MODELING AND SOLVING THE NONOGRAM PUZZLE USING CONSTRAINT PROGRAMMING

by

Trieu Hung Tran

A THESIS

Presented to the Faculty of

The College of Art and Sciences at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Bachelor of Science

Major: Computer Science

Under the Supervision of Professor Berthe Y. Choueiry (adviser) and
Professor Mohammad Rashedul Hasan (co-adviser)

Lincoln, Nebraska

October, 2019

MODELING AND SOLVING THE NONOGRAM PUZZLE USING CONSTRAINT PROGRAMMING

Trieu Hung Tran, B.S.

University of Nebraska, 2019

Adviser: Berthe Y. Choueiry (adviser) and Mohammad Rashedul Hasan (co-adviser)

Finding a solution for a Constraint Satisfaction Problem (CSP) is in general NP-complete. Backtrack search is a sound and complete algorithm for solving CSPs. Nonogram is an interesting and popular combinatorial puzzle. Its two obvious models, which use regular and global table constraints, present various practical limitations.

We study the reformulation of the regular constraint in terms of ternary-table constraints proposed in the literature. Then, we investigate the performance of search with various consistency algorithms on the two table-constraint models.

Empirical evaluations on Nonogram benchmark demonstrate the effectiveness of the ternary-tables model in terms of CPU time. We show that high-level singleton consistencies outperform all other consistency algorithms in terms of instances solved in a backtrack-free manner. However, GAC-based consistency algorithms outperform all others in terms of CPU time.

ACKNOWLEDGMENTS

I would like to thank my adviser, Dr. Berthe Y. Choueiry, for her guidance and advice during my two years in the Constraint Systems Laboratory. I appreciate the conversations that we had about research and her constructive feedback in the revision of this thesis. I would also like to thank my co-adviser, Dr. Mohammad Rashedul Hasan, for his continued support and for introducing me to Artificial Intelligence and Machine Learning. I feel honored to have both of them as my advisers. I would also like to thank Dr. Christian Bessière of Laboratoire d'Informatique d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) of the Université de Montpellier and the French CNRS for his insightful and constructive questions that allowed me to revise the focus and directions of my research several times.

I also wish to acknowledge the help provided by the people of the Constraint Systems Laboratory including Mr. Anthony (Tony) Schneider, Mr. Robert Woodward, Mr. Ian Howell, and Mr. Khang Phan. They developed the constraint solver Stampede, which was essential to my experiments. Tony helped me with the initial idea of implementing the constraint conversion in Stampede. Robert proposed and implemented the four higher-level consistency algorithms that I used in my experiments. Ian was always available to help with debugging and guiding me through Stampede's implementation. I verified the results of my experiments in collaboration with Khang, who is conducting similar studies. Their helpful explanations to my questions and recommendations on my research are greatly appreciated. It was both a delightful and rewarding experience to work with each of them.

Finally, I would like to express my deep gratitude to my family for giving me a chance to pursue my passion in Computer Science. I am also grateful to my girlfriend, Anh, for her support and encouragement throughout the process of this thesis.

This research was supported by a UCARE award from the University of Nebraska-Lincoln and by NSF grant RI-1619344. Experiments were conducted on the equipment at the Holland Computing Center at UNL.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the motivation behind our study of the Nonogram puzzle as a Constraint Satisfaction Problem (CSP). We summarize the current models and our project's contribution. Finally, we describe the organization of this thesis.

1.1 Motivation

Constraint Processing (CP) is a flexible computational framework for modeling and solving decision problems. This framework allows the users to state arbitrary constraints and variables in an expressive way and focuses on providing feedback about solutions and conflicts.

Puzzles are one common problem that can be modelled as a Constraint Satisfaction Problem (CSP). They have been an attractive and suitable tool to introduce the concepts of CP to general public. In the Constraint Systems Laboratory (ConSystLab),[1] many puzzle solvers have been implemented and visualized to support teaching in the classrooms and research including Minesweeper [Bayer *et al.*,

[1] http://consystlab.unl.edu

2006],[2] Game of Set [Swearingn *et al.*, 2011],[3] and Sudoku [Reeson *et al.*, 2007; Howell *et al.*, 2018].[4] However, Nonogram had not been studied yet.

Currently, two constraint models of the Nonogram puzzle as a CSP are provided on the XCSP3 website of benchmark problems.[5] One model uses the global-table constraints and the second regular constraints. While the global-tables model is costly, both in storage space and in processing time, the regular constraints model requires the use of a specialized Generalized Arc Consistency (GAC) propagator [Pesant, 2004].

We propose to investigate:

1. The reformulation of regular constraints into ternary-table constraints as proposed by Bessière *et al.* [2008].

2. The performance of backtrack search with various consistency algorithms for solving two of the models, namely, global-table and ternary-table constraints.

1.2 Contributions

Our investigations result in the generation of new models using only ternary-table constraints, which can now be used by other researchers.

We empirically assess the performance of the most commonly used algorithms for enforcing GAC. We show that, when the constraint arity is small, the GAC2001 algorithm [Bessière *et al.*, 2005] outperforms STR2 [Lecoutre, 2011] and CT [Demeulenaere *et al.*, 2016] algorithms. However, CT exhibits the best performance on constraints with large arity.

[2] http://minesweeper.unl.edu
[3] http://gameofset.unl.edu
[4] http://sudoku.unl.edu
[5] http://www.xcsp.org/series

Finally, we empirically characterize the performance of backtrack search with various types of consistency algorithms available on the Stampede constraint solver of the Constraint Systems Laboratory, including GAC (filtering domain or relation), singleton-type consistency, and higher-level relational consistency.

1.3 Thesis Organization

The rest of this thesis touches on different components of this study in each chapter. Chapter 2 reviews background information of Constraint Processing, introduces the Nonogram puzzle, and the consistency algorithms that we used. Chapter 3 discusses three different models of the Nonogram as a CSP. Chapter 4 introduces our experimental set-up and Chapter 5 discusses our results. Finally, Chapter 6 concludes the thesis and suggests directions for future research.

Chapter 2

Background

In this chapter, we review background information about the Constraint Satisfaction Problem (CSP). Then, we review the consistency algorithms and the search mechanisms that we use in this thesis. Finally, we introduce the Nonogram puzzle which is the focus of this thesis.

2.1 Constraint Satisfaction Problem (CSP)

A Constraint Satisfaction Problem (CSP) is defined by $\langle \mathcal{V}, \mathcal{D}, \mathcal{C} \rangle$ in which:

- $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$ is a set of *variables*.
- $\mathcal{D}$ is a set of *domain* values of which the *domain* of X_i is a subset (i.e., $dom(X_i) \subseteq \mathcal{D}$). $dom(X_i)$ is a set of all possible values that can be assigned to X_i.
- $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ is a set of constraints in which C_i is a relation over a subset of variables.

- The subset of variables that a constraint C_i is applied on is the *scope* of C_i $scope(C_i) \subseteq \mathcal{V}$.
- The cardinality of $scope(C_i)$ is called the *arity* of C_i.
- The relation of a constraint C_i lists all allowed combinations of values that can be assigned to the variables in the scope of C_i. It is calculated as the Cartesian product of the domains of all variables in $scope(C_i)$. Each tuple in this relation is called a *support* of the constraint.

A constraint defined over two variables is a *binary constraint*, and a constraint defined over three variables is a *ternary constraint*. A *global constraint* is a constraint whose scope is specified as an input parameter.

Although many graphical representations of a CSP exist, we use two such representations, namely, the primal graph and the dual graph.

- In the *primal graph*, the vertices are the CSP variables. There is an edge between two vertices if their corresponding variables belong to at least one constraint in the CSP.
- In the *dual graph*, the vertices are the constraints. An edge exists between two vertices if the scopes of the two corresponding constraints intersect on at least one variable. This edge represents an equality constraint between the two vertices indicating that the shared variables must have the same values in the two corresponding constraints.

2.2 Solving a CSP

Because CSPs are in general NP-complete, backtrack search is currently the only sound and complete algorithm for solving them. Typically, backtrack search proceeds

in a depth-first manner, extending a consistent partial solution by instantiating one variable at a time.

CSP solvers enforce consistency properties before search and after every variable instantiation. This process removes inconsistent values from the domains of unassigned variables based on the current partial solution. By doing so, we can prune sub-trees that contain no solution, thus, reducing thrashing. In Section 2.4, we review the various consistency algorithms tested in this thesis.

2.3 Variable Ordering Heuristics

Variable ordering heuristics control the order in which variables are instantiated during search. This order directly affects the size of the search tree and can be determined either statically before search or dynamically during search. Two common dynamic variable ordering heuristics are dom/deg [Bessière and Régin, 1996] and dom/wdeg [Boussemart *et al.*, 2004].

- The dom/deg [Bessière and Régin, 1996] heuristic chooses the variable with the smallest ratio of current domain size to degree (i.e., the number of constraints the variable belongs to) as the next variable to assign.

- The dom/wdeg [Boussemart *et al.*, 2004] heuristic chooses the variable with the smallest ratio of current domain size to weighted degree as the next variable to instantiate. The weighted degree of a variable is calculated by totaling the weights of all constraints that apply to the variable. The weight of every constraint is initialized to one (1) and incremented by one every time that constraint causes a domain wipe-out (i.e., removes the last value in the domain of a variable) when enforcing consistency properties.

2.4 Consistency Algorithms

The consistency algorithms are used to support the process of finding the solution for a CSP by eliminating some values and/or tuples that cannot be a part of any solutions. We utilize 14 algorithms which are already implemented in Stampede, the CSP solver developed by graduate students of the Constraint Systems Laboratory, including:

- GAC: CT [Demeulenaere *et al.*, 2016], GAC2001 [Bessière *et al.*, 2005], and STR2 [Lecoutre, 2011].
- Singleton consistency: SAC1 [Debruyne and Bessière, 1997], POAC1 [Balafrej *et al.*, 2014], and APOAC [Balafrej *et al.*, 2014], as well as the following algorithms ANPOAC, APOAC$_{around\cup}$, PrePeak^{+}-POAC1, A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018].
- Relational consistency: R(∗,3)C, R(∗,4)C, wR(∗,3)C, and wR(∗,4)C [Karakashian *et al.*, 2010].

2.4.1 Generalized Arc Consistency (GAC)

Generalized Arc Consistency (GAC) [Mackworth, 1977] is a property of a CSP ensuring that, for every constraint in the CSP, a value in the domain of a variable in the scope of that constraint can be consistently extended to all other variables in the scope of the same constraint. GAC is practically always used during backtrack search, and several algorithms for enforcing this property have been proposed and continue to be proposed. Stampede provides several GAC implementations.

GAC2001 [Bessière *et al.*, 2005] operates by filtering the domains of the variables, leaving the constraints unaltered. In order to reduce the effort of constraint checks,

it keeps track of the last found supporting structure of each variable-value pair.

Simple Tabular Reduction (STR2) [Lecoutre, 2011] dynamically maintains the tables of allowed tuples (i.e., supports). The algorithm not only filters out domain values using GAC, but also removes tuples that are no longer GAC-consistent.

Compact-Table (CT) [Demeulenaere *et al.*, 2016] adopts STR algorithms' idea of maintaining the tables of supports and improves performance by using a data structure called an RSparseBitSet, which is similar to the sparse set structure [Briggs and Torczon, 1993; le Clément *et al.*, 2013].

In this thesis, we use two different types of constraint queue implemented in Stampede for GAC-based algorithms:

1. FIFO queue: the constraints are processed in a 'First In First Out' manner. This type of queue is supported in Stampede for all three GAC algorithms used in this thesis.

2. Lexicographical queue: the constraints are processed in the lexicographical order of their names. This type of queue is currently not supported in the CT implementation of Stampede.

2.4.2 Singleton Arc Consistency (SAC)

Singleton Arc Consistency (SAC) [Debruyne and Bessière, 1997] is a property of a CSP ensuring that for every assignment of a value in the domain of a variable to the variable itself, the resulting CSP is generalized arc consistent. The process of checking a given value as described is called a *singleton test*. SAC algorithms, such as SAC1 [Debruyne and Bessière, 1997], remove domain values that fail the singleton test.

POAC1 [Balafrej *et al.*, 2014] improves the performance of SAC1 [Debruyne and Bessière, 1997] and enforces a stronger consistency (i.e., may yield more filtering of domain values). When running singleton tests on the domain of a given variable, POAC1 maintains a counter for each value in the domain of every other variable. If a given value is filtered out in every singleton test, it cannot possibly appear in a solution and consequently can be filtered.

APOAC [Balafrej *et al.*, 2014] is an adaptive version of POAC1 [Balafrej *et al.*, 2014]: it interrupts the process after a certain number of variables are processed. This number is dynamically determined by a learning process during search.

PrePeak$^+$-POAC1 [Woodward, 2018] triggers the high-level consistency (HLC) algorithm POAC1 [Balafrej *et al.*, 2014] only at some specific depths during search. The points of triggering are determined by tracking the number of backtracks per depth of the search tree.

ANPOAC, APOAC$_{around\cup}$, and A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018] enforce APOAC [Balafrej *et al.*, 2014] on restricted sub-problems in the CSP determined by some topological properties:

- ANPOAC restricts APOAC to the neighborhood of the variable.
- APOAC$_{around\cup}$ restricts APOAC to variables that appear in a cycle of the current variable.
- A$\cup_{cyc}^{bfsc}$POAC restricts APOAC to the union of variables in the Breadth-First Search (BFS) approximation of the minimum cycle basis (MCB).

2.4.3 Relational Consistency

In contrast to the variable-based consistency properties, which are defined over the domains of the variables, relational-consistency properties are defined over the rela-

tions of the constraints. For instance, m-wise consistency [Gyssens, 1986; Janssen *et al.*, 1989] is a property of a CSP ensuring that every tuple in the relation of a constraint can be consistently extended to $m-1$ other constraints.

m-wise consistency is denoted as R$(*, m)$C by Karakashian *et al.* [2010]. Algorithms enforcing R$(*, m)$C operate by removing inconsistent tuples from the constraints. wR$(*, m)$C is a weakened version of R$(*, m)$C by integrating redundant edges removal, which may result in less filtering of the constraints. In this thesis, we examine the performance of both R$(*, m)$C and wR$(*, m)$C for $m = 3$ and $m = 4$.

2.5 Nonogram

A Nonogram puzzle of size $r \times c$ ($r > 0$ and $c > 0$) is defined by:

- An $r \times c$ board of colorable cells in which each cell can be in one of the two states: blank or colored.
- A set of $(r + c)$ labels, one for every row and every column of the board. A label of a row or a column is defined as $L = [l_1, l_2, \ldots, l_n]$ where $n \in \mathbb{Z}^+$, $l_i \in \mathbb{Z}$ and $l_i \geq 0$ $\forall 1 \leq i \leq n$. The meaning of the labels is discussed in Section 2.5.1.

An example of a Nonogram puzzle and its solution are shown in Figure 2.1.

2.5.1 Solving a Nonogram puzzle

A label $L = [l_1, l_2, \ldots, l_n]$ of a row or a column of a Nonogram puzzle represents a pattern of the colored cells in that row or column. Every l_i represents a block of l_i contiguous colored cells. There must be at least one blank cell between two block of l_i and l_{i+1} colored cells, $\forall 1 \leq i < n$. The first l_1 colored cells can be preceded by multiple blank cells, and the last l_n colored cell can also be followed by multiple blank

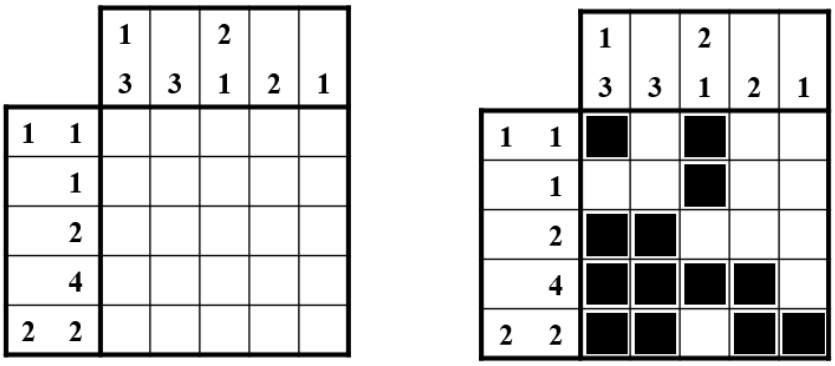

Figure 2.1: A Nonogram example and its solution

cells. The blank cells at the beginning and the end do not affect the correctness of the solution.

A Nonogram puzzle is usually left empty initially (i.e., all cells in the board are blank). A solution to the Nonogram puzzle is a way of coloring the cells in the board such that the colored cells form patterns that match the labels of the corresponding rows and columns. A row or column of a Nonogram puzzle by itself can have more than one solution differing in the number of blank cells at the beginning, at the end, and between consecutive blocks of colored cells.

For example, the solution of a Nonogram puzzle is shown on the right of Figure 2.1. Note that, without considering other rows and columns, there can be multiple ways to color the first row using the label $L_{R_1} = [1, 1]$. One of those ways is to color the second cell and the fourth cell of the row and leave the three other cells blank. However, that way of coloring is not consistent with the labels of other rows and columns. On the other hand, there is exactly one way to color the first column of the puzzle with label $L_{C_1} = [1, 3]$. It is because the label requires four colored cells with at least one blank cell which makes use of all five cells of the column.

2.5.2 A valid Nonogram puzzle

A valid Nonogram puzzle must have exactly one solution. Because there must be at least one blank cell between two consecutive blocks of colored cells, a row or column with label $L = [l_1, l_2, \ldots, l_n]$ must have at least $(\sum_{i=1}^{n} l_i) + n - 1$ cells. Moreover, because the total number of blank cells and colored cells is bounded by c in each row and by r in each column, a valid label must satisfy the following constraints:

- For a row, $(\sum_{i=1}^{n} l_i) + n - 1 \leq c$.
- For a column, $(\sum_{i=1}^{n} l_i) + n - 1 \leq r$.

Summary

In this chapter, we reviewed the background information of CSPs. We also described 14 consistency algorithms belonging to three main groups: GAC-based (three algorithms), singleton consistency (seven algorithms), and relational consistency (four algorithms). Then, we reviewed the method of backtracking to solve CSPs. Finally, we introduced and gave an example of the Nonogram puzzle. The special grid structure of the Nonogram puzzle is the main motivation behind our experiments on consistency algorithms that focus on exploring the structure of CSPs, such as ANPOAC, APOAC$_{around\cup}$, and A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018].

Chapter 3

Modeling

In this chapter, we discuss three constraint models for the Nonogram puzzle. These models only differ in how the constraints are formulated as, namely, global tables, regular, and ternary tables.

3.1 Variables and Domains

Consider the Nonogram show at the left of Figure 2.1 and its solution shown at the right of this figure. The cells of the Nonogram puzzle are modeled as CSP variables with Boolean domains. The variable $V_{i,j}$ represents the cell at row i and column j. $V_{i,j} \leftarrow 0$ indicates that the cell blank and $V_{i,j} \leftarrow 1$ indicates that the cell is colored.

The models with global-table and regular constraints have no other CSP variables. The model with ternary-table constraints results from reformulating the regular constraints [Bessière *et al.*, 2008] and requires adding additional variables as discussed in Section 3.4.

3.2 Global-Table Constraints

This model has one global constraint per row and one per column of the puzzle. Each constraint lists as tuples of *supports* all consistent combinations. This model is the most consuming in terms of space storage (i.e., the largest in space).

