question is what to do with variables that have reduced cost zero. There are two possibilities what to do with them:

- solve a RENS subproblem as in (4.1), where the already rounded variables are fixed;
- round the remaining variables by their fractionality, as done in the standard Feasibility Pump.

One should, however, keep in mind that solving a sub-MIP tends to be expensive, and use the RENS option only if the resulting sub-MIP would be small enough.

We modified the Feasibility Pump heuristic by replacing its rounding strategy by the one just described and called this variant *Lagrange Feasibility Pump*.

Computational Results

The Feasibility Pump and its variant were tested separately on the four test classes and compared against each other. Unfortunately, none of them yielded many solutions. No solution was found on the CPMPS test set, and only in two BINDATA1S instances, the Standard Feasibility Pump yielded a solution.

Both heuristics worked best on the easy coloring and the resource allocation problems. The standard version was 16 and 11 times, respectively, successful, while the Lagrange variant found 5 and 4 solutions, respectively. In the anna instance from COLORINGEASY, the solution time could be decreased from 6.26 (without heuristics) to 1.89 seconds.

It is striking that the Lagrange Feasibility Pump was beaten by the standard version. However, there was on instance where it led to a particular improvement, namely the games120 instance in the COLORINGEASY test set. This instance could be solved within 2.61 seconds (16.96 seconds with the master heuristics).

One should also mention the high running time the heuristics needed, but this was merely due to technical reasons: in order to solve LPs with modified objective functions, SCIP provides a *diving mode*. Unfortunately, this mode did not work in GCG, so the LP (3.2) had to solved via a sub-MIP which increased the computational effort. This is to be changed in the future, so that the heuristics will hopefully be faster.

		Standard Feas. Pump	Lagrange Feas. Pump
BindATA1S	Solutions found	2	0
	Best solution	0	0
	Running time	0:04:37	0:04:47
	Running time (avg.)	2.84	2.91
ColEasy	Solutions found	16	5
	Best solution	3	2
	Running time	0:33:07	0:39:41
	Running time (avg.)	8.88	7.91
ColHard	Solutions found	1	1
	Best solution	0	0
	Running time	1:47:35	2:01:51
	Running time (avg.)	9.92	11.02
CpmpS	Solutions found	0	0
	Best solution	0	0
	Running time	1:23:04	0:09:01
	Running time (avg.)	29.78	4.43
RapS	Solutions found	11	4
	Best solution	1	1
	Running time	0:03:12	0:03:36
	Running time (avg.)	3.92	4.24

Table 3.4.: Summarized Results for Feasibility Pump Heuristics

Ideas for further development

As the results show, the Lagrange Feasibility Pump still needs some development. During the experiments, the reduced costs often were zero, so that the rounding strategy was not too different to that of the standard version. A possibility to overcome this may be an *update strategy* for the dual solution π after each rounding.

Another idea for a Feasibility Pump would be to decompose in a different fashion. The standard version decomposes into integer and linear constraints; the Dantzig-Wolfe decomposition provides new possibilities for that; one could e. g. decompose into master constraints $Ax \geq b$ and the $x \in X$, where the latter can be represented by the convexity constraints. We already did some first experiments with that variant, but had no success until now, so that it is not treated in this thesis.

4 Experiments on the Default SCIP Heuristics

In addition to the heuristics from the last chapter, we also had a look on the heuristics which are by default implemented in SCIP. As noted in Section 2.2, these had previously only been run on the extended instance. Since many of them are successful when run during a SCIP solving process, it seems worth trying them in GCG as well. We therefore made some adjustments such that they work properly with GCG. In particular, those heuristics which use an LP feasible solution – which are most of them – need to obtain this solution from the GCG relaxator rather from the standard LP relaxation.

Almost all of the heuristics presented here as well as their impact on SCIP are intensively discussed in Berthold's thesis [Ber06]. We will therefore keep the mathematical descriptions short.

One distinguishes between *start heuristics* and *improvement heuristics*. In contrast to start heuristics, improvement heuristics also take already known feasible solutions into account and try to construct new, better feasible solutions out of them.

4.1 Rounding Heuristics

The first – and simplest – kind of start heuristics we regard are rounding heuristics. We assume we have an LP feasible solution $\tilde{x}$ available. If we round the variables to integer values, we will clearly obtain an integer feasible solution but might lose LP feasibility. We therefore need to round in such a way that we avoid the violation of the linear constraints. The following methods try to achieve this; some of them do not only round fractional variables, but shift them by some amount and also change the values of integral variables.

The original SCIP implementations for these heuristics which we adapted to GCG were done by Tobias Achterberg (Simple Rounding, Rounding, Shifting), Timo Berthold (RENS) and Gregor Hendel (ZI Rounding).

Simple Rounding

For each variable $\tilde{x}_j$, round it downwards if it is *trivially down-roundable* (i. e. all coefficients in the corresponding column are nonpositive) and upwards if it is *trivially up-roundable* (the coefficients are nonnegative). Not all variables may be trivially roundable – but if so, then this method will yield a feasible solution.

Rounding

Let us first define some notions: We define the number of *down-locks* of a variable $\tilde{x}_j$ as the number of positive coefficients of the corresponding column and analogously, the number of *up-locks* as the number of negative coefficients; the number of *locks* is the minimum of these two numbers.

If the current (partially) rounded solution violates a row, then choose a variable $\tilde{x}_j$ and round it such that the violation is reduced and the number of locks of $\tilde{x}_j$ is minimal. Otherwise, we regard for each $\tilde{x}_j$ the maximum ξ_j of its down-locks and up-locks, take the $\tilde{x}_j$ with maximal ξ_j and round it in the opposite direction of where this maximum is attained.

ZI Rounding

This heuristic was introduced and described in [Wal10] and can be viewed as an extension to Simple Rounding. For each $\tilde{x}_j$, we consider its fractionality

$$\mathrm{ZI}(\tilde{x}_j) = \min(\lceil \tilde{x}_j \rceil - \tilde{x}_j,\ \tilde{x}_j - \lfloor \tilde{x}_j \rfloor);$$

furthermore, we consider the integer infeasibility

$$\mathrm{ZI}(\tilde{x}) = \sum_j \mathrm{ZI}(\tilde{x}_j).$$

With the row slacks being $\tilde{s}_i = \sum_j a_{ij}\tilde{x}_j - b_i$, $\tilde{x}_j$ can be shifted upwards by at most $ub = \min_i\{-\frac{\tilde{s}_i}{a_{ij}} : a_{ij} < 0\}$ and downwards by at most $lb = \min_i\{\frac{\tilde{s}_i}{a_{ij}} : a_{ij} > 0\}$. Since the shifting is also limited by the variable's upper and lower bounds u_j and l_j, respectively, the greatest possible amount for shifting upwards is $UB = \min(u_j, ub)$ and for shifting downwards $LB = \min(l_j, lb)$.

We then try to shift $\tilde{x}_j$ upwards by UB or downwards by LB if one of these shiftings reduces $\mathrm{ZI}(\tilde{x})$, and in the direction in which the greater reduction is achieved. An extension of this rounding method also shifts integer values in order to improve the objective function value.

Shifting

Shifting basically works like Rounding, except that it also changes the values of integral variables. If there are violated rows of which none contain fractional variables, a random row is chosen and a score is calculated for each integral variable which depends upon how often and by what amount the variable has been shifted before; then, the variable with the lowest score is shifted.

RENS

Apart from these methods which just change the values of $\tilde{x}$, there is another heuristic which tries to calculate the best possible rounding w. r. t. the objective function. This heuristic is called RENS (***R****elaxation* ***E****nforced* ***N****eighborhood* ***S****earch*) and solves a sub-MIP, where all variables with integral $\tilde{x}_j$ are fixed to this value and the lower and upper bounds of the variables are changed to $\lfloor \tilde{x}_j \rfloor$ and $\lceil \tilde{x}_j \rceil$, respectively. That is, we solve the sub-MIP

$$\begin{array}{lrcll} \min & & c^T x & & \\ \text{s. t.} & & & & \\ & & Ax \geq b & & \\ & \lfloor \tilde{x}_j \rfloor \leq & x_j \leq & \lceil \tilde{x}_j \rceil & \text{for all } 1 \leq j \leq n \\ & & x \in & \mathbb{R}^n & \\ & & x_i \in & \mathbb{Z} & \text{for all } i \in I. \end{array} \tag{4.1}$$

We have already mentioned this heuristic in Section 3.3 and compared it to Extreme Points Crossover.

		SimR	Rou	ZI	Shi	General
BINDATA1S	Solutions found	0	23	2	71	
	Best solution					25
	Running time					0:00:07
	Running time (avg.)					1.01
COLORING	Solutions found	0	17	0	79	
	Best solution					33
	Running time					0:04:21
	Running time (avg.)					2.21
CPMPS	Solutions found	0	2	0	0	
	Best solution					1
	Running time					0:07:52
	Running time (avg.)					2.98
RAPS	Solutions found	0	119	129	18	
	Best solution					2
	Running time					0:00:04
	Running time (avg.)					1.00

Table 4.1.: Summarized Results for Rounding Heuristics (without RENS)

Computational Results

We will discuss the results for Simple Rounding, Rounding, ZI Rounding and Shifting here; for the results of RENS, we refer to Section 3.3 where it was compared to Extreme Points Crossover.

Not surprisingly, rounding heuristics turned out to be quite fast and effective heuristics. Except for the capacitated p-median instances where only two solutions in total could be found, they often succeeded in all the other instances.

It is striking that Simple Rounding never found a solution in any instance, and that Shifting outperformed Rounding in both the bin packing and coloring instances. This seems not surprising since Shifting is actually an extension to Rounding; but in the resource allocation instances, the methods behave differently. ZI Rounding, on the other problem classes not too successful, outperforms all the other; and Rounding finds only a few solutions less.

The rounding heuristics even led to a small success on the easy bin packing instances: Although the time to reach the optimum was on average and in total almost the same, they reduced the number of solving nodes.

They also were of use on the COLORING test set: in 33 cases, they found the best known solution or the optimum. Note that we obtained solutions on hard coloring instances where the GCG master heuristics failed. In particular, these are 4-FullIns_4, DSJR500.1, le450_25a, le450_25b and qg.order30.

4.2 Diving Heuristics

In Section 3.4, we already got to know a diving heuristic: the Feasibility Pump. It searched for a feasible solution by solving the LP with a modified objective function.

The other approach diving heuristics can take is to change the bounds of variables. This is what the heuristics we present in this chapter do. As these heuristics are quite successful in SCIP – see [Ber06] –, we decided to implement them into GCG as well. For this purpose, we modified the original SCIP implementations done by Tobias Achterberg.

The Idea

Feasible solutions are not only obtained by heuristics, but also by the relaxation; at several nodes in the branch-and-bound tree, the LP relaxation yields a solution which is not only LP feasible but also integer feasible.

During the branch-and-bound process, the search tree is usually kept balanced, i. e. the node selector usually selects higher nodes. Nodes at deeper depths are explored at later times during the solving process, although many branching could lead to an integer feasible LP solution. Therefore, *diving heuristics* try to complement the solver in this way: they iteratively perform branchings and solve LP relaxations, thus exploring deeper nodes in the branch-and-bound tree and potentially finding solutions which otherwise would be found much later in the branch-and-bound process. Finding these solutions earlier of course means that nodes which otherwise would have to be explored by the branch-and-bound solver may be pruned, and thus, diving heuristics can lead to an improvement of the solution time.

Typically, diving heuristics start with an LP feasible solution $\tilde{x}$ and iteratively select a fractional variable, change its bound and solve the LP again. This process stops until the LP gets infeasible, its optimum is worse than the best known solution or it yields an integer feasible solution – i. e. until the diving heuristic reaches a node that would be pruned.

There are several strategies to select a variable which have already been discussed in [Ber06]; we will therefore not go into further detail here. For GCG, we adapted the following seven diving heuristics:

- Coefficient Diving
- Fractional Diving
- Guided Diving
- Integer Diving
- Linesearch Diving
- Pseudocost Diving
- Vectorlength Diving.

Computational Results

We tested the diving heuristics on the four different problem sets and found out that their performances highly depend on the problem structure. This means that there were several heuristics which often produced feasible solutions on certain problem classes while on others, they were unable to find any solutions at all.

It should first be mentioned that all diving heuristic had not very much success on the bin packing and coloring instances; only a few solutions could be found. However, this was different on the capacitated p-median and resource allocation instances. On the CpmpS test set, Integer Diving and Vectorlength Diving were most effective, while on the RapS test set, Coefficient Diving and Pseudocost Diving clearly outperformed the other diving heuristics.

4.3 Improvement Heuristics

The heuristics we have discussed previously are *start heuristics*; they try to find feasible solutions, without the knowledge of other feasible solutions. In contrast to that, *improvement heuristics*

		CoD	Frd	GuiD	IntD	LinD	PsCD	VecD	General
BINDATA1S	Solutions found	0	1	0	0	0	0	0	
	Best solution								0
	Running time								0:01:25
	Running time (avg.)								1.38
COLORING	Solutions found	1	1	0	0	1	1	2	
	Best solution								0
	Running time								0:03:00
	Running time (avg.)								1.33
CPMPS	Solutions found	0	5	2	46	11	1	65	
	Best solution								7
	Running time								2:59:36
	Running time (avg.)								24.87
RAPS	Solutions found	485	0	0	0	0	75	0	
	Best solution								3
	Running time								0:04:56
	Running time (avg.)								2.79

Table 4.2.: Summarized Results for Diving Heuristics

try to improve existing solutions; they search the *neighborhood* of already known solutions for better solutions.

In SCIP, several such heuristics are implemented: *Crossover*, *Mutation*, *Local Branching* [FL03], *RINS* (***R**elaxation **I**nduced **N**eighborhood **S**earch*) [DRP05] and *DINS* (***D**istance **I**nduced **N**eighborhood **S**earch*) [Gho07]. These heuristics usually restrict the feasible set to the neighborhood by adding additional constraints to the MIP; furthermore, they add an *objective cutoff*. Solving the resulting sub-MIP may then possibly yield a solution which is better than the known ones.

The SCIP implementations of these heuristics were done by Timo Berthold (Crossover, Mutation, Local Branching, RINS) and Robert Waniek (DINS); we adjusted them slightly for the use with GCG.

Crossover

In Section 3.3, we introduced the Extreme Points Crossover heuristic which explores the neighborhood of the extreme points of a relaxation solution. This heuristic chose promising points and fixed all variables in which these points took the same value. The idea for this heuristic had its roots in the Crossover heuristic implemented in SCIP. Here, one considers not extreme points but rather known feasible solutions $\bar{x}^1, \dots, \bar{x}^\kappa$; if these solutions have a number of variables with identical values in common, it is likely that there are more, possibly better solutions with these variable values. One therefore considers the set of identical variables

$$\bar{I} = \{i \in I : \bar{x}_i^k = \bar{x}_i^1 \text{ for all } 1 \le k \le \kappa\},$$

solves MIP (1.2) with all variables in this set fixed.

Mutation

A similar heuristic is *Mutation*; the main difference to Crossover is that it works on only one feasible solution $\bar{x}$; it chooses the set $\bar{I} \subseteq I$ randomly, fixes all $i \in \bar{I}$ to $\bar{x}_i$ and solves the resulting sub-MIP.

Local Branching

Originally introduced as a branching strategy, *Local Branching* by Fischetti and Lodi [FL03] can also be used as an improvement heuristic. It searches the *k-neighborhood* of the incumbent solution $\bar{x}$, defined as

$$y \in P(A,b) : \sum_{i \in I} |\bar{x}_i - y_i| \leq k, \; y_i \in \mathbb{Z} \text{ for all } i \in I$$

which consists of all feasible solutions with Manhattan distance less or equal than k from $\bar{x}$; we also call this strategy a *soft fixing* for $\bar{x}$. In order to restrict the original MIP to this neighborhood, one needs to add a linearized expression of the constraint

$$\Delta^I(x, \bar{x}) \leq k$$

to it. This requires additional variables and constraints; how it is done is described in [Ber06] and [Lod03].

RINS

A heuristic similar to Crossover is *RINS*. It was introduced by Danna, Rothberg, and Le Pape [DRP05] and works on the incumbent $\bar{x}$ – which is feasible but has a possibly high objective function value – and the current relaxation solution $\tilde{x}$ – which has a low objective function value but is in general not integer feasible. RINS searches the crossed neighborhood of these two points – called the *Relaxation Induced Neighborhood* of $\bar{x}$ – for solutions which are integer feasible *and* have a low objective function value – or at least one which is lower than that of $\bar{x}$. The RINS sub-MIP is the original sub-MIP with the additional constraint

$$x_i = \bar{x}_i \text{ for all } i \in I \text{ with } \bar{x}_i = \tilde{x}_i.$$

DINS

Like RINS, the *DINS* heuristic described by Ghosh [Gho07] operates on the incumbent $\bar{x}$ and the current relaxation solution $\tilde{x}$. It is again based on the fact that the relaxation solution has a better objective value than the incumbent. Therefore, one expects that an improving MIP solution might be closer to $\tilde{x}$ than the incumbent, and looks for a solution satisfying

$$\sum_{i \in I} |x_i - \tilde{x}_i| \leq \sum_{i \in I} |\bar{x}_i - \tilde{x}_i|.$$

This (nonlinear) constraint is linearly expressed by using several soft fixing, hard fixing and rebounding schemes which we will not describe here; for further details, we refer to [Gho07].

Computational Results

As improvement heuristics need feasible solutions to find further solutions, we ran them together with the rounding heuristics (except RENS) from Section 4.1 on our four test sets.

Local Branching did not find solutions on any test set; it seems to be absolutely useless in a branch-and-price context.

The by far best improvement heuristic was DINS. It was at the same time the only heuristic that found solutions on bin packing, coloring and resource allocation instances. On the easy

		Cross	Mut	LB	RINS	DINS	General
BindData1S	Solutions found	0	0	0	0	6	
	Best solution						2
	Running time						0:00:47
	Running time (avg.)						1.31
ColEasy	Solutions found	0	0	0	1	15	
	Best solution						6
	Running time						0:00:38
	Running time (avg.)						1.50
ColHard	Solutions found	0	0	0	0	5	
	Best solution						2
	Running time						0:00:58
	Running time (avg.)						1.74
CpmpS	Solutions found	4	0	0	16	113	
	Best solution						23
	Running time						0:44:20
	Running time (avg.)						6.59
RapS	Solutions found	0	0	0	0	3	
	Best solution						0
	Running time						0:00:00
	Running time (avg.)						1.00

Table 4.3.: Summarized Results for Improvement Heuristics

coloring instances, it found the optimal solution in six cases; this meant that the homer instance could be solved about 500 seconds faster than with rounding heuristics only. On the other hand, RENS and Extreme Points Crossover still perform better on this instance. DINS could also improve the solutions found by the rounding heuristics on the hard coloring instance le450_25a.

Capacitated p-median problems were the test set where improvement heuristics were most successful. Again, DINS was the best heuristic there, followed by RINS which was also quite successful. Crossover found only 4 solutions. In 23 cases, the solutions found by improvement heuristics were the best known; however, this was at the cost of a high running time.

The only improvement heuristic for which it makes sense to use it in GCG is DINS, and probably also RINS. Crossover, Mutation and Local Branching are, on the other hand, quite useless.

5 General Results

In the previous chapters, we discussed the individual impact each heuristic had on GCG; we analyzed the heuristics w. r. t. their running time, the number solutions they found and the quality of the solutions.

In this chapter, we are going to compare the heuristics against each other; we will lay our focus on the improvement of the performance of GCG.

Furthermore, we ran several of our heuristics together on nine instances of the MIPLIB test set.

5.1 Our Heuristics vs. The Master Heuristics

We ran GCG both without any heuristics and with the master heuristics over all instances; for each type of heuristic, we investigated in how far it beat the master heuristics w. r. t.

- the number of solving nodes needed;
- the total running time;
- the time to find the first feasible solution.

The results are stated in Table 5.1. We will use the following abbreviations: **ResM** for Restricted Master Heuristics, **ColS** for Column Selection Heuristics, **Extr** for Extreme Points Crossover, **RENS** for RENS, **Rou** for Rounding Heuristics, **Div** for Diving Heuristics and **Imp** for Improvement Heuristics. Due to their little success, we will disregard the Feasibility Pump Heuristics here. The table indicates in how many instances each heuristic(s) needed less nodes and less than 80 percent of the time GCG with the master heuristics needed.

As the table shows, the heuristics particularly reduced the number of solving nodes. This especially holds for the bin packing instances where it was never possible to significantly reduce the total running time and only once the time to first. This is actually no surprise since GCG already copes well with these instances even without heuristics.

A similar observation could be made on the resource allocation instances; here, no heuristic could improve the total solution time in more than four instances – which is a pity since there are quite some of these instances where it is not possible to reach the optimum after one hour. On the other hand, the solutions found by GCG already were of good quality – with the master heuristics, the final gap was always below 0.10 percent.

The time to first was mainly reduced on the capacitated p-median instances; the highest effect here was achieved by Extreme Points Crossover and the Diving Heuristics. For all heuristics, these were the instances where they were most successful in reducing the time to first. On the coloring instances, the rounding heuristics also led to an improvement: in 20 cases, GCG was faster in finding a solution with them.

Although the overall solution time was not reduced too often, let us mention three cases where heuristics were particularly successful: RENS and Extreme Points Crossover reduced the time in 20 and 13 coloring instances, respectively, and the Column Selection Heuristics had their advantages on 13 capacitated p-median instances.

		ResM	ColS	Extr	RENS	Rou	Div	Imp
Bin	Nodes	6	6	7	17	13	6	15
	Time	0	0	0	0	0	0	0
	First	0	1	0	0	0	0	0
Col	Nodes	19	16	22	26	21	19	21
	Time	4	4	13	20	5	3	6
	First	9	9	8	0	20	2	19
CpmpS	Nodes	37	27	44	24	23	31	37
	Time	6	13	3	6	7	1	3
	First	24	25	32	24	21	34	12
RapS	Nodes	19	20	19	18	14	24	17
	Time	2	4	2	3	3	1	2
	First	0	2	0	0	1	0	0
Total	Nodes	81	69	73	85	71	80	90
	Time	12	21	18	29	15	5	11
	First	33	37	40	24	42	36	31

Table 5.1.: Improvements w. r. t. needed solving nodes, time and time to first

	GCG without heuristics					GCG with master heuristics				
Instance	**Primal**	**Gap**	**Nodes**	**Time**	**First**	**Primal**	**Gap**	**Nodes**	**Time**	**First**
10teams	924	0.00	143	89.53	89.50	924	0.00	23	7.02	7.02
aflow30a	1191	5.53	2650	3601.98	1184.89	1203	6.26	3532	3600.08	147.99
misc07	2810	0.00	1	80.87	80.86	2810	0.00	1	80.04	80.04
noswot	–	–	147	3600.26	–					
nsrand-ipx	67040	33.44	31615	3600.00	427.19	58240	15.91	41618	3600.11	4.71
p2756	3124	0.00	102	53.22	52.55	3124	0.00	141	68.42	8.03
rout	1077.56	0.00	779	646.67	23.59	1077.56	0.00	803	628.51	11.61
set1ch	–	–	318	3600.81	–	–	–	387	3600.53	–
tr12-30	–	–	5	3600.10	–	–	–	10	3600.13	–
Total			35760	5:14:33				46515	4:13:04	
Geom. Mean			182	810.42				104	251.73	

Table 5.2.: GCG without and with master heuristics on the Miplib instances

5.2 The performance on the MIPLIB instances

After testing our heuristics on the four test sets, we additionally tested them on some instances taken from MIPLIB [AKM06]. We switched on all heuristics previously discussed in this thesis, except for those which did not tend to find many solutions or were slow. Hence, we left out Feasibility Restricted Master, Feasibility Greedy Column Selection, Extreme Points Crossover, both Feasibility Pump heuristics, Crossover, Mutation, Local Branching and RINS.

We have only results for seven out of nine instances available; the runs on aflow40a and set1ch had to be aborted after exceeding the time limit of one hour.

On 10teams, misc07 and tr12-30, the heuristics were unable to find solutions but found solutions on the other instances. They were outperformed on nsrand-ipx, p2756 and rout by the master heuristics but found a solution on noswot after 0.44 seconds, where GCG with master heuristics crashed.

Although applying the heuristics on the MIPLIB instances did not save time, the fact that solutions were found shows that there lies some potential.

Instance	Nodes	Time	First	HTime	RMast	ColSel	Rou	Dive	RENS	DINS	Primal	HGap
	MIPLIB instances and primal heuristics											
10teams	143	107.01	106.98	18.35	0	0	0	0	0	0	–	∞
misc07	1	79.56	79.56	0.00	0	0	0	0	0	0	–	∞
noswot	3259	3600.49	0.44	2.61	1	0	1	1	0	3	-48	0.00
nsrand-ipx	40346	3600.05	125.39	226.33	8	0	0	0	1	0	57760	1.12
p2756	87	41.92	11.50	0.75	1	0	0	0	0	1	3195	2.27
rout	941	826.05	15.92	1.71	0	0	0	0	1	0	1119.23	3.87
tr12-30	7	3601.48	–	1.35	0	0	0	0	0	0	–	∞
Total	44784	3:17:36		0:04:11	10	0	1	1	2	4		
Geom. Mean	195	542.13		4.25								

Table 5.3.: GCG with selected primal heuristics on the MIPLIB instances

6 Summary and Conclusion

We implemented primal heuristics into the branch-and-price solver GCG, which itself is an extension of the Constraint Integer Programming solver SCIP. While most branch-and-price solvers are problem specific, the objective of GCG is to provide a generic branch-and-price solver which is not specifically tailored for any certain problem class.

Until this thesis, GCG, developed by Gerald Gamrath [Gam10], already was quite a successful solver which outperformed SCIP on several problems. However, there still remained the issue of finding feasible solutions. No special heuristics were implemented into GCG, only the standard SCIP heuristics for the master variables were available. The goal of this thesis was to complement GCG in that respect, and to provide it with heuristics that take both the original and the master variables into account.

In Chapter 3, we discussed heuristics specifically tailored for the branch-and-price scenario. These were Restricted Master Heuristics, Column Selection Heuristics, Extreme Points Crossover and the Lagrange Feasibility Pump.

For the Restricted Master Heuristics, we implemented two variants: *LP Restricted Master* and *Feasibility Restricted Master*. Although both versions were able to find solutions, the former – which restricts the master variables to those which are part of the current master LP solution – was the more successful. In fact, it turned out to be quite a fast and effective heuristic.

The same held for Column Selection Heuristics, which we implemented as *Row Greedy Column Selection*, *Feasibility Greedy Column Selection* and *Relaxation Based Column Selection*. The first two heuristics turned out to behave almost equivalently on our test sets; the third was inferior to the other. Nevertheless, they were also of use for GCG.

Extreme Points Crossover considers the extreme points of the current master solution and tries to detect identical valued variables among them; it fixes a number of those variables in the hope that this will lead to an integer solution which is also LP feasible. We compared this heuristic to RENS, which had already been implemented in SCIP and which we adjusted for the use with GCG. Both heuristics found solutions, but as they needed to solve sub-MIPs, they were also very time-consuming.

The Lagrange Feasibility Pump was a modification of another SCIP heuristic, with a modified rounding strategy that takes a Lagrangean Relaxation into account and rounds against reduced costs w. r. t. master constraints. However, this heuristic still needs some development.

In addition to that, we adjusted several of the default SCIP heuristics for GCG, namely rounding heuristics, diving heuristics and improvement heuristics, which we discussed in Chapter 4. These heuristics further helped improve the solution process, although each of them had its weaknesses on certain problem classes.

The heuristics presented in this thesis almost all were able to generate feasible solutions at quite a number of problem instances. Yet, there still lies some potential in them. Extreme Points Crossover needs to be accelerated in order to be of real use; and there are other ideas which have not been realized yet. For example, another Feasibility Pump could be developed that decomposes differently than the standard version. The work on those and further heuristics is going to be continued in the future.

A Notation

A	A matrix in $\mathbb{R}^{m\times n}$
D	A matrix in $\mathbb{R}^{l\times n}$
b	A vector in $\mathbb{R}^m$
d	A vector in $\mathbb{R}^l$
c	A vector in $\mathbb{R}^n$, usually representing the objective function of a MIP
x	A (variable) vector in $\mathbb{R}^n$
l_i	The lower bound of a variable x_i
u_i	The upper bound of a variable x_i
I	The index set of integer variables in $\{1,\dots,n\}$
$\bar{x}$	An integer feasible solution of a MIP
$\tilde{x}$	An LP feasible solution of a MIP
$\hat{x}$	The optimal or best known solution of a MIP
$P(A,b)$	A polyhedron $\{x \in \mathbb{R}^n : Ax \geq b\}$
P	An index set representing the extreme points of a polyhedron
$\{x^p\}_{p\in P}$	The extreme points of a polyhedron
R	An index set representing the extreme rays of a polyhedron
$\{x^r\}_{r\in R}$	The extreme rays of a polyhedron
G	The set of generators of a polyhedron, $G = P \cup R$
$\{x^g\}_{g\in G}$	The generators of a polyhedron
X	The set $\{x \in \mathbb{R}^n : Dx \geq d, x_i \in \mathbb{Z} \text{ for all } i \in I\}$
K	The number of blocks of X
X_k	A block of X
x^k	The components of x corresponding to the block X_k
D^k	The k-th block of the matrix D
d^k	The components of d_k corresponding to the block k
A^k	The columns of A corresponding to the block k
P_k	The extreme points of X_k
R_k	The extreme rays of X_k
G_k	The generators of X_k
a^p	A coefficient of the master problem
c^p	An objective coefficient of the master problem
λ_{kg}	A master variable belonging to block k and generator g
λ	The vector of master variables
$\bar{\lambda}$	An integer feasible master solution
$\tilde{\lambda}$	An LP feasible master solution
L	The number of classed of identical blocks
ν_{lp}	An aggregated master variable
π	A dual solution vector
γ_{PD}	The primal-dual gap
γ_P	The primal gap
μ	The shifted geometric mean

B Tables

In this appendix, we present our detailed computations we performed. For each heuristic, we list

- the time that was needed to find the first feasible solution when the heuristic was switched on;
- the time the heuristic itself spent;
- the number of solutions found by this heuristic;
- the best primal bound obtained by the heuristic;
- the primal gap between that bound and the best known solution.

In some cases, we also list the number of nodes and the total running time of GCG.

The rounding, diving and improvement heuristics are each grouped together; the heuristic time listed is the total time all heuristics of the corresponding kind needed, and we list for each individual heuristic the number of solutions it found. The abbreviations used are:

SRou Simple Rounding

Rou Rounding

Zi ZI Rounding

Shi Shifting

CoD Coefficient Diving

FrD Fractional Diving

GuiD Guided Diving

IntD Integer Diving

LinD Linesearch Diving

PsCD Pseudocost Diving

VecD Vectorlength Diving

Cross Crossover

LocB Local Branching

Mut Mutation

Finally, we list our computations made with GCG only and GCG with master heuristics. Here we list additionally the primal bound reached at termination, and the primal-dual gap at that time.

A red time indicates that the computation was aborted due to a reached time limit; if the best primal bound and its primal gap are green, this means that the primal gap is zero and the heuristic found the best known solution during the solution process.

Note that some computations are not listed here; these are the ones where heuristics barely yielded any solutions.

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
N1C1W1_A	0.60	0.01	1	25	0.00	0.61	0.02	1	34	36.00	0.82	0.02	1	29	16.00
N1C1W1_K	0.45	0.02	1	26	0.00	0.48	0.01	1	34	30.77	0.64	0.02	1	29	11.54
N1C1W2_B	0.66	0.00	1	31	3.33	0.71	0.00	0	–	∞	0.90	0.02	1	36	20.00
N1C1W2_L	0.77	0.00	0	–	∞	0.77	0.00	0	–	∞	1.02	0.00	0	–	∞
N1C1W4_C	0.78	0.00	0	–	∞	0.74	0.00	0	–	∞	1.03	0.00	0	–	∞
N1C1W4_M	0.67	0.01	1	41	0.00	0.69	0.01	0	–	∞	0.94	0.01	0	–	∞
N1C2W1_D	0.47	0.00	0	–	∞	0.46	0.00	0	–	∞	0.60	0.00	0	–	∞
N1C2W1_N	1.07	0.01	1	21	0.00	1.14	0.02	1	29	38.10	1.40	0.02	1	23	9.52
N1C2W2_E	0.58	0.01	1	33	0.00	0.62	0.01	0	–	∞	0.83	0.01	1	39	18.18
N1C2W2_O	0.49	0.00	1	30	3.45	0.53	0.01	0	–	∞	0.66	0.02	1	37	27.59
N1C2W4_F	0.53	0.00	0	–	∞	0.52	0.00	0	–	∞	0.69	0.00	0	–	∞
N1C2W4_P	0.68	0.00	0	–	∞	0.70	0.00	0	–	∞	0.86	0.00	0	–	∞
N1C3W1_G	0.85	0.02	1	16	6.67	0.84	0.02	1	20	33.33	1.12	0.05	1	16	6.67
N1C3W1_Q	1.07	0.04	1	21	5.00	1.04	0.03	1	28	40.00	1.34	0.04	1	23	15.00
N1C3W2_H	1.08	0.01	1	23	0.00	1.10	0.02	1	34	47.83	1.40	0.03	1	25	8.70
N1C3W2_R	1.13	0.07	1	20	5.26	1.12	0.04	1	26	36.84	1.40	0.04	1	20	5.26
N1C3W4_I	0.65	0.02	1	24	4.35	0.67	0.01	0	–	∞	0.96	0.11	1	29	26.09
N1C3W4_S	0.61	0.01	1	23	4.55	0.65	0.01	0	–	∞	0.83	0.03	1	27	22.73
N2C1W1_A	3.74	0.02	1	49	2.08	3.78	0.03	1	65	35.42	5.26	0.44	1	54	12.50
N2C1W1_K	2.81	0.01	1	56	1.82	2.96	0.02	1	75	36.36	3.74	0.04	1	64	16.36
N2C1W2_B	4.16	0.02	1	62	1.64	4.07	0.02	1	88	44.26	5.25	0.04	1	70	14.75
N2C1W2_L	3.31	0.02	1	62	0.00	3.31	0.01	1	92	48.39	4.35	0.12	1	73	17.74
N2C1W4_C	3.14	0.00	0	–	∞	3.20	0.00	0	–	∞	3.99	0.00	0	–	∞
N2C1W4_M	3.42	0.01	1	73	1.39	3.49	0.00	0	–	∞	4.42	0.02	1	83	15.28
N2C2W1_D	7.51	0.02	1	42	0.00	7.83	0.27	1	54	28.57	9.50	0.06	1	45	7.14
N2C2W1_N	5.79	0.02	1	43	0.00	5.83	0.04	1	58	34.88	7.48	0.05	1	47	9.30
N2C2W2_E	2.95	0.02	1	54	0.00	2.95	0.02	0	–	∞	3.80	0.04	1	65	20.37
N2C2W2_O	4.09	0.01	1	50	0.00	4.36	0.02	0	–	∞	5.50	0.11	1	61	22.00
N2C2W4_F	4.94	0.00	0	–	∞	4.98	0.00	0	–	∞	6.55	0.00	0	–	∞
N2C2W4_P	3.86	0.01	1	60	0.00	3.91	0.01	0	–	∞	5.15	0.17	1	70	16.67
N2C3W1_G	3.79	0.37	1	37	12.12	3.95	0.52	1	43	30.30	4.53	0.11	1	34	3.03
N2C3W1_Q	3.59	0.19	1	38	11.76	3.65	0.20	1	48	41.18	4.53	0.15	1	36	5.88
N2C3W2_H	4.18	0.16	1	40	5.26	4.63	0.56	1	56	47.37	6.43	1.17	1	39	2.63
N2C3W2_R	5.00	0.12	1	42	5.00	4.87	0.10	1	56	40.00	6.28	0.18	1	44	10.00
N2C3W4_I	6.83	0.10	1	45	2.27	6.66	0.03	1	66	50.00	8.71	0.18	1	49	11.36
N2C3W4_S	10.64	0.18	1	43	2.38	10.44	0.06	1	58	38.10	13.15	0.26	1	44	4.76
N3C1W1_A	13.22	0.04	1	107	1.90	13.34	0.15	1	144	37.14	20.19	0.78	1	118	12.38
N3C1W1_K	13.70	0.04	1	102	0.00	13.97	0.28	1	137	34.31	19.99	0.40	1	113	10.78
N3C1W2_B	10.75	0.00	0	–	∞	10.68	0.00	0	–	∞	16.28	0.00	0	–	∞
N3C1W2_L	14.28	0.00	0	–	∞	14.29	0.00	0	–	∞	20.85	0.00	0	–	∞
N3C1W4_C	17.12	0.04	1	148	1.37	17.22	0.05	1	191	30.82	24.61	0.20	1	158	8.22
N3C1W4_M	17.35	0.00	0	–	∞	17.38	0.00	0	–	∞	24.47	0.00	0	–	∞

Continue next page

	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
Instance	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**First**	**HTime**	**NSols**	**Primal**	**HGap**
N3C2W1_D	10.46	0.04	1	87	2.35	10.86	0.49	1	121	42.35	18.66	3.54	1	102	20.00
N3C2W1_N	11.45	0.03	1	92	1.10	11.86	0.40	1	124	36.26	17.56	0.50	1	105	15.38
N3C2W2_E	13.77	0.04	1	116	0.00	13.93	0.08	1	165	42.24	20.77	0.44	1	139	19.83
N3C2W2_O	13.29	0.03	1	107	0.00	13.42	0.11	1	158	47.66	20.03	0.37	1	133	24.30
N3C2W4_F	14.68	0.03	1	115	0.00	14.82	0.13	1	163	41.74	22.12	0.50	1	137	19.13
N3C2W4_P	18.72	0.04	1	124	1.64	18.89	0.12	1	159	30.33	27.21	0.54	1	140	14.75
N3C3W1_G	14.28	0.71	1	70	7.69	15.95	2.26	1	81	24.62	30.24	12.37	1	70	7.69
N3C3W1_Q	15.01	0.44	1	77	5.48	14.78	0.16	1	95	30.14	21.23	1.49	1	76	4.11
N3C3W2_H	37.39	0.32	1	85	3.66	38.56	1.20	1	113	37.80	53.41	5.05	1	86	4.88
N3C3W2_R	16.24	0.35	1	82	3.80	18.06	1.97	1	108	36.71	24.73	2.97	1	83	5.06
N3C3W4_I	17.44	0.14	1	94	2.17	17.52	0.20	1	138	50.00	26.10	2.06	1	110	19.57
N3C3W4_S	35.46	0.26	1	87	3.57	36.05	0.58	1	114	35.71	49.12	3.24	1	88	4.76
Total		0:00:04	44				0:00:10	34				0:00:38	43		
Geom. Mean		1.00					1.03					1.18			

Table B.1.: Restricted Master Heuristics on the test set BINDATA1S

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	**HTime**	**NSols**	**Primal**	**HGap**	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**First**	**HTime**	**NSols**	**Primal**	**HGap**
1-FullIns_3	1.31	0.01	1	4	0.00	1.26	0.01	1	6	50.00	1.60	0.02	1	5	25.00
2-FullIns_3	2.40	0.06	1	6	20.00	3.47	0.01	0	–	∞	4.66	0.14	0	–	∞
3-FullIns_3	2.65	0.20	1	8	33.33	7.24	0.01	0	–	∞	9.80	0.19	0	–	∞
4-FullIns_3	4.36	0.25	1	10	42.86	12.63	0.01	0	–	∞	17.23	0.23	0	–	∞
5-FullIns_3	5.89	0.26	1	12	50.00	21.11	0.02	0	–	∞	29.09	0.43	0	–	∞
anna	1.11	0.04	1	14	27.27	6.31	0.01	0	–	∞	1.53	0.02	1	16	45.45
david	0.63	0.02	1	14	27.27	1.37	0.01	0	–	∞	0.86	0.01	1	16	45.45
games120	18.77	0.34	0	–	∞	18.42	0.01	0	–	∞	25.61	0.10	0	–	∞
homer	1709.67	1.69	0	–	∞	1708.44	0.02	0	–	∞	–	0.07	0	–	∞
huck	0.42	0.03	1	13	18.18	1.74	0.01	0	–	∞	2.46	0.01	0	–	∞
jean	0.39	0.02	1	12	20.00	1.38	0.01	0	–	∞	1.95	0.01	0	–	∞
miles1000	18.32	0.00	0	–	∞	18.29	0.00	0	–	∞	24.52	0.00	0	–	∞
miles1500	46.55	0.00	0	–	∞	47.16	0.00	0	–	∞	61.85	0.00	0	–	∞
miles250	13.97	0.27	0	–	∞	13.51	0.01	0	–	∞	18.32	0.18	0	–	∞
miles500	3.18	0.05	1	22	10.00	10.64	0.01	0	–	∞	4.25	0.04	1	26	30.00
miles750	5.63	0.00	0	–	∞	5.68	0.00	0	–	∞	7.68	0.00	0	–	∞
mulsol.i.1	59.77	0.05	1	50	2.04	65.76	0.01	0	–	∞	83.64	0.12	1	69	40.82
mulsol.i.2	61.76	0.25	1	34	9.68	294.78	0.04	0	–	∞	87.22	0.78	1	45	45.16
mulsol.i.3	64.91	0.32	1	34	9.68	345.52	0.16	0	–	∞	90.43	0.11	1	39	25.81
mulsol.i.4	73.03	0.38	1	36	16.13	392.02	0.16	0	–	∞	101.69	0.68	1	38	22.58
mulsol.i.5	72.45	0.33	1	35	12.90	244.31	0.22	0	–	∞	100.92	0.13	1	40	29.03
myciel3	0.09	0.01	1	4	0.00	0.14	0.00	0	–	∞	0.15	0.01	1	4	0.00
myciel4	2.55	0.08	2	5	0.00	2.57	0.67	1	6	20.00	3.16	1.74	1	5	0.00
queen5_5	0.24	0.00	0	–	∞	0.26	0.00	0	–	∞	0.29	0.00	0	–	∞
queen6_6	3.54	0.01	1	8	14.29	9.07	0.00	0	–	∞	4.52	0.01	1	10	42.86
queen7_7	16.70	0.00	0	–	∞	16.54	0.00	0	–	∞	21.25	0.00	0	–	∞
queen8_12	17.48	1.02	1	17	41.67	184.01	0.01	0	–	∞	22.53	0.22	1	18	50.00
zeroin.i.1	56.23	0.06	1	51	4.08	196.11	0.01	0	–	∞	81.12	0.30	1	66	34.69
zeroin.i.2	50.92	0.40	1	35	16.67	373.69	0.02	0	–	∞	70.30	0.69	1	39	30.00
zeroin.i.3	43.56	0.59	1	35	16.67	300.83	0.02	0	–	∞	60.45	0.46	1	41	36.67
Total		0:00:06	23				0:00:01	2				0:00:06	16		
Geom. Mean		1.02					1.00					1.02			

Table B.2.: Restricted Master Heuristics on the test set ColoringEasy

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
1-FullIns_4	169.78	0.14	1	5	0.00	–	0.35	0	–	∞	208.01	1.59	1	7	40.00
2-Insertions_3	6.86	1.07	3	4	0.00	6.93	30.37	1	5	0.00	8.49	18.35	1	4	0.00
4-FullIns_4	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
ash331GPIA	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
DSJC125.9	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
DSJR500.1	–	9.51	0	–	∞	–	4.29	0	–	∞	–	22.66	0	–	∞
fpsol2.i.1	582.71	0.68	1	72	0.00	–	0.03	0	–	∞	705.81	1.25	1	83	0.00
fpsol2.i.2	360.51	2.90	1	44	0.00	–	0.67	0	–	∞	–	3.65	0	–	∞
fpsol2.i.3	356.90	2.42	1	41	0.00	–	0.90	0	–	∞	509.84	7.48	1	45	0.00
inithx.i.1	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
inithx.i.2	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
inithx.i.3	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
le450_25a	–	7.55	0	–	∞	–	0.38	0	–	∞	–	3.93	0	–	∞
le450_25b	–	15.30	0	–	∞	–	0.39	0	–	∞	–	3.00	0	–	∞
qg.order30	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
qg.order40	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
queen8_8	32.19	0.16	1	10	11.11	1068.01	0.01	0	–	∞	41.10	0.22	1	13	30.00
queen9_9	153.34	0.13	1	11	0.00	–	0.14	0	–	∞	192.32	2.17	1	14	0.00
queen10_10	328.73	0.82	1	13	0.00	–	0.18	0	–	∞	407.86	3.27	1	15	0.00
school1	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
school1_nsh	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
wap05a	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
will199GPIA	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
Total		0:00:40	10				0:00:37	1				0:01:07	7		
Geom. Mean		1.48					1.24					1.87			