In the Nonogram in Figure 2.1, the label of the first row, $L_{R_1} = [1, 1]$, yields a constraint whose scope consists of the five cells $\{V_{1,1}, V_{1,2}, V_{1,3}, V_{1,4}, V_{1,5}\}$ of the top row and definition is the following set of tuples:

$$\{(1,0,1,0,0), (1,0,0,1,0), (1,0,0,0,1), (0,1,0,1,0), (0,1,0,0,1), (0,0,1,0,1)\}$$

3.3 Regular Constraints

The Nonogram puzzle can be modeled using the regular constraint introduced by Pesant [2004]. A regular constraint is defined by `regular(`$\mathbf{x}, M$`)` where:

- $\mathbf{x}$ is a sequence of finite-domain variables and
- M is a *deterministic finite automaton* (DFA) described as $M = (Q, \Sigma, \delta, q_0, F)$ in which, Q is a finite set of states, Σ is the alphabet, $\delta : Q \times \Sigma \to Q$ is the transition between states, q_0 is the initial state, and $F \subseteq Q$ is the finite set of final states.

3.3.1 Construction of the DFA

In a Nonogram puzzle, each row and column is modeled by a regular constraint [Pesant, 2004]. Thus, in a puzzle of size $r \times c$, there are $(r + c)$ regular constraints. The regular constraint of a given row or column with the label $L = [l_1, l_2, \ldots, l_n]$ is generated according to the following steps:

STEP 1: **x** is the sequence of all cells from left to right in a row and top to bottom in a column.

STEP 2: The first state of the DFA is the state q_0 with a self-loop labelled 0 because it is always possible to add blank cells at the beginning as long as the colored cells form a valid solution. This first state, shown in Figure 3.1, is the beginning of a 'current DFA,' which is expanded in the following steps.

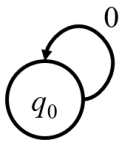

Figure 3.1: The first state of the current DFA

STEP 3: For every integer l_i in L:

1. Form a chain of l_i new states connected to each other by a directed edge labelled 1. The last state of the current DFA is connected to the first state in the chain by a directed edge labelled 1. For example, with $l_1 = 2$, we have the situation shown in Figure 3.2. For $l_1 = k > 2$, the chain is illustrated in Figure 3.3. Doing so forces the row or column of the puzzle

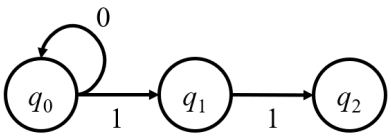

Figure 3.2: The current DFA for $l_1 = 2$

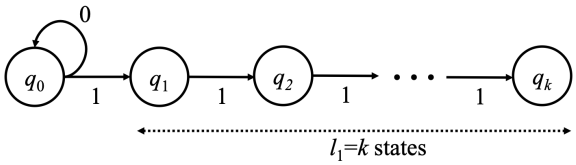

Figure 3.3: The current DFA for $l_1 = k > 2$

to have a sequence of l_i contiguous colored cells.

2. Next, there are two cases to consider:

- Case $i = n$, l_i is the last integer in L and no new states need to be added (to Q). At this point, we add a self-loop labelled 0 on the last state of the current DFA because, similar to the first state, adding blank cells at the end of a consistent solution does not change its correctness. This situation is illustrated in Figure 3.4 where q_f is the final state of the current DFA.

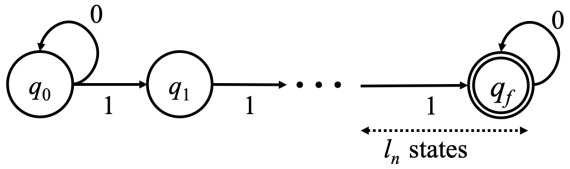

Figure 3.4: The current DFA for $i = n$ with the last state q_f

- Case $i < n$, we must add at least one blank cell, in the puzzle, between two contiguous blocks of colored cells. This requirement is achieved by adding a *new* state with a self-loop labelled 0. Then, we connect the current DFA to this state by a directed edge labelled 0. Figure 3.5 illustrates the situation for a current DFA with $l_1 = k$ and $n > 1$.

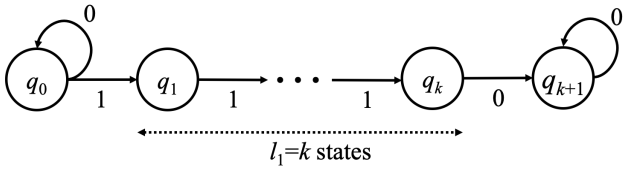

Figure 3.5: The current DFA for $l_1 = k$ and $n > 1$

STEP 4: The last state in the DFA is the single final state q_f, which is the last state of the chain of states corresponding to l_n.

For each state $q_i \in Q$, i denotes the timestamp of its creation.

3.3.2 Example of the DFA of a Regular Constraint

Figure 3.6 shows the DFA of the first row $L_{R_1} = [1, 1]$ in the Nonogram example of Figure 2.1. $Q_{R_1} = \{q_{r,0}, q_{r,1}, q_{r,2}, q_{r,3}\}$ has four states. The alphabet Σ_{R_1} is $\{0, 1\}$,

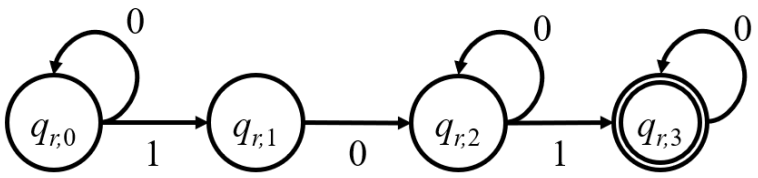

Figure 3.6: DFA of the first row in the example

which is the set of values labelling the directed edges. The initial state is $q_{r,0}$. The set of final states F_{R_1} consists of only one state $\{q_{r,3}\}$. According to the DFA, the row of the Nonogram can start with no, a single, or multiple 0's (i.e., blank cells), then transition to $q_{r,1}$ by coloring a single cell because the edge $(q_{r,0}, q_{r,1})$ is labelled 1. Because the edge $(q_{r,1}, q_{r,2})$ is labelled 0, the following cell must be blank. Because $q_{r,2}$ has a self-loop labelled 0, we can have multiple blank cells. Because the edge $(q_{r,2}, q_{r,3})$ is labelled 1, we must color one more single cell. Finally, the self-loop on $q_{r,3}$ is labelled 0 and indicates that we can follow with multiple blank cells.

Figure 3.7 shows the DFA of the first column $L_{C_1} = [1, 3]$ in the Nonogram example of Figure 2.1. The set of states $Q_{C_1} = \{q_{c,0}, q_{c,1}, q_{c,2}, q_{c,3}, q_{c,4}, q_{c,5}\}$. The alphabet Σ_{C_1}

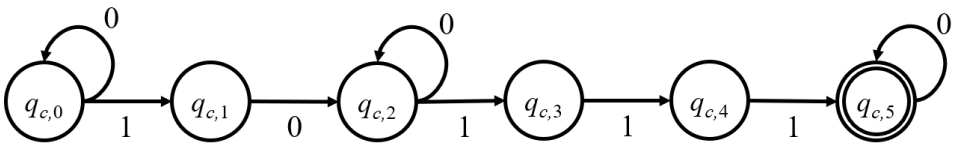

Figure 3.7: DFA of the first column in the example

is again $\{0, 1\}$. The initial state is $q_{c,0}$. The set of final states $F_{C_1} = \{q_{c,5}\}$. Similar to the first row, the first column of the Nonogram can start with no, a single, or multiple 0's (i.e., blank cells), then transition to $q_{c,1}$ by coloring a single cell because

the edge $(q_{c,0}, q_{c,1})$ is labelled 1. Then, we can have multiple blank cells because the edge $(q_{c,1}, q_{c,2})$ is labelled 0 and $q_{c,2}$ has a self-loop labelled 0. Because the three next states $q_{c,3}$, $q_{c,4}$, and $q_{c,5}$ form a chain connected by directed edges labelled 1, we must color the next three contiguous cells. Finally, we can follow with multiple blank cells because the self-loop on $q_{c,5}$ is labelled 0.

3.3.3 Size of the DFA

According to the **STEP** 3 in Section 3.3.1:

- The DFA starts with a single initial state q_0.
- Every l_i, where $0 \leq i < n$, is represented by a chain of l_i states connected with directed edges labelled 1 and a single state with a self-loop labelled 0.
- l_n is represented by exactly l_n states.

Hence, the total number of states of the DFA:

$$|Q| = \left(\sum_{i=1}^{n} l_i \right) + n. \tag{3.1}$$

Along with Section 2.5.2, the number of states $|Q|$ must satisfy the following constraints:

- For a row, $|Q| \leq c + 1$.
- For a column, $|Q| \leq r + 1$.

3.4 Ternary-Table Constraints

We build the ternary-table constraint model of the Nonogram puzzle by reformulating each of the regular constraints of the model in Section 3.4 into a set of ternary constraints [Bessière *et al.*, 2008].

3.4.1 The Slide Constraint

A regular constraint on a row or a column of a Nonogram `regular(x,` M`)` can be encoded as the constraint $\textsc{Slide}_2(F, [a_0, x_1, a_1, \ldots, x_n, a_n])$ [Bessière *et al.*, 2008] where:

- $n = c$ if the regular constraint is defined over a row of the Nonogram and $n = r$ if it is defined over a column.
- $a_0, a_1, \ldots, a_n$ are auxiliary CSP variables that are added to the CSP variables representing the states of the DFA. The domain for every auxiliary variable a_i, $dom(a_i)$, is a subset of Q (i.e., $dom(a_i) \subseteq Q$) where Q is the set of all states in the DFA, so $|dom(a_i)| \leq |Q| \leq \texttt{max}(r, c) + 1$.
  - The domain of a_0 is the singleton of the initial state (i.e., $dom(a_0) = \{q_0\}$).
  - The domain of a_i where $0 < i < n$ is the set of all states (i.e., $dom(a_i) = Q$).
  - The domain of a_n is the set of final states.
- F is a constraint ensuring that, for every tuple (u, w, v) assigned to the variables a_i, x_{i+1}, and a_{i+1}, where $u \in dom(a_i)$, $w \in \{0, 1\}$, and $v \in dom(a_{i+1})$, we have $v = \delta(u, w)$. In other words, (u, w, v) must represent a valid transition in the DFA in which u and v are two states that are connected by a directed edge labelled w.

3.4.2 The Slide Constraint as Ternary-Table Constraints

We can rewrite every SLIDE constraint [Bessière *et al.*, 2008] on $\mathbf{x}$ (where $|\mathbf{x}| = n$) as a set of n ternary constraints. Each ternary constraint is defined over three variables $\{a_i, x_{i+1}, a_{i+1}\}$ where $0 \leq i \leq n-1$, and its set of tuples is defined as

$$\{(u, w, v) \mid (u \in dom(a_i)) \wedge (w \in \{0, 1\}) \wedge (v \in dom(a_{i+1})) \wedge (v = \delta(u, w))\}$$

This process is illustrated in Figure 3.8.

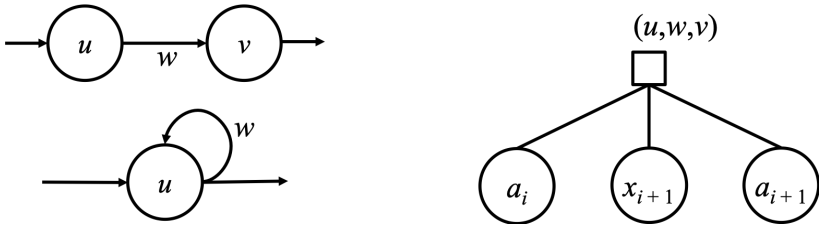

Figure 3.8: Generation of a tuple of a ternary constraint

For example, the ternary constraints C_1 and C_2 on the first two cells of any row are shown in Figure 3.9. Notice that two contiguous ternary constraints intersect on

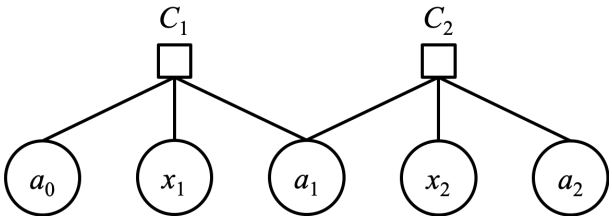

Figure 3.9: Example of ternary constraints

exactly one auxiliary variable.

3.4.3 Size of the CSP Model

Given a Nonogram puzzle of size $r \times c$, every regular constraint [Pesant, 2004] on a row of c cells is encoded into c ternary constraints with $c+1$ auxiliary variables. Also, every regular constraint on a column of r cells is encoded into r ternary constraints with $r+1$ auxiliary variables. The total number of variables is $(3rc+r+c)$, which consists of $r \times c$ variables representing the cells of the Nonogram and $(2rc + r + c)$ auxiliary variables. The total number of ternary constraints obtained from reformulating $(r+c)$ regular constraints is $2rc$ in which $(r \times c)$ constraints represent r rows and the other $(r \times c)$ constraints represent c columns.

The number of tuples of every ternary constraints is bounded by the number of transitions in the corresponding DFA. Because there are $|Q|$ states in the DFA of the regular constraint of a row or column and two possible labels $\{0, 1\}$ for the edges between states, the number of transitions is bounded by $2|Q| \leq 2\texttt{max}(r, c) + 2$. Therefore, the number of tuples of every ternary constraint is also bounded by $2\texttt{max}(r, c) + 2$.

3.4.4 Examples of the Conversion

Figure 3.10 shows the constraint sub-network of ternary table constraints modeling the first row of the Nonogram of Figure 2.1.

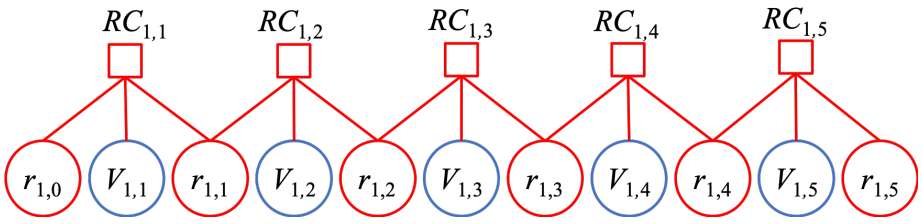

Figure 3.10: Constraint network with ternary tables for the first row of Figure 2.1

The CSP variables are $V_{1,1}, V_{1,2}, \ldots, V_{1,5}$ (shown in blue) corresponding to the five cells in the first row and $r_{1,0}, r_{1,1}, \ldots, r_{1,5}$ (shown in red) corresponding to the auxiliary variables. $V_{1,1}, V_{1,2}, \ldots, V_{1,5}$ are Boolean variables. The domains of the auxiliary variables are as follows:

$$dom(r_{1,0}) = \{q_{r,0}\}$$
$$dom(r_{1,1}) = \{q_{r,0}, q_{r,1}, q_{r,2}, q_{r,3}\}$$
$$dom(r_{1,2}) = \{q_{r,0}, q_{r,1}, q_{r,2}, q_{r,3}\}$$
$$dom(r_{1,3}) = \{q_{r,0}, q_{r,1}, q_{r,2}, q_{r,3}\}$$
$$dom(r_{1,4}) = \{q_{r,0}, q_{r,1}, q_{r,2}, q_{r,3}\}$$
$$dom(r_{1,5}) = \{q_{r,3}\}$$

The tuples of five ternary constraints $RC_{1,1}, RC_{1,2}, \ldots, RC_{1,5}$ are listed in Table 3.1. Every ternary constraint $RC_{1,i}$ over the variables $\{r_{1,i-1}, V_{1,i}, r_{1,i}\}$ where

Table 3.1: Tuples of the ternary constraints for the first row of Figure 2.1

Constraint	**Tuples**
$RC_{1,1}$	$\{(q_{r,0}, 0, q_{r,0}), (q_{r,0}, 1, q_{r,1})\}$
$RC_{1,2}$ $RC_{1,3}$ $RC_{1,4}$	$\{(q_{r,0}, 0, q_{r,0}), (q_{r,0}, 1, q_{r,1}), (q_{r,1}, 0, q_{r,2}),$ $(q_{r,2}, 0, q_{r,2}), (q_{r,2}, 1, q_{r,3}), (q_{r,3}, 0, q_{r,3})\}$
$RC_{1,5}$	$\{(q_{r,2}, 1, q_{r,3}), (q_{r,3}, 0, q_{r,3})\}$

$2 \leq i \leq 4$ has six tuples representing six valid transitions in the DFA shown in Figure 3.6. Because $dom(r_{1,0}) = \{q_{r,0}\}$ consists of only one value (i.e., $q_{r,0}$), the constraint $RC_{1,1}$ can only accept two transitions, namely, $(q_{r,0}, 0, q_{r,0})$ and $(q_{r,0}, 1, q_{r,1})$. The last constraint in the row $RC_{1,5}$ is defined over $\{r_{1,4}, V_{1,5}, r_{1,5}\}$. It has a similar requirement because $dom(r_{1,5}) = \{q_{r,3}\}$ consists of only one value (i.e., $q_{r,3}$). Hence, $RC_{1,5}$ has the following allowed tuples $(q_{r,2}, 1, q_{r,3})$ and $(q_{r,3}, 0, q_{r,3})$.

Similarly, Figure 3.11 shows the constraint sub-network of the first column of the Nonogram of Figure 2.1.

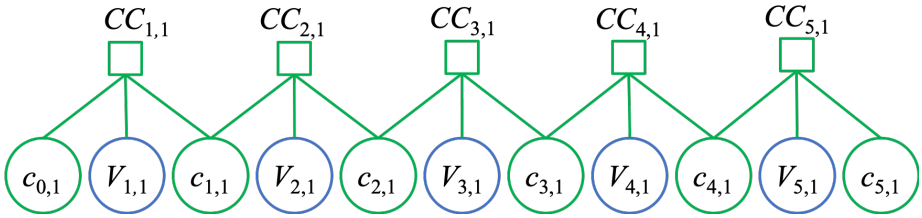

Figure 3.11: Constraint network with ternary tables for the first column of Figure 2.1

The CSP Boolean variables are $V_{1,1}, V_{2,1}, \ldots, V_{5,1}$ (shown in blue) corresponding to the five cells in the first column and $c_{0,1}, c_{1,1}, \ldots, c_{5,1}$ (shown in green) corresponding to the auxiliary variables. The domains of the auxiliary variables are as follows:

$$dom(c_{0,1}) = \{q_{c,0}\}$$
$$dom(c_{1,1}) = \{q_{c,0}, q_{c,1}, q_{c,2}, q_{c,3}, q_{c,4}, q_{c,5}\}$$
$$dom(c_{2,1}) = \{q_{c,0}, q_{c,1}, q_{c,2}, q_{c,3}, q_{c,4}, q_{c,5}\}$$
$$dom(c_{3,1}) = \{q_{c,0}, q_{c,1}, q_{c,2}, q_{c,3}, q_{c,4}, q_{c,5}\}$$
$$dom(c_{4,1}) = \{q_{c,0}, q_{c,1}, q_{c,2}, q_{c,3}, q_{c,4}, q_{c,5}\}$$
$$dom(c_{5,1}) = \{q_{c,5}\}$$

The tuples of five ternary constraints $CC_{1,1}, CC_{2,1}, \ldots, CC_{5,1}$ are listed in Table 3.2. Every ternary constraint $CC_{i,1}$ over the variables $\{c_{i-1,1}, V_{i,1}, c_{i,1}\}$ where $2 \leq i \leq 4$ has eight tuples representing eight valid transitions in the DFA shown in Figure 3.7. The constraint $CC_{1,1}$ also accepts two transitions, namely, $(q_{c,0}, 0, q_{c,0})$ and $(q_{c,0}, 1, q_{c,1})$ because $dom(c_{0,1}) = \{q_{c,0}\}$. The last constraint in the column $CC_{5,1}$ defined over $\{c_{4,1}, V_{5,1}, c_{5,1}\}$ has the following allowed tuples $(q_{c,4}, 1, q_{c,5})$ and $(q_{c,5}, 0, q_{c,5})$ because $dom(c_{5,1}) = \{q_{c,5}\}$.