Table B.3.: Restricted Master Heuristics on the test set ColoringHard

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
p550-1	5.19	0.03	0	–	∞	5.52	0.24	0	–	∞	10.82	3.90	0	–	∞
p550-3	6.03	0.04	0	–	∞	7.11	1.12	0	–	∞	13.50	5.62	0	–	∞
p550-5	6.74	0.00	0	–	∞	6.70	0.00	0	–	∞	8.85	0.00	0	–	∞
p550-7	7.35	0.05	0	–	∞	8.00	0.69	0	–	∞	13.64	3.82	0	–	∞
p550-9	5.10	0.05	0	–	∞	5.82	0.64	0	–	∞	10.31	3.54	0	–	∞
p1250-1	7.44	0.02	0	–	∞	7.46	0.11	0	–	∞	2.49	0.08	1	956	149.61
p1250-3	2.76	0.03	0	–	∞	2.85	0.14	0	–	∞	4.79	1.25	0	–	∞
p1250-5	5.68	0.03	0	–	∞	5.84	0.16	0	–	∞	5.65	2.70	1	1410	228.67
p1250-7	8.59	0.10	0	–	∞	8.63	0.23	0	–	∞	12.80	5.42	1	445	0.00
p1250-9	7.00	0.02	0	–	∞	6.93	0.11	0	–	∞	5.13	1.43	1	989	126.83
p1650-1	1.56	0.02	0	–	∞	1.62	0.09	0	–	∞	3.08	1.34	1	1391	366.78
p1650-3	2.27	0.01	0	–	∞	2.40	0.18	0	–	∞	3.84	1.86	1	1425	353.82
p1650-5	6.93	0.02	0	–	∞	7.16	0.18	0	–	∞	3.07	1.04	1	1061	202.28
p1650-7	4.03	0.02	0	–	∞	4.35	0.17	0	–	∞	3.22	0.83	1	1029	185.04
p1650-9	3.44	0.02	0	–	∞	3.60	0.12	0	–	∞	3.25	0.45	1	904	142.36
p2050-1	1.76	0.01	0	–	∞	1.90	0.10	0	–	∞	1.92	0.06	1	991	272.56
p2050-3	1.96	0.01	0	–	∞	2.07	0.10	0	–	∞	2.73	0.33	1	1420	356.59
p2050-5	2.65	0.01	0	–	∞	2.70	0.12	0	–	∞	3.08	0.30	1	1199	236.80
p2050-7	2.92	0.01	0	–	∞	2.98	0.08	0	–	∞	3.60	0.68	1	1159	223.74
p2050-9	4.02	0.02	0	–	∞	4.10	0.13	0	–	∞	4.08	0.48	1	1140	176.70
p10100-11	58.64	0.11	0	–	∞	65.76	7.09	0	–	∞	97.76	17.29	0	–	∞
p10100-13	29.99	0.10	0	–	∞	32.71	2.80	0	–	∞	60.06	19.95	0	–	∞
p10100-15	118.70	0.10	0	–	∞	123.46	4.32	0	–	∞	188.61	22.59	0	–	∞
p10100-17	34.69	0.11	0	–	∞	41.85	7.25	0	–	∞	63.20	16.05	0	–	∞
p10100-19	24.03	0.09	0	–	∞	26.25	2.31	0	–	∞	56.06	24.12	0	–	∞
p25100-11	77.18	0.18	0	–	∞	80.37	3.29	0	–	∞	110.04	4.71	0	–	∞
p25100-13	62.04	0.36	0	–	∞	63.50	2.03	0	–	∞	88.40	22.52	0	–	∞
p25100-15	61.17	0.05	0	–	∞	64.03	2.76	0	–	∞	88.75	4.98	0	–	∞
p25100-17	21.10	0.04	0	–	∞	23.09	1.93	0	–	∞	33.00	4.51	0	–	∞
p25100-19	61.48	0.24	0	–	∞	63.70	2.34	0	–	∞	88.98	7.86	0	–	∞
p33100-11	48.90	0.03	0	–	∞	49.77	0.70	0	–	∞	70.37	3.57	0	–	∞
p33100-13	21.99	0.02	0	–	∞	22.72	0.49	0	–	∞	13.00	5.50	1	2098	370.40
p33100-15	36.15	0.04	0	–	∞	36.97	0.83	0	–	∞	18.70	10.83	1	479	1.05
p33100-17	13.49	0.03	0	–	∞	14.60	1.09	0	–	∞	22.16	3.82	0	–	∞
p33100-19	52.31	0.14	0	–	∞	53.46	1.40	0	–	∞	74.08	2.88	0	–	∞
p40100-11	22.27	0.10	0	–	∞	24.70	2.55	0	–	∞	15.48	7.60	1	415	0.00
p40100-13	14.65	0.07	0	–	∞	16.72	2.11	0	–	∞	13.30	5.30	1	2216	437.86
p40100-15	34.84	0.07	0	–	∞	35.29	0.65	0	–	∞	14.02	4.33	1	496	0.00
p40100-17	96.58	0.07	0	–	∞	98.24	2.00	0	–	∞	135.15	3.34	0	–	∞
p40100-19	159.88	0.09	0	–	∞	161.67	1.96	0	–	∞	12.20	6.23	2	451	0.22
p15150-21	115.25	0.14	0	–	∞	146.55	30.64	0	–	∞	222.85	65.36	0	–	∞
p15150-23	78.31	0.13	0	–	∞	104.50	25.92	0	–	∞	164.04	58.90	0	–	∞

Continue next page

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
p15150-25	74.46	0.15	0	–	∞	78.74	4.08	0	–	∞	163.63	61.72	0	–	∞
p15150-27	757.61	0.65	0	–	∞	771.63	12.22	0	–	∞	1098.31	76.72	0	–	∞
p15150-29	48.74	0.00	0	–	∞	48.81	0.00	0	–	∞	62.73	0.00	0	–	∞
p36150-21	460.29	0.41	0	–	∞	478.27	18.63	0	–	∞	667.87	37.41	0	–	∞
p36150-23	1760.77	0.31	0	–	∞	1794.88	31.66	0	–	∞	1730.07	106.75	1	867	0.00
p36150-25	42.65	0.06	0	–	∞	53.63	11.08	0	–	∞	71.58	14.99	0	–	∞
p36150-27	195.92	0.24	0	–	∞	204.06	8.06	0	–	∞	283.38	15.50	0	–	∞
p36150-29	89.39	0.06	0	–	∞	98.97	9.47	0	–	∞	134.57	12.96	0	–	∞
p37150-21	170.77	0.06	0	–	∞	177.88	7.22	0	–	∞	247.00	14.45	0	–	∞
p37150-23	997.17	0.27	0	–	∞	1016.23	15.26	0	–	∞	1412.32	59.49	0	–	∞
p37150-25	38.51	0.06	0	–	∞	38.84	0.21	0	–	∞	66.02	15.30	0	–	∞
p37150-27	1524.12	0.26	0	–	∞	1554.54	30.87	0	–	∞	–	35.07	0	–	∞
p37150-29	49.75	0.13	0	–	∞	61.09	11.51	0	–	∞	81.36	15.10	0	–	∞
p50150-21	524.22	0.24	0	–	∞	534.29	10.80	0	–	∞	753.75	27.29	0	–	∞
p50150-23	605.49	0.26	0	–	∞	616.17	10.89	0	–	∞	866.79	27.10	0	–	∞
p50150-25	21.55	0.04	0	–	∞	29.40	7.71	0	–	∞	36.29	7.75	0	–	∞
p50150-27	109.12	0.19	0	–	∞	114.74	5.82	0	–	∞	161.75	13.58	0	–	∞
p50150-29	182.50	0.40	0	–	∞	188.24	5.99	0	–	∞	261.41	9.99	0	–	∞
p60150-21	59.07	0.03	0	–	∞	62.96	3.57	0	–	∞	88.15	5.50	0	–	∞
p60150-23	435.22	0.20	0	–	∞	444.54	8.68	0	–	∞	494.11	30.29	1	560	0.00
p60150-25	121.65	0.32	0	–	∞	121.62	0.50	0	–	∞	176.89	8.51	0	–	∞
p60150-27	580.62	0.19	0	–	∞	592.07	11.52	0	–	∞	626.63	26.47	1	1943	148.47
p60150-29	25.79	0.03	0	–	∞	27.30	1.53	0	–	∞	39.24	14.90	1	3600	628.74
p20200-32	512.08	0.27	0	–	∞	568.04	56.56	0	–	∞	890.73	205.59	0	–	∞
p20200-34	419.25	0.27	0	–	∞	470.16	51.29	0	–	∞	718.16	158.21	0	–	∞
p20200-36	130.45	0.22	0	–	∞	164.05	33.87	0	–	∞	299.56	131.67	0	–	∞
p50200-36	237.42	0.19	0	–	∞	253.69	17.03	0	–	∞	360.67	44.30	0	–	∞
p50200-37	163.94	0.30	0	–	∞	183.31	19.74	0	–	∞	250.67	34.71	0	–	∞
p50200-39	109.07	0.20	0	–	∞	130.22	21.33	0	–	∞	170.55	28.49	0	–	∞
p66200-31	68.64	0.13	0	–	∞	68.67	0.36	0	–	∞	124.47	36.47	0	–	∞
p66200-36	189.65	0.32	0	–	∞	201.89	12.47	0	–	∞	288.39	35.05	0	–	∞
p66200-38	1063.36	0.26	0	–	∞	1092.03	28.48	0	–	∞	1497.26	56.18	0	–	∞
p80200-33	385.26	0.24	0	–	∞	404.80	18.94	0	–	∞	557.92	48.83	1	3502	526.48
p80200-34	368.25	0.17	0	–	∞	380.81	12.64	0	–	∞	505.87	21.42	0	–	∞
p80200-38	103.93	0.11	0	–	∞	115.16	11.70	0	–	∞	163.47	22.42	0	–	∞
Total		0:00:10	0				0:10:22	0				0:30:17	26		
Geom. Mean		1.00					3.68					9.51			

Table B.4.: Restricted Master Heuristics on the test set CpmpS

Instance	LP Restricted Master					Feasibility Restricted Master ($\alpha = 0.25$)					Feasibility Restricted Master ($\alpha = 0.50$)				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
new1_2_32	229.76	0.08	1	35744	0.08	238.26	0.11	1	31025	13.27	229.77	0.15	1	33014	7.71
new1_2_64	419.23	0.05	1	35751	0.06	419.98	0.08	1	29134	18.55	417.91	0.11	1	34009	4.93
new1_6_32	271.55	0.10	1	55487	0.01	282.80	0.13	1	51114	7.89	272.48	0.18	1	52346	5.67
new1_6_64	395.21	0.06	1	55487	0.01	409.95	0.08	1	48838	12.00	395.46	0.12	1	53846	2.97
new1_10_32	855.82	0.17	1	75716	0.05	887.12	0.25	1	65313	13.79	857.83	0.39	1	70150	7.40
new1_10_64	1402.50	0.10	1	75732	0.03	1403.79	0.13	1	59028	22.08	1402.71	0.23	1	68111	10.09
new2_1_32	141.57	0.05	1	3108	0.03	147.21	0.07	1	2893	6.95	141.55	0.08	1	2901	6.69
new2_1_64	234.37	0.04	1	3108	0.03	242.91	0.04	1	3003	3.41	234.46	0.04	1	3003	3.41
new2_3_32	196.94	0.07	1	4201	0.02	204.46	0.08	1	3907	7.02	197.21	0.10	1	3926	6.57
new2_3_64	335.78	0.04	1	4199	0.07	347.74	0.06	1	3834	8.76	336.41	0.07	1	4083	2.83
new2_7_32	411.02	0.11	1	6143	0.07	427.05	0.14	1	5111	16.85	411.43	0.20	1	5707	7.16
new2_7_64	735.67	0.07	1	6146	0.03	765.41	0.08	1	5715	7.04	737.91	0.11	1	5726	6.86
new3_1_32	842.44	0.08	1	71967	0.04	874.13	0.10	1	67293	6.53	843.72	0.14	1	70090	2.65
new3_1_64	1756.18	0.06	1	71967	0.04	1819.82	0.06	1	64999	9.72	1759.66	0.08	1	70905	1.52
new3_4_32	1338.36	0.13	1	107468	0.02	1388.32	0.18	1	99106	7.80	1333.59	0.28	1	101831	5.27
new3_4_64	2905.93	0.07	1	107491	0.00	3010.27	0.09	1	94948	11.67	2909.45	0.12	1	106087	1.31
new3_5_32	1818.74	0.16	1	120468	0.03	1889.00	0.22	1	80059	33.55	1821.12	0.35	1	114277	5.16
new3_5_64	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
new4_1_32	41.30	0.00	0	–	∞	42.91	0.00	0	–	∞	41.35	0.00	0	–	∞
new4_1_64	52.83	0.00	0	–	∞	55.07	0.00	0	–	∞	52.83	0.00	0	–	∞
new4_5_32	53.98	0.05	1	37016	0.00	113.19	0.04	0	–	∞	54.06	0.08	1	36540	1.29
new4_5_64	65.44	0.04	1	37002	0.04	68.31	0.05	1	36777	0.65	65.25	0.05	1	36822	0.52
new4_10_32	137.56	0.08	1	54842	0.00	150.56	0.07	0	–	∞	137.26	0.15	1	53460	2.52
new4_10_64	228.42	0.06	1	54842	0.00	238.20	0.06	1	53776	1.94	227.87	0.09	1	53895	1.73
new5_2_32	352.24	0.07	1	47892	0.05	367.77	0.11	1	33367	30.36	351.31	0.15	1	45694	4.63
new5_2_64	747.43	0.04	1	47914	0.00	780.12	0.06	1	47165	1.56	748.02	0.09	1	47210	1.47
new5_7_32	477.19	0.14	1	81925	0.05	497.82	0.17	1	67468	17.67	478.02	0.30	1	77532	5.43
new5_7_64	611.01	0.07	1	81969	0.02	637.83	0.09	1	73961	9.78	611.56	0.12	1	77051	6.01
new5_10_32	680.27	0.20	1	99930	0.09	694.80	0.25	1	82863	17.16	679.92	0.41	1	95265	4.76
new5_10_64	1345.89	0.11	1	99964	0.07	1381.70	0.15	1	82350	17.67	1345.73	0.24	1	97433	2.59
new6_1_32	568.26	0.08	1	71382	0.06	687.39	0.15	1	67266	5.82	568.94	0.14	1	69853	2.20
new6_1_64	1277.70	0.04	1	71385	0.06	1299.63	0.05	1	64433	9.79	1279.20	0.07	1	64433	9.79
new6_2_32	1131.09	0.10	1	82922	0.02	1151.27	0.14	1	68918	16.91	1130.29	0.23	1	80209	3.30
new6_2_64	2024.56	0.06	1	82923	0.02	2055.95	0.06	1	63491	23.45	2022.76	0.10	1	81611	1.60
new6_5_32	1323.66	0.16	1	119184	0.04	1343.16	0.20	1	84830	28.83	1320.06	0.36	1	112796	5.37
new6_5_64	1844.97	0.07	1	119211	0.02	1862.78	0.09	1	111549	6.44	1830.76	0.13	1	111555	6.44
new7_1_32	250.21	0.09	1	42653	0.07	254.83	0.09	1	38997	8.64	248.68	0.11	1	40691	4.67
new7_1_64	454.86	0.06	1	42685	0.00	463.78	0.07	1	38465	9.89	454.83	0.10	1	41829	2.01
new7_4_32	595.09	0.12	1	60674	0.07	607.27	0.17	1	54851	9.66	595.83	0.27	1	57627	5.09
new7_4_64	2193.16	0.10	1	60649	0.02	2236.72	0.11	1	46323	23.64	2192.73	0.17	1	59007	2.73
new7_9_32	1343.91	0.20	1	92362	0.02	1367.47	0.28	1	77874	15.70	1342.08	0.51	1	85571	7.37
new7_9_64	2976.42	0.13	1	92377	0.00	3023.77	0.18	1	74498	19.36	2978.21	0.27	1	89657	2.95
Total		0:00:03	39				0:00:04	37				0:00:06	39		
Geom. Mean		1.00					1.00					1.00			

Table B.5.: Restricted Master Heuristics on the test set RAPS

Instance	Row Greedy Column Selection					Feasibility Greedy Column Selection					Relaxation Based Column Selection				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
N1C1W1_A	0.64	0.01	1	28	12.00	0.61	0.01	1	28	12.00	0.60	0.00	0	–	∞
N1C1W1_K	0.49	0.01	1	30	15.38	0.46	0.01	1	30	15.38	0.49	0.00	0	–	∞
N1C1W2_B	0.71	0.00	0	–	∞	0.71	0.00	0	–	∞	0.66	0.00	0	–	∞
N1C1W2_L	0.76	0.00	0	–	∞	0.75	0.00	0	–	∞	0.76	0.00	0	–	∞
N1C1W4_C	0.70	0.00	0	–	∞	0.76	0.00	0	–	∞	0.71	0.00	0	–	∞
N1C1W4_M	0.70	0.00	0	–	∞	0.69	0.00	0	–	∞	0.71	0.00	0	–	∞
N1C2W1_D	0.45	0.00	0	–	∞	0.42	0.00	0	–	∞	0.47	0.00	0	–	∞
N1C2W1_N	1.03	0.01	1	24	14.29	1.12	0.00	1	24	14.29	1.07	0.00	0	–	∞
N1C2W2_E	0.60	0.00	0	–	∞	0.57	0.00	0	–	∞	0.59	0.00	0	–	∞
N1C2W2_O	0.49	0.00	0	–	∞	0.51	0.00	0	–	∞	0.50	0.00	0	–	∞
N1C2W4_F	0.50	0.00	0	–	∞	0.51	0.00	0	–	∞	0.51	0.00	0	–	∞
N1C2W4_P	0.70	0.00	0	–	∞	0.69	0.00	0	–	∞	0.71	0.00	0	–	∞
N1C3W1_G	0.82	0.00	1	17	13.33	0.83	0.00	1	17	13.33	0.86	0.00	0	–	∞
N1C3W1_Q	1.00	0.00	1	25	25.00	1.01	0.01	1	25	25.00	1.04	0.00	0	–	∞
N1C3W2_H	1.10	0.00	1	26	13.04	1.07	0.01	1	26	13.04	1.08	0.00	0	–	∞
N1C3W2_R	1.10	0.01	1	21	10.53	1.09	0.00	1	21	10.53	1.19	0.00	0	–	∞
N1C3W4_I	0.66	0.01	1	26	13.04	0.62	0.01	1	26	13.04	0.67	0.00	0	–	∞
N1C3W4_S	0.61	0.01	1	26	18.18	0.62	0.01	1	26	18.18	0.64	0.01	0	–	∞
N2C1W1_A	3.73	0.07	1	56	16.67	3.77	0.07	1	56	16.67	3.88	0.02	0	–	∞
N2C1W1_K	2.83	0.09	1	62	12.73	2.83	0.08	1	62	12.73	3.05	0.02	0	–	∞
N2C1W2_B	4.09	0.10	1	67	9.84	4.12	0.09	1	67	9.84	4.25	0.02	0	–	∞
N2C1W2_L	3.27	0.11	1	70	12.90	3.25	0.10	1	70	12.90	3.33	0.01	0	–	∞
N2C1W4_C	3.15	0.00	0	–	∞	3.17	0.00	0	–	∞	3.19	0.00	0	–	∞
N2C1W4_M	3.45	0.12	1	75	4.17	3.40	0.12	1	75	4.17	3.47	0.01	0	–	∞
N2C2W1_D	7.51	0.06	1	49	16.67	7.65	0.05	1	49	16.67	7.44	0.03	1	42	0.00
N2C2W1_N	5.73	0.06	1	52	20.93	5.68	0.06	1	52	20.93	5.91	0.03	0	–	∞
N2C2W2_E	2.92	0.08	1	60	11.11	2.88	0.08	1	60	11.11	2.91	0.01	1	54	0.00
N2C2W2_O	4.17	0.08	1	59	18.00	4.21	0.07	1	59	18.00	4.40	0.01	0	–	∞
N2C2W4_F	4.89	0.00	0	–	∞	5.03	0.00	0	–	∞	4.96	0.00	0	–	∞
N2C2W4_P	3.76	0.09	1	66	10.00	3.81	0.10	1	66	10.00	3.76	0.01	1	60	0.00
N2C3W1_G	3.47	0.04	1	41	24.24	3.40	0.03	1	41	24.24	3.72	0.03	0	–	∞
N2C3W1_Q	3.37	0.04	1	39	14.71	3.50	0.04	1	39	14.71	3.97	0.03	0	–	∞
N2C3W2_H	4.10	0.05	1	43	13.16	4.10	0.04	1	43	13.16	4.36	0.03	0	–	∞
N2C3W2_R	4.71	0.05	1	45	12.50	4.74	0.05	1	45	12.50	5.05	0.02	0	–	∞
N2C3W4_I	6.62	0.05	1	50	13.64	6.80	0.06	1	50	13.64	6.86	0.03	0	–	∞
N2C3W4_S	10.28	0.05	1	47	11.90	10.36	0.05	1	47	11.90	10.58	0.03	0	–	∞
N3C1W1_A	13.17	1.10	1	120	14.29	13.22	1.08	1	120	14.29	14.74	0.22	0	–	∞
N3C1W1_K	13.67	1.05	1	117	14.71	13.55	1.03	1	117	14.71	14.44	0.14	0	–	∞
N3C1W2_B	10.74	0.00	0	–	∞	10.53	0.00	0	–	∞	10.68	0.00	0	–	∞
N3C1W2_L	14.42	0.00	0	–	∞	14.13	0.00	0	–	∞	14.26	0.00	0	–	∞
N3C1W4_C	17.23	2.11	1	152	4.11	17.18	2.09	1	152	4.11	17.54	0.11	0	–	∞
N3C1W4_M	17.14	0.00	0	–	∞	17.51	0.00	0	–	∞	17.43	0.00	0	–	∞

Continue next page

Instance	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
	Row Greedy Column Selection					Feasibility Greedy Column Selection					Relaxation Based Column Selection				
N3C2W1_D	10.36	0.67	1	100	17.65	10.33	0.66	1	100	17.65	12.43	0.12	0	–	∞
N3C2W1_N	11.42	0.78	1	105	15.38	11.38	0.77	1	105	15.38	11.56	0.04	0	–	∞
N3C2W2_E	13.90	1.40	1	130	12.07	13.78	1.38	1	130	12.07	13.76	0.05	1	116	0.00
N3C2W2_O	13.48	1.12	1	122	14.02	13.39	1.12	1	122	14.02	13.34	0.04	1	107	0.00
N3C2W4_F	14.61	1.33	1	129	12.17	14.57	1.30	1	129	12.17	14.63	0.05	1	115	0.00
N3C2W4_P	19.08	1.51	1	134	9.84	18.57	1.48	1	134	9.84	19.17	0.09	0	–	∞
N3C3W1_G	13.59	1.21	1	77	18.46	13.53	1.12	1	77	18.46	17.83	0.34	0	–	∞
N3C3W1_Q	14.58	0.97	1	84	15.07	14.39	0.94	1	84	15.07	19.14	0.36	0	–	∞
N3C3W2_H	37.48	0.62	1	96	17.07	37.44	0.60	1	96	17.07	46.37	0.28	0	–	∞
N3C3W2_R	16.01	0.58	1	96	21.52	15.96	0.56	1	96	21.52	18.29	0.28	0	–	∞
N3C3W4_I	17.42	0.80	1	109	18.48	17.34	0.79	1	109	18.48	18.89	0.23	0	–	∞
N3C3W4_S	35.20	0.62	1	93	10.71	34.87	0.60	1	93	10.71	37.03	0.30	0	–	∞
Total		0:00:17	40				0:00:16	40				0:00:03	6		
Geom. Mean		1.04					1.04					1.00			

Table B.6.: Column Selection Heuristics on the test set BINDATA1S

Instance	Row Greedy Column Selection					Feasibility Greedy Column Selection					Relaxation Based Column Selection				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
1-FullIns_3	1.32	0.00	1	5	25.00	1.22	0.00	1	5	25.00	1.41	0.00	0	–	∞
2-FullIns_3	2.30	0.00	1	6	20.00	2.29	0.00	1	6	20.00	3.47	0.00	0	–	∞
3-FullIns_3	2.43	0.00	1	9	50.00	2.39	0.01	1	9	50.00	7.19	0.00	0	–	∞
4-FullIns_3	4.07	0.04	1	10	42.86	4.06	0.01	1	10	42.86	12.58	0.02	0	–	∞
5-FullIns_3	21.04	0.03	0	–	∞	21.24	0.03	0	–	∞	21.04	0.01	0	–	∞
anna	1.10	0.05	1	14	27.27	1.11	0.02	1	14	27.27	6.33	0.01	0	–	∞
david	1.33	0.00	0	–	∞	1.35	0.00	0	–	∞	1.38	0.00	0	–	∞
games120	18.44	0.01	0	–	∞	18.47	0.02	0	–	∞	18.46	0.00	0	–	∞
homer	1711.37	5.75	0	–	∞	1706.40	3.65	0	–	∞	1704.82	2.00	0	–	∞
huck	0.53	0.00	1	15	36.36	0.52	0.01	1	15	36.36	1.79	0.01	0	–	∞
jean	0.45	0.01	1	13	30.00	0.48	0.01	1	13	30.00	1.42	0.01	0	–	∞
miles1000	18.09	0.00	0	–	∞	18.23	0.00	0	–	∞	17.94	0.00	0	–	∞
miles1500	47.19	0.00	0	–	∞	47.54	0.00	0	–	∞	45.73	0.00	0	–	∞
miles250	13.50	0.03	0	–	∞	13.50	0.02	0	–	∞	13.55	0.03	0	–	∞
miles500	3.12	0.07	1	27	35.00	3.14	0.05	1	27	35.00	10.51	0.02	0	–	∞
miles750	5.64	0.00	0	–	∞	5.65	0.00	0	–	∞	5.56	0.00	0	–	∞
mulsol.i.1	59.28	0.23	1	59	20.41	59.51	0.21	1	59	20.41	65.81	0.06	0	–	∞
mulsol.i.2	61.52	1.09	1	41	32.26	61.33	0.82	1	41	32.26	294.67	0.25	0	–	∞
mulsol.i.3	64.59	1.16	1	41	32.26	64.66	0.89	1	41	32.26	346.50	0.22	0	–	∞
mulsol.i.4	72.40	1.12	1	42	35.48	72.47	0.84	1	42	35.48	393.13	0.25	0	–	∞
mulsol.i.5	72.11	0.74	1	39	25.81	72.17	0.55	1	39	25.81	244.23	0.20	0	–	∞
myciel3	0.12	0.00	0	–	∞	0.13	0.00	0	–	∞	0.10	0.00	0	–	∞
myciel4	2.60	0.13	2	5	0.00	2.53	0.05	2	5	0.00	11.42	0.00	0	–	∞
queen5_5	0.25	0.00	0	–	∞	0.23	0.00	0	–	∞	0.23	0.00	0	–	∞
queen6_6	9.02	0.00	0	–	∞	9.13	0.00	0	–	∞	9.10	0.00	0	–	∞
queen7_7	16.62	0.00	0	–	∞	16.45	0.00	0	–	∞	16.69	0.00	0	–	∞
queen8_12	183.24	0.06	0	–	∞	183.57	0.03	0	–	∞	182.84	0.02	0	–	∞
zeroin.i.1	56.48	1.20	1	55	12.24	57.25	1.12	1	55	12.24	195.94	0.18	0	–	∞
zeroin.i.2	50.02	1.16	1	37	23.33	50.37	0.91	1	37	23.33	373.06	0.27	0	–	∞
zeroin.i.3	43.00	1.15	1	37	23.33	42.91	0.90	1	37	23.33	300.86	0.21	0	–	∞
Total		0:00:14	18				0:00:10	18				0:00:03	0		
Geom. Mean		1.09					1.05					1.02			

Table B.7.: Column Selection Heuristics on the test set ColoringEasy

Instance	Row Greedy Column Selection					Feasibility Greedy Column Selection					Relaxation Based Column Selection				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
1-FullIns_4	169.99	0.23	1	6	0.00	169.76	0.13	1	6	0.00	–	0.09	0	–	∞
2-Insertions_3	6.88	1.75	2	4	0.00	6.87	1.23	2	4	0.00	1307.77	0.53	0	–	∞
4-FullIns_4	–	0.19	0	–	∞	–	0.12	0	–	∞	–	0.11	0	–	∞
ash331GPIA	–	0.06	0	–	∞	–	0.04	0	–	∞	–	0.07	0	–	∞
DSJC125.9	1824.44	0.11	1	53	0.00	1824.54	0.11	1	53	0.00	–	0.05	0	–	∞
DSJR500.1	–	1.45	0	–	∞	–	0.96	0	–	∞	–	0.94	0	–	∞
fpsol2.i.1	576.67	16.86	1	84	0.00	581.09	15.46	1	84	0.00	–	3.93	0	–	∞
fpsol2.i.2	360.48	6.20	1	43	0.00	358.65	4.64	1	43	0.00	–	1.26	0	–	∞
fpsol2.i.3	354.88	5.22	1	43	0.00	354.27	3.90	1	43	0.00	–	1.06	0	–	∞
inithx.i.1	1854.75	3.44	1	79	0.00	1852.52	3.13	1	79	0.00	–	1.40	0	–	∞
inithx.i.2	–	1.20	0	–	∞	–	0.93	0	–	∞	–	0.56	0	–	∞
inithx.i.3	–	1.24	0	–	∞	–	0.94	0	–	∞	–	0.52	0	–	∞
le450_25a	–	1.68	0	–	∞	–	1.37	0	–	∞	–	1.02	0	–	∞
le450_25b	–	1.66	0	–	∞	–	1.37	0	–	∞	–	1.01	0	–	∞
qg.order30	–	0.89	0	–	∞	–	0.82	0	–	∞	–	0.64	0	–	∞
qg.order40	–	2.72	0	–	∞	–	2.57	0	–	∞	–	2.07	0	–	∞
queen8_8	32.15	0.09	1	12	20.00	32.26	0.07	1	12	20.00	1059.91	0.03	0	–	∞
queen9_9	153.37	0.07	1	13	0.00	153.27	0.04	1	13	0.00	–	0.02	0	–	∞
queen10_10	328.53	0.05	2	13	0.00	328.73	0.04	2	13	0.00	–	0.03	0	–	∞
school1	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
school1_nsh	–	0.00	0	–	∞	–	0.00	0	–	∞	–	0.00	0	–	∞
wap05a	–	2.13	0	–	∞	–	2.47	0	–	∞	–	1.43	0	–	∞
will199GPIA	–	0.16	0	–	∞	–	0.10	0	–	∞	–	0.15	0	–	∞
Total		0:00:47	12				0:00:40	12				0:00:16	0		
Geom. Mean		1.66					1.51					1.14			

Table B.8.: Column Selection Heuristics on the test set ColoringHard

Instance	Row Greedy Column Selection					Feasibility Greedy Column Selection					Relaxation Based Column Selection				
	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap	First	HTime	NSols	Primal	HGap
new1_2_32	229.29	0.02	4	1444	95.96	238.39	0.02	4	1444	95.96	237.98	0.06	3	35130	1.79
new1_2_64	419.37	0.02	1	2331	93.48	435.16	0.03	1	2331	93.48	434.62	0.01	1	35650	0.34
new1_6_32	272.15	0.01	1	1083	98.05	281.81	0.01	1	1083	98.05	282.01	0.03	1	55341	0.28
new1_6_64	394.82	0.02	2	2390	95.69	408.79	0.02	2	2390	95.69	794.77	0.02	0	–	∞
new1_10_32	856.93	0.02	1	1336	98.24	889.04	0.02	1	1336	98.24	1068.40	0.06	0	–	∞
new1_10_64	1401.88	0.02	1	2464	96.75	1454.88	0.01	1	2464	96.75	1451.90	0.02	1	75635	0.16
new2_1_32	141.45	0.01	1	113	96.37	147.24	0.00	1	113	96.37	147.19	0.01	1	3091	0.58
new2_1_64	233.82	0.00	1	284	90.87	242.92	0.00	1	284	90.87	242.38	0.01	1	3100	0.29
new2_3_32	197.24	0.01	1	127	96.98	204.88	0.01	1	127	96.98	204.38	0.03	1	4060	3.38
new2_3_64	336.19	0.01	1	250	94.05	349.20	0.01	1	250	94.05	348.51	0.01	1	4188	0.33
new2_7_32	411.77	0.03	3	133	97.84	426.46	0.04	3	133	97.84	426.11	0.13	1	6112	0.57
new2_7_64	738.17	0.01	1	225	96.34	766.09	0.01	1	225	96.34	975.46	0.02	0	–	∞
new3_1_32	842.66	0.02	2	3120	95.67	875.21	0.02	2	3120	95.67	3012.13	0.03	0	–	∞
new3_4_32	1339.62	0.01	1	3225	97.00	1387.70	0.02	1	3225	97.00	3134.05	0.06	0	–	∞
new3_4_64	2899.51	0.01	1	5804	94.60	3007.59	0.02	1	5804	94.60	3005.94	0.02	1	107280	0.20
new3_5_32	1817.51	0.01	1	2304	98.09	1887.20	0.02	1	2304	98.09	1884.90	0.06	1	120244	0.20
new3_5_64	3600.07	0.01	1	6188	0.00	3600.02	0.01	1	6188	0.00	3600.03	0.02	1	120498	0.00
new4_1_32	41.30	0.00	0	–	∞	43.05	0.00	0	–	∞	42.95	0.00	0	–	∞
new4_1_64	52.76	0.00	0	–	∞	54.72	0.00	0	–	∞	54.56	0.00	0	–	∞
new4_5_32	53.94	0.01	1	950	97.43	56.14	0.01	1	950	97.43	55.83	0.01	1	36081	2.53
new4_5_64	65.06	0.01	1	1739	95.30	68.02	0.01	1	1739	95.30	67.74	0.01	1	36968	0.13
new4_10_32	137.32	0.01	1	874	98.41	142.98	0.02	1	874	98.41	142.75	0.02	1	54770	0.13
new4_10_64	227.71	0.02	1	1613	97.06	237.32	0.01	1	1613	97.06	237.40	0.01	1	52745	3.82
new5_2_32	352.58	0.01	1	1662	96.53	366.66	0.01	1	1662	96.53	365.98	0.01	1	43032	10.19
new5_2_64	746.75	0.01	1	3266	93.18	775.52	0.00	1	3266	93.18	773.78	0.01	1	47789	0.26
new5_7_32	476.54	0.04	3	1762	97.85	495.70	0.04	3	1762	97.85	495.69	0.16	2	81426	0.64
new5_7_64	611.01	0.02	1	3220	96.07	634.44	0.01	1	3220	96.07	635.69	0.03	1	81742	0.29
new5_10_32	678.81	0.04	2	1743	98.26	705.39	0.04	2	1743	98.26	704.84	0.12	2	97919	2.11
new5_10_64	1344.34	0.01	1	3201	96.80	1397.48	0.01	1	3201	96.80	1401.14	0.05	1	99645	0.37
new6_1_32	567.70	0.02	2	2655	96.28	588.60	0.01	2	2655	96.28	588.86	0.02	1	68147	4.59
new6_1_64	1275.15	0.01	1	5677	92.05	1324.60	0.01	1	5677	92.05	1321.74	0.01	1	65141	8.80
new6_2_32	1130.01	0.01	1	3002	96.38	1173.30	0.01	1	3002	96.38	1176.43	0.02	1	82780	0.20
new6_2_64	2023.09	0.01	1	5147	93.79	2093.56	0.00	1	5147	93.79	2099.09	0.02	1	82874	0.08
new6_5_32	1319.71	0.01	1	3594	96.98	1370.60	0.02	1	3594	96.99	1370.24	0.06	1	119007	0.19
new6_5_64	1826.61	0.02	1	6403	94.63	1895.63	0.01	1	6403	94.63	1893.70	0.03	1	119168	0.05
new7_1_64	454.64	0.00	1	3544	91.70	471.50	0.00	1	3544	91.70	471.96	0.01	1	42631	0.13
new7_4_32	594.39	0.02	2	1703	97.20	618.14	0.03	2	1703	97.20	1216.06	0.04	0	–	∞
new7_4_64	2197.73	0.00	1	3480	94.26	2281.77	0.01	1	3480	94.26	2275.96	0.02	1	46786	22.84
new7_9_32	1340.18	0.02	1	1498	98.38	1391.38	0.02	1	1498	98.38	1390.84	0.06	1	90596	1.93
new7_9_64	2976.21	0.02	2	3545	96.16	3084.20	0.01	1	3545	96.16	3077.59	0.02	1	92174	0.22
Total		0:00:00	51				0:00:00	50				0:00:01	36		
Geom. Mean		1.00					1.00					1.00			

Table B.9.: Column Selection Heuristics on the test set RapS

	RENS							Extreme Points Crossover						
Instance	Nodes	Time	First	HTime	NSols	Primal	HGap	Nodes	Time	First	HTime	NSols	Primal	HGap
N1C1W1_A	1	0.69	0.69	0.04	1	25	0.00	7	7.10	7.08	6.24	1	26	4.00
N1C1W1_K	1	0.49	0.48	0.02	1	26	0.00	1	2.95	2.95	2.36	1	26	0.00
N1C1W2_B	1	0.85	0.84	0.12	1	30	0.00	4	8.31	8.27	7.37	1	31	3.33
N1C1W2_L	1	0.73	0.72	0.00	0	–	∞	1	1.03	1.01	0.00	0	–	∞
N1C1W4_C	1	0.82	0.80	0.00	0	–	∞	1	1.03	0.99	0.00	0	–	∞
N1C1W4_M	1	0.73	0.72	0.04	1	41	0.00	1	4.61	4.61	3.64	1	41	0.00
N1C2W1_D	1	0.47	0.46	0.00	0	–	∞	1	0.61	0.60	0.00	0	–	∞
N1C2W1_N	1	1.98	1.98	0.87	1	21	0.00	1	7.42	7.42	6.02	1	21	0.00
N1C2W2_E	1	0.64	0.63	0.03	1	33	0.00	1	4.52	4.51	3.70	1	33	0.00
N1C2W2_O	1	0.53	0.52	0.02	1	29	0.00	1	1.66	1.65	1.00	1	29	0.00
N1C2W4_F	1	0.54	0.53	0.00	0	–	∞	1	0.74	0.72	0.00	0	–	∞
N1C2W4_P	1	0.73	0.72	0.00	0	–	∞	1	0.96	0.93	0.00	0	–	∞
N1C3W1_G	10	2.04	2.03	1.15	0	–	∞	10	6.46	6.44	5.39	1	16	6.67
N1C3W1_Q	1	1.29	1.29	0.25	1	20	0.00	12	6.74	6.70	5.38	1	21	5.00
N1C3W2_H	1	1.66	1.65	0.55	1	23	0.00	1	6.01	6.00	4.65	1	23	0.00
N1C3W2_R	23	2.11	2.11	0.97	0	–	∞	23	6.95	6.95	5.45	0	–	∞
N1C3W4_I	1	0.73	0.72	0.07	1	23	0.00	5	7.46	7.44	6.62	1	24	4.35
N1C3W4_S	1	0.75	0.75	0.16	1	22	0.00	12	7.34	7.34	6.49	0	–	∞
N2C1W1_A	1	3.87	3.86	0.09	1	48	0.00	21	38.54	38.52	33.47	0	–	∞
N2C1W1_K	12	3.46	3.45	0.44	0	–	∞	12	39.73	39.71	35.81	0	–	∞
N2C1W2_B	1	4.48	4.46	0.43	1	61	0.00	4	39.48	39.35	34.16	1	64	4.92
N2C1W2_L	1	3.34	3.32	0.10	1	62	0.00	4	49.51	49.41	45.06	1	64	3.23
N2C1W4_C	1	3.23	3.13	0.00	0	–	∞	1	4.15	4.00	0.00	0	–	∞
N2C1W4_M	1	3.56	3.53	0.10	1	72	0.00	5	30.35	30.18	25.80	1	73	1.39
N2C2W1_D	22	11.67	11.66	3.99	0	–	∞	22	38.38	38.15	28.72	1	45	7.14
N2C2W1_N	1	7.09	7.08	1.27	1	43	0.00	20	38.25	38.03	30.75	1	45	4.65
N2C2W2_E	1	3.00	2.98	0.08	1	54	0.00	6	35.03	34.92	31.28	1	55	1.85
N2C2W2_O	1	4.14	4.12	0.07	1	50	0.00	15	34.15	34.13	28.53	0	–	∞
N2C2W4_F	1	5.15	5.08	0.00	0	–	∞	1	6.62	6.51	0.00	0	–	∞
N2C2W4_P	1	3.90	3.88	0.08	1	60	0.00	3	75.13	75.04	70.07	1	62	3.33
N2C3W1_G	32	9.79	9.78	5.97	0	–	∞	32	23.02	23.00	18.08	0	–	∞
N2C3W1_Q	41	10.37	10.36	6.44	0	–	∞	41	24.36	23.68	19.27	1	36	5.88
N2C3W2_H	30	9.38	9.37	5.09	0	–	∞	30	27.65	27.31	22.12	1	42	10.53
N2C3W2_R	30	7.71	7.69	2.71	0	–	∞	30	32.95	32.58	26.43	1	42	5.00
N2C3W4_I	23	8.48	8.47	1.80	0	–	∞	23	42.79	42.54	34.00	1	47	6.82
N2C3W4_S	32	12.88	12.87	2.18	0	–	∞	32	46.93	46.59	33.73	1	45	7.14
N3C1W1_A	1	14.18	14.11	0.84	1	105	0.00	25	207.11	205.03	185.74	1	117	11.43
N3C1W1_K	1	13.99	13.93	0.36	1	102	0.00	14	201.19	200.15	180.61	1	110	7.84
N3C1W2_B	1	11.04	10.47	0.00	0	–	∞	1	17.28	16.52	0.00	0	–	∞
N3C1W2_L	1	14.68	14.11	0.00	0	–	∞	1	21.66	20.91	0.00	0	–	∞
N3C1W4_C	1	17.57	17.47	0.54	1	146	0.00	6	156.35	155.44	131.02	1	154	5.48
N3C1W4_M	1	17.68	17.12	0.00	0	–	∞	1	25.20	24.46	0.00	0	–	∞

Continue next page

Instance	RENS							Extreme Points Crossover						
	Nodes	Time	First	HTime	NSols	Primal	HGap	Nodes	Time	First	HTime	NSols	Primal	HGap
N3C2W1_D	1	10.72	10.66	0.36	1	85	0.00	34	148.78	148.67	130.91	0	–	∞
N3C2W1_N	1	11.78	11.72	0.31	1	91	0.00	2	158.96	158.86	141.58	0	–	∞
N3C2W2_E	1	14.17	14.09	0.42	1	116	0.00	3	200.31	200.18	179.66	0	–	∞
N3C2W2_O	1	13.60	13.52	0.37	1	107	0.00	2	153.65	153.54	133.91	0	–	∞
N3C2W4_F	1	14.99	14.92	0.38	1	115	0.00	2	266.35	266.23	244.66	0	–	∞
N3C2W4_P	1	19.16	19.08	0.45	1	122	0.00	6	264.70	264.57	237.36	0	–	∞
N3C3W1_G	83	38.10	38.05	20.48	0	–	∞	83	143.36	143.27	119.26	0	–	∞
N3C3W1_Q	64	36.65	36.60	17.86	0	–	∞	64	143.34	143.24	117.45	0	–	∞
N3C3W2_H	38	61.44	61.38	15.07	0	–	∞	38	245.98	245.88	185.82	0	–	∞
N3C3W2_R	41	32.97	32.91	15.10	0	–	∞	41	217.41	217.32	192.67	0	–	∞
N3C3W4_I	30	21.85	21.79	3.26	0	–	∞	30	261.56	261.45	235.59	0	–	∞
N3C3W4_S	41	47.34	47.28	10.82	0	–	∞	41	277.47	274.97	228.90	1	90	7.14
Total	590	0:09:05		0:02:01	28			780	1:03:39		0:53:46	27		
Geom. Mean		5.09		1.57					24.07		16.48			

Table B.10.: RENS vs. Extreme Points Crossover on the test set BINDATA1S

	RENS							Extreme Points Crossover						
Instance	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**
1-FullIns_3	11	1.31	1.31	0.00	0	–	∞	1	1.62	1.61	0.04	1	4	0.00
2-FullIns_3	21	3.54	3.54	0.00	0	–	∞	21	4.38	4.38	0.01	0	–	∞
3-FullIns_3	45	7.19	7.19	0.01	0	–	∞	1	3.18	3.17	0.12	1	6	0.00
4-FullIns_3	1	9.11	9.11	5.05	1	7	0.00	1	5.92	5.91	0.88	1	7	0.00
5-FullIns_3	1	12.40	12.38	6.75	1	8	0.00	1	14.50	14.48	7.30	1	8	0.00
anna	1	1.20	1.19	0.05	1	11	0.00	1	2.80	2.78	1.32	1	11	0.00
david	19	1.35	1.34	0.05	0	–	∞	1	0.99	0.98	0.19	1	11	0.00
games120	1	1.55	1.53	0.48	1	9	0.00	1	3.67	3.65	2.28	1	9	0.00
homer	1	20.25	20.20	5.94	1	13	0.00	1	42.77	42.69	24.71	1	13	0.00
huck	1	0.44	0.43	0.04	1	11	0.00	1	1.12	1.11	0.56	1	11	0.00
jean	1	0.36	0.35	0.03	1	10	0.00	1	0.66	0.65	0.18	1	10	0.00
miles1000	1	24.54	18.16	0.00	0	–	∞	1	32.36	24.00	0.00	0	–	∞
miles1500	1	77.13	47.50	0.00	0	–	∞	1	104.91	60.40	0.00	0	–	∞
miles250	1	1.49	1.48	0.58	1	8	0.00	1	3.28	3.26	2.02	1	8	0.00
miles500	1	3.31	3.28	0.21	1	20	0.00	54	19.61	9.64	5.54	1	21	5.00
miles750	1	7.85	5.72	0.00	0	–	∞	1	10.18	7.63	0.00	0	–	∞
mulsol.i.1	1	61.35	61.16	1.40	1	49	0.00	12	122.65	113.40	32.44	1	52	6.12
mulsol.i.2	1	64.48	64.37	2.21	1	31	0.00	102	421.30	421.14	30.73	0	–	∞
mulsol.i.3	1	68.90	68.79	3.26	1	31	0.00	100	498.54	498.39	39.12	0	–	∞
mulsol.i.4	1	74.14	74.03	1.00	1	31	0.00	100	566.53	566.37	45.59	0	–	∞
mulsol.i.5	1	76.73	76.61	3.95	1	31	0.00	70	353.05	352.90	27.44	0	–	∞
myciel3	7	0.13	0.13	0.00	0	–	∞	5	0.19	0.14	0.02	1	4	0.00
myciel4	595	92.46	12.08	0.49	0	–	∞	505	104.74	3.21	0.10	1	5	0.00
queen5_5	1	0.20	0.20	0.00	0	–	∞	1	0.28	0.28	0.00	0	–	∞
queen6_6	1	4.92	4.91	1.53	1	7	0.00	14	31.24	7.44	3.13	1	8	14.29
queen7_7	1	16.81	16.78	0.00	0	–	∞	1	20.58	20.54	0.00	0	–	∞
queen8_12	186	847.06	218.24	32.01	0	–	∞	186	1081.03	273.87	35.11	0	–	∞
zeroin.i.1	1	59.14	58.95	1.50	1	49	0.00	86	289.60	289.37	28.07	0	–	∞
zeroin.i.2	1	51.86	51.76	0.91	1	30	0.00	137	512.20	84.89	17.79	1	31	3.33
zeroin.i.3	1	45.47	45.37	1.81	1	30	0.00	1	82.75	82.60	24.66	1	30	0.00
Total	907	0:27:16		0:01:09	18			1409	1:12:16		0:05:29	18		
Geom. Mean		11.04		1.59					21.53		3.86			