Table 3.2: Tuples of the ternary constraints for the first column of Figure 2.1

Constraint	**Tuples**
$CC_{1,1}$	$\{(q_{c,0}, 0, q_{c,0}), (q_{c,0}, 1, q_{c,1})\}$
$CC_{2,1}$ $CC_{3,1}$ $CC_{4,1}$	$\{(q_{c,0}, 0, q_{c,0}), (q_{c,0}, 1, q_{c,1}), (q_{c,1}, 0, q_{c,2}), (q_{c,2}, 0, q_{c,2}),$ $(q_{c,2}, 1, q_{c,3}), (q_{c,3}, 1, q_{c,4}), (q_{c,4}, 1, q_{c,5}), (q_{c,5}, 0, q_{c,5})\}$
$CC_{5,1}$	$\{(q_{c,4}, 1, q_{c,5}), (q_{c,5}, 0, q_{c,5})\}$

3.4.5 Generation of the XCSP3 Files

Without loss of generality, we replace the values in the domains of the auxillary variables with integers in the XCSP3 files of the ternary-table constraint models that we generate. For example, we refer to the first state in the DFA of a regular constraint as state 0 instead of q_0.

Given an XCSP3 file of a CSP instance with regular constraints [Pesant, 2004] representing an $r \times c$ Nonogram puzzle, we generate the XCSP3 file of the corresponding model with ternary constraints as follow:

STEP 1: Generate a new CSP instance with $|\mathcal{V}| = r \times c$ Boolean variables. These variables represent the cells of the Nonogram puzzle.

STEP 2: For each of the $r \times c$ regular constraints in the original regular-constraint model:

1. Add to $\mathcal{V}$ a new auxiliary variable a_0 with a domain consisting of exactly one value 0 (i.e., the initial state of the DFA).
2. Associate to each state $q_i \in Q$ of the DFA, the integer i.
3. For each variable x_i in the scope of the considered regular constraint:[1]

[1]The variables in the scope of a regular constraint are the Boolean variables corresponding to the cells of the puzzle.

a) If the variable is the last variable in the scope of the regular constraint, add to $\mathcal{V}$ a new auxiliary variable a_i whose domain is the singleton $\{|Q|-1\}$. Otherwise, add to $\mathcal{V}$ a new auxiliary variable a_i and set its domain to $\{0, 1, \ldots, |Q|-1\}$.

b) Add to $\mathcal{C}$ a new constraint whose scope $\{a_{i-1}, x_i, a_i\}$ consists of x_i and the two most recent auxiliary variables (i.e., a_{i-1} and a_i) added to $\mathcal{V}$. The set of allowed tuples of the new constraint is set to the intersection of the Cartesian product of the domains of these three variables $dom(a_{i-1}) \times dom(x_i) \times dom(a_i)$ and the set of 'transitions' specified in the regular constraint (after replacing q_i by i).

STEP 4: Export the new CSP instance to XML file using XCSP3 format.[2]

3.5 Graphical Representation

The primal graph of the global (both regular and table) constraints model consists of vertices arranged in a grid. All the vertices of any given column (row) are connected in a clique. As for dual graph, it is composed of a complete bipartite graph whose vertices are, on one side, the dual variables corresponding to "row" constraints and, on the other side, those corresponding to the "column" constraints.[3]

The graphical representations of the ternary-tables model are more unusual. Figure 3.12 and Figure 3.13 show the primal and dual constraint graphs, respectively, of the model using ternary-tables. The figures show in red the constraints along the rows, in green, the constraints along the columns, and in blue the vertices/edges corresponding to the cells of the Nonogram.

[2] http://www.xcsp.org/specifications

[3] Similar to the graphs of a Sudoku puzzle.

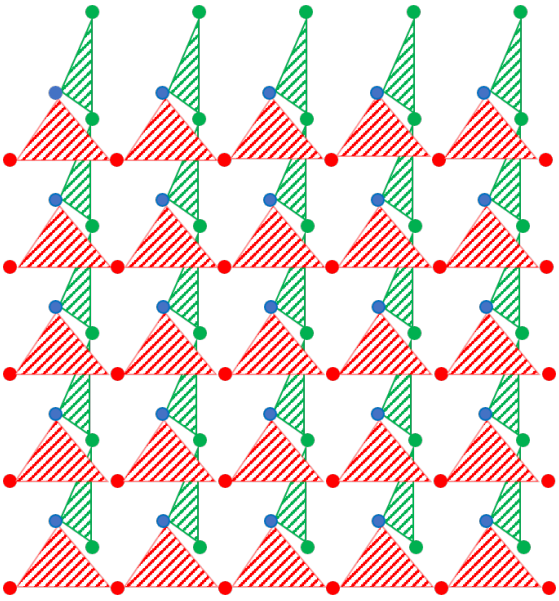

Figure 3.12: Primal graph (ternary-tables)

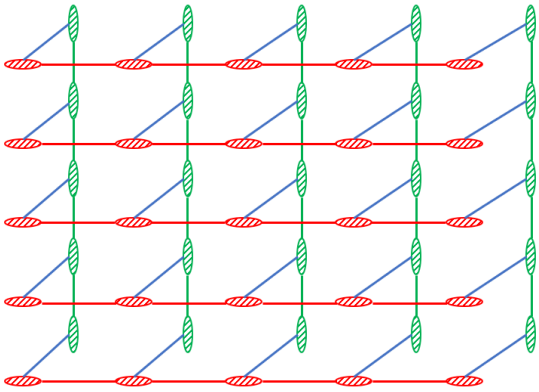

Figure 3.13: Dual graph (ternary-tables)

Summary

In this chapter, we discussed three constraint models for the Nonogram puzzle as summarized in Table 3.3 for a puzzle with r rows and c columns:

Table 3.3: Three constraint models of an $r \times c$ Nonogram puzzle

Model	**#Variables**	**Domains**	**Constraints**
Global-table	$r \times c$	$\{0, 1\}$	r constraints of arity c with $\mathcal{O}(2^c)$ tuples, c constraints of arity r with $\mathcal{O}(2^r)$ tuples
Regular	$r \times c$	$\{0, 1\}$	$r + c$ regular constraints
Ternary-table	$r \times c$ $2rc + r + c$	$\{0, 1\}$ $\subseteq \{0, 1, \ldots, \lvert Q \rvert\}$	$2rc$ ternary constraints, each with $\mathcal{O}(2\texttt{max}(r, c) + 2)$ tuples

Chapter 4

Experimental Setup

In this chapter, we describe our experimental setup for testing different models of the Nonogram. We list the datasets of Nonogram instances used for testing and different factors in two experiments that we conducted. All the experiments were run on Crane, a computer cluster of Intel Xeon E5-2670 2.60GHz provided by the Holland Computing Center (HCC) at University of Nebraska-Lincoln (UNL).

4.1 Datasets

We use the Nonogram benchmarks available on the XCSP3 website of benchmark problems.[1,2] Two series are available on the site:

1. 174 instances with *regular model* using one regular constraint [Pesant, 2004] per row and one per column.

2. 170 instances with *global-tables model* using one large table constraint per row and one per column of enumerated support tuples. Table A.1 in Appendix A

[1] http://www.xcsp.org/series,
[2] The original puzzles were contributed by Gilles Pesant.

lists detailed information of each instance.

We apply the process described in Section 3.4.5 on the set of 174 regular-model instances to obtain 174 corresponding instances with *ternary-table constraint model* and store them in XCSP3 format. Table A.2 in Appendix A lists detailed information of each instance. The CSP parameters of the three sets of instances are shown in Table 4.1.

Table 4.1: Parameters of the constraint models

Parameter		**Regular**	**Global-table**	**Ternary-table**
$\mid \mathcal{V} \mid$		[256, 1,056]	[256, 1,056]	[800, 3,233]
$\mid \mathcal{C} \mid$		[32, 65]	[32, 65]	[512, 2,112]
Domain size	min	2	2	1
	max			[9, 33]
Arity	min	[15, 32]	[15, 32]	3
	max	[16, 35]	[16, 35]	
# Tuples/constraint	min	N/A	[1, 21]	[1, 2]
	max		[66, 54,264]	[11, 46]

4.2 Tested Factors

We evaluate the performance of solving the above two datasets of the Nonogram puzzle using different combinations of the following three factors:

Consistency algorithms: 14 algorithms discussed in Section 2.4, namely:

- GAC: CT [Demeulenaere *et al.*, 2016], GAC2001 [Bessière *et al.*, 2005], and STR2 [Lecoutre, 2011].
- Singleton consistency: SAC1 [Debruyne and Bessière, 1997], POAC1 [Balafrej *et al.*, 2014], and APOAC [Balafrej *et al.*, 2014], as well as the following algorithms ANPOAC, APOAC$_{around\cup}$, PrePeak^{+}-POAC1, A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018].

- Relational consistency: R(∗,3)C, R(∗,4)C, wR(∗,3)C, and wR(∗,4)C [Karakashian *et al.*, 2010].

Ordering heuristics: Two variable ordering heuristics discussed in Section 2.3: dom/deg [Bessière and Régin, 1996] and dom/wdeg [Boussemart *et al.*, 2004].

Propagation queue: Two constraint queue types discussed in Section 2.4.1: FIFO queue and lexicographical queue.

The tested factors are summarized in Table 4.2.

Table 4.2: Tested factors

Factor	**Variation**
Algorithm	GAC2001, STR2, CT, SAC1, POAC1, APOAC, PrePeak^{+}-POAC1 ANPOAC, APOAC$_{around\cup}$, A$\cup_{cyc}^{bfsc}$POAC, R(∗,3)C, R(∗,4)C, wR(∗,3)C, wR(∗,4)C
Ordering heuristic	dom/deg, dom/wdeg
Propagation queue	FIFO, lexicographical

4.3 Conducted Experiments

We set up two experiments to examine the performance of backtrack search with various consistency algorithms discussed in Section 2.4 on two models of the Nonogram puzzle as a CSP, namely, the global-tables model and the ternary-tables model.

4.3.1 Comparing GAC-Based Algorithms

First, we compare the performance of three GAC-based algorithms, namely, GAC2001 [Bessière *et al.*, 2005], STR2 [Lecoutre, 2011], and CT [Demeulenaere *et al.*, 2016] on both datasets for each ordering heuristic and each propagation queue. Because

lexicographical queue is not supported in the CT implementation of STAMPEDE, there are 20 different combinations of the factors in total: ten on the ternary-table constraint model and the other ten on the global-table constraint model.

We set a time limit of two hours (i.e., 7,200 seconds) per instance with 12GB of memory.

4.3.2 Comparing All Consistency Algorithms

In this experiment, we use dom/wdeg [Boussemart *et al.*, 2004] as the variable ordering heuristic and FIFO propagation queue because they exhibit the best performance in the previous experiment (as we discuss in Section 5.1). Then, we apply all 14 consistency algorithms, one at a time, on the instances of the two datasets introduced in Section 4.1.

Singleton-based consistency algorithms enforce GAC. In STAMPEDE:

- SAC1 [Debruyne and Bessière, 1997], along with ANPOAC, APOAC$_{around\cup}$, PREPEAK^{+}-POAC1, and A$\cup^{bfsc}_{cyc}$POAC [Woodward, 2018] are implemented using STR2.
- POAC1 and APOAC [Balafrej *et al.*, 2014] are implemented using GAC2001.

As a result, the performance of the HLC algorithm reflects the performance of the GAC algorithm that it uses.

We set the time limit per instance to four hours (i.e., 14,400 seconds) and limit the memory to 12GB.

4.4 Evaluation Criteria

We summarize the performance of solving the puzzle by reporting:

- The number of instances solved (#Solved) out of the number of available instances in the dataset
- The average number of backtracks (avg. # BT) computed over instances solved by all algorithms
- The average number of nodes visited (avg. # NV) computed over instances solved by all algorithms
- The average CPU time, in seconds, over instances solved by all algorithms
- The average CPU time in seconds computer over all instances. If a combination fails to terminate within the time limit, we add the corresponding time limit (i.e., 7,200 seconds in the first experiment and 14,400 seconds in the second experiment) to the CPU time of that algorithm before computing the average. We indicate with the sign '>' that the algorithm did not terminate on one or more instances.
- The number of instances in the dataset on which each algorithm has the best performance (i.e., the smallest CPU time)
- The number of instances in the dataset that an algorithm can solve in a backtrack-free manner (i.e., #BT is 0)

The column for each criterion in the summary tables (Tables 5.2 to 5.5 in Chapter 5) are automatically formatted with a background color-gradient from green to yellow then red, with green being the best performance and red the worst.

Finally, we statistically rank the performance of the tested algorithms using paired t-tests with the average CPU time on all instances.

Summary

In this chapter, we reviewed the two datasets of Nonogram instances used in our experiments, namely, the global-tables and the ternary-tables. Then, we articulated the three different factors that yield the tested algorithm combination, namely, consistency algorithms, ordering heuristics, and propagation queue. Finally, we described two large experiments that we conducted in order to rank the consistency algorithms in terms of their performance for solving the Nonogram puzzle.

Chapter 5

Empirical Evaluation

In this chapter, we report the results of our experiments for assessing the performance of various consistency algorithms as lookahead during backtrack search on two constraint models of the Nonogram puzzle, namely, the model with global-table constraints and the one with ternary-table constraints.

Note that the experiments discussed in Section 5.1 are given two (2) hours per instance and those reported in Section 5.2 are given four (4) hours per instance, which results in slight differences in the conclusions of the statistical tests.

5.1 Comparing GAC-Based Algorithms

In this section, we discuss the results of our first experiment using the three algorithms for enforcing GAC, namely, GAC2001 [Bessière *et al.*, 2005], STR2 [Lecoutre, 2011], and CT [Demeulenaere *et al.*, 2016]. All three algorithms yield the same filtering. However, GAC2001 does not touch the table constraints whereas STR2 and CT filter the constraints first then the variable domains. The goal of this experiment is to:

1. Ensure the correctness of all three algorithms by verifying that they traverse

the same tree when ties are broken in the same way.

2. Evaluate whether or not the performance of these algorithms is affected by the arity of the constraints. Indeed, such an experiment was never carried out before and this aspect is in our context (arity of the constraints in the global-tables model varies in [15,35]).

We test the ten (10) combinations shown in Table 5.1, where 'lex' designates lexicographical ordering of the constraints in the constraint propagation-queue:

Table 5.1: GAC algorithms tested

Algorithm	**Ordering heuristic (dynamic)**	**Propagation-queue management**
CT	dom/deg don/wdeg	FIFO
STR2	dom/deg dom/wdeg	lex FIFO
GAC2001	dom/deg dom/wdeg	lex FIFO

Sections 5.1.1 and 5.1.2 discuss the results on the models with ternary-table constraints and global-table constraints, respectively. From examining the respective results shown in Table 5.2 and Table 5.3, we conclude that:

- The ordering dom/wdeg [Boussemart *et al.*, 2004] significantly outperforms dom/deg [Bessière and Régin, 1996] (overwhelming green for dom/wdeg versus reddish for dom/deg) in every criterion and for both models.

- Instance #69 is not solved by any of the GAC algorithms when the runtime is limited to two (2) hours.

- Solving the model with global-table constraints takes one or two orders of magnitude more time than solving the model with ternary-table constraints. A

careful examination of the processing time shows that significant effort is spent on loading the huge tables (up to 54,264 tuples per constraint, see Table 4.1) and setting the corresponding data structures.

5.1.1 Ternary-Tables Model

Table 5.2 summarizes the performance of the GAC-based algorithms on the instances of ternary-tables model. Further, Tables B.1 and B.2 in Appendix B provide the

Table 5.2: All combinations of GAC-based algorithms on ternary-tables model

# Instances 174 in total: 169 solved by all and 173 solved by at least one									
				On instances solved by all algorithms					
Ordering heuristic	**Propagation queue**	**Algorithm**	**# Solved**	**avg. # BT**	**avg. # NV**	**avg. CPU time (s)**	**avg. CPU time on all instances (s)**	**Best on x instances**	**# Instances solved backtrack free**
dom/deg	lex	STR2	169	352,601.96	385,977.19	49.48	>254.95	0	55
		GAC2001	170	352,601.96	385,977.19	40.61	>232.28	34	55
	FIFO	CT	171	352,601.96	385,977.19	21.91	>175.96	0	55
		STR2	169	352,601.96	385,977.19	34.77	>240.67	0	55
		GAC2001	171	352,601.96	385,977.19	28.16	>190.67	57	55
dom/wdeg	lex	STR2	173	20,998.86	24,070.84	4.66	>46.05	0	52
		GAC2001	173	20,998.86	24,070.84	3.56	>44.95	20	52
	FIFO	CT	173	3,016.45	5,121.00	2.00	>43.49	1	55
		STR2	173	8,153.52	10,428.38	2.49	>43.91	1	56
		GAC2001	173	3,016.45	5,121.00	1.39	>42.90	60	55

detailed results on all tested instances.

Overall, we make the following general observations:

- Everything else being equal, GAC2001 [Bessière *et al.*, 2005] outperforms CT

[Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] on the largest number of solved instances and the number of instances on which it is the 'quickest' (i.e., best).

- All configurations included, GAC2001 [Bessière *et al.*, 2005] configurations exhibit the best performance on 171 out of the 173 solved instances.
- STR2 [Lecoutre, 2011] limits the number of instances solved by all algorithms to 169. Indeed, it is the algorithm that exhibits the worst CPU time under all tested configurations.

5.1.1.1 Using dom/deg ordering heuristic

We make the following observations:

- All algorithms traverse the same tree (i.e., have the same avg. # BT and the same avg. # NV) and solve the same number of instances (55) in a backtrack-free manner.
- None of the three algorithms can solve the instances #69, #137, and #167 within two (2) hours. Further, STR2 [Lecoutre, 2011] does not terminate on instances #44 and #153 within this time limit.
- A detailed analysis of the CPU time of GAC2001 [Bessière *et al.*, 2005] and CT [Demeulenaere *et al.*, 2016] shows that GAC2001 performs best on all instances except for the most difficult ones (about 23 instances), on which CT outperforms GAC2001. As a result, CT shows the smallest average CPU time on instances solved by all algorithms with dom/deg [Bessière and Régin, 1996] and FIFO.

5.1.1.2 Using dom/wdeg ordering heuristic

It is well known that dom/wdeg [Boussemart *et al.*, 2004], in general, outperforms other ordering heuristics. As argued above, this result is confirmed in this experiment.

This experiment shows that, in general, FIFO seems to outperform lex.[1] A better ordering of the queue would have used the weights of the constraints for the propagation queue. However, this option is currently not readily available on STAMPEDE.[2] As a result, the best one can do under our circumstances is to use dom/wdeg [Boussemart *et al.*, 2004] and FIFO.

Under these conditions (i.e., dom/wdeg and FIFO), STR2 [Lecoutre, 2011] traverses a significantly larger search tree than CT [Demeulenaere *et al.*, 2016] and GAC2001 [Bessière *et al.*, 2005]. The CPU time of STR2 is also larger. It appears that STR2 is a poor choice for the model with ternary-table constraints. Unfortunately, it is the algorithm used to implement the high-level consistency algorithms explored in Section 5.2.

5.1.1.3 Pairwise Statistical Testing

Comparing pairwise the ten (10) configurations of Table 5.1 using paired t-test, we obtain the 'comparability' graph shown in Figure 5.1, which shows that GAC2001 [Bessière *et al.*, 2005] is a preferable choice on ternary-table constraints. Note that a statistical dominance cannot be established if a path between two algorithms does not exist.

[1] lex is not implemented with CT [Demeulenaere *et al.*, 2016] on STAMPEDE.
[2] We ran out of time for providing it.