Table B.11.: RENS vs. Extreme Points Crossover on the test set ColoringEasy

Instance	RENS Nodes	RENS Time	RENS First	RENS HTime	RENS NSols	RENS Primal	RENS HGap	Extreme Points Crossover Nodes	Extreme Points Crossover Time	Extreme Points Crossover First	Extreme Points Crossover HTime	Extreme Points Crossover NSols	Extreme Points Crossover Primal	Extreme Points Crossover HGap
1-FullIns_4	120	1800.20	–	0.01	0	–	∞	98	1800.02	–	1.09	0	–	∞
2-Insertions_3	1054	1800.02	1310.44	0.62	0	–	∞	971	1800.12	1642.03	0.06	0	–	∞
4-FullIns_4	1	1800.41	–	0.00	0	–	∞	1	1804.08	–	0.00	0	–	∞
ash331GPIA	1	1803.13	–	0.00	0	–	∞	2	1803.91	–	0.00	0	–	∞
DSJC125.9	1	1823.38	–	0.00	0	–	∞	1	1819.84	–	0.00	0	–	∞
DSJR500.1	88	1800.56	–	48.46	0	–	∞	74	1800.72	–	77.62	0	–	∞
fpsol2.i.1	291	1789.56	–	9.97	0	–	∞	189	1727.67	–	155.12	0	–	∞
fpsol2.i.2	106	1800.13	–	14.30	0	–	∞	103	1784.01	–	75.18	0	–	∞
fpsol2.i.3	1	366.49	366.22	7.61	1	30	0.00	87	1778.14	–	67.35	0	–	∞
inithx.i.1	1	1849.70	–	0.00	0	–	∞	1	2311.55	–	0.00	0	–	∞
inithx.i.2	1	1970.19	–	0.00	0	–	∞	1	2035.73	–	0.00	0	–	∞
inithx.i.3	1	1966.64	–	0.00	0	–	∞	1	2029.59	–	0.00	0	–	∞
le450_25a	146	1800.19	–	95.61	0	–	∞	165	1794.05	–	360.03	0	–	∞
le450_25b	139	1800.21	–	119.98	0	–	∞	153	1798.64	691.34	367.61	1	29	0.00
qg.order30	1	3348.49	–	0.00	0	–	∞	1	3728.99	–	0.00	0	–	∞
qg.order40	1	9142.25	–	0.00	0	–	∞	1	1536.40	–	0.00	0	–	∞
queen8_8	389	1078.13	1078.11	10.51	0	–	∞	389	1396.34	1396.31	10.60	0	–	∞
queen9_9	232	1800.02	1799.69	16.97	0	–	∞	304	1800.00	–	22.40	0	–	∞
queen10_10	105	1800.00	–	24.69	0	–	∞	139	1798.21	–	31.92	0	–	∞
school1	1	1792.78	–	0.00	0	–	∞	1	1771.91	–	0.00	0	–	∞
school1_nsh	1	1794.63	–	0.00	0	–	∞	1	1778.34	–	0.00	0	–	∞
wap05a	1	5458.88	–	0.00	0	–	∞	1	6307.01	–	0.00	0	–	∞
will199GPIA	1	1802.82	–	0.00	0	–	∞	1	1800.54	–	0.00	0	–	∞
Total	2683	14:29:48		0:05:48	1			2685	13:20:05		0:19:28	1		
Geom. Mean		1918.10		3.47					1962.19		5.39			

Table B.12.: RENS vs. Extreme Points Crossover on the test set ColoringHard

Instance	RENS							Extreme Points Crossover						
	Nodes	Time	First	HTime	NSols	Primal	HGap	Nodes	Time	First	HTime	NSols	Primal	HGap
p550-1	14	9.33	5.21	0.03	0	–	∞	14	12.17	5.13	0.07	1	813	14.03
p550-3	3	6.42	5.71	0.04	1	751	0.00	3	8.57	7.57	0.15	1	751	0.00
p550-5	1	6.75	6.75	0.00	0	–	∞	1	8.85	8.84	0.00	0	–	∞
p550-7	21	19.22	7.38	0.03	0	–	∞	19	22.27	6.68	0.11	1	793	0.76
p550-9	9	8.63	5.12	0.03	0	–	∞	9	11.51	5.47	0.09	1	729	1.96
p1250-1	139	13.17	7.27	0.02	0	–	∞	154	22.20	3.07	0.69	1	424	10.70
p1250-3	109	11.44	2.71	0.03	0	–	∞	109	15.15	3.38	0.64	1	435	7.41
p1250-5	95	13.32	5.68	0.03	0	–	∞	95	18.76	8.98	1.59	0	–	∞
p1250-7	2035	229.26	8.71	0.03	0	–	∞	2035	299.41	4.13	1.28	1	1207	171.24
p1250-9	122	27.08	6.92	0.03	0	–	∞	122	37.80	5.56	2.03	1	584	33.94
p1650-1	5	1.81	1.56	0.03	0	–	∞	5	2.40	1.93	0.15	1	328	10.07
p1650-3	11	2.58	2.26	0.03	0	–	∞	11	3.57	2.21	0.34	1	341	8.60
p1650-5	61	8.54	6.93	0.03	0	–	∞	46	9.83	2.79	0.71	1	390	11.11
p1650-7	88	9.68	4.10	0.02	0	–	∞	88	13.46	3.38	1.03	1	410	13.57
p1650-9	101	14.27	3.45	0.02	0	–	∞	101	20.50	4.60	1.98	1	903	142.09
p2050-1	41	4.38	1.79	0.02	0	–	∞	44	6.49	1.97	0.13	1	273	2.63
p2050-3	14	3.03	1.96	0.02	0	–	∞	14	4.13	2.50	0.17	1	341	9.65
p2050-5	5	2.85	2.16	0.03	1	357	0.28	5	4.05	3.08	0.31	1	357	0.28
p2050-7	6	3.07	2.27	0.02	1	359	0.28	6	4.11	3.06	0.18	1	359	0.28
p2050-9	337	41.55	4.09	0.04	0	–	∞	246	43.83	3.99	0.40	1	432	4.85
p10100-11	31	60.73	18.66	0.41	1	1007	0.10	23	100.53	27.05	3.35	1	1017	1.09
p10100-13	12	45.64	29.92	0.09	0	–	∞	5	39.08	23.69	0.62	1	1037	1.07
p10100-15	192	316.66	19.91	0.77	1	1101	0.92	265	542.05	26.67	1.45	1	1121	2.75
p10100-17	76	91.43	34.64	0.10	0	–	∞	76	129.11	27.55	5.73	1	1572	52.03
p10100-19	21	44.80	24.54	0.52	0	–	∞	22	68.46	30.81	8.60	1	1052	2.04
p25100-11	3090	540.57	77.04	0.09	0	–	∞	3090	722.90	104.87	3.25	0	–	∞
p25100-13	13448	1800.01	62.23	0.11	0	–	∞	9912	1800.01	87.56	7.26	0	–	∞
p25100-15	608	150.74	61.34	0.17	0	–	∞	608	205.02	85.51	4.27	0	–	∞
p25100-17	51	25.35	21.03	0.08	0	–	∞	51	37.24	31.56	3.97	0	–	∞
p25100-19	5854	1111.14	61.47	0.10	0	–	∞	6250	1575.42	13.27	4.99	1	636	15.43
p33100-11	605	81.42	48.60	0.10	0	–	∞	565	104.27	12.26	3.56	1	483	16.67
p33100-13	165	35.78	21.83	0.10	0	–	∞	101	33.39	9.63	2.31	1	481	7.85
p33100-15	584	102.58	36.11	0.10	0	–	∞	584	139.61	51.59	4.05	0	–	∞
p33100-17	332	50.76	13.43	0.09	0	–	∞	312	73.40	8.73	0.76	1	460	6.73
p33100-19	3188	497.58	52.06	0.09	0	–	∞	3188	670.81	75.83	7.95	0	–	∞
p40100-11	2698	350.09	22.32	0.08	0	–	∞	3426	609.08	9.95	3.36	1	424	2.17
p40100-13	916	99.52	14.70	0.09	0	–	∞	996	141.15	8.89	1.02	1	423	2.67
p40100-15	1031	165.64	34.60	0.06	0	–	∞	904	203.08	13.64	3.95	1	523	5.44
p40100-17	1975	385.47	96.50	0.10	0	–	∞	1975	520.68	134.93	7.16	0	–	∞
p40100-19	2822	487.85	159.06	0.09	0	–	∞	2441	517.00	16.46	9.24	1	567	26.00
p15150-21	77	156.09	114.99	0.25	0	–	∞	77	221.84	73.11	13.50	1	1420	10.25
p15150-23	13	81.06	78.69	0.39	0	–	∞	13	119.78	116.61	13.82	0	–	∞

Continue next page

Instance	RENS							Extreme Points Crossover						
	Nodes	Time	First	HTime	NSols	Primal	HGap	Nodes	Time	First	HTime	NSols	Primal	HGap
p15150-25	124	230.38	50.36	0.78	1	1204	0.92	124	321.49	73.26	8.94	1	1438	20.54
p15150-27	714	1800.13	761.99	3.13	0	–	∞	521	1800.16	1016.50	15.52	0	–	∞
p15150-29	1	48.91	48.70	0.00	0	–	∞	1	62.23	61.91	0.00	0	–	∞
p36150-21	5375	1800.03	457.87	0.25	0	–	∞	3959	1800.03	36.08	12.57	1	769	0.00
p36150-23	3162	1800.00	1753.10	0.26	0	–	∞	2272	1800.11	–	24.07	0	–	∞
p36150-25	85	49.01	42.61	0.25	0	–	∞	98	70.80	33.58	3.90	1	673	4.67
p36150-27	3258	1800.04	195.55	0.26	0	–	∞	2276	1800.03	275.50	16.27	0	–	∞
p36150-29	228	111.16	88.85	0.26	0	–	∞	111	85.72	32.42	6.83	1	677	1.96
p37150-21	393	208.59	169.63	0.25	0	–	∞	393	285.35	37.06	9.40	1	1012	48.60
p37150-23	3683	1800.02	995.82	0.26	0	–	∞	2525	1800.07	1343.84	25.48	0	–	∞
p37150-25	73	47.20	38.28	0.23	0	–	∞	66	58.68	30.29	2.64	1	642	2.07
p37150-27	2954	1800.01	1523.16	0.26	0	–	∞	1986	1800.05	–	20.40	0	–	∞
p37150-29	1131	342.42	49.57	0.25	0	–	∞	682	301.82	31.26	4.06	1	666	2.62
p50150-21	6494	1800.01	520.57	0.25	0	–	∞	4669	1800.03	719.76	35.87	2	651	0.00
p50150-23	4762	1510.24	601.31	0.23	0	–	∞	4016	1800.04	824.72	17.35	0	–	∞
p50150-25	64	38.50	21.78	0.24	0	–	∞	53	47.13	26.24	2.70	1	499	2.25
p50150-27	3050	1141.26	108.70	0.26	0	–	∞	3050	1550.51	158.69	16.34	0	–	∞
p50150-29	7948	1800.02	181.67	0.25	0	–	∞	5721	1800.03	27.63	9.79	3	555	0.00
p60150-21	281	112.52	59.03	0.25	0	–	∞	226	136.44	37.62	12.17	1	613	11.05
p60150-23	4772	1800.01	433.97	0.25	0	–	∞	3826	1800.03	40.94	18.60	1	570	2.70
p60150-25	9813	1800.02	120.46	0.24	0	–	∞	7352	1800.02	32.26	19.82	1	483	10.78
p60150-27	3383	1800.03	580.46	0.26	0	–	∞	2435	1800.04	801.15	47.03	0	–	∞
p60150-29	356	96.06	25.91	0.25	0	–	∞	356	139.08	36.47	12.88	1	522	5.67
p20200-32	609	1800.39	512.25	0.46	0	–	∞	407	1800.26	694.39	22.83	0	–	∞
p20200-34	638	1805.76	418.56	0.47	0	–	∞	443	1800.11	562.49	12.38	0	–	∞
p20200-36	37	206.42	108.09	0.56	1	1384	0.14	37	271.90	142.53	8.49	1	1384	0.14
p50200-36	1271	673.52	236.80	0.45	0	–	∞	1271	898.31	323.24	16.75	0	–	∞
p50200-37	2727	1054.51	163.89	0.46	0	–	∞	2727	1365.92	223.25	15.42	0	–	∞
p50200-39	1410	692.47	108.72	0.47	0	–	∞	1256	829.11	78.70	5.66	1	733	1.52
p66200-31	1388	414.33	68.53	0.45	0	–	∞	1266	524.22	62.27	7.98	1	598	4.00
p66200-36	4705	1800.04	189.63	0.46	0	–	∞	3319	1800.04	270.86	31.07	0	–	∞
p66200-38	3464	1800.01	1059.16	0.45	0	–	∞	3134	1800.01	65.80	10.48	1	630	0.00
p80200-33	4727	1800.06	383.78	0.45	0	–	∞	3675	1800.01	58.33	10.63	2	560	0.00
p80200-34	2102	1800.01	369.60	0.46	0	–	∞	1534	1800.04	520.89	48.51	0	–	∞
p80200-38	1146	527.11	103.41	0.43	0	–	∞	1146	715.47	158.42	22.71	0	–	∞
Total	127334	12:33:48		0:00:19	7			104944	13:40:08		0:11:21	55		
Geom. Mean		140.59		1.01					173.21		4.75			

Table B.13.: RENS vs. Extreme Points Crossover on the test set CPMPS

	RENS							Extreme Points Crossover						
Instance	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**
new1_2_32	79	2579.84	230.01	0.09	1	35765	0.02	79	2705.03	231.29	0.99	1	35770	0.00
new1_2_64	21	1560.37	420.25	0.07	1	35765	0.02	25	1738.37	420.47	0.47	1	35770	0.00
new1_6_32	34	1367.45	273.23	0.12	1	55487	0.01	23	1047.66	273.56	1.01	1	55471	0.04
new1_6_64	23	2180.91	395.83	0.10	1	55487	0.01	23	2187.94	398.12	1.07	1	55487	0.01
new1_10_32	27	1685.70	857.08	0.20	1	75717	0.05	33	1888.08	860.59	2.41	1	75730	0.03
new1_10_64	7	2169.72	1402.79	0.15	1	75733	0.03	7	2173.76	1406.51	1.66	1	75733	0.03
new2_1_32	11	223.17	141.81	0.07	1	3109	0.00	11	223.65	142.25	0.36	1	3109	0.00
new2_1_64	7	415.91	234.33	0.06	1	3109	0.00	7	418.27	235.92	0.33	1	3109	0.00
new2_3_32	27	572.01	197.38	0.10	1	4202	0.00	27	572.70	198.05	0.70	1	4202	0.00
new2_3_64	1	338.55	338.53	0.08	1	4202	0.00	1	337.72	337.71	0.97	1	4202	0.00
new2_7_32	102	3600.01	411.88	0.15	1	6146	0.00	99	3600.01	413.93	1.81	1	6147	0.00
new2_7_64	13	1507.61	740.56	0.12	1	6146	0.03	13	1505.96	740.49	1.24	1	6146	0.03
new3_1_32	19	2429.22	845.81	0.12	1	71998	0.00	19	2429.90	847.49	1.30	1	71998	0.00
new3_1_64	1	1763.45	1763.43	0.13	1	71998	0.00	1	1769.34	1769.31	0.83	1	71998	0.00
new3_4_32	31	3600.00	1341.37	0.18	1	107468	0.02	28	3600.00	1347.08	2.85	1	107468	0.02
new3_4_64	5	3599.78	2909.56	0.15	1	107491	0.00	5	3599.80	2927.59	2.80	1	107491	0.00
new3_5_32	25	3600.01	1822.40	0.20	1	120493	0.01	29	3600.03	1827.75	3.25	1	120504	0.00
new3_5_64	1	3600.04	–	0.00	0	–	∞	1	3600.04	–	0.00	0	–	∞
new4_1_32	1	41.66	41.39	0.00	0	–	∞	1	41.80	41.53	0.00	0	–	∞
new4_1_64	1	52.75	52.59	0.00	0	–	∞	1	53.06	52.90	0.00	0	–	∞
new4_5_32	13	166.19	54.14	0.13	1	37016	0.00	13	166.49	54.58	0.52	1	37016	0.00
new4_5_64	13	178.13	65.37	0.10	1	37016	0.00	13	178.47	65.68	0.43	1	37016	0.00
new4_10_32	3	150.99	137.53	0.20	1	54842	0.00	3	152.09	138.64	1.37	1	54842	0.00
new4_10_64	3	277.27	227.72	0.15	1	54842	0.00	3	278.90	229.30	0.99	1	54837	0.01
new5_2_32	13	603.35	352.97	0.09	1	47896	0.04	13	618.35	354.27	0.77	1	47903	0.02
new5_2_64	1	748.97	748.95	0.08	1	47914	0.00	1	750.61	750.58	0.63	1	47914	0.00
new5_7_32	94	3600.00	478.19	0.17	1	81934	0.00	105	3600.00	479.99	2.26	1	81940	0.00
new5_7_64	11	1342.78	614.62	0.16	1	81969	0.02	11	1341.80	614.79	1.74	1	81982	0.00
new5_10_32	71	3600.04	679.60	0.25	1	99978	0.05	73	3600.01	687.25	3.64	1	99974	0.06
new5_10_64	33	3600.02	1352.14	0.21	1	99983	0.05	34	3600.01	1351.60	2.93	1	100009	0.02
new6_1_32	23	2092.24	572.90	0.11	1	71399	0.04	39	2949.94	571.46	1.14	1	71407	0.03
new6_1_64	11	2992.91	1278.09	0.10	1	71402	0.03	11	3002.07	1284.18	0.96	1	71402	0.03
new6_2_32	7	1936.86	1134.01	0.14	1	82941	0.00	7	1935.14	1133.01	1.53	1	82941	0.00
new6_2_64	3	2381.58	2031.95	0.10	1	82942	0.00	3	2383.47	2033.39	1.24	1	82942	0.00
new6_5_32	33	3600.03	1321.26	0.20	1	119184	0.04	30	3600.04	1324.31	3.35	1	119206	0.02
new6_5_64	11	3599.69	1831.67	0.18	1	119211	0.02	13	3599.87	1838.43	2.57	1	119220	0.01
new7_1_32	102	3022.85	249.83	0.09	1	42680	0.01	102	3041.11	251.94	1.26	1	42680	0.01
new7_1_64	3	533.58	455.15	0.08	1	42685	0.00	3	536.31	457.62	0.65	1	42685	0.00
new7_4_32	29	1740.43	596.52	0.15	1	60698	0.03	29	1744.31	599.17	2.50	1	60698	0.03
new7_4_64	29	3600.01	2207.40	0.14	1	60698	0.00	28	3600.01	2207.95	2.69	1	60698	0.00
new7_9_32	19	2379.57	1343.59	0.29	1	92370	0.01	19	2386.03	1350.70	6.41	1	92369	0.01
new7_9_64	19	3600.00	2985.98	0.23	1	92378	0.00	19	3600.00	2989.78	3.62	1	92378	0.00
Total	980	22:57:15		0:00:05	39			1005	23:15:58		0:01:07	39		
Geom. Mean		1048.62		1.00					1059.19		1.52			

Table B.14.: RENS vs. Extreme Points Crossover on the test set RAPS

Instance	Standard Feasibility Pump							Lagrange Feasibility Pump						
	Nodes	Time	First	HTime	NSols	Primal	HGap	Nodes	Time	First	HTime	NSols	Primal	HGap
1-FullIns_3	11	2.16	1.97	0.38	2	6	50.00	11	1.89	1.74	0.17	1	6	50.00
2-FullIns_3	21	10.78	9.08	6.08	1	7	40.00	21	5.18	3.58	0.71	1	7	40.00
3-FullIns_3	45	16.78	16.78	6.98	0	–	∞	45	11.80	11.80	2.25	0	–	∞
4-FullIns_3	78	35.36	8.38	17.87	5	10	42.86	78	22.25	22.23	5.23	0	–	∞
5-FullIns_3	116	48.65	48.63	19.04	0	–	∞	116	40.18	40.16	11.30	0	–	∞
anna	1	1.89	1.57	0.33	1	11	0.00	1	1.93	1.58	0.37	1	11	0.00
david	1	1.61	1.40	0.72	1	11	0.00	19	4.73	4.72	2.84	0	–	∞
games120	84	28.76	28.74	2.46	0	–	∞	1	2.61	2.24	1.17	1	9	0.00
homer	350	1800.01	–	50.17	0	–	∞	357	1800.53	–	46.13	0	–	∞
huck	38	4.42	4.41	1.87	0	–	∞	38	4.67	4.66	2.22	0	–	∞
jean	31	17.92	17.91	15.84	0	–	∞	31	4.59	4.58	2.62	0	–	∞
miles1000	1	34.91	26.16	0.00	0	–	∞	1	33.99	25.57	0.00	0	–	∞
miles1500	1	116.82	68.76	0.00	0	–	∞	1	111.75	64.98	0.00	0	–	∞
miles250	85	21.03	21.01	2.15	0	–	∞	85	21.38	21.36	3.12	0	–	∞
miles500	54	25.05	25.00	9.76	0	–	∞	54	23.11	23.07	8.38	0	–	∞
miles750	1	10.88	8.23	0.00	0	–	∞	1	10.50	7.94	0.00	0	–	∞
mulsol.i.1	12	324.72	324.51	226.63	0	–	∞	12	522.84	522.64	427.84	0	–	∞
mulsol.i.2	102	687.09	342.36	265.14	1	33	6.45	102	793.56	793.40	384.08	0	–	∞
mulsol.i.3	100	769.13	355.04	270.76	1	32	3.23	100	817.94	419.25	337.44	1	33	6.45
mulsol.i.4	100	783.18	783.01	219.19	0	–	∞	100	803.23	803.06	259.70	0	–	∞
mulsol.i.5	70	611.09	610.94	258.47	0	–	∞	70	648.23	648.07	306.52	0	–	∞
myciel3	5	0.28	0.22	0.10	1	4	0.00	7	0.36	0.36	0.19	0	–	∞
myciel4	629	135.66	3.58	0.37	2	7	40.00	595	121.18	14.85	0.50	0	–	∞
queen5_5	1	0.30	0.30	0.00	0	–	∞	1	0.32	0.31	0.00	0	–	∞
queen6_6	13	17.45	5.44	0.86	1	10	42.86	13	17.64	13.05	1.76	0	–	∞
queen7_7	1	22.14	22.10	0.00	0	–	∞	1	21.25	21.22	0.00	0	–	∞
queen8_12	186	1151.94	269.68	7.53	0	–	∞	186	1105.87	261.87	10.99	0	–	∞
zeroin.i.1	86	697.89	697.65	413.93	0	–	∞	86	649.50	649.25	374.58	0	–	∞
zeroin.i.2	137	644.08	643.92	107.36	0	–	∞	137	622.67	622.51	104.83	0	–	∞
zeroin.i.3	102	513.40	513.24	83.32	0	–	∞	102	503.23	503.08	86.37	0	–	∞
Total	2462	2:22:15		0:33:07	16			2372	2:25:28		0:39:41	5		
Geom. Mean	23	47.68		8.88				22	41.46		7.91			