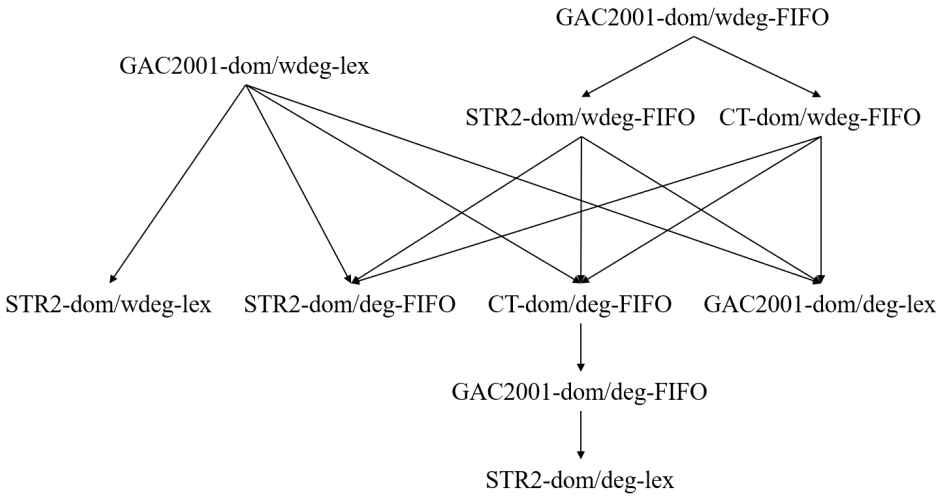

Figure 5.1: Comparing the ten configurations of GAC-based algorithms on the ternary-tables model

5.1.2 Global-Tables Model

Table 5.3 summarizes the performance of the ten (10) configurations of GAC algorithms on global-tables model. Tables B.3 and B.4 in Appendix B provide the detailed results on all tested instances. Out of the 170 instances available,[3] only 122 are solved by all tested algorithms and 130 instances solved by at least one algorithm.

Some of our observations on the global-tables model are the same as on the ternary-tables model (Section 5.1.1), including:

- All algorithms traverse the same tree (i.e., have the same avg. # BT and # NV) except for STR2 [Lecoutre, 2011] with dom/wdeg [Boussemart *et al.*, 2004] and FIFO, which traverses a slightly smaller tree and takes slightly less CPU time.

- With dom/deg [Bessière and Régin, 1996] ordering heuristic, all configurations solve the same (42) instances in a backtrack-free manner.

[3]Global-table constraints require significant storage space.

Table 5.3: All combinations of GAC-based algorithms on global-tables model

# Instances 170 in total: 122 solved by all and 130 solved by at least one										
Ordering heuristic	**Propagation queue**	**Algorithm**	**# Solved**	**On instances solved by all algorithms**			**avg. CPU time on all instances (s)**	**Best on x instances**	**# Instances solved backtrack free**
				avg. # BT	**avg. # NV**	**avg. CPU time (s)**			
dom/deg	lex	STR2	125	129,419.43	148,550.63	602.75	>2,538.21	7	42
		GAC2001	125	129,419.43	148,550.63	741.75	>2,639.53	1	42
	FIFO	CT	126	129,419.43	148,550.63	569.62	>2,482.60	11	42
		STR2	126	129,419.43	148,550.63	598.70	>2,542.20	43	42
		GAC2001	125	129,419.43	148,550.63	758.16	>2,649.82	3	42
dom/wdeg	lex	STR2	128	4,748.34	5,774.34	565.88	>2,374.16	7	42
		GAC2001	128	4,748.34	5,774.34	570.92	>2,388.80	1	41
	FIFO	CT	129	1,514.03	2,174.25	564.98	>2,463.42	39	44
		STR2	128	1,370.44	2,034.14	565.13	>2,437.29	27	44
		GAC2001	129	1,514.03	2,174.25	566.46	>2,465.64	1	44

- As stated above, dom/wdeg [Boussemart *et al.*, 2004] outperforms dom/deg [Bessière and Régin, 1996] and improves every algorithm in almost every evaluation criterion.

However, contrary to the results on the model with ternary tables (Section 5.1.1), CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] exhibit a better performance than GAC2001 [Bessière *et al.*, 2005] on the model with global tables, both in CPU time and the number of instances on which they perform best. The difference in performance is more visible with the dom/deg [Bessière and Régin, 1996] ordering heuristic:

- Using FIFO propagation queue, CT [Demeulenaere *et al.*, 2016] and STR2

[Lecoutre, 2011] solve one more instance (instance #51) than GAC2001 [Bessière *et al.*, 2005]. They also have smaller average CPU time.

- Using lexicographical propagation queue, STR2 [Lecoutre, 2011] has a smaller CPU time than GAC2001 [Bessière *et al.*, 2005] although they solve the same number of instances (125).

We observe similar behaviors in the case of dom/wdeg [Boussemart *et al.*, 2004] ordering heuristic. Although GAC2001 [Bessière *et al.*, 2005] solves one more instance (instance #108) than STR2 [Lecoutre, 2011] using dom/wdeg, it has the largest average CPU time out of three algorithms.

Using the t-test, we obtain the 'comparability' graph shown in Figure 5.2, which shows that CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] are preferable over GAC2001 [Bessière *et al.*, 2005] on global-tables model.

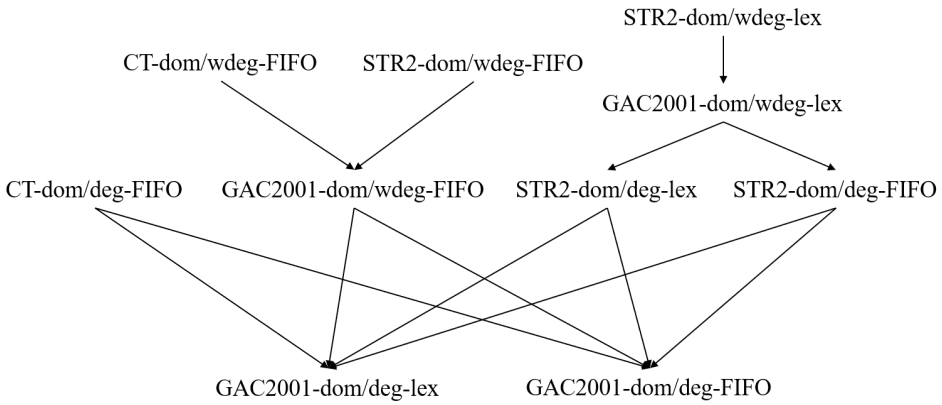

Figure 5.2: Comparing the ten configurations of GAC-based algorithms on the global-tables model

5.1.3 Conclusions about GAC Algorithms

Based on our observations drawn in Section 5.1.1 and Section 5.1.2, GAC algorithms seem to perform better on the new ternary-tables model of the Nonogram puzzle

than the global-tables model. In the remaining of this thesis, we make the following choices:

1. **Consistency algorithm:** GAC2001 [Bessière *et al.*, 2005] outperforms both CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] on the ternary-tables model and the opposite holds for the global-tables model. However, some of the high-level consistency algorithms that we test in the remainder of this chapter are implemented in Stampede with STR2.

2. **Ordering heuristic:** For both ternary and global tables models of the Nonogram puzzle, dom/wdeg [Boussemart *et al.*, 2004] clearly outperforms dom/deg [Bessière and Régin, 1996] on all evaluation criteria considered. In the remainder of this thesis, we only use dom/wdeg.

3. **Propagation queue:** The propagation-queue management does not seem to be a dramatically significant factor. Under, dom/wdeg [Boussemart *et al.*, 2004], FIFO seems to have a slightly better performance than lex. We suspect that a queue management based on the constraint weights may yield better results, but this question remains to be investigated.

Therefore, in the following experiment (Section 5.2), we decide to use dom/wdeg [Boussemart *et al.*, 2004] and FIFO in combination with the 14 consistency algorithms as specified in Section 4.2.

5.2 Comparing All Consistency Algorithms

In this section, we use dom/wdeg [Boussemart *et al.*, 2004] and FIFO to evaluate the performance of 14 consistency algorithms on the two datasets of Nonogram puzzles. Because all combinations in this experiment differ only on the consistency algorithms

being used, for simplicity, we refer to the combinations applied on each dataset only by the used consistency algorithm.

5.2.1 Ternary-Tables Model

The relational consistency algorithms R(∗,3)C, R(∗,4)C, and their weakened versions [Karakashian *et al.*, 2010] result in the same domain filtering as GAC because any two ternary-table constraints intersect on at most one variable. However, they take significantly more time. Thus, we omit them from the results.

Table 5.4 summarizes the performance of the other ten (10) algorithms on the instances of ternary-tables model. Tables B.5 and B.6 in Appendix B provide the detailed results on all instances. Overall, every instance is solved by at least one

Table 5.4: All algorithms performance on ternary-tables model

# Instaces 174 in total: 166 solved by all and 174 solved by at least one							
		On instances solved by all algorithms					
Algorithm	**# Solved**	**avg. # BT**	**avg. # NV**	**avg. CPU time (s)**	**avg. CPU time on all instances (s)**	**Best on x instances**	**# Instances solved backtrack free**
APOAC$_{around\cup}$	173	709.88	2,746.05	2.43	>85.14	1	170
PrePeak^{+}-POAC1	173	9,066.67	11,400.07	3.42	>86.10	0	56
A$\cup_{cyc}^{bfsc}$POAC	173	4,534.17	6,706.55	3.71	>86.38	0	105
ANPOAC	173	9,066.67	11,400.07	3.62	>86.29	0	56
SAC1	166	0.22	2,023.35	2.85	>664.78	0	165
APOAC	166	312.36	2,339.93	2.03	>664.01	0	165
POAC1	166	0.22	2,023.36	1.97	>663.95	8	165
CT	174	5,401.43	7,616.10	2.15	24.39	1	55
GAC2001	173	5,401.43	7,616.10	1.56	>84.28	164	55
STR2	174	9,066.67	11,400.07	2.59	34.75	0	56

algorithm, and 166 out of 174 instances are solved by all algorithms.

According to the numerical results shown in Table 5.4, we rank the algorithms based on the following criteria, where $x \succ y$ indicates that algorithm x is better than algorithm y:

- *Best on number of instances*, shown in parenthesis:

$$\text{GAC2001 (164)} \succ \text{POAC1 (8)} \succ \text{CT (1)} \succ \text{APOAC}_{around\cup} \text{ (1)}$$

  GAC2001 [Bessière *et al.*, 2005] dominates all other algorithms in this specific criterion with 164 instances on which it has the best performance. The second position belongs to POAC1 [Balafrej *et al.*, 2014] with only eight (8) instances.

- *Number of instances solved*, shown in parenthesis:

$$\begin{gathered}\text{CT, STR2 (174)} \\ \succ \text{GAC2001,ANPOAC,PrePeak}^{+}\text{-POAC1,A}\cup_{cyc}^{bfsc}\text{POAC,APOAC}_{around\cup} \text{ (173)} \\ \succ \text{POAC1, APOAC, SAC1 (166)}\end{gathered}$$

  Note that only CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] could solve instance #69 in the allocated four hours. Because CT, STR2, and GAC2001 [Bessière *et al.*, 2005] have similar pruning power as we discussed in Section 5.1, and because STR2's *real* CPU time[4] on instance #69 was close to the time limit, we suspect that GAC2001's *real* CPU time on instance #69 might be close to the time limit by above.

  In order to verify that, we rerun all algorithms on instance #69 with dom/wdeg [Boussemart *et al.*, 2004] and FIFO. We also increase the time limit to ten (10)

[4]This unstable CPU time is reported by Stampede and is different from the CPU time that we report in the detailed results in Appendix B, which was calculated using the number of instructions.

hours per instance and allocate 12GB of memory. As we expected, other than the two algorithms CT and STR2 that can solve instance #69 in the original experiment, there are two other algorithms that have a completion time close to four hours, namely, GAC2001 and PrePeak$^+$-POAC1 [Woodward, 2018].

- *Best average CPU time* on instances solved by all algorithms:

$$\text{GAC2001} \succ \text{POAC1} \succ \text{APOAC} \succ \text{CT} \succ \text{APOAC}_{around\cup}$$
$$\succ \text{STR2} \succ \text{SAC1} \succ \text{PrePeak}^+\text{-POAC1} \succ \text{ANPOAC} \succ \text{A}\cup_{cyc}^{bfsc}\text{POAC}$$

GAC2001 [Bessière *et al.*, 2005] has the smallest average CPU time (1.56 seconds) overall. Except for APOAC$_{around\cup}$ [Woodward, 2018], the other three algorithms to enforce High-Level Consistency proposed by Woodward [2018] take more time on average than others to solve a single instance in ternary-tables model. Given that STR2's [Lecoutre, 2011] performance is not very good on ternary-tables model (also discussed in Section 5.1), the fact that these algorithms use STR2 in their implementation to enforce GAC (mentioned in Section 4.3.2) might be the main reason behind their poor performance.

- *Number of instances solved backtrack free*, shown in parenthesis:

$$\text{APOAC}_{around\cup}\ (170) \succ \text{POAC1, APOAC, SAC1}\ (165) \succ \text{A}\cup_{cyc}^{bfsc}\text{POAC}\ (105)$$
$$\succ \text{PrePeak}^+\text{-POAC1, ANPOAC, STR2}\ (56) \succ \text{CT, GAC2001}\ (55)$$

We observe that APOAC$_{around\cup}$ [Woodward, 2018] solves the most instances in a backtrack-free manner while other structure-exploring algorithms, including ANPOAC and A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018], fail to do so. In addition, all singleton consistency algorithms perform better or at least as good as GAC-based algorithms in this specific criterion.

The above implies that using singleton consistency algorithms and focusing on the structure on which APOAC$_{around\cup}$ [Woodward, 2018] is interested may be effective to explore the special structure of the ternary-tables model of the Nonogram puzzle.

Comparing pairwise the ten (10) configurations and using paired t-test, we obtain the 'comparability' graph shown in Figure 5.3. The 'comparability' graph shows that

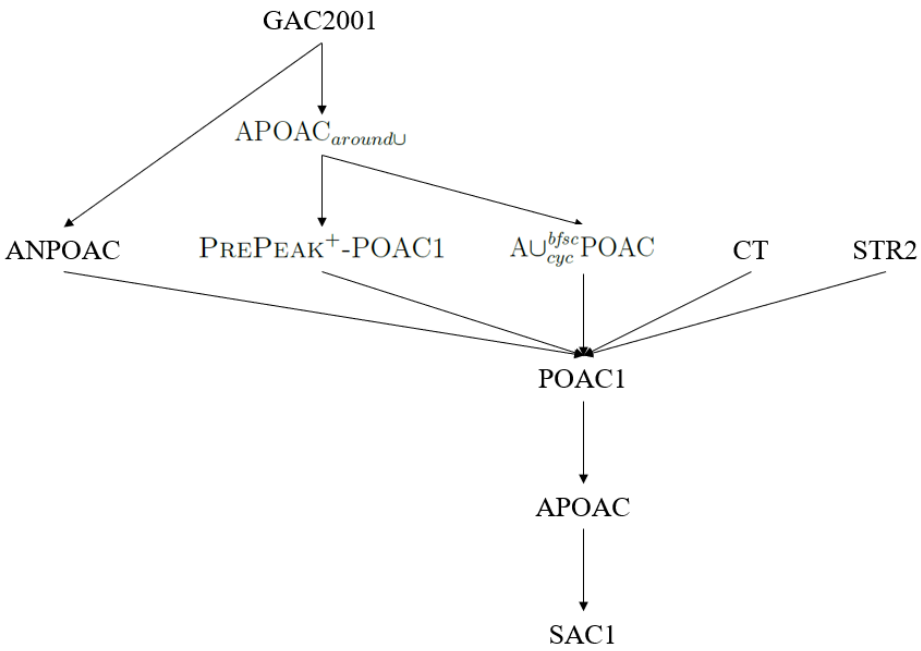

Figure 5.3: Comparing ten algorithms on the ternary-table constraint model

GAC2001 [Bessière *et al.*, 2005] statistically dominates all except two algorithms, namely, CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011]. Although these three GAC-based consistency algorithms are not statistical distinguishable, GAC2001 has better performance on the ternary-tables model with smaller average numbers of backtrack and nodes visited compared to STR2, and significant smaller average CPU time than both CT and STR2.

5.2.2 Global-Tables Model

Table 5.5 summarizes the performance of all algorithms on the global-tables model. Tables B.7, B.8, and B.9 in Appendix B provide the detailed results on all tested instances. Similar to our observation in Section 5.1.2, all algorithms take more time

Table 5.5: All algorithms performance on global-tables model

# Instaces 170 in total: 54 solved by all and 134 solved by at least one							
		On instances solved by all algorithms					
Algorithm	**# Solved**	**avg. # BT**	**avg. # NV**	**avg. CPU time (s)**	**avg. CPU time on all instances (s)**	**Best on x instances**	**# Instances solved backtrack free**
APOAC$_{around\cup}$	134	0.00	392.65	120.74	>4,222.14	0	127
PrePeak^{+}-POAC1	134	55.94	458.72	120.68	>4,222.30	2	45
A$\cup_{cyc}^{bfsc}$POAC	132	31.54	430.02	120.95	>4,263.02	0	69
ANPOAC	133	55.94	458.72	120.80	>4,300.25	0	45
SAC1	129	0.00	392.65	120.65	>4,429.40	14	126
APOAC	128	0.00	392.65	120.72	>4,515.33	0	122
POAC1	129	0.00	392.65	120.72	>4,551.54	0	127
CT	133	48.02	450.13	120.63	>4,299.11	45	45
GAC2001	133	48.02	450.13	120.68	>4,301.55	4	45
STR2	133	55.94	458.72	120.61	>4,299.59	69	45
R(∗,3)C	77	462.52	945.61	1,022.28	>9,027.53	0	36
R(∗,4)C	58	1.94	395.04	1,165.82	>9,974.33	0	57
wR(∗,3)C	75	462.52	945.61	1,021.96	>9,012.27	0	35
wR(∗,4)C	58	1.94	395.04	1,166.36	>9,974.50	0	57

to complete on the global-tables model than on the ternary-tables model. Out of the 170 instances available, only 134 instances are solved by at least one algorithm and only 54 instances are completed by all algorithms in the allocated four hours.

Overall, the four relational consistency algorithms, namely, R(∗,3)C, wR(∗,3)C,

R(∗,4)C, and wR(∗,4)C [Karakashian *et al.*, 2010] are not particularly effective on the global-tables model. They are way too costly to be effective. This observation is visible by the red background color in the last four rows of the summary table (Table 5.5).

According to the numerical results shown in Table 5.5, we rank the algorithms based on the following criteria:

- *Best on number of instances*, shown in parenthesis:

$$\text{STR2 (69)} \succ \text{CT (45)}$$
$$\succ \text{SAC1 (14)} \succ \text{GAC2001 (4)} \succ \text{PrePeak}^{+}\text{-POAC1 (2)}$$

  Once again, CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] are more effective than GAC2001 [Bessière *et al.*, 2005] on the global-tables model. GAC2001, on the other hand, fails to maintain its dominance as in Table 5.4.

- *Number of instances solved*, shown in parenthesis:

$$\text{APOAC}_{around\cup}\text{, PrePeak}^{+}\text{-POAC1 (134)}$$
$$\succ \text{ANPOAC, CT, GAC2001, STR2 (133)} \succ \text{A}\cup_{cyc}^{bfsc}\text{POAC (132)}$$
$$\succ \text{SAC1, POAC1 (129)} \succ \text{APOAC (128)}$$
$$\succ \text{R(∗,3)C (77)} \succ \text{wR(∗,3)C (75)} \succ \text{R(∗,4)C, wR(∗,4)C (58)}$$

  Although having the largest amount of solved instances in Table 5.4, GAC-based algorithms are outperformed on one instance (instance #107) by $\text{APOAC}_{around\cup}$ and PrePeak^{+}-POAC1 [Woodward, 2018]. The other two algorithms proposed by Woodward [2018], namely, ANPOAC and $\text{A}\cup_{cyc}^{bfsc}\text{POAC}$, also perform really well with 133 and 132 solved instances, respectively.