Table B.15.: Feasibility Pump Heuristics on the test set ColoringEasy

	Standard Feasibility Pump							Lagrange Feasibility Pump						
Instance	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**	**Nodes**	**Time**	**First**	**HTime**	**NSols**	**Primal**	**HGap**
new1_2_32	90	2849.51	1307.51	2.38	0	–	∞	90	2806.93	1287.43	2.47	0	–	∞
new1_2_64	18	1552.10	723.92	2.28	0	–	∞	18	1531.74	714.44	3.22	0	–	∞
new1_6_32	27	1325.96	413.80	4.91	0	–	∞	27	1308.11	408.81	6.11	0	–	∞
new1_6_64	24	2595.82	790.50	4.11	0	–	∞	24	2561.27	780.35	4.06	0	–	∞
new1_10_32	27	1727.57	1066.71	8.41	0	–	∞	27	1706.10	1052.45	8.87	0	–	∞
new1_10_64	7	2277.17	1742.78	7.11	0	–	∞	7	2242.62	1716.26	8.52	0	–	∞
new2_1_32	11	231.18	185.95	2.01	0	–	∞	11	228.38	183.77	2.47	0	–	∞
new2_1_64	7	428.91	242.73	1.73	1	2874	7.56	7	422.59	277.55	1.91	0	–	∞
new2_3_32	17	399.60	240.81	2.88	0	–	∞	17	395.41	238.71	4.42	0	–	∞
new2_3_64	3	370.48	370.47	2.47	0	–	∞	3	367.17	367.16	3.49	0	–	∞
new2_7_32	114	3600.02	2453.83	5.42	0	–	∞	113	3600.02	2426.86	9.02	0	–	∞
new2_7_64	13	1548.00	972.98	4.90	0	–	∞	13	1527.01	959.50	4.69	0	–	∞
new3_1_32	42	3599.84	2975.97	4.20	0	–	∞	34	3564.18	2928.47	4.03	0	–	∞
new3_1_64	4	2771.10	2771.05	2.62	0	–	∞	4	2723.39	2723.34	2.60	0	–	∞
new3_4_32	32	3600.01	1387.55	7.86	1	99019	7.88	34	3600.01	3055.38	7.50	0	–	∞
new3_4_64	7	3599.81	2984.23	4.30	1	106960	0.49	7	3599.83	3599.71	4.37	0	–	∞
new3_5_32	22	3600.02	3599.68	9.56	0	–	∞	22	3600.02	3599.68	11.36	0	–	∞
new3_5_64	1	3600.05	0.03	0.03	1	0	0.00	1	3600.06	0.03	0.04	1	0	0.00
new4_1_32	1	42.24	41.99	0.00	0	–	∞	1	41.57	41.33	0.00	0	–	∞
new4_1_64	1	53.58	53.42	0.00	0	–	∞	1	52.94	52.78	0.00	0	–	∞
new4_5_32	13	208.47	112.85	3.76	0	–	∞	13	205.19	111.09	3.57	0	–	∞
new4_5_64	13	173.96	68.36	2.87	1	33577	9.29	13	170.77	65.42	1.56	1	0	100.00
new4_10_32	3	159.36	152.47	5.91	0	–	∞	3	157.34	150.64	6.12	0	–	∞
new4_10_64	3	287.68	247.29	3.88	0	–	∞	3	281.34	228.73	3.02	1	0	100.00
new5_2_32	14	652.84	479.54	3.10	0	–	∞	14	642.16	471.55	3.25	0	–	∞
new5_2_64	4	1096.72	1096.69	2.69	0	–	∞	4	1081.74	1081.71	2.54	0	–	∞
new5_7_32	98	3600.01	1782.03	7.07	0	–	∞	100	3600.00	1754.12	8.66	0	–	∞
new5_7_64	12	1413.62	955.47	4.20	0	–	∞	12	1393.03	941.25	4.26	0	–	∞
new5_10_32	72	3600.00	2425.93	11.25	0	–	∞	71	3600.00	2394.40	14.43	0	–	∞
new5_10_64	43	3600.02	2677.87	8.58	0	–	∞	43	3600.00	2634.69	8.46	0	–	∞
new6_1_32	40	3251.99	1692.99	3.52	0	–	∞	40	3200.59	1665.78	3.57	0	–	∞
new6_1_64	20	3599.89	2582.76	2.65	0	–	∞	20	3599.87	2541.58	2.66	0	–	∞
new6_2_32	7	1943.93	1167.28	5.62	1	74525	10.15	7	1941.68	1631.12	6.74	0	–	∞
new6_2_64	3	2439.07	2079.53	2.92	1	82832	0.13	3	2537.94	2456.48	2.91	0	–	∞
new6_5_32	29	3600.05	1366.97	9.63	1	109294	8.31	29	3600.05	3599.76	10.36	0	–	∞
new6_5_64	12	3599.78	1883.75	4.58	1	0	100.00	14	3599.81	1924.46	4.55	1	0	100.00
new7_1_32	59	1943.43	1121.12	2.77	0	–	∞	59	1924.13	1109.04	2.88	0	–	∞
new7_1_64	3	551.29	470.47	2.43	1	38794	9.12	3	563.09	510.03	2.50	0	–	∞
new7_4_32	30	1634.04	1208.04	5.08	0	–	∞	30	1672.79	1251.84	6.75	0	–	∞
new7_4_64	33	3600.00	3599.75	4.69	0	–	∞	31	3600.00	3599.78	6.36	0	–	∞
new7_9_32	17	2544.99	1922.68	10.22	0	–	∞	17	2491.07	1886.49	13.70	0	–	∞
new7_9_64	19	3599.96	3079.14	8.03	1	83419	9.70	13	3600.00	3599.57	8.58	0	–	∞
Total	1015	24:07:54		0:03:12	11			1003	24:02:21		0:03:36	4		
Geom. Mean	13	1299.76		3.92				13	1291.22		4.24			

Table B.16.: Feasibility Pump Heuristics on the test set RAPS

Instance	Nodes	Time	First	HTime	SRou	Rou	Zi	Shi	Primal	HGap
				Rounding Heuristics						
N1C1W1_A	4	0.62	0.61	0.00	0	1	0	1	25	0.00
N1C1W1_K	5	0.45	0.44	0.00	0	1	0	1	26	0.00
N1C1W2_B	4	0.69	0.67	0.00	0	0	0	1	33	10.00
N1C1W2_L	1	0.77	0.75	0.00	0	0	0	0	–	∞
N1C1W4_C	1	0.77	0.76	0.00	0	0	0	0	–	∞
N1C1W4_M	1	0.69	0.69	0.00	0	1	0	1	41	0.00
N1C2W1_D	1	0.50	0.49	0.00	0	0	0	0	–	∞
N1C2W1_N	13	1.11	1.09	0.00	0	1	0	1	21	0.00
N1C2W2_E	4	0.59	0.58	0.00	0	0	1	1	33	0.00
N1C2W2_O	2	0.49	0.49	0.01	0	0	0	1	30	3.45
N1C2W4_F	1	0.56	0.55	0.00	0	0	0	0	–	∞
N1C2W4_P	1	0.71	0.69	0.00	0	0	0	0	–	∞
N1C3W1_G	10	0.82	0.80	0.00	0	0	0	1	19	26.67
N1C3W1_Q	12	1.02	0.99	0.01	0	0	0	2	21	5.00
N1C3W2_H	2	1.09	1.08	0.00	0	1	0	1	23	0.00
N1C3W2_R	23	1.23	1.12	0.01	0	0	0	2	20	5.26
N1C3W4_I	4	0.66	0.65	0.00	0	1	0	1	23	0.00
N1C3W4_S	6	0.62	0.61	0.00	0	1	0	1	22	0.00
N2C1W1_A	21	3.92	3.73	0.06	0	0	0	2	50	4.17
N2C1W1_K	11	3.08	2.90	0.04	0	0	1	1	55	0.00
N2C1W2_B	4	4.10	4.05	0.00	0	0	0	1	67	9.84
N2C1W2_L	2	3.35	3.33	0.01	0	1	0	1	62	0.00
N2C1W4_C	1	3.25	3.16	0.00	0	0	0	0	–	∞
N2C1W4_M	1	3.43	3.43	0.01	0	1	0	1	72	0.00
N2C2W1_D	22	7.73	7.53	0.05	0	0	0	3	49	16.67
N2C2W1_N	20	5.93	5.75	0.02	0	0	0	1	51	18.60
N2C2W2_E	1	2.89	2.89	0.00	0	1	0	1	54	0.00
N2C2W2_O	1	4.17	4.17	0.01	0	1	0	0	50	0.00
N2C2W4_F	1	5.03	4.96	0.00	0	0	0	0	–	∞
N2C2W4_P	2	3.88	3.86	0.02	0	1	0	1	60	0.00
N2C3W1_G	32	3.92	3.48	0.11	0	0	0	4	35	6.06
N2C3W1_Q	41	4.01	3.44	0.10	0	0	0	4	39	14.71
N2C3W2_H	30	4.37	4.09	0.06	0	0	0	3	43	13.16
N2C3W2_R	29	5.08	4.76	0.04	0	1	0	3	40	0.00
N2C3W4_I	20	6.93	6.76	0.08	0	1	0	2	44	0.00
N2C3W4_S	27	10.66	10.42	0.05	0	1	0	1	42	0.00
N3C1W1_A	13	14.08	13.08	0.31	0	1	0	2	105	0.00
N3C1W1_K	14	14.64	13.76	0.28	0	0	0	1	103	0.98
N3C1W2_B	1	11.08	10.49	0.00	0	0	0	0	–	∞
N3C1W2_L	1	14.91	14.35	0.00	0	0	0	0	–	∞
N3C1W4_C	1	16.92	16.89	0.06	0	1	0	0	146	0.00
N3C1W4_M	1	17.66	17.11	0.00	0	0	0	0	–	∞
N3C2W1_D	28	12.61	10.50	0.45	0	1	0	1	85	0.00
N3C2W1_N	1	11.44	11.43	0.03	0	1	0	0	91	0.00
N3C2W2_E	1	13.67	13.64	0.06	0	1	0	1	116	0.00
N3C2W2_O	1	13.51	13.47	0.06	0	1	0	1	107	0.00
N3C2W4_F	1	14.72	14.68	0.06	0	1	0	1	115	0.00
N3C2W4_P	1	18.70	18.68	0.05	0	1	0	0	122	0.00
N3C3W1_G	83	18.89	14.40	1.51	0	0	0	4	70	7.69
N3C3W1_Q	64	19.70	14.48	0.99	0	0	0	4	81	10.96
N3C3W2_H	38	47.03	37.59	0.73	0	0	0	3	95	15.85
N3C3W2_R	41	18.64	16.10	0.70	0	0	0	4	90	13.92
N3C3W4_I	30	19.32	17.47	0.58	0	0	0	3	99	7.61
N3C3W4_S	41	37.58	35.35	0.75	0	0	0	2	94	11.90
Total	722	0:07:14		0:00:07	0	23	2	71		
Geom. Mean		4.29		1.01						

Table B.17.: Rounding Heuristics on the test set BINDATA1S

Instance	Nodes	Time	First	Rounding Heuristics HTime	SRou	Rou	Zi	Shi	Primal	HGap
1-FullIns_3	8	1.39	1.33	0.00	0	1	0	1	4	0.00
1-FullIns_4	198	1800.08	913.38	0.41	0	0	0	2	6	0.00
2-FullIns_3	21	3.53	2.29	0.02	0	0	0	1	6	20.00
2-Insertions_3	1110	1800.02	6.89	0.26	0	0	0	1	4	0.00
3-FullIns_3	45	7.29	2.44	0.06	0	0	0	2	7	16.67
4-FullIns_3	62	10.29	4.01	0.22	0	1	0	1	7	0.00
4-FullIns_4	1	1805.08	1804.75	0.40	0	0	0	1	12	0.00
5-FullIns_3	98	19.26	5.65	0.44	0	1	0	3	8	0.00
anna	46	5.60	1.08	0.15	0	1	0	2	11	0.00
ash331GPIA	1	1800.19	1800.13	0.09	0	0	0	1	6	0.00
david	12	1.07	0.63	0.04	0	1	0	2	11	0.00
DSJC125.9	1	1823.86	–	0.35	0	0	0	0	–	∞
DSJR500.1	89	1800.22	102.93	4.78	0	0	0	2	15	0.00
fpsol2.i.1	285	1792.71	579.11	74.97	0	0	0	3	70	0.00
fpsol2.i.2	86	1800.02	358.65	18.53	0	0	0	3	34	0.00
fpsol2.i.3	112	1800.03	355.35	15.33	0	0	0	2	35	0.00
games120	71	15.30	1.05	0.28	0	1	0	2	9	0.00
homer	418	1718.47	14.27	10.35	0	0	0	2	15	15.38
huck	30	1.57	0.45	0.07	0	1	0	2	11	0.00
inithx.i.1	1	1862.54	–	10.25	0	0	0	0	–	∞
inithx.i.2	1	1972.02	1971.34	1.14	0	0	0	1	40	0.00
inithx.i.3	1	1966.15	1965.59	1.00	0	0	0	1	38	0.00
jean	14	0.97	0.40	0.03	0	1	0	1	10	0.00
le450_25a	144	1800.10	247.89	23.24	0	0	0	2	34	0.00
le450_25b	138	1800.13	236.12	22.92	0	0	0	3	32	0.00
miles1000	1	24.82	18.11	0.00	0	0	0	0	–	∞
miles1500	1	77.57	45.85	0.00	0	0	0	0	–	∞
miles250	77	12.83	0.98	0.23	0	1	0	3	8	0.00
miles500	39	9.59	3.11	0.72	0	1	0	2	20	0.00
miles750	1	7.76	5.59	0.00	0	0	0	0	–	∞
mulsol.i.1	11	67.78	59.86	1.57	0	1	0	1	49	0.00
mulsol.i.2	94	287.31	61.88	8.25	0	1	0	4	31	0.00
mulsol.i.3	84	332.31	65.11	7.56	0	1	0	5	31	0.00
mulsol.i.4	90	377.73	72.67	8.04	0	1	0	3	31	0.00
mulsol.i.5	63	236.29	72.21	5.67	0	1	0	2	31	0.00
myciel3	5	0.12	0.08	0.00	0	0	0	1	4	0.00
myciel4	505	80.81	2.56	0.09	0	0	0	1	5	0.00
qg.order30	1	3384.25	3383.48	1.57	0	0	0	1	39	0.00
qg.order40	1	9278.29	–	4.89	0	0	0	0	–	∞
queen10_10	77	1800.01	–	0.74	0	0	0	0	–	∞
queen5_5	1	0.23	0.23	0.00	0	0	0	0	–	∞
queen6_6	13	12.90	9.12	0.01	0	0	0	0	–	∞
queen7_7	1	16.66	16.63	0.00	0	0	0	0	–	∞
queen8_12	186	804.61	87.37	1.77	0	0	0	2	17	41.67
queen8_8	309	1149.74	596.63	1.09	0	0	0	1	13	44.44
queen9_9	230	1800.05	858.95	1.10	0	0	0	2	14	0.00
school1	1	1792.87	–	0.00	0	0	0	0	–	∞
school1_nsh	1	1794.57	–	0.00	0	0	0	0	–	∞
wap05a	1	5544.10	–	5.93	0	0	0	0	–	∞
will199GPIA	1	1800.89	–	0.27	0	0	0	0	–	∞
zeroin.i.1	63	167.62	57.43	9.03	0	1	0	1	49	0.00
zeroin.i.2	128	370.38	50.58	9.83	0	1	0	5	30	0.00
zeroin.i.3	102	310.24	43.55	8.03	0	0	0	4	31	3.33
Total	5080	16:22:30		0:04:21	0	17	0	79		
Geom. Mean		168.04		2.21						

Table B.18.: Rounding Heuristics on the test set COLORING

Instance	Nodes	Time	First	HTime	SRou	Rou	Zi	Shi	Primal	HGap
				Rounding Heuristics						
new1_2_32	78	2058.85	188.24	0.16	0	9	4	1	35649	0.34
new1_2_64	18	1250.17	347.03	0.03	0	2	3	0	35675	0.27
new1_6_32	27	1061.46	222.86	0.08	0	2	2	1	55442	0.10
new1_6_64	33	2606.27	326.46	0.12	0	2	2	1	55442	0.10
new1_10_32	27	1375.16	700.26	0.16	0	5	4	1	75621	0.18
new1_10_64	7	1819.11	1155.15	0.03	0	2	1	1	75733	0.03
new2_1_32	11	183.69	116.83	0.04	0	2	3	1	3103	0.19
new2_1_64	7	342.16	193.48	0.00	0	0	2	0	3103	0.19
new2_3_32	17	317.25	161.83	0.03	0	2	4	0	4195	0.17
new2_3_64	3	295.94	279.21	0.01	0	0	2	0	4173	0.69
new2_7_32	115	3587.33	337.05	0.54	0	14	6	1	6132	0.26
new2_7_64	13	1237.86	608.19	0.03	0	0	2	1	6126	0.36
new3_1_32	34	2854.55	691.29	0.13	0	5	7	0	71903	0.13
new3_1_64	4	2216.00	1445.30	0.00	0	1	1	0	71719	0.39
new3_4_32	33	3600.01	1097.63	0.14	0	5	5	1	107268	0.21
new3_4_64	7	3599.94	2390.62	0.02	0	2	2	0	107491	0.00
new3_5_32	29	3600.01	1489.36	0.21	0	9	4	1	120476	0.02
new3_5_64	17	3599.99	3067.29	0.04	0	3	2	1	120447	0.05
new4_1_32	1	33.46	33.25	0.00	0	0	0	0	–	∞
new4_1_64	1	43.05	42.92	0.00	0	0	0	0	–	∞
new4_5_32	13	164.29	43.47	0.03	0	1	2	0	36980	0.10
new4_5_64	13	137.26	52.91	0.00	0	1	2	0	36980	0.10
new4_10_32	3	121.83	110.94	0.02	0	0	1	0	54784	0.11
new4_10_64	3	225.50	185.14	0.01	0	0	1	0	54784	0.11
new5_2_32	14	517.28	288.20	0.01	0	3	3	0	47811	0.21
new5_2_64	4	876.50	614.81	0.01	0	1	1	0	47789	0.26
new5_7_32	121	3600.01	389.36	0.77	0	11	8	1	81857	0.13
new5_7_64	11	1098.08	503.17	0.04	0	3	3	1	81895	0.11
new5_10_32	90	3600.00	553.40	0.55	0	3	7	0	99940	0.09
new5_10_64	39	3600.01	1108.51	0.18	0	0	4	0	99930	0.10
new6_1_32	40	2598.58	466.09	0.11	0	2	3	0	71298	0.18
new6_1_64	16	2943.65	1051.43	0.04	0	1	2	0	71309	0.16
new6_2_32	7	1548.56	925.66	0.01	0	2	2	1	82767	0.21
new6_2_64	3	1945.02	1659.32	0.01	0	1	1	0	82874	0.08
new6_5_32	35	3600.01	1077.39	0.24	0	5	5	0	119196	0.00
new6_5_64	11	3436.97	1507.97	0.05	0	3	2	1	119192	0.03
new7_1_32	68	1717.22	203.80	0.17	0	3	9	0	42490	0.46
new7_1_64	3	440.47	375.67	0.01	0	1	1	1	42562	0.29
new7_4_32	33	1612.81	485.80	0.17	0	6	5	1	60680	0.06
new7_4_64	21	3600.00	1793.17	0.08	0	5	4	1	60547	0.28
new7_9_32	17	2003.44	1091.33	0.12	0	2	4	0	92275	0.11
new7_9_64	11	3599.89	2417.79	0.04	0	0	3	0	92160	0.24
Total	1058	21:51:09		0:00:04	0	119	129	18		
Geom. Mean	13	1113.91		1.00						

Table B.19.: Rounding Heuristics on the test set RapS

Instance	Nodes	Time	First	HTime	Diving Heuristics CoD	FrD	GuiD	IntD	LinD	PsCD	VecD	Primal	HGap
p550-1	14	12.46	6.78	0.29	0	0	0	0	0	0	0	–	∞
p550-3	3	8.42	7.82	0.00	0	0	0	0	0	0	0	–	∞
p550-5	1	8.92	8.90	0.00	0	0	0	0	0	0	0	–	∞
p550-7	21	25.44	9.58	0.00	0	0	0	0	0	0	0	–	∞
p550-9	9	11.40	6.71	0.00	0	0	0	0	0	0	0	–	∞
p1250-1	252	43.72	5.86	3.24	0	0	1	1	0	1	1	488	27.42
p1250-3	133	21.11	2.93	0.22	0	1	0	0	0	0	0	512	26.42
p1250-5	380	79.18	8.97	6.84	0	0	0	0	0	0	2	591	37.76
p1250-7	3997	592.08	3.08	1.37	0	1	0	0	0	0	0	692	55.51
p1250-9	225	70.33	15.22	3.21	0	0	0	0	1	0	2	627	43.81
p1650-1	5	2.31	1.98	0.00	0	0	0	0	0	0	0	–	∞
p1650-3	13	3.74	3.12	0.30	0	0	0	0	0	0	0	–	∞
p1650-5	77	12.74	5.41	0.74	0	0	0	0	0	0	0	–	∞
p1650-7	120	16.75	4.65	0.15	0	0	0	0	0	0	0	–	∞
p1650-9	450	57.46	6.54	0.64	0	0	0	1	0	0	0	533	42.90
p2050-1	72	9.66	2.26	0.08	0	0	0	0	0	0	0	–	∞
p2050-3	14	4.00	2.56	0.00	0	0	0	0	0	0	0	–	∞
p2050-5	6	3.98	3.48	0.00	0	0	0	0	0	0	0	–	∞
p2050-7	6	3.94	3.78	0.00	0	0	0	0	0	0	0	–	∞
p2050-9	799	105.12	10.06	0.96	0	0	0	1	0	0	0	519	25.97
p10100-11	21	167.41	24.91	1.11	0	1	0	0	0	0	0	1290	28.23
p10100-13	12	61.25	39.67	0.00	0	0	0	0	0	0	0	–	∞
p10100-15	210	522.24	78.37	25.41	0	0	0	0	0	0	1	1299	19.07
p10100-17	191	359.78	49.56	16.73	0	0	0	0	0	0	1	1149	11.12
p10100-19	20	65.78	31.48	4.35	0	0	0	0	0	0	0	–	∞
p25100-11	6318	1437.87	24.50	20.75	0	0	0	2	0	0	0	805	47.98
p25100-13	9512	1800.01	9.12	27.36	0	1	0	1	0	0	0	744	33.09
p25100-15	2583	728.79	23.91	95.67	0	0	0	1	0	0	2	855	46.66
p25100-17	100	68.84	25.52	13.66	0	0	0	0	0	0	1	849	56.64
p25100-19	6368	1800.01	21.06	27.39	0	0	0	1	0	0	0	727	28.45
p33100-11	2026	341.76	18.04	14.25	0	0	0	2	0	0	0	564	36.23
p33100-13	613	152.33	18.45	36.82	0	0	0	0	0	0	1	750	68.16
p33100-15	539	143.97	17.81	34.75	0	0	0	0	0	0	2	714	50.63
p33100-17	763	143.99	14.70	29.95	0	0	0	0	0	0	1	707	64.04
p33100-19	5691	1037.30	18.49	17.22	0	0	0	2	0	0	0	732	64.49
p40100-11	11073	1800.00	17.43	137.74	0	0	0	0	0	0	1	552	33.01
p40100-13	2056	284.41	11.55	5.37	0	0	0	0	0	0	0	–	∞
p40100-15	8979	1613.34	20.50	94.21	0	0	0	1	0	0	1	623	25.60
p40100-17	3402	836.03	27.47	22.32	0	0	0	2	0	0	0	583	32.50
p40100-19	8764	1800.00	18.42	169.86	0	0	0	1	1	0	1	610	34.66
p15150-21	104	359.41	110.73	57.82	0	0	0	0	0	0	0	–	∞
p15150-23	10	97.70	95.29	3.80	0	0	0	0	0	0	0	–	∞
p15150-25	150	402.06	120.20	28.81	0	0	1	0	1	0	1	1521	27.49
p15150-27	414	1807.93	126.04	144.06	0	0	0	0	1	0	1	1794	35.29
p15150-29	1	62.25	61.94	0.00	0	0	0	0	0	0	0	–	∞
p36150-21	2403	1800.02	54.31	368.70	0	0	0	1	0	0	1	961	0.00
p36150-23	2039	1800.03	62.93	347.11	0	0	0	0	0	0	5	968	0.00
p36150-25	161	139.65	47.23	52.74	0	0	0	0	0	0	1	1048	62.99
p36150-27	1833	1800.02	74.18	364.46	0	0	0	0	1	0	3	1252	63.66
p36150-29	314	225.60	47.76	68.78	0	0	0	1	0	0	1	861	29.67
p37150-21	920	569.87	72.13	52.99	0	0	0	2	0	0	0	1004	47.43
p37150-23	2182	1800.02	62.98	353.32	0	0	0	1	1	0	3	970	39.77
p37150-25	202	173.83	51.16	70.72	0	0	0	0	0	0	1	1126	79.01
p37150-27	1931	1800.02	54.81	315.99	0	0	0	0	0	0	5	1098	45.62
p37150-29	3313	1279.91	52.71	273.25	0	0	0	1	1	0	1	987	52.08
p50150-21	4226	1800.01	30.15	64.02	0	1	0	1	0	0	0	804	28.43
p50150-23	3723	1800.03	45.67	62.78	0	0	0	1	0	0	0	837	42.35
p50150-25	51	62.36	27.64	15.47	0	0	0	0	0	0	0	–	∞
p50150-27	2580	1800.05	52.64	321.94	0	0	0	0	0	0	2	960	62.99
p50150-29	5074	1800.06	40.90	343.67	0	0	0	1	0	0	1	692	20.98
p60150-21	1796	707.09	44.35	63.28	0	0	0	2	0	0	0	736	33.33

Continue next page

Instance	Nodes	Time	First	HTime	Diving Heuristics CoD	FrD	GuiD	IntD	LinD	PsCD	VecD	Primal	HGap
p60150-23	3514	1800.05	45.60	287.94	0	0	0	1	0	0	1	770	31.18
p60150-25	6018	1800.03	46.12	353.53	0	0	0	2	0	0	1	583	32.80
p60150-27	2464	1800.03	98.41	363.08	0	0	0	3	1	0	1	966	0.00
p60150-29	1378	392.42	38.41	19.91	0	0	0	0	0	0	0	–	∞
p20200-32	204	1800.22	282.31	168.48	0	0	0	1	0	0	3	1866	0.00
p20200-34	185	1800.00	374.74	149.27	0	0	0	1	0	0	2	1840	32.18
p20200-36	81	455.65	216.42	43.15	0	0	0	0	0	0	1	2403	73.88
p50200-36	1525	1800.00	113.91	618.60	0	0	0	1	0	0	2	1018	41.00
p50200-37	1656	1800.04	96.48	559.95	0	0	0	1	0	0	4	1034	0.00
p50200-39	1553	1800.00	112.24	586.40	0	0	0	1	2	0	3	1067	44.58
p66200-31	2882	1800.04	84.74	776.67	0	0	0	1	1	0	1	860	48.53
p66200-36	2234	1800.05	88.11	619.20	0	0	0	2	0	0	1	831	0.00
p66200-38	2029	1800.01	92.57	643.06	0	0	0	1	0	0	1	940	51.13
p80200-33	2282	1800.02	85.83	605.12	0	0	0	1	0	0	1	789	38.91
p80200-34	952	1800.00	428.47	577.60	0	0	0	1	0	0	0	1177	0.00
p80200-38	2215	1559.83	76.10	217.95	0	0	0	2	0	0	0	809	36.66
Total	140437	17:49:14		2:59:36	0	5	2	46	11	1	65		
Geom. Mean	387	267.23		24.87									

Table B.20.: Diving Heuristics on the test set CPMPS

Instance	Nodes	Time	First	HTime	Diving Heuristics CoD	FrD	GuiD	IntD	LinD	PsCD	VecD	Primal	HGap
new1_2_32	68	3170.26	395.58	16.45	31	0	0	0	0	0	0	31581	11.71
new1_2_64	18	1513.22	702.58	0.00	0	0	0	0	0	0	0	–	∞
new1_6_32	23	1125.77	397.12	0.80	0	0	0	0	0	0	0	–	∞
new1_6_64	23	3024.43	767.80	5.15	0	0	0	0	0	0	0	–	∞
new1_10_32	29	2933.55	1023.62	13.04	0	0	0	0	0	0	0	–	∞
new1_10_64	7	2218.86	1693.55	0.00	0	0	0	0	0	0	0	–	∞
new2_1_32	11	222.12	177.99	0.00	0	0	0	0	0	0	0	–	∞
new2_1_64	7	417.73	272.31	0.00	0	0	0	0	0	0	0	–	∞
new2_3_32	17	385.44	230.29	0.43	0	0	0	0	0	0	0	–	∞
new2_3_64	3	358.01	358.00	0.00	0	0	0	0	0	0	0	–	∞
new2_7_32	120	3600.00	523.15	45.63	21	0	0	0	0	0	0	5370	12.64
new2_7_64	13	1501.03	939.45	0.00	0	0	0	0	0	0	0	–	∞
new3_1_32	46	3600.01	1247.06	9.17	35	0	0	0	0	13	0	65947	8.40
new3_1_64	4	2823.68	2823.61	0.00	0	0	0	0	0	0	0	–	∞
new3_4_32	25	3600.00	2208.91	9.04	71	0	0	0	0	5	0	97646	9.13
new3_4_64	7	3599.88	3599.74	0.00	0	0	0	0	0	0	0	–	∞
new3_5_32	20	3600.00	2977.66	10.33	55	0	0	0	0	16	0	109694	8.96
new3_5_64	1	3600.09	–	0.00	0	0	0	0	0	0	0	–	∞
new4_1_32	1	41.56	41.30	0.00	0	0	0	0	0	0	0	–	∞
new4_1_64	1	52.66	52.50	0.00	0	0	0	0	0	0	0	–	∞
new4_5_32	11	227.25	107.77	3.00	0	0	0	0	0	0	0	–	∞
new4_5_64	11	152.50	83.90	2.60	0	0	0	0	0	0	0	–	∞
new4_10_32	3	149.99	143.23	0.00	0	0	0	0	0	0	0	–	∞
new4_10_64	3	276.27	236.74	0.00	0	0	0	0	0	0	0	–	∞
new5_2_32	14	629.65	460.57	0.00	0	0	0	0	0	0	0	–	∞
new5_2_64	4	1089.59	1089.55	0.00	0	0	0	0	0	0	0	–	∞
new5_7_32	68	3600.00	619.68	56.07	19	0	0	0	0	0	0	73443	10.39
new5_7_64	12	1394.30	934.37	7.41	0	0	0	0	0	0	0	–	∞
new5_10_32	68	3600.00	986.01	60.74	49	0	0	0	0	0	0	90144	9.84
new5_10_64	16	3600.00	2279.10	10.14	57	0	0	0	0	21	0	90891	0.00
new6_1_32	44	3490.38	1940.30	0.69	0	0	0	0	0	0	0	–	∞
new6_1_64	14	3599.86	2619.79	2.88	0	0	0	0	0	0	0	–	∞
new6_2_32	7	1905.38	1620.57	0.00	0	0	0	0	0	0	0	–	∞
new6_2_64	3	2463.75	2380.04	0.00	0	0	0	0	0	0	0	–	∞
new6_5_32	19	3600.00	2102.71	15.14	60	0	0	0	0	6	0	108407	0.00
new6_5_64	12	3599.76	2554.76	5.17	0	0	0	0	0	0	0	–	∞
new7_1_32	50	1402.35	317.87	8.71	21	0	0	0	0	14	0	38748	9.22
new7_1_64	3	535.16	482.62	0.00	0	0	0	0	0	0	0	–	∞
new7_4_32	30	1585.62	1170.22	0.91	0	0	0	0	0	0	0	–	∞
new7_4_64	10	3600.00	2953.90	3.01	29	0	0	0	0	0	0	54340	0.00
new7_9_32	21	3260.31	1615.42	10.16	37	0	0	0	0	0	0	82002	11.23
new7_9_64	19	3600.00	3424.67	0.00	0	0	0	0	0	0	0	–	∞
Total	886	24:39:10		0:04:56	485	0	0	0	0	75	0		
Geom. Mean	11	1303.21		2.79									

Table B.21.: Diving Heuristics on the test set RapS

Instance	Nodes	Time	First	HTime	Cross	Mut	LocB	RINS	DINS	Primal	HGap
					Improvement Heuristics						
1-FullIns_3	8	1.71	1.63	0.00	0	0	0	0	0	–	∞
2-FullIns_3	21	4.56	2.93	0.04	0	0	0	0	0	–	∞
3-FullIns_3	45	9.84	3.18	0.11	0	0	0	0	0	–	∞
4-FullIns_3	62	14.15	5.20	0.20	0	0	0	0	0	–	∞
5-FullIns_3	98	26.84	7.55	0.41	0	0	0	0	0	–	∞
anna	46	8.08	1.54	0.32	0	0	0	0	1	12	9.09
david	11	1.58	0.87	0.07	0	0	0	0	1	11	0.00
games120	41	10.36	1.47	0.25	0	0	0	0	1	9	0.00
homer	271	1262.70	18.59	5.53	0	0	0	1	1	13	0.00
huck	11	1.13	0.61	0.08	0	0	0	0	1	11	0.00
jean	14	1.30	0.53	0.04	0	0	0	0	0	–	∞
miles1000	1	33.72	25.23	0.00	0	0	0	0	0	–	∞
miles1500	1	111.66	63.67	0.00	0	0	0	0	0	–	∞
miles250	71	16.57	1.29	0.23	0	0	0	0	1	8	0.00
miles500	39	13.30	4.18	0.24	0	0	0	0	0	–	∞
miles750	1	10.49	7.89	0.00	0	0	0	0	0	–	∞
mulsol.i.1	11	98.61	86.01	1.80	0	0	0	0	0	–	∞
mulsol.i.2	94	400.55	86.55	4.32	0	0	0	0	2	32	3.23
mulsol.i.3	71	401.84	92.83	3.62	0	0	0	0	2	31	0.00
mulsol.i.4	90	524.32	102.01	3.36	0	0	0	0	0	–	∞
mulsol.i.5	63	328.40	101.43	2.32	0	0	0	0	0	–	∞
myciel3	5	0.17	0.11	0.00	0	0	0	0	0	–	∞
myciel4	505	109.69	3.12	1.25	0	0	0	0	0	–	∞
queen5_5	1	0.31	0.31	0.00	0	0	0	0	0	–	∞
queen6_6	13	15.95	11.37	0.00	0	0	0	0	0	–	∞
queen7_7	1	21.23	21.20	0.00	0	0	0	0	0	–	∞
queen8_12	186	1106.28	119.70	1.48	0	0	0	0	1	17	41.67
zeroin.i.1	63	237.89	82.40	3.72	0	0	0	0	0	–	∞
zeroin.i.2	128	516.80	70.25	5.04	0	0	0	0	2	31	3.33
zeroin.i.3	102	431.66	60.95	4.44	0	0	0	0	2	31	3.33
Total	2074	1:35:21		0:00:38	0	0	0	1	15		
Geom. Mean	23	30.59		1.50							

Table B.22.: Improvement Heuristics on the test set ColoringEasy

Instance	Nodes	Time	First	Improvement Heuristics HTime	Cross	Mut	LocB	RINS	DINS	Primal	HGap
1-FullIns_4	136	1800.03	1138.19	0.48	0	0	0	0	0	–	∞
2-Insertions_3	1003	1800.67	8.44	1.19	0	0	0	0	0	–	∞
4-FullIns_4	1	1801.13	1800.81	0.00	0	0	0	0	0	–	∞
ash331GPIA	1	1800.94	1800.82	0.00	0	0	0	0	0	–	∞
DSJC125.9	1	1822.30	–	0.00	0	0	0	0	0	–	∞
DSJR500.1	80	1801.15	134.78	1.77	0	0	0	0	0	–	∞
fpsol2.i.1	197	1729.75	709.44	23.59	0	0	0	0	1	69	0.00
fpsol2.i.2	101	1784.06	515.20	5.69	0	0	0	0	1	37	8.82
fpsol2.i.3	101	1782.38	504.31	5.21	0	0	0	0	0	–	∞
inithx.i.1	1	2315.07	–	0.00	0	0	0	0	0	–	∞
inithx.i.2	1	2044.45	2043.33	0.00	0	0	0	0	0	–	∞
inithx.i.3	1	2030.31	2029.48	0.00	0	0	0	0	0	–	∞
le450_25a	196	1800.09	336.34	10.48	0	0	0	0	1	33	0.00
le450_25b	155	1800.02	321.62	6.08	0	0	0	0	0	–	∞
qg.order30	1	3735.79	3734.95	0.00	0	0	0	0	0	–	∞
qg.order40	1	1538.40	–	0.00	0	0	0	0	0	–	∞
queen8_8	309	1490.17	772.21	2.50	0	0	0	0	1	12	33.33
queen9_9	260	1800.02	1067.38	1.42	0	0	0	0	1	14	7.69
queen10_10	141	1798.79	–	0.00	0	0	0	0	0	–	∞
school1	1	1771.82	–	0.00	0	0	0	0	0	–	∞
school1_nsh	1	1778.49	–	0.00	0	0	0	0	0	–	∞
wap05a	1	6372.10	–	0.00	0	0	0	0	0	–	∞
will199GPIA	1	1800.98	–	0.00	0	0	0	0	0	–	∞
Total	2691	13:23:18		0:00:58	0	0	0	0	5		
Geom. Mean	12	1970.02		1.74							

Table B.23.: Improvement Heuristics on the test set ColoringHard

Instance	Nodes	Time	First	HTime	Cross	Mut	LocB	RINS	DINS	Primal	HGap
				Improvement Heuristics							
p550-1	14	12.12	6.76	0.00	0	0	0	0	0	–	∞
p550-3	3	8.31	7.76	0.00	0	0	0	0	0	–	∞
p550-5	1	8.78	8.76	0.00	0	0	0	0	0	–	∞
p550-7	21	25.27	9.58	0.00	0	0	0	0	0	–	∞
p550-9	9	11.33	6.63	0.00	0	0	0	0	0	–	∞
p1250-1	137	17.25	9.40	0.34	0	0	0	0	2	383	0.00
p1250-3	109	15.06	3.52	0.21	0	0	0	0	0	–	∞
p1250-5	95	17.50	7.47	0.00	0	0	0	0	0	–	∞
p1250-7	2034	310.96	11.48	8.78	0	0	0	1	0	445	0.00
p1250-9	122	37.41	9.15	1.25	0	0	0	0	0	–	∞
p1650-1	5	2.29	1.98	0.00	0	0	0	0	0	–	∞
p1650-3	11	3.29	2.88	0.00	0	0	0	0	0	–	∞
p1650-5	61	11.11	8.90	0.07	0	0	0	0	0	–	∞
p1650-7	88	12.71	5.36	0.00	0	0	0	0	0	–	∞
p1650-9	101	19.23	4.52	0.36	0	0	0	0	0	–	∞
p2050-1	41	5.59	2.24	0.00	0	0	0	0	0	–	∞
p2050-3	14	3.93	2.49	0.00	0	0	0	0	0	–	∞
p2050-5	6	3.98	3.50	0.00	0	0	0	0	0	–	∞
p2050-7	6	3.98	3.82	0.00	0	0	0	0	0	–	∞
p2050-9	362	58.74	5.27	1.49	0	0	0	0	1	412	0.00
p10100-11	46	116.41	78.76	0.00	0	0	0	0	0	–	∞
p10100-13	12	60.70	39.15	0.00	0	0	0	0	0	–	∞
p10100-15	213	409.90	162.60	3.74	0	0	0	0	0	–	∞
p10100-17	76	122.75	46.16	0.00	0	0	0	0	0	–	∞
p10100-19	21	59.28	31.54	0.00	0	0	0	0	0	–	∞
p25100-11	3167	781.10	103.15	39.34	0	0	0	0	5	547	0.55
p25100-13	8645	1800.00	83.03	39.70	0	0	0	0	3	575	0.17
p25100-15	631	205.93	81.65	2.94	0	0	0	1	3	583	0.00
p25100-17	51	33.30	27.57	0.00	0	0	0	0	0	–	∞
p25100-19	6755	1800.01	81.61	39.52	0	0	0	0	6	551	0.00
p33100-11	532	108.67	65.24	2.28	0	0	0	1	2	414	0.00
p33100-13	248	73.05	29.42	3.38	0	0	0	0	3	448	0.45
p33100-15	248	71.57	48.19	2.87	0	0	0	0	5	476	0.42
p33100-17	332	70.31	17.93	2.27	0	0	0	0	0	–	∞
p33100-19	3828	745.49	69.46	44.49	0	0	0	2	1	445	0.00
p40100-11	2017	427.91	29.78	33.53	0	0	0	0	2	415	0.00
p40100-13	1045	197.56	19.45	36.46	0	0	0	0	3	412	0.00
p40100-15	1166	290.82	46.60	39.11	0	0	0	0	2	502	1.21
p40100-17	1962	526.33	74.08	45.08	0	0	0	0	1	458	4.09
p40100-19	3368	799.47	216.19	34.81	1	0	0	0	4	451	0.22
p15150-21	77	209.02	153.07	0.00	0	0	0	0	0	–	∞
p15150-23	13	105.80	102.62	0.00	0	0	0	0	0	–	∞
p15150-25	124	314.03	99.65	0.43	0	0	0	0	0	–	∞
p15150-27	525	1800.16	1003.18	4.99	0	0	0	0	0	–	∞
p15150-29	1	61.09	60.77	0.00	0	0	0	0	0	–	∞
p36150-21	3642	1800.00	614.95	74.46	0	0	0	1	8	707	0.00
p36150-23	2278	1800.06	–	0.00	0	0	0	0	0	–	∞
p36150-25	84	64.00	55.43	0.40	0	0	0	0	1	644	0.16
p36150-27	2138	1800.02	264.17	76.63	0	0	0	1	1	754	0.00
p36150-29	286	178.85	119.19	4.01	0	0	0	0	3	665	0.15
p37150-21	373	285.47	228.79	5.08	0	0	0	0	6	682	0.15
p37150-23	2351	1800.00	1339.59	126.93	0	0	0	1	3	669	0.00
p37150-25	73	62.02	49.45	0.46	0	0	0	0	0	–	∞
p37150-27	1986	1800.05	–	0.00	0	0	0	0	0	–	∞
p37150-29	645	308.98	64.85	7.48	0	0	0	0	3	649	0.00
p50150-21	4381	1800.00	712.64	72.95	0	0	0	0	0	–	∞
p50150-23	3803	1800.03	821.07	79.25	0	0	0	1	3	567	0.53
p50150-25	50	46.75	27.51	0.78	0	0	0	0	1	488	0.00
p50150-27	3368	1800.00	146.78	87.02	1	0	0	1	2	580	0.00
p50150-29	5131	1800.02	247.31	87.36	1	0	0	1	3	552	0.00
p60150-21	255	148.98	81.06	5.34	0	0	0	0	1	553	0.18

Continue next page

Instance	Nodes	Time	First	HTime	Improvement Heuristics Cross	Mut	LocB	RINS	DINS	Primal	HGap
p60150-23	3181	1800.03	585.93	115.40	0	0	0	0	2	565	1.25
p60150-25	7744	1800.01	165.33	88.56	0	0	0	1	3	445	2.06
p60150-27	2293	1800.04	776.82	111.33	1	0	0	0	1	770	0.00
p60150-29	356	133.35	34.51	3.90	0	0	0	0	0	–	∞
p20200-32	411	1800.50	673.00	5.85	0	0	0	0	0	–	∞
p20200-34	432	1800.18	546.08	6.32	0	0	0	0	2	1391	0.00
p20200-36	32	229.42	163.44	0.00	0	0	0	0	0	–	∞
p50200-36	1341	1057.05	311.92	130.42	0	0	0	0	1	706	0.71
p50200-37	2281	1271.18	213.18	111.37	0	0	0	0	3	755	0.27
p50200-39	2796	1784.67	138.86	166.96	0	0	0	0	1	727	0.69
p66200-31	1168	637.25	86.74	138.71	0	0	0	0	4	575	0.00
p66200-36	3024	1800.09	250.79	162.52	0	0	0	1	3	580	0.00
p66200-38	2178	1800.26	1405.79	145.36	0	0	0	1	3	625	0.00
p80200-33	3267	1800.01	516.62	151.57	0	0	0	1	5	557	0.00
p80200-34	1352	1800.01	482.46	168.93	0	0	0	0	1	852	0.00
p80200-38	1471	1077.73	137.95	138.09	0	0	0	1	6	593	0.17
Total	102514	14:17:46		0:44:20	4	0	0	16	113		
Geom. Mean	278	179.81		6.59							

Table B.24.: Improvement Heuristics on the test set CPMPS

Instance	GCG without heuristics Primal	Gap	Nodes	Time	First	GCG with master heuristics Primal	Gap	Nodes	Time	First
N1C1W1_A	25	0.00	7	0.65	0.65	25	0.00	1	0.60	0.60
N1C1W1_K	26	0.00	6	0.50	0.50	26	0.00	1	0.46	0.46
N1C1W2_B	30	0.00	4	0.68	0.67	30	0.00	1	0.67	0.67
N1C1W2_L	31	0.00	1	0.79	0.77	31	0.00	1	0.79	0.79
N1C1W4_C	36	0.00	1	0.73	0.71	36	0.00	1	0.74	0.74
N1C1W4_M	41	0.00	2	0.72	0.71	41	0.00	1	0.66	0.66
N1C2W1_D	21	0.00	1	0.46	0.45	21	0.00	1	0.47	0.47
N1C2W1_N	21	0.00	15	1.05	1.05	21	0.00	1	1.04	1.04
N1C2W2_E	33	0.00	5	0.62	0.61	33	0.00	1	0.57	0.57
N1C2W2_O	29	0.00	2	0.50	0.50	29	0.00	2	0.53	0.49
N1C2W4_F	32	0.00	1	0.54	0.52	32	0.00	1	0.50	0.50
N1C2W4_P	28	0.00	1	0.75	0.74	28	0.00	1	0.67	0.66
N1C3W1_G	15	0.00	10	0.86	0.86	15	0.00	10	0.91	0.85
N1C3W1_Q	20	0.00	12	1.05	1.05	20	0.00	12	1.10	1.04
N1C3W2_H	23	0.00	15	1.08	1.07	23	0.00	1	1.24	1.24
N1C3W2_R	19	0.00	23	1.18	1.17	19	0.00	17	1.20	1.09
N1C3W4_I	23	0.00	5	0.67	0.67	23	0.00	4	0.65	0.62
N1C3W4_S	22	0.00	12	0.65	0.65	22	0.00	11	0.65	0.60
N2C1W1_A	48	0.00	21	3.86	3.85	48	0.00	1	3.68	3.68
N2C1W1_K	55	0.00	12	3.00	2.98	55	0.00	5	3.13	2.81
N2C1W2_B	61	0.00	4	4.16	4.15	61	0.00	4	4.17	3.99
N2C1W2_L	62	0.00	4	3.34	3.33	62	0.00	1	3.21	3.20
N2C1W4_C	77	0.00	1	3.29	3.21	77	0.00	1	3.13	3.12
N2C1W4_M	72	0.00	5	3.49	3.47	72	0.00	4	3.52	3.39
N2C2W1_D	42	0.00	22	7.69	7.68	42	0.00	1	7.48	7.47
N2C2W1_N	43	0.00	20	5.84	5.83	43	0.00	1	5.81	5.70
N2C2W2_E	54	0.00	6	2.99	2.97	54	0.00	1	2.71	2.71
N2C2W2_O	50	0.00	15	4.36	4.35	50	0.00	1	4.06	4.05
N2C2W4_F	57	0.00	1	5.09	5.02	57	0.00	1	4.88	4.87
N2C2W4_P	60	0.00	3	3.86	3.85	60	0.00	1	3.75	3.75
N2C3W1_G	33	0.00	32	3.81	3.80	33	0.00	37	4.28	4.27
N2C3W1_Q	34	0.00	41	3.93	3.92	34	0.00	41	4.07	3.44
N2C3W2_H	38	0.00	30	4.29	4.28	38	0.00	32	4.42	4.01
N2C3W2_R	40	0.00	30	5.00	4.99	40	0.00	20	5.08	4.81
N2C3W4_I	44	0.00	23	6.80	6.78	44	0.00	1	6.68	6.68
N2C3W4_S	42	0.00	32	10.54	10.53	42	0.00	31	10.66	10.31
N3C1W1_A	105	0.00	25	14.49	14.41	105	0.00	18	14.45	12.69
N3C1W1_K	102	0.00	14	14.17	14.10	102	0.00	1	13.89	13.34
N3C1W2_B	126	0.00	1	11.17	10.58	126	0.00	1	10.75	10.69
N3C1W2_L	136	0.00	1	14.67	14.13	136	0.00	1	14.23	14.17
N3C1W4_C	146	0.00	6	17.53	17.42	146	0.00	6	17.72	16.73
N3C1W4_M	149	0.00	1	17.86	17.28	149	0.00	1	17.29	17.22
N3C2W1_D	85	0.00	34	12.41	12.35	85	0.00	24	12.08	10.12
N3C2W1_N	91	0.00	2	11.53	11.48	91	0.00	2	11.49	11.10
N3C2W2_E	116	0.00	3	13.92	13.84	116	0.00	1	13.37	13.32
N3C2W2_O	107	0.00	2	13.28	13.21	107	0.00	1	12.93	12.88
N3C2W4_F	115	0.00	2	14.58	14.50	115	0.00	1	14.36	14.31
N3C2W4_P	122	0.00	6	19.17	19.08	122	0.00	6	19.32	19.26
N3C3W1_G	65	0.00	83	17.44	17.39	65	0.00	105	19.79	13.80
N3C3W1_Q	73	0.00	64	18.67	18.62	73	0.00	81	20.72	14.62
N3C3W2_H	82	0.00	38	46.25	46.19	82	0.00	35	40.34	36.71
N3C3W2_R	79	0.00	41	17.93	17.87	79	0.00	35	18.07	18.03
N3C3W4_I	92	0.00	30	18.54	18.48	92	0.00	33	20.17	17.50
N3C3W4_S	84	0.00	41	36.48	36.42	84	0.00	57	44.48	35.69
Total			819	0:07:08				661	0:07:13	
Geom. Mean			7	4.26				3	4.28	