On the other hand, among the four relational consistency algorithms, algorithms enforcing 3-wise consistency show a better performance than the ones that enforce 4-wise consistency.

- *Best average CPU time* on instances solved by all algorithms:

$$\text{STR2} \succ \text{CT} \succ \text{SAC1} \succ \text{PrePeak}^{+}\text{-POAC1} \succ \text{GAC2001}$$
$$\succ \text{POAC1} \succ \text{APOAC} \succ \text{APOAC}_{around\cup} \succ \text{ANPOAC} \succ \text{A}\cup_{cyc}^{bfsc}\text{POAC}$$
$$\succ \text{wR}(*,3)\text{C} \succ \text{R}(*,3)\text{C} \succ \text{R}(*,4)\text{C} \succ \text{wR}(*,4)\text{C}$$

Except for the relational consistency algorithms, the difference between the average CPU time of all other algorithms is not significant (approximately 0.34 seconds between STR2 [Lecoutre, 2011] and A$\cup_{cyc}^{bfsc}$POAC [Woodward, 2018]). Hence, although we can construct an ordering given the small differences between their average CPU time, we cannot claim that any of them is significantly better than the others. Similar situation also applies on the ordering of the relational consistency algorithms R$(*, m)$C [Karakashian *et al.*, 2010] and their weakened versions.

- *Number of instances solved backtrack free*, shown in parenthesis:

$$\text{APOAC}_{around\cup}, \text{POAC1 (127)} \succ \text{SAC1 (126)} \succ \text{APOAC (122)}$$
$$\succ \text{A}\cup_{cyc}^{bfsc}\text{POAC (69)} \succ \text{R(*,4)C, wR(*,4)C (57)}$$
$$\succ \text{PrePeak}^{+}\text{-POAC1, ANPOAC, STR2, CT, GAC2001 (45)}$$
$$\succ \text{R(*,3)C (36)} \succ \text{wR(*,3)C (35)}$$

The case that we observed in Section 5.2.1 also applies to the global-tables model, in which we have singleton consistency algorithms and especially APOAC$_{around\cup}$

[Woodward, 2018] with the largest number of instances solved in a backtrack free manner.

The results show that although having the smallest amount of solved instances and the largest average CPU time, 4-wise consistency algorithms, namely, R(∗,4)C and wR(∗,4)C [Karakashian *et al.*, 2010], solve a larger number of instances (57) in a backtrack-free manner than GAC-based and the 3-wise consistency algorithms. They also have a significantly small average number of backtracks (1.94) and nodes visited (395.04) compared to other algorithms.

Among all algorithms that use STR2 [Lecoutre, 2011] to enforce GAC during search (mentioned in Section 4.3.2), PrePeak$^+$-POAC1 and ANPOAC [Woodward, 2018] are affected the most by the performance of STR2 because their average number of backtracks and nodes visited are exactly the same as STR2's in both models of the Nonogram puzzle (Table 5.4 and Table 5.5).

Comparing pairwise the fourteen (14) tested algorithms using paired t-test, we obtain the 'comparability' graph shown in Figure 5.4. From the 'comparability' graph, we observe the following:

- Relational consistency algorithms [Karakashian *et al.*, 2010] are statistically the worst, among which 3-wise consistency algorithms outperform the 4-wise ones.
- The ordering between GAC-based algorithm is as expected according to our observation from Section 5.1.2, i.e., CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011] statistically dominate GAC2001 [Bessière *et al.*, 2005] on global-tables model.
- Other than ANPOAC and A$\cup^{bfsc}_{cyc}$POAC [Woodward, 2018] which perform worse than CT [Demeulenaere *et al.*, 2016] and STR2 [Lecoutre, 2011], the other

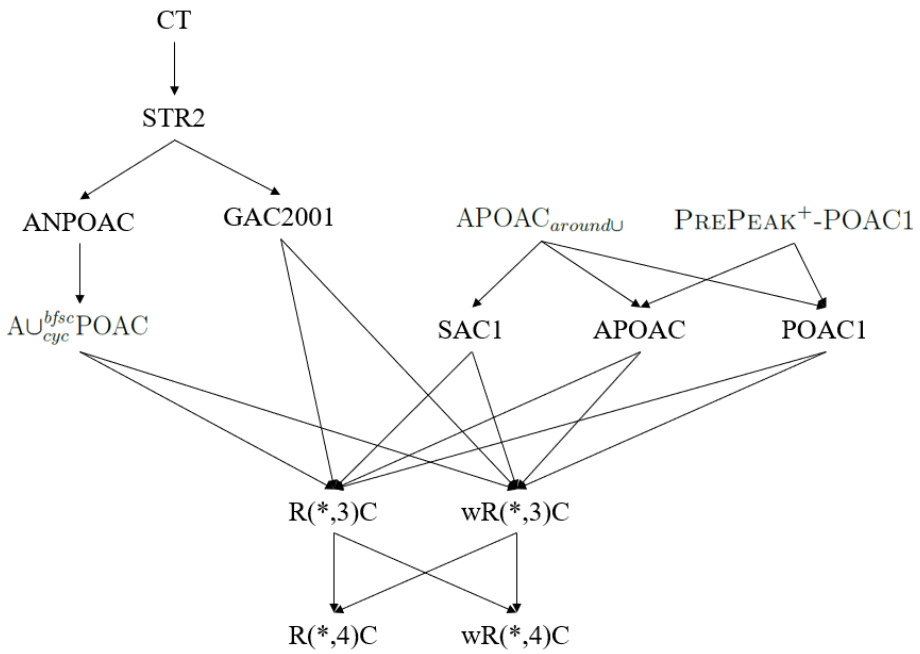

Figure 5.4: Comparing 14 consistency algorithms on the global-tables model

singleton consistency algorithms and the three GAC-based algorithms are not statistically distinguishable.

Summary

In this chapter, we evaluated the performance of all algorithms on solving the instances of two models of the Nonogram puzzle, namely, the ternary-tables model and the global-tables model. We selected dom/wdeg [Boussemart *et al.*, 2004] ordering heuristic over dom/deg [Bessière and Régin, 1996] and FIFO propagation queue over lexicographical because of their good performance on both datasets in our first experiment with GAC-based algorithms. We also showed that relational consistency algorithms [Karakashian *et al.*, 2010] are not effective on either of the models. Finally, we concluded that although GAC-based algorithm dominate all others on ternary-tables

model, they are not statistically distinguishable from most singleton consistency algorithms on global-tables model.

Chapter 6

Conclusions and Future Work

In this chapter, we conclude the thesis and suggest directions for future research.

6.1 Conclusions

The Nonogram is an interesting puzzle to model as a CSP and solve with CP techniques. From the modeling perspective, the model with global tables is too costly in terms of storage space, and the model with regular constraints requires the availability of a specialized propagator. These two limitations make the reformulation the regular constraint into a set of ternary-tables constraints as proposed by Bessière *et al.* [2008] particularly interesting.

In this thesis, we studied this reformulation and provided clear and detailed instructions on how to generate the ternary-tables models in XCSP3 format.

Empirically evaluating the performance of backtrack search with various consistency algorithms for solving the models of the puzzle given with global tables and with ternary tables, we confirm that dom/wdeg variable ordering heuristic [Boussemart *et al.*, 2004] is more effective than dom/deg [Bessière and Régin, 1996], and that

FIFO and lexicographical propagation queues are not distinguishable. Regarding the performance of all tested consistency algorithms, we conclude that:

1. Among the three GAC-based algorithms, namely, CT [Demeulenaere *et al.*, 2016], STR2 [Lecoutre, 2011], and GAC2001 [Bessière *et al.*, 2005], GAC2001 is the best on the ternary-tables model; meanwhile, CT and STR2 outperform GAC2001 on models with larger arity constraints, i.e. the global-tables model.

2. Including higher-level consistency and relational consistency algorithms, GAC-based algorithms, especially GAC2001 [Bessière *et al.*, 2005], are currently the most effective on the ternary-tables model of the Nonogram puzzle.

3. The global-tables model is not attractive either in terms of storage or in terms of being amenable to effective solving.

4. Relational consistency algorithms are not effective for solving the Nonogram puzzle in either of the constraint models being tested.

5. Most importantly, singleton-based consistency algorithms solve the largest number of instances of both models in a backtrack-free manner. This result is particularly significant on the ternary-tables model. We strongly believe that more attention should be devoted to these algorithms in the future.

While High-Level Consistency (HLC) algorithms focusing on the topological properties of the CSP did not show to be the best in either of the tested models,[1] we hope that our promising results can inspire the studies of other similar algorithms that effectively explore the special structure of the Nonogram puzzle.

[1] APOAC$_{around\cup}$ [Woodward, 2018] shows excellent performance on instances #28, #137, #141, and #153 and deserves closer examination.

We empirically established that consistency algorithms perform better on the new ternary-tables constraint model of the Nonogram puzzle compared to the global-tables one.

6.2 Future Work

Our experiments open up many directions for future research:

1. Both experiments conducted in this thesis were on two out of three models of the Nonogram puzzle, namely, the ternary-tables and global-tables models. It would be interesting to compare the performance of search on these two models with the one using the specialized propagator on the regular constraint model [Pesant, 2004]. We can also extend the implementation of the tested consistency algorithms to regular constraints in order to verify their performance on the regular constraint model.

2. Many of the tested HLC algorithms use STR2 [Lecoutre, 2011] to enforce GAC during search. Given that STR2 is worse than GAC2001 [Bessière *et al.*, 2005] on the ternary-tables model and is worse than CT [Demeulenaere *et al.*, 2016] on the global-tables model, the performance of these HLC algorithms could be improved by using a different GAC algorithm.

3. From the special structure of the primal graph of the ternary-tables model (Figure 3.12), we can force a static variable ordering heuristic to first instantiate the Boolean variables (shown in blue) representing the cells of the puzzle before instantiating the auxiliary variables (shown in green and red). By doing so, we suspect that search will benefit in some aspects.

4. Observing that singleton consistency algorithms, especially APOAC$_{around\cup}$ [Woodward, 2018], solve the largest number of instances of both models in a backtrack-free manner, it is reasonable to study and examine other algorithms that effectively explore similar structures of the Nonogram puzzle.

5. It would be useful to extend and verify the performance of the tested consistency algorithms and the new ternary-tables model of the Nonogram puzzle with 2-way branching.

6. We believe that performance could be improved by using the constraints' weight to order the propagation queue of the consistency algorithms instead of using FIFO.

Appendix A

Dataset Information

In this appendix, we provide detailed information about the two datasets tested described in Section 4.1, including:

- Table A.1 contains 170 instances with *global-table constraint model* also from the XCSP3 website of benchmark problems.[1]
- Table A.2 contains 174 instances with *ternary-table constraint model* that we generated using the Nonogram benchmark of 174 instances with *regular constraint model* from the XCSP3 website of benchmark problems.[2]

[1] http://www.xcsp.org/series
[2] http://www.xcsp.org/series

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
001	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	1,264.06	1	2	2.00	2
002	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
003	20	20	40	400	20	20.00	20	1.00	1.00	1.00	3,003	286.70	9	2	2.00	2
004	24	24	48	576	24	24.00	24	1.00	1.00	1.00	19,448	1,300.96	11	2	2.00	2
005	24	24	48	576	24	24.00	24	1.00	1.00	1.00	31,824	6,688.50	1	2	2.00	2
006	24	24	48	576	24	24.00	24	1.00	1.00	1.00	8,568	1,052.58	1	2	2.00	2
007	33	32	65	1,056	33	32.49	32	-	-	-	-	-	-	2	2.00	2
008	24	24	48	576	24	24.00	24	1.00	1.00	1.00	27,132	2,371.88	1	2	2.00	2
009	35	29	64	1,015	35	31.72	29	1.00	1.00	1.00	8,855	1,064.12	1	2	2.00	2
010	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,001	244.10	1	2	2.00	2
011	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,005	684.08	1	2	2.00	2
012	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	54,264	5,242.17	1	2	2.00	2
013	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,005	382.96	1	2	2.00	2
014	24	24	48	576	24	24.00	24	1.00	1.00	1.00	27,132	2,836.75	1	2	2.00	2
015	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	11,440	897.50	1	2	2.00	2
016	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
017	20	20	40	400	20	20.00	20	1.00	1.00	1.00	5,005	558.43	1	2	2.00	2
018	24	24	48	576	24	24.00	24	1.00	1.00	1.00	27,132	3,117.44	1	2	2.00	2
019	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
020	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
021	24	24	48	576	24	24.00	24	1.00	1.00	1.00	8,568	771.40	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
022	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	26,334	2,990.12	1	2	2.00	2
023	24	24	48	576	24	24.00	24	1.00	1.00	1.00	1,140	443.10	1	2	2.00	2
024	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
025	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	42,504	1,736.14	1	2	2.00	2
026	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
027	24	24	48	576	24	24.00	24	1.00	1.00	1.00	2,380	587.50	1	2	2.00	2
028	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	20,349	2,984.73	1	2	2.00	2
029	24	24	48	576	24	24.00	24	1.00	1.00	1.00	24,310	2,471.46	1	2	2.00	2
030	20	20	40	400	20	20.00	20	1.00	1.00	1.00	816	157.95	1	2	2.00	2
031	24	24	48	576	24	24.00	24	1.00	1.00	1.00	1,540	250.58	1	2	2.00	2
032	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,002	421.25	1	2	2.00	2
033	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
034	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	1,024.67	18	2	2.00	2
035	24	24	48	576	24	24.00	24	1.00	1.00	1.00	1,330	179.08	1	2	2.00	2
036	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	12,650	1,495.44	1	2	2.00	2
037	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,365	291.03	1	2	2.00	2
038	35	28	63	980	35	31.11	28	-	-	-	-	-	-	2	2.00	2
039	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
040	32	20	52	640	32	24.62	20	-	-	-	-	-	-	2	2.00	2
041	16	16	32	256	16	16.00	16	1.00	1.00	1.00	165	44.91	1	2	2.00	2
042	16	16	32	256	16	16.00	16	1.00	1.00	0.99	330	99.97	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
043	16	16	32	256	16	16.00	16	1.00	1.00	0.99	715	190.25	11	2	2.00	2
044	24	24	48	576	24	24.00	24	1.00	1.00	1.00	12,376	2,592.67	1	2	2.00	2
045	16	16	32	256	16	16.00	16	1.00	1.00	0.99	715	226.22	5	2	2.00	2
047	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,820	302.68	1	2	2.00	2
048	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,001	240.45	1	2	2.00	2
049	16	16	32	256	16	16.00	16	1.00	1.00	0.99	715	189.50	1	2	2.00	2
050	16	16	32	256	16	16.00	16	1.00	1.00	0.99	495	134.34	1	2	2.00	2
051	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	2,324.69	1	2	2.00	2
052	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
053	24	24	48	576	24	24.00	24	1.00	1.00	1.00	12,376	2,273.10	1	2	2.00	2
054	24	24	48	576	24	24.00	24	1.00	1.00	1.00	4,368	478.48	1	2	2.00	2
055	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	33,649	1,925.03	1	2	2.00	2
056	16	16	32	256	16	16.00	16	1.00	1.00	0.99	364	114.72	1	2	2.00	2
057	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	922.54	1	2	2.00	2
058	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	33,649	4,309.77	1	2	2.00	2
059	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,985	676.06	1	2	2.00	2
060	24	24	48	576	24	24.00	24	1.00	1.00	1.00	31,824	6,614.12	1	2	2.00	2
062	16	16	32	256	16	16.00	16	1.00	1.00	0.99	715	115.03	1	2	2.00	2
064	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,365	279.73	1	2	2.00	2
066	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,820	320.68	14	2	2.00	2
067	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
068	16	16	32	256	16	16.00	16	1.00	1.00	0.99	330	67.84	8	2	2.00	2
069	33	25	58	825	33	28.45	25	-	-	-	-	-	-	2	2.00	2
070	24	24	48	576	24	24.00	24	1.00	1.00	1.00	19,448	1,148.42	1	2	2.00	2
071	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
072	24	24	48	576	24	24.00	24	1.00	1.00	1.00	18,564	1,818.96	1	2	2.00	2
073	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,820	234.18	1	2	2.00	2
074	24	16	40	384	24	19.20	16	1.00	1.00	0.99	4,368	258.05	1	2	2.00	2
075	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
076	20	20	40	400	20	20.00	20	1.00	1.00	1.00	5,005	698.28	1	2	2.00	2
077	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
078	24	24	48	576	24	24.00	24	1.00	1.00	1.00	12,376	1,314.48	1	2	2.00	2
079	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
080	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,001	147.55	1	2	2.00	2
081	24	24	48	576	24	24.00	24	1.00	1.00	1.00	3,060	334.77	1	2	2.00	2
082	28	28	56	784	28	28.00	28	1.00	1.00	1.00	5,985	686.36	1	2	2.00	2
083	16	16	32	256	16	16.00	16	1.00	1.00	1.00	78	41.38	7	2	2.00	2
085	24	24	48	576	24	24.00	24	1.00	1.00	1.00	2,380	244.96	1	2	2.00	2
087	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	698.33	1	2	2.00	2
088	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
089	20	20	40	400	20	20.00	20	1.00	1.00	1.00	4,368	727.20	1	2	2.00	2
090	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
091	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
092	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
093	16	16	32	256	16	16.00	16	1.00	1.00	1.00	286	48.59	1	2	2.00	2
094	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	2,790.54	1	2	2.00	2
095	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
096	24	24	48	576	24	24.00	24	1.00	1.00	1.00	27,132	3,415.67	1	2	2.00	2
097	16	16	32	256	16	16.00	16	1.00	1.00	1.00	91	21.13	1	2	2.00	2
098	20	20	40	400	20	20.00	20	1.00	1.00	1.00	560	133.58	1	2	2.00	2
099	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,005	1,200.02	13	2	2.00	2
100	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	54,264	5,867.64	15	2	2.00	2
101	29	15	44	435	29	19.77	15	1.00	1.00	0.99	26,334	2,162.91	1	2	2.00	2
102	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,985	838.25	1	2	2.00	2
103	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
104	20	20	40	400	20	20.00	20	1.00	1.00	1.00	560	69.50	1	2	2.00	2
105	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	2,300	265.36	1	2	2.00	2
106	16	16	32	256	16	16.00	16	1.00	1.00	0.99	495	167.91	1	2	2.00	2
107	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	12,650	1,684.88	1	2	2.00	2
108	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	33,649	4,338.89	1	2	2.00	2
109	16	16	32	256	16	16.00	16	1.00	1.00	0.99	364	157.41	12	2	2.00	2
110	20	20	40	400	20	20.00	20	1.00	1.00	1.00	715	110.83	1	2	2.00	2
111	24	24	48	576	24	24.00	24	1.00	1.00	1.00	2,380	344.88	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
112	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
113	20	20	40	400	20	20.00	20	1.00	1.00	1.00	4,368	396.78	1	2	2.00	2
114	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	1,077.65	1	2	2.00	2
115	20	20	40	400	20	20.00	20	1.00	1.00	1.00	286	57.15	1	2	2.00	2
116	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	20,349	1,297.75	1	2	2.00	2
117	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,002	441.95	14	2	2.00	2
118	24	24	48	576	24	24.00	24	1.00	1.00	1.00	8,568	1,414.81	17	2	2.00	2
119	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
120	24	24	48	576	24	24.00	24	1.00	1.00	1.00	8,008	741.23	1	2	2.00	2
121	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	27,132	2,846.97	1	2	2.00	2
122	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
123	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,002	252.38	1	2	2.00	2
124	24	24	48	576	24	24.00	24	1.00	1.00	1.00	3,060	553.50	1	2	2.00	2
125	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	33,649	3,374.50	1	2	2.00	2
126	16	16	32	256	16	16.00	16	1.00	1.00	1.00	210	66.69	12	2	2.00	2
128	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
129	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	2,278.23	1	2	2.00	2
130	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
131	24	24	48	576	24	24.00	24	1.00	1.00	1.00	4,845	622.02	1	2	2.00	2
132	20	20	40	400	20	20.00	20	1.00	1.00	1.00	455	73.90	1	2	2.00	2
133	24	24	48	576	24	24.00	24	1.00	1.00	1.00	816	119.42	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
134	24	24	48	576	24	24.00	24	1.00	1.00	1.00	1,820	308.75	1	2	2.00	2
135	28	24	52	672	28	25.85	24	1.00	1.00	1.00	3,060	320.73	1	2	2.00	2
136	24	24	48	576	24	24.00	24	1.00	1.00	1.00	8,568	1,648.38	1	2	2.00	2
137	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
138	24	24	48	576	24	24.00	24	1.00	1.00	1.00	24,310	1,329.98	1	2	2.00	2
139	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
140	20	20	40	400	20	20.00	20	1.00	1.00	1.00	816	213.48	1	2	2.00	2
141	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	14,950	1,661.59	1	2	2.00	2
142	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	1,197.75	1	2	2.00	2
143	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
144	16	16	32	256	16	16.00	16	1.00	1.00	0.99	715	150.50	1	2	2.00	2
146	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,002	606.13	1	2	2.00	2
147	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,002	422.63	1	2	2.00	2
148	24	24	48	576	24	24.00	24	1.00	1.00	1.00	15,504	2,730.35	1	2	2.00	2
149	16	16	32	256	16	16.00	16	1.00	1.00	1.00	210	50.41	1	2	2.00	2
150	24	24	48	576	24	24.00	24	1.00	1.00	1.00	4,845	1,298.19	1	2	2.00	2
151	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
152	20	20	40	400	20	20.00	20	1.00	1.00	1.00	1,365	328.43	1	2	2.00	2
153	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
154	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
155	16	16	32	256	16	16.00	16	1.00	1.00	1.00	66	16.50	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
157	20	20	40	400	20	20.00	20	1.00	1.00	1.00	2,380	311.95	1	2	2.00	2
158	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,985	910.02	1	2	2.00	2
160	24	24	48	576	24	24.00	24	1.00	1.00	1.00	1,140	155.42	1	2	2.00	2
161	16	16	32	256	16	16.00	16	1.00	1.00	0.99	495	74.59	1	2	2.00	2
162	24	24	48	576	24	24.00	24	1.00	1.00	1.00	2,380	312.69	1	2	2.00	2
163	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	10,626	1,205.19	1	2	2.00	2
164	16	16	32	256	16	16.00	16	1.00	1.00	1.00	120	27.53	1	2	2.00	2
165	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	1,654.33	1	2	2.00	2
166	24	24	48	576	24	24.00	24	1.00	1.00	1.00	12,376	1,143.83	1	2	2.00	2
167	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	1,039.81	21	2	2.00	2
168	20	20	40	400	20	20.00	20	1.00	1.00	1.00	3,003	721.55	1	2	2.00	2
169	24	24	48	576	24	24.00	24	1.00	1.00	1.00	6,188	1,628.23	1	2	2.00	2
170	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
171	16	16	32	256	16	16.00	16	1.00	1.00	0.99	364	116.28	1	2	2.00	2
172	20	20	40	400	20	20.00	20	1.00	1.00	1.00	4,368	514.98	1	2	2.00	2
173	24	24	48	576	24	24.00	24	1.00	1.00	1.00	31,824	2,765.40	17	2	2.00	2
174	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2
175	24	24	48	576	24	24.00	24	1.00	1.00	1.00	5,985	937.96	1	2	2.00	2
176	16	16	32	256	16	16.00	16	1.00	1.00	1.00	220	65.53	1	2	2.00	2
177	32	32	64	1,024	32	32.00	32	1.00	1.00	1.00	12,650	717.20	1	2	2.00	2
178	32	24	56	768	32	27.43	24	1.00	1.00	1.00	8,008	581.00	1	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.1: Detailed information of global-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
179	24	24	48	576	24	24.00	24	1.00	1.00	1.00	924	151.75	1	2	2.00	2
180	32	32	64	1,024	32	32.00	32	-	-	-	-	-	-	2	2.00	2

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
001	24	24	1,152	1,776	3	3.00	3	0.97	0.84	0.50	20	8.62	1	17	5.04	1
002	32	32	2,048	3,136	3	3.00	3	0.98	0.90	0.50	26	13.53	1	22	7.79	1
003	20	20	800	1,240	3	3.00	3	0.96	0.92	0.67	19	11.32	2	13	6.90	1
004	24	24	1,152	1,776	3	3.00	3	0.96	0.93	0.67	22	12.96	2	15	7.90	1
005	24	24	1,152	1,776	3	3.00	3	0.97	0.88	0.50	25	13.49	1	21	7.50	1
006	24	24	1,152	1,776	3	3.00	3	0.97	0.92	0.50	24	14.24	1	20	8.24	1
007	33	32	2,112	3,233	3	3.00	3	0.98	0.96	0.50	34	21.36	2	30	12.39	1
008	24	24	1,152	1,776	3	3.00	3	0.98	0.92	0.50	32	13.19	2	25	7.42	1
009	35	29	2,030	3,109	3	3.00	3	0.98	0.92	0.50	41	15.55	1	32	9.33	1
010	20	20	800	1,240	3	3.00	3	0.96	0.84	0.50	20	9.12	1	16	5.46	1
011	24	24	1,152	1,776	3	3.00	3	0.97	0.83	0.50	22	9.57	1	18	5.68	1
012	32	32	2,048	3,136	3	3.00	3	0.98	0.89	0.50	40	15.87	1	30	9.21	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
013	24	24	1,152	1,776	3	3.00	3	0.98	0.92	0.50	25	15.65	1	22	9.51	1
014	24	24	1,152	1,776	3	3.00	3	0.97	0.91	0.50	24	13.38	1	19	7.57	1
015	32	32	2,048	3,136	3	3.00	3	0.98	0.98	0.96	38	31.85	2	33	19.21	1
016	32	32	2,048	3,136	3	3.00	3	0.98	0.96	0.86	38	24.50	2	33	13.80	1
017	20	20	800	1,240	3	3.00	3	0.96	0.82	0.50	24	8.62	1	18	4.94	1
018	24	24	1,152	1,776	3	3.00	3	0.97	0.81	0.50	27	8.90	1	22	5.11	1
019	32	32	2,048	3,136	3	3.00	3	0.98	0.93	0.50	35	18.90	1	31	10.54	1
020	32	32	2,048	3,136	3	3.00	3	0.98	0.88	0.50	31	13.75	1	25	8.01	1
021	24	24	1,152	1,776	3	3.00	3	0.98	0.79	0.50	25	7.00	1	24	4.29	1
022	32	32	2,048	3,136	3	3.00	3	0.97	0.81	0.50	30	8.90	1	24	5.38	1
023	24	24	1,152	1,776	3	3.00	3	0.97	0.90	0.50	23	10.51	1	20	6.31	1
024	32	32	2,048	3,136	3	3.00	3	0.98	0.96	0.88	36	21.60	2	33	12.65	1
025	32	32	2,048	3,136	3	3.00	3	0.98	0.87	0.50	32	10.04	1	30	6.10	1
026	32	32	2,048	3,136	3	3.00	3	0.98	0.97	0.90	46	29.04	2	33	15.97	1
027	24	24	1,152	1,776	3	3.00	3	0.98	0.96	0.89	26	18.31	2	25	11.49	1
028	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	27	14.58	1	23	8.74	1
029	24	24	1,152	1,776	3	3.00	3	0.97	0.92	0.50	33	16.03	1	24	8.80	1
030	20	20	800	1,240	3	3.00	3	0.97	0.85	0.50	20	8.21	1	18	5.03	1
031	24	24	1,152	1,776	3	3.00	3	0.97	0.86	0.50	23	8.95	1	21	5.48	1
032	20	20	800	1,240	3	3.00	3	0.97	0.87	0.50	22	9.63	1	17	5.51	1
033	32	32	2,048	3,136	3	3.00	3	0.98	0.90	0.50	35	15.62	1	28	9.51	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
034	24	24	1,152	1,776	3	3.00	3	0.97	0.95	0.83	28	18.27	2	22	10.40	1
035	24	24	1,152	1,776	3	3.00	3	0.96	0.85	0.50	19	8.71	1	15	5.44	1
036	32	32	2,048	3,136	3	3.00	3	0.97	0.85	0.50	26	11.01	1	21	6.66	1
037	20	20	800	1,240	3	3.00	3	0.96	0.84	0.50	21	9.50	1	16	5.61	1
038	35	28	1,960	3,003	3	3.00	3	0.97	0.94	0.83	26	15.40	2	20	8.90	1
039	32	32	2,048	3,136	3	3.00	3	0.98	0.88	0.50	33	13.04	1	26	7.34	1
040	32	20	1,280	1,972	3	3.00	3	0.97	0.86	0.50	24	11.43	1	21	6.57	1
041	16	16	512	800	3	3.00	3	0.96	0.79	0.50	16	5.78	1	13	3.68	1
042	16	16	512	800	3	3.00	3	0.95	0.86	0.50	15	8.71	1	12	5.31	1
043	16	16	512	800	3	3.00	3	0.95	0.90	0.75	20	10.15	2	14	5.73	1
044	24	24	1,152	1,776	3	3.00	3	0.96	0.87	0.50	21	10.81	1	16	6.15	1
045	16	16	512	800	3	3.00	3	0.96	0.90	0.75	20	9.60	2	16	5.50	1
046	19	20	760	1,179	3	3.00	3	0.97	0.90	0.50	20	10.83	2	18	6.40	1
047	20	20	800	1,240	3	3.00	3	0.98	0.94	0.80	22	14.06	2	21	8.60	1
048	20	20	800	1,240	3	3.00	3	0.98	0.94	0.80	22	13.79	2	21	8.48	1
049	16	16	512	800	3	3.00	3	0.95	0.88	0.50	18	8.88	1	14	5.15	1
050	16	16	512	800	3	3.00	3	0.94	0.84	0.50	16	7.73	1	12	4.56	1
051	24	24	1,152	1,776	3	3.00	3	0.97	0.90	0.50	22	11.32	1	20	6.39	1
052	32	32	2,048	3,136	3	3.00	3	0.98	0.95	0.75	37	21.51	2	28	11.89	1
053	24	24	1,152	1,776	3	3.00	3	0.97	0.84	0.50	22	11.26	1	19	6.44	1
054	24	24	1,152	1,776	3	3.00	3	0.98	0.89	0.50	24	10.73	1	22	6.53	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
055	32	32	2,048	3,136	3	3.00	3	0.98	0.92	0.50	30	13.37	1	28	8.20	1
056	16	16	512	800	3	3.00	3	0.95	0.88	0.50	15	8.48	1	12	5.01	1
057	24	24	1,152	1,776	3	3.00	3	0.96	0.89	0.50	19	10.21	1	16	5.93	1
058	32	32	2,048	3,136	3	3.00	3	0.97	0.89	0.50	23	11.30	1	20	6.66	1
059	24	24	1,152	1,776	3	3.00	3	0.98	0.88	0.50	28	12.07	1	24	7.28	1
060	24	24	1,152	1,776	3	3.00	3	0.97	0.92	0.50	34	15.90	1	24	8.56	1
061	33	32	2,112	3,233	3	3.00	3	0.97	0.93	0.50	21	12.34	2	19	7.82	1
062	16	16	512	800	3	3.00	3	0.96	0.84	0.50	17	7.84	1	15	4.71	1
064	20	20	800	1,240	3	3.00	3	0.96	0.87	0.50	26	9.91	1	19	5.72	1
065	20	32	1,280	1,972	3	3.00	3	0.98	0.88	0.50	25	10.78	1	23	6.27	1
066	20	20	800	1,240	3	3.00	3	0.96	0.91	0.75	20	10.06	2	16	5.87	1
067	32	32	2,048	3,136	3	3.00	3	0.97	0.91	0.75	24	11.96	2	19	6.85	1
068	16	16	512	800	3	3.00	3	0.96	0.86	0.50	21	7.74	2	15	4.66	1
069	33	25	1,650	2,533	3	3.00	3	0.98	0.93	0.50	40	20.33	1	31	10.57	1
070	24	24	1,152	1,776	3	3.00	3	0.97	0.93	0.50	26	14.28	1	21	8.25	1
071	32	32	2,048	3,136	3	3.00	3	0.98	0.93	0.50	39	21.90	1	30	11.82	1
072	24	24	1,152	1,776	3	3.00	3	0.98	0.96	0.86	31	19.34	2	25	11.26	1
073	20	20	800	1,240	3	3.00	3	0.95	0.77	0.50	18	6.72	1	13	4.14	1
074	24	16	768	1,192	3	3.00	3	0.97	0.85	0.50	27	9.29	1	21	5.57	1
075	32	32	2,048	3,136	3	3.00	3	0.98	0.90	0.50	33	15.47	1	31	9.35	1
076	20	20	800	1,240	3	3.00	3	0.96	0.87	0.50	29	12.68	1	20	6.90	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
077	32	32	2,048	3,136	3	3.00	3	0.98	0.85	0.50	29	13.50	1	25	7.86	1
078	24	24	1,152	1,776	3	3.00	3	0.97	0.84	0.50	25	9.28	1	19	5.40	1
079	32	32	2,048	3,136	3	3.00	3	0.97	0.84	0.50	30	10.04	1	24	5.78	1
080	20	20	800	1,240	3	3.00	3	0.95	0.82	0.50	15	7.07	1	11	4.46	1
081	24	24	1,152	1,776	3	3.00	3	0.96	0.84	0.50	17	8.08	1	13	5.12	1
082	28	28	1,568	2,408	3	3.00	3	0.97	0.90	0.50	19	10.47	1	17	6.60	1
083	16	16	512	800	3	3.00	3	0.95	0.89	0.67	12	7.52	2	11	4.77	1
085	24	24	1,152	1,776	3	3.00	3	0.96	0.86	0.50	18	8.27	1	16	5.12	1
086	28	32	1,792	2,748	3	3.00	3	0.97	0.89	0.50	22	10.25	1	19	6.21	1
087	24	24	1,152	1,776	3	3.00	3	0.97	0.83	0.50	21	9.07	1	17	5.47	1
088	32	32	2,048	3,136	3	3.00	3	0.98	0.90	0.50	34	15.13	1	30	8.59	1
089	20	20	800	1,240	3	3.00	3	0.96	0.90	0.50	21	11.40	1	16	6.50	1
090	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	32	15.82	1	28	9.06	1
091	32	32	2,048	3,136	3	3.00	3	0.98	0.93	0.50	37	16.25	1	29	9.33	1
092	32	32	2,048	3,136	3	3.00	3	0.98	0.93	0.50	39	18.11	1	33	10.62	1
093	16	16	512	800	3	3.00	3	0.95	0.81	0.50	14	6.60	1	12	4.21	1
094	24	24	1,152	1,776	3	3.00	3	0.97	0.93	0.50	32	16.26	1	22	8.98	1
095	32	32	2,048	3,136	3	3.00	3	0.98	0.90	0.50	39	17.52	1	28	9.50	1
096	24	24	1,152	1,776	3	3.00	3	0.97	0.85	0.50	28	12.00	1	20	6.71	1
097	16	16	512	800	3	3.00	3	0.93	0.76	0.50	11	4.61	1	9	3.10	1
098	20	20	800	1,240	3	3.00	3	0.96	0.89	0.50	19	9.56	1	15	5.86	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
099	24	24	1,152	1,776	3	3.00	3	0.97	0.94	0.86	28	14.95	2	23	8.64	1
100	32	32	2,048	3,136	3	3.00	3	0.98	0.96	0.89	37	18.96	2	31	11.25	1
101	29	15	870	1,349	3	3.00	3	0.97	0.88	0.50	32	13.49	1	25	7.61	1
102	24	24	1,152	1,776	3	3.00	3	0.97	0.89	0.50	22	10.57	1	18	6.23	1
103	32	32	2,048	3,136	3	3.00	3	0.98	0.92	0.50	30	16.03	1	29	8.90	1
104	20	20	800	1,240	3	3.00	3	0.96	0.82	0.50	19	7.73	1	16	4.95	1
105	32	32	2,048	3,136	3	3.00	3	0.98	0.87	0.50	27	11.18	1	24	7.17	1
106	16	16	512	800	3	3.00	3	0.96	0.89	0.50	21	9.60	2	17	5.48	1
107	32	32	2,048	3,136	3	3.00	3	0.96	0.88	0.50	18	11.57	1	14	6.92	1
108	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	25	11.68	1	24	6.95	1
109	16	16	512	800	3	3.00	3	0.95	0.91	0.80	15	9.38	2	12	5.46	1
110	20	20	800	1,240	3	3.00	3	0.96	0.87	0.50	21	9.51	1	17	5.86	1
111	24	24	1,152	1,776	3	3.00	3	0.97	0.87	0.50	27	11.12	1	22	6.78	1
112	32	32	2,048	3,136	3	3.00	3	0.98	0.88	0.50	35	14.47	1	27	8.40	1
113	20	20	800	1,240	3	3.00	3	0.97	0.88	0.50	21	10.05	1	18	5.89	1
114	24	24	1,152	1,776	3	3.00	3	0.97	0.88	0.50	24	11.16	1	20	6.54	1
115	20	20	800	1,240	3	3.00	3	0.96	0.89	0.50	18	10.62	1	16	6.73	1
116	32	32	2,048	3,136	3	3.00	3	0.98	0.93	0.50	29	16.93	1	24	10.37	1
117	20	20	800	1,240	3	3.00	3	0.97	0.92	0.75	25	11.32	2	19	6.56	1
118	24	24	1,152	1,776	3	3.00	3	0.97	0.93	0.75	26	12.50	2	21	7.28	1
119	32	32	2,048	3,136	3	3.00	3	0.98	0.94	0.75	34	14.19	2	27	8.26	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
120	24	24	1,152	1,776	3	3.00	3	0.98	0.96	0.90	27	19.80	2	25	12.07	1
121	32	32	2,048	3,136	3	3.00	3	0.98	0.97	0.93	36	26.76	2	33	16.43	1
122	32	32	2,048	3,136	3	3.00	3	0.98	0.96	0.83	36	21.66	2	33	11.79	1
123	20	20	800	1,240	3	3.00	3	0.97	0.86	0.50	23	9.51	1	18	5.61	1
124	24	24	1,152	1,776	3	3.00	3	0.97	0.86	0.50	22	9.74	1	18	5.77	1
125	32	32	2,048	3,136	3	3.00	3	0.97	0.87	0.50	27	10.75	1	21	6.35	1
126	16	16	512	800	3	3.00	3	0.96	0.89	0.67	19	8.62	2	15	5.18	1
128	32	32	2,048	3,136	3	3.00	3	0.98	0.88	0.50	32	15.09	1	25	8.78	1
129	24	24	1,152	1,776	3	3.00	3	0.97	0.87	0.50	23	11.06	1	19	6.36	1
130	32	32	2,048	3,136	3	3.00	3	0.98	0.88	0.50	27	12.55	1	25	7.21	1
131	24	24	1,152	1,776	3	3.00	3	0.97	0.80	0.50	21	8.80	1	18	5.33	1
132	20	20	800	1,240	3	3.00	3	0.95	0.84	0.50	16	8.06	1	13	5.17	1
133	24	24	1,152	1,776	3	3.00	3	0.96	0.86	0.50	17	8.96	1	14	5.77	1
134	24	24	1,152	1,776	3	3.00	3	0.96	0.84	0.50	19	8.18	1	16	5.04	1
135	28	24	1,344	2,068	3	3.00	3	0.97	0.90	0.50	20	10.39	1	17	6.41	1
136	24	24	1,152	1,776	3	3.00	3	0.96	0.84	0.50	19	9.95	1	16	5.75	1
137	32	32	2,048	3,136	3	3.00	3	0.97	0.89	0.50	27	13.36	1	23	7.70	1
138	24	24	1,152	1,776	3	3.00	3	0.97	0.90	0.50	24	12.29	1	21	7.24	1
139	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	31	14.96	1	27	8.56	1
140	20	20	800	1,240	3	3.00	3	0.95	0.89	0.50	16	9.27	1	13	5.52	1
141	32	32	2,048	3,136	3	3.00	3	0.96	0.90	0.50	19	10.30	1	15	6.18	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
142	24	24	1,152	1,776	3	3.00	3	0.96	0.85	0.50	21	10.35	1	17	6.03	1
143	32	32	2,048	3,136	3	3.00	3	0.97	0.85	0.50	24	11.73	1	20	6.68	1
144	16	16	512	800	3	3.00	3	0.96	0.88	0.50	22	10.12	1	16	5.69	1
146	20	20	800	1,240	3	3.00	3	0.96	0.88	0.50	20	11.04	1	15	6.30	1
147	20	20	800	1,240	3	3.00	3	0.96	0.87	0.50	19	10.52	1	15	6.15	1
148	24	24	1,152	1,776	3	3.00	3	0.96	0.85	0.50	23	10.85	1	18	6.21	1
149	16	16	512	800	3	3.00	3	0.95	0.83	0.50	15	7.10	1	12	4.41	1
150	24	24	1,152	1,776	3	3.00	3	0.97	0.91	0.50	24	12.27	1	21	7.16	1
151	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	27	13.41	1	24	7.71	1
152	20	20	800	1,240	3	3.00	3	0.95	0.82	0.50	19	7.86	1	14	4.69	1
153	32	32	2,048	3,136	3	3.00	3	0.96	0.82	0.50	24	9.06	1	18	5.36	1
154	32	32	2,048	3,136	3	3.00	3	0.97	0.85	0.50	25	11.19	1	19	6.42	1
155	16	16	512	800	3	3.00	3	0.95	0.84	0.50	11	6.14	1	10	4.17	1
157	20	20	800	1,240	3	3.00	3	0.96	0.90	0.50	23	12.57	1	18	7.34	1
158	24	24	1,152	1,776	3	3.00	3	0.97	0.89	0.50	24	12.95	1	19	7.63	1
160	24	24	1,152	1,776	3	3.00	3	0.96	0.88	0.50	17	8.85	1	14	5.55	1
161	16	16	512	800	3	3.00	3	0.94	0.85	0.50	16	7.84	1	12	4.75	1
162	24	24	1,152	1,776	3	3.00	3	0.96	0.86	0.50	19	9.73	1	15	5.97	1
163	32	32	2,048	3,136	3	3.00	3	0.97	0.86	0.50	25	11.15	1	21	6.87	1
164	16	16	512	800	3	3.00	3	0.96	0.88	0.50	15	8.34	1	13	5.37	1
165	24	24	1,152	1,776	3	3.00	3	0.97	0.87	0.50	29	13.39	1	21	7.40	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Table A.2: Detailed information of ternary-table instances