Table B.25.: GCG without and with master heuristics on the test set BINDATA1S

Instance	GCG without heuristics Primal	Gap	Nodes	Time	First	GCG with master heuristics Primal	Gap	Nodes	Time	First
1-FullIns_3	4	0.00	11	1.37	1.37	4	0.00	1	1.29	1.27
1-FullIns_4	–	–	120	1800.20	–	6	65.14	123	1800.06	170.09
2-FullIns_3	5	0.00	21	3.48	3.48	5	0.00	23	3.38	3.27
2-Insertions_3	4	65.05	1054	1800.11	1310.60	4	65.05	1011	1800.29	6.83
3-FullIns_3	6	0.00	45	7.20	7.19	6	0.00	51	7.08	4.97
4-FullIns_3	7	0.00	78	12.55	12.54	7	0.00	70	11.89	11.89
4-FullIns_4	–	–	1	1800.23	–	–	–	1	1801.83	–
5-FullIns_3	8	0.00	116	21.22	21.21	8	0.00	95	24.06	24.05
anna	11	0.00	59	6.26	6.24	11	0.00	72	7.29	1.05
ash331GPIA	–	–	1	1800.71	–	–	–	1	1803.23	–
david	11	0.00	19	1.36	1.36	11	0.00	28	1.76	0.60
DSJC125.9	–	–	1	1823.54	–	45	3.83	1	1824.28	1791.88
DSJR500.1	–	–	89	1801.21	–	–	–	71	1801.68	–
fpsol2.i.1	–	–	293	1789.68	–	96	0.00	255	1788.06	240.29
fpsol2.i.2	–	–	108	1800.06	–	–	–	111	1800.03	–
fpsol2.i.3	–	–	115	1800.10	–	–	–	156	1800.05	–
games120	9	0.00	84	18.56	18.55	9	0.00	85	16.96	14.48
homer	13	0.00	418	1715.51	1715.24	13	0.00	450	1786.38	1786.32
huck	11	0.00	38	1.83	1.82	11	0.00	31	1.63	0.41
inithx.i.1	–	–	1	1867.62	–	–	–	1	1870.64	–
inithx.i.2	–	–	1	1970.90	–	–	–	1	1805.56	–
inithx.i.3	–	–	1	1974.51	–	–	–	1	1802.81	–
jean	10	0.00	31	1.41	1.40	10	0.00	39	1.65	0.38
le450_25a	–	–	152	1800.11	–	–	–	143	1800.11	–
le450_25b	–	–	142	1800.05	–	–	–	144	1800.12	–
miles1000	42	0.00	1	24.70	18.32	42	0.00	1	18.48	18.38
miles1500	73	0.00	1	76.35	46.58	73	0.00	1	46.07	45.80
miles250	8	0.00	85	13.63	13.61	8	0.00	93	17.84	14.47
miles500	20	0.00	54	10.73	10.70	20	0.00	27	6.28	2.55
miles750	31	0.00	1	7.79	5.66	31	0.00	14	13.41	6.48
mulsol.i.1	49	0.00	12	66.81	66.68	49	0.00	15	72.74	45.73
mulsol.i.2	31	0.00	102	300.99	300.88	31	0.00	91	257.33	50.24
mulsol.i.3	31	0.00	100	351.34	351.23	31	0.00	114	366.30	53.37
mulsol.i.4	31	0.00	100	399.74	399.62	31	0.00	98	338.66	338.58
mulsol.i.5	31	0.00	70	248.06	247.96	31	0.00	96	295.32	63.52
myciel3	4	0.00	7	0.12	0.11	4	0.00	5	0.13	0.09
myciel4	5	0.00	595	91.56	11.40	5	0.00	521	85.55	2.59
qg.order30	–	–	1	3338.60	–	–	–	1	1831.43	–
qg.order40	–	–	1	9117.18	–	–	–	1	9333.99	–
queen5_5	5	0.00	1	0.22	0.22	5	0.00	1	0.23	0.23
queen6_6	7	0.00	13	12.97	9.16	7	0.00	13	14.52	3.59
queen7_7	7	0.00	1	16.71	16.69	7	0.00	1	20.05	20.05
queen8_12	12	0.00	186	811.91	185.02	12	0.00	86	263.89	16.14
queen8_8	9	0.00	389	1069.19	1069.17	9	0.00	58	275.88	32.15
queen9_9	–	–	230	1800.01	–	–	–	203	1800.00	–
queen10_10	–	–	103	1800.01	–	16	59.97	90	1800.00	330.70
school1	–	–	1	1792.89	–	–	–	1	1792.84	–
school1_nsh	–	–	1	1794.61	–	–	–	1	1794.61	–
wap05a	–	–	1	5543.95	–	–	–	1	5536.58	–
will199GPIA	–	–	1	1800.03	–	–	–	1	1802.05	–
zeroin.i.1	49	0.00	86	199.66	199.50	49	0.00	66	164.93	39.08
zeroin.i.2	30	0.00	137	379.08	378.97	30	0.00	92	290.77	36.75
zeroin.i.3	30	0.00	102	306.15	306.05	30	0.00	93	293.77	293.70
Total			5381	16:19:54				4750	15:28:15	
Geom. Mean			20	174.07				18	162.41	

Table B.26.: GCG without and with master heuristics on the test set COLORING

Instance	GCG without heuristics					GCG with master heuristics				
	Primal	Gap	Nodes	Time	First	Primal	Gap	Nodes	Time	First
p550-1	713	0.00	14	9.32	5.22	713	0.00	17	11.43	4.04
p550-3	751	0.00	3	6.34	5.90	751	0.00	3	6.89	5.73
p550-5	664	0.00	1	6.74	6.73	664	0.00	1	7.26	7.26
p550-7	787	0.00	21	19.29	7.30	787	0.00	23	21.25	8.82
p550-9	715	0.00	9	8.65	5.13	715	0.00	7	8.38	4.21
p1250-1	383	0.00	139	13.25	7.28	383	0.00	142	13.51	5.26
p1250-3	405	0.00	109	11.37	2.66	405	0.00	96	11.67	4.18
p1250-5	429	0.00	95	13.25	5.65	429	0.00	185	24.77	5.16
p1250-7	445	0.00	2035	229.20	8.69	445	0.00	1561	180.42	12.34
p1250-9	436	0.00	122	27.20	6.99	436	0.00	76	19.51	6.78
p1650-1	298	0.00	5	1.70	1.47	298	0.00	5	1.84	1.59
p1650-3	314	0.00	11	2.51	2.21	314	0.00	11	2.61	2.29
p1650-5	351	0.00	61	8.47	6.85	351	0.00	61	9.08	1.66
p1650-7	361	0.00	88	9.57	4.07	361	0.00	88	9.78	1.85
p1650-9	373	0.00	101	14.23	3.44	373	0.00	240	28.88	7.72
p2050-1	266	0.00	41	4.32	1.74	266	0.00	41	4.49	1.85
p2050-3	311	0.00	14	3.08	1.96	311	0.00	14	3.14	1.85
p2050-5	356	0.00	6	2.94	2.60	356	0.00	5	2.93	2.18
p2050-7	358	0.00	6	3.06	2.94	358	0.00	6	3.06	2.22
p2050-9	412	0.00	337	41.24	4.00	412	0.00	206	27.29	2.72
p10100-11	1006	0.00	46	86.62	58.85	1006	0.00	38	71.14	44.95
p10100-13	1026	0.00	12	45.37	29.59	1026	0.00	10	45.47	35.03
p10100-15	1091	0.00	213	299.11	119.21	1091	0.00	205	289.58	46.88
p10100-17	1034	0.00	76	91.24	34.47	1034	0.00	75	94.66	71.80
p10100-19	1031	0.00	21	44.42	24.06	1031	0.00	21	48.67	28.95
p25100-11	544	0.00	3090	541.06	77.05	544	0.00	3001	538.37	92.00
p25100-13	557	1.61	13476	1800.00	61.97	559	1.97	12907	1800.00	129.90
p25100-15	583	0.00	608	150.44	61.01	583	0.00	455	118.23	34.19
p25100-17	542	0.00	51	25.39	21.12	542	0.00	40	21.61	20.38
p25100-19	551	0.00	5854	1111.00	61.52	551	0.00	8961	1646.15	100.15
p33100-11	414	0.00	605	81.21	48.59	414	0.00	615	81.59	27.78
p33100-13	446	0.00	165	35.77	21.79	446	0.00	206	35.48	10.59
p33100-15	474	0.00	584	102.35	35.94	474	0.00	291	52.15	20.60
p33100-17	431	0.00	332	50.61	13.39	431	0.00	404	66.64	15.30
p33100-19	445	0.00	3188	497.77	52.11	445	0.00	4027	596.22	124.16
p40100-11	415	0.00	2698	350.03	22.19	415	0.00	2581	355.17	13.08
p40100-13	412	0.00	916	99.34	14.47	412	0.00	1042	111.64	7.30
p40100-15	496	0.00	1031	165.65	34.41	496	0.00	1119	176.49	7.32
p40100-17	440	0.00	1975	386.07	96.41	440	0.00	2153	398.30	30.31
p40100-19	450	0.00	2822	487.79	159.00	450	0.00	2649	459.89	143.20
p15150-21	1288	0.00	77	156.29	114.99	1288	0.00	175	314.52	174.13
p15150-23	1279	0.00	13	80.84	78.48	1279	0.00	12	86.78	86.74
p15150-25	1193	0.00	124	229.83	74.36	1193	0.00	113	232.43	87.46
p15150-27	1328	0.48	714	1800.05	757.57	1333	1.11	913	1800.17	665.80
p15150-29	1219	0.00	1	48.95	48.75	1219	0.00	1	47.49	47.46
p36150-21	738	6.58	5379	1800.03	457.39	767	10.99	4998	1800.02	965.20
p36150-23	728	7.96	3146	1800.00	1759.47	–	–	3039	1800.00	–
p36150-25	643	0.00	85	48.79	42.43	643	0.00	77	43.23	23.36
p36150-27	759	1.57	3247	1800.04	195.98	755	1.05	3008	1800.03	688.33
p36150-29	664	0.00	228	111.18	88.91	664	0.00	204	94.16	53.63
p37150-21	681	0.00	393	208.51	169.55	681	0.00	213	107.02	58.36
p37150-23	742	13.79	3705	1800.09	991.70	747	14.54	3557	1800.04	1525.44
p37150-25	629	0.00	73	47.28	38.32	629	0.00	72	42.39	32.60
p37150-27	794	9.51	2946	1800.01	1525.65	777	7.01	3436	1800.03	837.08
p37150-29	649	0.00	1131	341.23	49.45	649	0.00	980	299.45	58.68
p50150-21	612	3.16	6475	1800.01	521.02	650	9.97	6023	1800.04	1411.78
p50150-23	564	0.00	4762	1513.07	602.77	564	0.00	4201	1278.62	305.76
p50150-25	488	0.00	64	38.16	21.54	488	0.00	66	39.45	25.51
p50150-27	579	0.00	3050	1143.59	108.59	579	0.00	2812	1074.68	143.17
p50150-29	548	1.76	7958	1800.02	181.30	554	2.93	7436	1800.01	450.15
p60150-21	552	0.00	281	112.86	59.22	552	0.00	314	122.54	57.38

Continue next page

Instance	GCG without heuristics					GCG with master heuristics				
	Primal	Gap	Nodes	Time	First	Primal	Gap	Nodes	Time	First
p60150-23	557	0.87	4798	1800.01	432.30	555	0.00	4154	1470.74	323.17
p60150-25	436	0.67	9805	1800.01	120.31	436	0.45	10449	1800.00	94.59
p60150-27	782	7.94	3401	1800.00	577.87	808	11.57	3387	1800.00	780.50
p60150-29	494	0.00	356	96.07	25.66	494	0.00	299	84.36	25.92
p20200-32	1427	0.39	609	1800.65	512.32	1424	0.14	586	1800.20	412.09
p20200-34	1386	0.17	638	1805.85	418.09	1385	0.00	556	1778.54	784.80
p20200-36	1382	0.00	32	180.23	130.15	1382	0.00	26	180.82	130.84
p50200-36	701	0.00	1271	671.86	235.87	701	0.00	1885	884.95	98.83
p50200-37	753	0.00	2727	1052.47	163.35	753	0.00	2862	1160.73	291.65
p50200-39	722	0.00	1410	691.04	108.46	722	0.00	1312	705.32	160.24
p66200-31	575	0.00	1388	415.13	68.33	575	0.00	965	341.63	129.13
p66200-36	580	0.43	4714	1800.03	188.69	580	0.31	4555	1800.03	224.20
p66200-38	681	13.41	3465	1800.05	1058.72	686	14.26	3344	1800.08	1029.79
p80200-33	559	1.05	4715	1800.01	383.66	565	2.26	4458	1800.00	230.35
p80200-34	847	0.90	2110	1800.01	367.97	847	0.95	2076	1800.03	350.72
p80200-38	592	0.00	1146	528.20	103.66	592	0.00	918	499.75	185.85
Total			127418	12:33:29				127070	12:34:55	
Geom. Mean			306	140.63				299	140.92	

Table B.27.: GCG without and with master heuristics on the test set CPMPS

Instance	GCG without heuristics Primal	Gap	Nodes	Time	First	GCG with master heuristics Primal	Gap	Nodes	Time	First
new1_2_32	35771	0.00	90	2791.85	1279.95	35771	0.00	77	2685.33	238.97
new1_2_64	35771	0.00	18	1519.80	705.77	35771	0.00	21	1437.23	398.39
new1_6_32	55495	0.00	27	1299.15	399.56	55495	0.00	26	1247.59	282.17
new1_6_64	55495	0.00	24	2540.78	769.86	55495	0.00	23	1892.14	396.80
new1_10_32	75756	0.00	27	1684.84	1032.55	75756	0.00	39	3104.39	893.63
new1_10_64	75756	0.00	7	2223.48	1697.12	75756	0.00	7	2218.33	1402.11
new2_1_32	3109	0.00	11	223.48	179.04	3109	0.00	15	254.72	147.44
new2_1_64	3109	0.00	7	417.42	272.57	3109	0.00	11	429.51	197.28
new2_3_32	4202	0.00	17	386.76	230.70	4202	0.00	33	564.08	204.97
new2_3_64	4202	0.00	3	358.28	358.26	4202	0.00	4	542.54	336.34
new2_7_32	6147	0.04	117	3600.00	2400.03	6148	0.02	101	3600.00	428.63
new2_7_64	6148	0.00	13	1511.53	946.02	6148	0.00	17	1474.45	738.63
new3_1_32	71998	0.00	34	3552.58	2915.83	71998	0.00	21	2415.66	875.63
new3_1_64	71998	0.00	4	2716.48	2716.43	71998	0.00	3	2627.46	1758.18
new3_4_32	107478	0.03	34	3600.01	3035.24	107491	0.01	29	3600.00	1391.21
new3_4_64	107491	0.00	7	3599.84	3599.73	107491	0.00	13	3599.83	2904.38
new3_5_32	120495	0.02	24	3600.02	3599.60	120486	0.03	23	3600.05	1888.61
new3_5_64	–	–	1	3600.04	–	120501	0.00	13	3599.86	3419.08
new4_1_32	22044	0.00	1	41.80	41.53	22044	0.00	1	37.72	37.71
new4_1_64	22044	0.00	1	52.94	52.78	22044	0.00	1	48.91	48.91
new4_5_32	37016	0.00	13	203.79	108.53	37016	0.00	13	165.52	55.63
new4_5_64	37016	0.00	13	169.52	84.00	37016	0.00	13	165.18	65.15
new4_10_32	54842	0.00	3	150.98	144.22	54842	0.00	3	170.92	141.76
new4_10_64	54842	0.00	3	278.67	238.81	54842	0.00	3	289.07	228.29
new5_2_32	47914	0.00	14	634.04	463.79	47914	0.00	11	601.03	366.14
new5_2_64	47914	0.00	4	1073.56	1073.52	47914	0.00	1	768.31	768.30
new5_7_32	81970	0.04	102	3600.00	1743.92	81936	0.09	89	3600.01	517.48
new5_7_64	81982	0.00	12	1380.58	930.51	81982	0.00	13	1424.88	612.71
new5_10_32	100031	0.00	72	3600.00	2379.37	100031	0.00	83	3600.04	647.75
new5_10_64	100020	0.02	42	3600.00	2617.22	100026	0.02	29	3600.02	1529.06
new6_1_32	71426	0.00	40	3191.29	1656.21	71426	0.00	44	3385.86	591.26
new6_1_64	71426	0.00	20	3599.90	2537.13	71426	0.00	9	2081.21	1277.64
new6_2_32	82942	0.00	7	1901.51	1616.45	82942	0.00	7	1489.69	997.75
new6_2_64	82942	0.00	3	2386.95	2305.81	82942	0.00	3	2381.12	2031.48
new6_5_32	119194	0.05	29	3600.04	3599.76	119202	0.04	36	3600.03	1509.89
new6_5_64	119233	0.00	12	3599.78	2509.46	119233	0.00	17	3599.88	1833.62
new7_1_32	42685	0.00	59	1907.67	1098.78	42685	0.00	59	2071.89	259.25
new7_1_64	42685	0.00	3	535.52	482.97	42685	0.00	3	534.21	455.67
new7_4_32	60719	0.00	30	1597.67	1178.51	60719	0.00	31	2157.24	617.59
new7_4_64	60663	0.11	27	3600.00	3599.79	60716	0.01	30	3600.01	1968.87
new7_9_32	92380	0.00	17	2470.79	1862.96	92380	0.00	17	2531.93	1391.06
new7_9_64	92380	0.00	19	3599.97	3436.09	92378	0.05	1	3600.23	3599.62
Total			1011	23:53:23				993	23:26:38	
Geom. Mean			13	1277.99				12	1270.24	

Table B.28.: GCG without and with master heuristics on the test set RapS

List of Algorithms

List of Tables

Bibliography

[AB07] Tobias Achterberg and Timo Berthold. Improving the Feasibility Pump. *Discrete Optimization*, Special Issue 4(1):77–86, 2007.

[ABH+] Tobias Achterberg, Timo Berthold, Stefan Heinz, Thorsten Koch, and Kati Wolter. SCIP – Solving Constraint Integer Programs, documentation. `http://scip.zib.de`.

[Ach07] Tobias Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007. `http://opus.kobv.de/zib/volltexte/2009/1153/`.

[Ach09] Tobias Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1), 2009. `http://mpc.zib.de/index.php/MPC/article/view/4`.

[AKM06] Tobias Achterberg, Thorsten Koch, and Alexander Martin. MIPLIB 2003. *Operations Research Letters*, 34(4):361–372, 2006.

[Ber06] Timo Berthold. Primal Heuristics for Mixed Integer Programs. Master's thesis, Technische Universität Berlin, 2006. `http://opus.kobv.de/zib/volltexte/2007/1054/`.

[BJN+98] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46(3):316–329, May-June 1998.

[CFM10] Alberto Caprara, Fabio Furini, and Enrico Malaguti. Exact algorithms for the temporal knapsack problem. Technical report OR-10-7, DEIS, University of Bologna, 2010.

[CFT99] Alberto Caprara, Matteo Fischetti, and Paolo Toth. A heuristic method for the set covering problem. *Operations Research*, 47(5):730–743, September-October 1999.

[CPL09] ILOG CPLEX 12.1, 2009. `http://www.ilog.com/products/cplex`.

[DL05] Jacques Desrosiers and Marco E. Lübbecke. A Primer in Column Generation. In Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors, *Column Generation*, pages 1–32. Springer US, 2005.

[Dob82] Gregory Dobson. Worst-Case Analysis of Greedy Heuristics for Integer Programming with Nonnegative Data. *Mathematics of Operations Research*, 7(4):515–531, November 1982.

[DRP05] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102:71–90, 2005.

[DW60] George B. Dantzig and Philip Wolfe. Decomposition Principle for Linear Programs. *Operations Research*, 8(1):101–111, 1960.

[FGL05] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104:91–104, 2005.

[FL03] Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003.

[FS09] Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1:201–222, 2009.

[Gam10] Gerald Gamrath. Generic Branch-Cut-and-Price. Master's thesis, Technische Universität Berlin, 2010.

[Gho07] Shubhashis Ghosh. DINS, a MIP Improvement Heuristic. In Matteo Fischetti and David Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 310–323. Springer Berlin / Heidelberg, 2007.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co Ltd, January 1979.

[GL10] Gerald Gamrath and Marco E. Lübbecke. Experiments with a Generic Dantzig-Wolfe Decomposition for Integer Programs. In Paola Festa, editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 239–252. Springer Berlin / Heidelberg, 2010.

[Gur10] Gurobi Optimizer 3.0, 2010. `http://www.gurobi.com`.

[JMS+10] Cédric Joncour, Sophie Michel, Ruslan Sadykov, Dimitri Sverdlov, and François Vanderbeck. Column Generation based Primal Heuristics. *Electronic Notes in Discrete Mathematics*, 36:695 – 702, 2010. ISCO 2010 - International Symposium on Combinatorial Optimization.

[JMT] David S. Johnson, Anuj Mehrotra, and Michael Trick. Color 02 03 04: Graph Coloring and its Generalizations. `http://mat.gsia.cmu.edu/COLOR04/`.

[Lod03] Andrea Lodi. Local branching: a tutorial. presented at MIC2003, Kyoto, 2003.

[Mit] Hans D. Mittelmann. Mixed Integer Linear Programming Benchmark (serial codes). `http://plato.asu.edu/ftp/milpf.html`.

[NW88] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[SK] Armin Scholl and Robert Klein. Bin packing instances: Data set 1. `http://www.wiwi.uni-jena.de/Entscheidung/binpp/`.

[Wal10] Chris Wallace. ZI round, a MIP rounding heuristic. *Journal of Heuristics*, 16(5):715–722, 2010.

[Wed95] Dag Wedelin. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, 57:283–301, 1995.