Instance #	# Rows	# Cols	$\|\mathcal{C}\|$	$\|\mathcal{V}\|$	Arity			Tightness			τ^*			$\|dom\|^*$		
					max	mean	min	max	mean	min	max	mean	min	max	mean	min
166	24	24	1,152	1,776	3	3.00	3	0.97	0.91	0.50	25	11.91	1	20	7.01	1
167	24	24	1,152	1,776	3	3.00	3	0.97	0.92	0.50	26	14.68	2	20	8.34	1
168	20	20	800	1,240	3	3.00	3	0.96	0.86	0.50	22	10.65	1	17	5.98	1
169	24	24	1,152	1,776	3	3.00	3	0.96	0.85	0.50	24	11.19	1	18	6.36	1
170	32	32	2,048	3,136	3	3.00	3	0.98	0.91	0.50	34	17.39	1	28	9.84	1
171	16	16	512	800	3	3.00	3	0.95	0.86	0.50	17	8.09	1	13	4.83	1
172	20	20	800	1,240	3	3.00	3	0.96	0.90	0.50	25	11.16	1	18	6.39	1
173	24	24	1,152	1,776	3	3.00	3	0.97	0.93	0.75	30	14.41	2	22	7.97	1
174	32	32	2,048	3,136	3	3.00	3	0.98	0.92	0.50	40	17.79	1	29	9.78	1
175	24	24	1,152	1,776	3	3.00	3	0.97	0.85	0.50	27	10.37	1	21	6.14	1
176	16	16	512	800	3	3.00	3	0.97	0.92	0.80	19	9.66	2	17	6.03	1
177	32	32	2,048	3,136	3	3.00	3	0.98	0.98	0.92	35	27.31	2	33	17.52	1
178	32	24	1,536	2,360	3	3.00	3	0.98	0.89	0.50	36	12.18	1	29	7.49	1
179	24	24	1,152	1,776	3	3.00	3	0.97	0.83	0.50	29	8.82	1	22	5.43	1
180	32	32	2,048	3,136	3	3.00	3	0.98	0.97	0.91	40	27.25	2	33	16.64	1

*τ is the number of tuples per constraint, $|dom|$ is the domain size

Appendix B

Detailed Experimental Results

This appendix contains the detailed results of the experiments discussed in Chapter 5. In particularly, we report the results of the first experiment with GAC-based algorithms and then report the results of the second experiment with all consistency algorithms.

B.1 Comparing GAC-Based Algorithms

In this section, we report the performance of the three GAC-based algorithms on two models of the Nonogram puzzle, namely, the ternary-tables and the global-tables models.

B.1.1 Ternary-Tables Model

Table B.1 and Table B.2 report the performance of three GAC-based algorithms, namely, CT [Demeulenaere *et al.*, 2016], STR2 [Lecoutre, 2011], and GAC2001 [Bessière *et al.*, 2005] with dom/deg [Bessière and Régin, 1996] and dom/wdeg [Boussemart *et al.*, 2004] variable ordering heuristic, respectively, on ternary-tables model.

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
001	6,328	6,328	6,328	6,328	6,328	9,212	9,212	9,212	9,212	9,212	3.25	2.43	1.89	2.46	1.77
002	0	0	0	0	0	3,140	3,140	3,140	3,140	3,140	3.58	2.20	3.54	3.57	2.18
003	125	125	125	125	125	1,379	1,379	1,379	1,379	1,379	0.71	0.50	0.68	0.69	0.49
004	624	624	624	624	624	2,425	2,425	2,425	2,425	2,425	1.39	0.95	1.31	1.35	0.91
005	48	48	48	48	48	1,838	1,838	1,838	1,838	1,838	1.32	0.88	1.29	1.30	0.87
006	8,111	8,111	8,111	8,111	8,111	10,908	10,908	10,908	10,908	10,908	2.93	2.22	2.02	2.55	1.90
007	16,983	16,983	16,983	16,983	16,983	22,280	22,280	22,280	22,280	22,280	8.30	6.17	5.63	6.62	4.70
008	77,172	77,172	77,172	77,172	77,172	84,211	84,211	84,211	84,211	84,211	10.13	8.26	6.00	9.17	7.40
009	0	0	0	0	0	3,109	3,109	3,109	3,109	3,109	3.72	2.37	3.71	3.72	2.37
010	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.64	0.43	0.63	0.64	0.43
011	19,714	19,714	19,714	19,714	19,714	22,926	22,926	22,926	22,926	22,926	3.41	2.59	2.12	2.72	2.00
012	2,227	2,227	2,227	2,227	2,227	5,456	5,456	5,456	5,456	5,456	3.86	2.48	3.75	3.82	2.42
013	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.35	0.93	1.35	1.35	0.93
014	16,715,578	16,715,578	16,715,578	16,715,578	16,715,578	19,482,586	19,482,586	19,482,586	19,482,586	19,482,586	3,222.60	2,603.63	1,331.38	2,307.70	1,833.74
015	1,235	1,235	1,235	1,235	1,235	4,426	4,426	4,426	4,426	4,426	4.63	3.32	4.58	4.61	3.31
016	74	74	74	74	74	3,217	3,217	3,217	3,217	3,217	4.12	2.78	4.10	4.12	2.78
017	0	0	0	0	0	1,256	1,256	1,256	1,256	1,256	0.64	0.43	0.63	0.64	0.43
018	0	0	0	0	0	1,780	1,780	1,780	1,780	1,780	1.21	0.77	1.20	1.21	0.77

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
019	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.82	2.46	3.80	3.82	2.46
020	2,638	2,638	2,638	2,638	2,638	5,919	5,919	5,919	5,919	5,919	3.83	2.40	3.68	3.76	2.35
021	157	157	157	157	157	1,948	1,948	1,948	1,948	1,948	1.19	0.74	1.17	1.18	0.74
022	476	476	476	476	476	3,625	3,625	3,625	3,625	3,625	3.47	2.07	3.43	3.46	2.06
023	273,602	273,602	273,602	273,602	273,602	310,708	310,708	310,708	310,708	310,708	84.59	71.37	29.89	51.44	41.96
024	7	7	7	7	7	3,147	3,147	3,147	3,147	3,147	3.95	2.60	3.94	3.94	2.60
025	6,978	6,978	6,978	6,978	6,978	10,252	10,252	10,252	10,252	10,252	4.08	2.60	3.82	4.00	2.54
026	0	0	0	0	0	3,137	3,137	3,137	3,137	3,137	4.39	3.07	4.37	4.38	3.07
027	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.40	0.99	1.41	1.41	0.99
028	44,890	44,890	44,890	44,890	44,890	51,294	51,294	51,294	51,294	51,294	21.16	17.37	8.44	11.83	9.19
029	43	43	43	43	43	1,824	1,824	1,824	1,824	1,824	1.38	0.95	1.36	1.37	0.94
030	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.63	0.43	0.63	0.64	0.43
031	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.20	0.76	1.19	1.20	0.76
032	0	0	0	0	0	1,245	1,245	1,245	1,245	1,245	0.65	0.44	0.64	0.64	0.44
033	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.76	2.39	3.75	3.76	2.39
034	15,299	15,299	15,299	15,299	15,299	18,322	18,322	18,322	18,322	18,322	3.93	3.14	2.37	3.01	2.32
035	10,287	10,287	10,287	10,287	10,287	12,916	12,916	12,916	12,916	12,916	2.93	2.20	1.82	2.29	1.66
036	129	129	129	129	129	3,279	3,279	3,279	3,279	3,279	3.59	2.19	3.50	3.55	2.15

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
037	0	0	0	0	0	1,242	1,242	1,242	1,242	1,242	0.65	0.44	0.64	0.64	0.44
038	0	0	0	0	0	3,005	3,005	3,005	3,005	3,005	3.37	2.11	3.34	3.37	2.11
039	2,264	2,264	2,264	2,264	2,264	5,482	5,482	5,482	5,482	5,482	3.80	2.37	3.65	3.75	2.33
040	3,577	3,577	3,577	3,577	3,577	5,735	5,735	5,735	5,735	5,735	1.87	1.27	1.64	1.75	1.17
041	145	145	145	145	145	965	965	965	965	965	0.29	0.20	0.28	0.29	0.20
042	0	0	0	0	0	802	802	802	802	802	0.30	0.22	0.30	0.30	0.22
043	4,839	4,839	4,839	4,839	4,839	6,155	6,155	6,155	6,155	6,155	0.69	0.54	0.50	0.63	0.49
044	-	52,780,865	52,780,865	-	52,780,865	-	57,555,968	57,555,968	-	57,555,968	-	4,754.28	2,743.54	-	3,668.45
045	0	0	0	0	0	813	813	813	813	813	0.31	0.23	0.31	0.31	0.23
046	7	7	7	7	7	1,190	1,190	1,190	1,190	1,190	0.60	0.42	0.60	0.60	0.42
047	0	0	0	0	0	1,242	1,242	1,242	1,242	1,242	0.70	0.50	0.70	0.70	0.51
048	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.70	0.50	0.70	0.70	0.50
049	812	812	812	812	812	1,685	1,685	1,685	1,685	1,685	0.38	0.28	0.33	0.36	0.26
050	0	0	0	0	0	801	801	801	801	801	0.29	0.21	0.29	0.30	0.21
051	1,197	1,197	1,197	1,197	1,197	3,064	3,064	3,064	3,064	3,064	1.36	0.89	1.27	1.30	0.85
052	105	105	105	105	105	3,255	3,255	3,255	3,255	3,255	4.03	2.66	3.98	4.01	2.65
053	622	622	622	622	622	2,433	2,433	2,433	2,433	2,433	1.29	0.85	1.26	1.28	0.84
054	1,876	1,876	1,876	1,876	1,876	3,751	3,751	3,751	3,751	3,751	1.43	0.97	1.30	1.35	0.90

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
055	0	0	0	0	0	3,137	3,137	3,137	3,137	3,137	3.61	2.23	3.58	3.60	2.22
056	504	504	504	504	504	1,363	1,363	1,363	1,363	1,363	0.36	0.26	0.32	0.34	0.25
057	18,553,364	18,553,364	18,553,364	18,553,364	18,553,364	19,366,117	19,366,117	19,366,117	19,366,117	19,366,117	1,743.85	1,469.07	713.35	1,033.60	876.84
058	6,647	6,647	6,647	6,647	6,647	11,349	11,349	11,349	11,349	11,349	7.16	5.08	4.44	5.24	3.45
059	19	19	19	19	19	1,799	1,799	1,799	1,799	1,799	1.27	0.84	1.26	1.27	0.84
060	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.35	0.92	1.34	1.35	0.92
061	20,637	20,637	20,637	20,637	20,637	24,208	24,208	24,208	24,208	24,208	7.13	5.29	5.07	5.69	4.01
062	16	16	16	16	16	819	819	819	819	819	0.30	0.21	0.30	0.30	0.22
064	32,375	32,375	32,375	32,375	32,375	35,552	35,552	35,552	35,552	35,552	3.04	2.44	1.62	2.19	1.70
065	1,411,610	1,411,610	1,411,610	1,411,610	1,411,610	1,512,076	1,512,076	1,512,076	1,512,076	1,512,076	225.20	189.95	96.85	159.57	131.75
066	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.65	0.44	0.64	0.65	0.44
067	6,205	6,205	6,205	6,205	6,205	9,661	9,661	9,661	9,661	9,661	3.92	2.43	3.68	3.83	2.36
068	123	123	123	123	123	925	925	925	925	925	0.30	0.21	0.30	0.30	0.21
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.33	0.90	1.33	1.33	0.90
071	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	4.09	2.73	4.07	4.09	2.74
072	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.44	1.02	1.43	1.44	1.02
073	0	0	0	0	0	1,247	1,247	1,247	1,247	1,247	0.62	0.41	0.61	0.62	0.41

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
074	120	120	120	120	120	1,314	1,314	1,314	1,314	1,314	0.61	0.42	0.60	0.60	0.42
075	1,507	1,507	1,507	1,507	1,507	4,681	4,681	4,681	4,681	4,681	3.76	2.36	3.71	3.75	2.36
076	386	386	386	386	386	1,665	1,665	1,665	1,665	1,665	0.77	0.56	0.71	0.74	0.53
077	0	0	0	0	0	3,143	3,143	3,143	3,143	3,143	3.59	2.21	3.57	3.59	2.20
078	708	708	708	708	708	2,510	2,510	2,510	2,510	2,510	1.27	0.81	1.22	1.26	0.81
079	196	196	196	196	196	3,389	3,389	3,389	3,389	3,389	3.55	2.13	3.43	3.48	2.07
080	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.61	0.41	0.61	0.61	0.41
081	189	189	189	189	189	1,976	1,976	1,976	1,976	1,976	1.20	0.76	1.18	1.20	0.76
082	0	0	0	0	0	2,408	2,408	2,408	2,408	2,408	2.14	1.33	2.11	2.14	1.34
083	0	0	0	0	0	801	801	801	801	801	0.29	0.21	0.30	0.29	0.21
085	1,973	1,973	1,973	1,973	1,973	4,073	4,073	4,073	4,073	4,073	1.93	1.38	1.45	1.66	1.13
086	21	21	21	21	21	2,776	2,776	2,776	2,776	2,776	2.76	1.71	2.70	2.76	1.68
087	1,413	1,413	1,413	1,413	1,413	3,293	3,293	3,293	3,293	3,293	1.39	0.92	1.28	1.34	0.88
088	1,521	1,521	1,521	1,521	1,521	4,730	4,730	4,730	4,730	4,730	3.82	2.43	3.70	3.77	2.38
089	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.66	0.46	0.66	0.66	0.45
090	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.77	2.41	3.75	3.77	2.42
091	6,441	6,441	6,441	6,441	6,441	9,971	9,971	9,971	9,971	9,971	5.64	4.00	4.27	4.66	3.10
092	357	357	357	357	357	3,515	3,515	3,515	3,515	3,515	3.83	2.54	3.79	3.82	2.45

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
093	0	0	0	0	0	803	803	803	803	803	0.30	0.20	0.29	0.29	0.20
094	169	169	169	169	169	1,960	1,960	1,960	1,960	1,960	1.36	0.93	1.35	1.39	0.93
095	343,380	343,380	343,380	343,380	343,380	375,570	375,570	375,570	375,570	375,570	84.45	68.69	37.01	64.56	51.19
096	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.28	0.84	1.27	1.27	0.84
097	122	122	122	122	122	939	939	939	939	939	0.29	0.20	0.28	0.29	0.20
098	256	256	256	256	256	1,507	1,507	1,507	1,507	1,507	0.67	0.46	0.65	0.66	0.45
099	468	468	468	468	468	2,304	2,304	2,304	2,304	2,304	1.50	1.05	1.37	1.42	0.97
100	275	275	275	275	275	3,428	3,428	3,428	3,428	3,428	4.00	2.61	3.86	3.92	2.54
101	0	0	0	0	0	1,349	1,349	1,349	1,349	1,349	0.83	0.59	0.83	0.83	0.59
102	9,133	9,133	9,133	9,133	9,133	11,592	11,592	11,592	11,592	11,592	3.30	2.58	1.88	2.32	1.70
103	0	0	0	0	0	3,140	3,140	3,140	3,140	3,140	3.65	2.28	3.63	3.68	2.27
104	0	0	0	0	0	1,242	1,242	1,242	1,242	1,242	0.62	0.42	0.62	0.63	0.42
105	819	819	819	819	819	4,021	4,021	4,021	4,021	4,021	3.86	2.43	3.61	3.72	2.32
106	0	0	0	0	0	801	801	801	801	801	0.31	0.22	0.31	0.31	0.22
107	6,218	6,218	6,218	6,218	6,218	9,717	9,717	9,717	9,717	9,717	4.94	3.33	4.04	4.53	2.96
108	0	0	0	0	0	3,139	3,139	3,139	3,139	3,139	3.51	2.11	3.48	3.53	2.11
109	528	528	528	528	528	1,428	1,428	1,428	1,428	1,428	0.39	0.29	0.34	0.36	0.26
110	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.65	0.45	0.65	0.66	0.45

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
111	1,628	1,628	1,628	1,628	1,628	3,459	3,459	3,459	3,459	3,459	1.33	0.88	1.28	1.31	0.87
112	4,226	4,226	4,226	4,226	4,226	7,518	7,518	7,518	7,518	7,518	4.35	2.85	3.95	4.20	2.71
113	139	139	139	139	139	1,381	1,381	1,381	1,381	1,381	0.64	0.44	0.64	0.64	0.44
114	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.27	0.83	1.25	1.26	0.83
115	859	859	859	859	859	2,133	2,133	2,133	2,133	2,133	0.71	0.50	0.69	0.70	0.49
116	2,190	2,190	2,190	2,190	2,190	5,493	5,493	5,493	5,493	5,493	4.89	3.32	4.09	4.42	2.94
117	42,645	42,645	42,645	42,645	42,645	48,645	48,645	48,645	48,645	48,645	6.50	5.23	3.25	5.04	4.05
118	615,560	615,560	615,560	615,560	615,560	644,696	644,696	644,696	644,696	644,696	59.28	51.03	30.99	48.86	40.99
119	12,717	12,717	12,717	12,717	12,717	16,470	16,470	16,470	16,470	16,470	5.11	3.47	4.07	4.42	2.87
120	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.44	1.03	1.45	1.44	1.03
121	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	4.27	2.95	4.26	4.26	2.95
122	251	251	251	251	251	3,403	3,403	3,403	3,403	3,403	3.95	2.58	3.90	3.94	2.57
123	5,154	5,154	5,154	5,154	5,154	6,635	6,635	6,635	6,635	6,635	0.94	0.69	0.81	0.92	0.67
124	18,921,598	18,921,598	18,921,598	18,921,598	18,921,598	20,287,097	20,287,097	20,287,097	20,287,097	20,287,097	2,218.24	1,859.68	1,006.14	1,662.04	1,383.85
125	184,402	184,402	184,402	184,402	184,402	202,353	202,353	202,353	202,353	202,353	35.58	28.18	15.37	23.07	17.61
126	410	410	410	410	410	1,232	1,232	1,232	1,232	1,232	0.33	0.25	0.32	0.33	0.24
128	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.73	2.35	3.70	3.73	2.35
129	6,458	6,458	6,458	6,458	6,458	8,726	8,726	8,726	8,726	8,726	2.12	1.53	1.60	1.85	1.30

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
130	1,460	1,460	1,460	1,460	1,460	4,670	4,670	4,670	4,670	4,670	3.71	2.29	3.56	3.62	2.21
131	117	117	117	117	117	1,896	1,896	1,896	1,896	1,896	1.21	0.77	1.20	1.20	0.77
132	9	9	9	9	9	1,251	1,251	1,251	1,251	1,251	0.63	0.42	0.62	0.62	0.42
133	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.20	0.76	1.19	1.20	0.76
134	46,256	46,256	46,256	46,256	46,256	51,310	51,310	51,310	51,310	51,310	6.81	5.43	3.47	4.93	3.82
135	61,548	61,548	61,548	61,548	61,548	68,266	68,266	68,266	68,266	68,266	14.66	12.28	6.32	9.44	7.65
136	5,239	5,239	5,239	5,239	5,239	7,496	7,496	7,496	7,496	7,496	1.95	1.38	1.48	1.70	1.16
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.26	0.83	1.25	1.26	0.82
139	397	397	397	397	397	3,561	3,561	3,561	3,561	3,561	3.76	2.35	3.64	3.69	2.30
140	61,200	61,200	61,200	61,200	61,200	71,668	71,668	71,668	71,668	71,668	14.84	12.09	5.68	9.38	7.52
141	1,511,558	1,511,558	1,511,558	1,511,558	1,511,558	1,582,762	1,582,762	1,582,762	1,582,762	1,582,762	220.64	186.79	86.37	122.88	102.62
142	82,293	82,293	82,293	82,293	82,293	90,614	90,614	90,614	90,614	90,614	12.96	10.53	5.57	8.69	6.89
143	74,199	74,199	74,199	74,199	74,199	85,196	85,196	85,196	85,196	85,196	17.65	13.53	8.19	11.48	8.22
144	0	0	0	0	0	800	800	800	800	800	0.32	0.23	0.32	0.31	0.23
146	1,327	1,327	1,327	1,327	1,327	2,642	2,642	2,642	2,642	2,642	0.74	0.52	0.70	0.72	0.51
147	599	599	599	599	599	1,867	1,867	1,867	1,867	1,867	0.69	0.48	0.67	0.68	0.48
148	3,158	3,158	3,158	3,158	3,158	5,131	5,131	5,131	5,131	5,131	1.49	1.01	1.32	1.38	0.92

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
149	0	0	0	0	0	800	800	800	800	800	0.30	0.21	0.30	0.30	0.21
150	656	656	656	656	656	2,550	2,550	2,550	2,550	2,550	1.42	0.96	1.30	1.35	0.89
151	1,881	1,881	1,881	1,881	1,881	5,125	5,125	5,125	5,125	5,125	3.92	2.45	3.62	3.72	2.30
152	18,724	18,724	18,724	18,724	18,724	20,896	20,896	20,896	20,896	20,896	1.90	1.47	1.13	1.41	1.07
153	-	-	58,790,242	-	58,790,242	-	-	61,601,993	-	61,601,993	-	-	2,570.04	-	3,149.91
154	15,760	15,760	15,760	15,760	15,760	20,073	20,073	20,073	20,073	20,073	5.36	3.59	4.25	4.82	3.12
155	0	0	0	0	0	805	805	805	805	805	0.29	0.20	0.28	0.28	0.20
157	4,924	4,924	4,924	4,924	4,924	6,490	6,490	6,490	6,490	6,490	1.10	0.83	0.90	1.04	0.77
158	3,193	3,193	3,193	3,193	3,193	5,113	5,113	5,113	5,113	5,113	1.58	1.10	1.41	1.50	1.03
160	339	339	339	339	339	2,170	2,170	2,170	2,170	2,170	1.36	0.90	1.28	1.36	0.90
161	565	565	565	565	565	1,402	1,402	1,402	1,402	1,402	0.34	0.25	0.32	0.33	0.24
162	31,361	31,361	31,361	31,361	31,361	35,250	35,250	35,250	35,250	35,250	6.04	4.86	3.19	4.50	3.52
163	5,066	5,066	5,066	5,066	5,066	8,607	8,607	8,607	8,607	8,607	5.16	3.52	3.89	4.19	2.68
164	0	0	0	0	0	802	802	802	802	802	0.30	0.22	0.30	0.30	0.22
165	921	921	921	921	921	2,752	2,752	2,752	2,752	2,752	1.44	0.99	1.34	1.37	0.93
166	23,979	23,979	23,979	23,979	23,979	29,682	29,682	29,682	29,682	29,682	8.18	6.54	3.86	5.87	4.58
167	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
168	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.67	0.47	0.66	0.67	0.47

Table B.1: GAC algorithms on ternary-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
169	29	29	29	29	29	1,809	1,809	1,809	1,809	1,809	1.24	0.80	1.23	1.24	0.80
170	203,829	203,829	203,829	203,829	203,829	222,994	222,994	222,994	222,994	222,994	28.99	22.72	13.88	20.72	15.60
171	3,806	3,806	3,806	3,806	3,806	4,875	4,875	4,875	4,875	4,875	0.53	0.40	0.40	0.46	0.34
172	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.66	0.46	0.66	0.66	0.46
173	355	355	355	355	355	2,163	2,163	2,163	2,163	2,163	1.43	0.98	1.36	1.40	0.96
174	38	38	38	38	38	3,182	3,182	3,182	3,182	3,182	3.81	2.45	3.80	3.82	2.45
175	68	68	68	68	68	1,851	1,851	1,851	1,851	1,851	1.23	0.80	1.22	1.23	0.79
176	154	154	154	154	154	958	958	958	958	958	0.31	0.23	0.31	0.31	0.23
177	0	0	0	0	0	3,137	3,137	3,137	3,137	3,137	4.28	2.97	4.29	4.29	2.98
178	0	0	0	0	0	2,368	2,368	2,368	2,368	2,368	2.14	1.39	2.11	2.13	1.37
179	602	602	602	602	602	2,417	2,417	2,417	2,417	2,417	1.27	0.82	1.22	1.24	0.80
180	16	16	16	16	16	3,159	3,159	3,159	3,159	3,159	4.28	2.96	4.27	4.28	2.96

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
001	1,451	1,451	2,752	2,752	2,752	3,249	3,249	4,552	4,552	4,552	1.28	0.83	1.28	1.30	0.85
002	0	0	0	0	0	3,139	3,139	3,139	3,139	3,139	3.63	2.24	3.57	3.59	2.21
003	0	0	11	11	11	1,241	1,241	1,254	1,254	1,254	0.68	0.48	0.68	0.68	0.48
004	187	187	0	0	0	1,973	1,973	1,778	1,778	1,778	1.35	0.91	1.30	1.30	0.87
005	1,745	1,745	56	56	56	3,585	3,585	1,843	1,843	1,843	1.43	0.98	1.30	1.31	0.88
006	1,401	1,401	272	272	272	3,275	3,275	2,076	2,076	2,076	1.51	1.05	1.34	1.36	0.93
007	72	72	0	0	0	3,311	3,311	3,236	3,236	3,236	4.24	2.79	4.20	4.22	2.78
008	4,793	4,793	6,846	5,336	6,846	6,714	6,714	8,945	7,360	8,945	1.72	1.23	1.63	1.75	1.30
009	0	0	0	0	0	3,109	3,109	3,109	3,109	3,109	3.74	2.39	3.73	3.74	2.39
010	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.65	0.44	0.64	0.65	0.44
011	1,940	1,940	923	923	923	3,850	3,850	2,785	2,785	2,785	1.51	1.01	1.27	1.30	0.85
012	693	693	367	367	367	3,859	3,859	3,527	3,527	3,527	3.78	2.40	3.73	3.75	2.37
013	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.36	0.94	1.36	1.36	0.94
014	24,278	24,278	10,999	8,667	10,999	26,932	26,932	13,288	10,916	13,288	2.88	2.22	1.74	1.84	1.42

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
015	125	125	102	75	102	3,275	3,275	3,258	3,225	3,258	4.57	3.27	4.57	4.57	3.28
016	3	3	3	3	3	3,145	3,145	3,143	3,143	3,143	4.14	2.81	4.12	4.14	2.80
017	0	0	0	0	0	1,245	1,245	1,245	1,245	1,245	0.64	0.44	0.64	0.64	0.43
018	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.22	0.78	1.21	1.22	0.78
019	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.85	2.48	3.82	3.85	2.48
020	751	751	566	566	566	3,915	3,915	3,729	3,729	3,729	3.69	2.30	3.65	3.68	2.29
021	19	19	19	19	19	1,798	1,798	1,798	1,798	1,798	1.19	0.74	1.18	1.19	0.74
022	425	425	516	516	516	3,570	3,570	3,678	3,678	3,678	3.49	2.08	3.46	3.50	2.09
023	154	154	543	543	543	1,946	1,946	2,348	2,348	2,348	1.28	0.84	1.26	1.29	0.84
024	7	7	4	4	4	3,148	3,148	3,143	3,143	3,143	3.97	2.62	3.96	3.98	2.62
025	7,405	7,405	5,857	4,498	5,857	10,635	10,635	9,060	7,715	9,060	4.02	2.57	3.72	3.81	2.40
026	0	0	0	0	0	3,137	3,137	3,137	3,137	3,137	4.41	3.09	4.40	4.41	3.09
027	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.42	1.00	1.42	1.42	1.01
028	10,772	10,772	56,645	43,636	56,645	14,233	14,233	62,243	48,617	62,243	5.29	3.71	8.30	9.36	8.40

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
029	0	0	107	107	107	1,777	1,777	1,893	1,893	1,893	1.38	0.96	1.38	1.39	0.97
030	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.64	0.44	0.64	0.64	0.44
031	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.21	0.77	1.21	1.21	0.77
032	15	15	0	0	0	1,266	1,266	1,241	1,241	1,241	0.67	0.46	0.64	0.65	0.44
033	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.79	2.42	3.77	3.79	2.42
034	388	388	462	462	462	2,226	2,226	2,279	2,279	2,279	1.55	1.11	1.45	1.47	1.04
035	1,655	1,655	768	768	768	3,565	3,565	2,604	2,604	2,604	1.47	0.99	1.24	1.28	0.82
036	440	440	154	154	154	3,621	3,621	3,297	3,297	3,297	3.75	2.33	3.52	3.56	2.16
037	7	7	0	0	0	1,251	1,251	1,242	1,242	1,242	0.66	0.45	0.65	0.66	0.45
038	0	0	0	0	0	3,005	3,005	3,005	3,005	3,005	3.40	2.14	3.37	3.40	2.14
039	183	183	1,225	1,225	1,225	3,350	3,350	4,410	4,411	4,410	3.67	2.27	3.60	3.66	2.25
040	732	732	1,301	1,301	1,301	2,753	2,753	3,326	3,326	3,326	1.62	1.07	1.57	1.61	1.05
041	156	156	75	75	75	965	965	879	879	879	0.29	0.21	0.29	0.29	0.20
042	0	0	0	0	0	802	802	802	802	802	0.31	0.23	0.31	0.31	0.23

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
043	117	117	200	200	200	941	941	1,032	1,032	1,032	0.33	0.25	0.33	0.34	0.25
044	346	346	206	206	206	2,149	2,149	2,009	2,009	2,009	1.28	0.84	1.25	1.27	0.82
045	0	0	151	151	151	802	802	962	962	962	0.31	0.23	0.32	0.32	0.23
046	7	7	7	7	7	1,189	1,189	1,190	1,190	1,190	0.61	0.43	0.61	0.61	0.43
047	0	0	0	0	0	1,242	1,242	1,242	1,242	1,242	0.71	0.51	0.71	0.71	0.51
048	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.71	0.51	0.71	0.71	0.51
049	324	324	226	238	226	1,169	1,169	1,056	1,069	1,056	0.34	0.25	0.32	0.33	0.24
050	0	0	0	0	0	801	801	801	801	801	0.30	0.22	0.30	0.30	0.22
051	917	917	468	468	468	2,720	2,720	2,266	2,266	2,266	1.29	0.85	1.25	1.26	0.82
052	107	107	626	626	626	3,261	3,261	3,808	3,808	3,808	4.07	2.70	4.05	4.09	2.73
053	163	163	161	161	161	1,952	1,952	1,954	1,954	1,954	1.28	0.84	1.26	1.28	0.84
054	56	56	732	732	732	1,839	1,839	2,534	2,534	2,534	1.26	0.82	1.27	1.29	0.85
055	0	0	0	0	0	3,139	3,139	3,137	3,137	3,137	3.64	2.25	3.60	3.62	2.24
056	365	365	334	342	334	1,187	1,187	1,152	1,162	1,152	0.33	0.24	0.32	0.33	0.24

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
057	2,147	2,147	2,047	2,047	2,047	4,053	4,053	3,974	3,974	3,974	1.43	0.96	1.33	1.40	0.93
058	1,320	1,320	257	257	257	4,498	4,498	3,416	3,416	3,416	3.68	2.26	3.49	3.53	2.13
059	19	19	19	19	19	1,799	1,799	1,799	1,799	1,799	1.28	0.85	1.28	1.28	0.85
060	0	0	0	0	0	1,778	1,778	1,777	1,777	1,777	1.36	0.94	1.36	1.36	0.94
061	22,980	22,980	8,092	11,631	8,092	26,860	26,860	11,509	15,106	11,509	7.86	5.93	4.44	5.01	3.19
062	16	16	16	16	16	820	820	819	819	819	0.31	0.22	0.30	0.31	0.22
064	1,001	1,001	700	891	700	2,296	2,296	1,988	2,198	1,988	0.73	0.52	0.69	0.73	0.50
065	1,375	1,375	1,735	1,732	1,735	3,426	3,426	3,812	3,808	3,812	1.73	1.15	1.60	1.70	1.13
066	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.66	0.45	0.65	0.66	0.45
067	719	719	1,879	1,879	1,879	3,894	3,894	5,080	5,080	5,080	3.59	2.19	3.56	3.61	2.20
068	123	123	123	123	123	925	925	925	925	925	0.31	0.22	0.30	0.30	0.22
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.34	0.91	1.34	1.34	0.92
071	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	4.12	2.76	4.10	4.12	2.76

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
072	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.45	1.03	1.44	1.45	1.03
073	136	136	99	99	99	1,383	1,383	1,349	1,349	1,349	0.63	0.42	0.62	0.63	0.42
074	216	216	133	133	133	1,420	1,420	1,333	1,333	1,333	0.63	0.44	0.61	0.62	0.43
075	394	394	2,229	2,229	2,229	3,541	3,541	5,394	5,394	5,394	3.75	2.36	3.78	3.82	2.42
076	290	290	48	48	48	1,551	1,551	1,300	1,300	1,300	0.73	0.53	0.71	0.72	0.51
077	318	318	360	360	360	3,478	3,478	3,528	3,528	3,528	3.64	2.25	3.62	3.64	2.26
078	482	482	462	462	462	2,275	2,275	2,249	2,249	2,249	1.26	0.81	1.23	1.25	0.80
079	1,446	1,446	332	332	332	4,699	4,699	3,515	3,515	3,515	3.69	2.25	3.47	3.51	2.11
080	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.62	0.42	0.62	0.62	0.42
081	198	198	220	220	220	1,987	1,987	2,007	2,007	2,007	1.22	0.78	1.20	1.21	0.77
082	0	0	0	0	0	2,408	2,408	2,408	2,408	2,408	2.16	1.35	2.14	2.14	1.35
083	0	0	0	0	0	801	801	801	801	801	0.30	0.22	0.30	0.30	0.21
085	770	770	1,059	1,059	1,059	2,597	2,597	2,911	2,911	2,911	1.30	0.85	1.26	1.30	0.84
086	21	21	21	21	21	2,776	2,776	2,776	2,776	2,776	2.78	1.73	2.73	2.77	1.73

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
087	307	307	308	308	308	2,108	2,108	2,103	2,103	2,103	1.24	0.80	1.22	1.24	0.80
088	1,561	1,561	1,238	1,235	1,238	4,742	4,742	4,404	4,401	4,404	3.81	2.38	3.70	3.75	2.39
089	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.67	0.46	0.66	0.67	0.46
090	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.83	2.42	3.77	3.79	2.44
091	482	482	0	0	0	3,633	3,633	3,140	3,140	3,140	3.85	2.50	3.77	3.83	2.42
092	334	334	357	357	357	3,488	3,488	3,514	3,514	3,514	3.95	2.48	3.82	3.90	2.51
093	0	0	0	0	0	802	802	803	803	803	0.29	0.21	0.29	0.29	0.21
094	115	115	78	78	78	1,902	1,902	1,865	1,865	1,865	1.37	0.97	1.36	1.41	0.94
095	3,903	3,903	2,625	2,625	2,625	7,152	7,152	5,875	5,875	5,875	3.99	2.58	4.00	3.98	2.57
096	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.29	0.85	1.28	1.29	0.85
097	149	149	308	308	308	960	960	1,125	1,125	1,125	0.29	0.20	0.29	0.30	0.21
098	319	319	294	294	294	1,577	1,577	1,546	1,546	1,546	0.68	0.47	0.66	0.67	0.46
099	240	240	74	74	74	2,028	2,028	1,857	1,857	1,857	1.39	0.96	1.34	1.36	0.92
100	283	283	184	348	184	3,442	3,442	3,334	3,509	3,334	4.05	2.64	3.90	3.97	2.56

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
101	0	0	0	0	0	1,349	1,349	1,349	1,349	1,349	0.84	0.60	0.83	0.84	0.60
102	12,189	12,189	5,130	4,359	5,130	14,460	14,460	7,087	6,276	7,087	2.94	2.31	1.52	1.61	1.19
103	11	11	11	11	11	3,152	3,152	3,153	3,153	3,153	3.68	2.30	3.68	3.69	2.32
104	0	0	2	0	2	1,242	1,242	1,248	1,242	1,248	0.64	0.42	0.63	0.64	0.43
105	2,817	2,817	1,375	1,195	1,375	6,071	6,071	4,558	4,374	4,558	4.04	2.56	3.63	3.69	2.29
106	0	0	0	0	0	801	801	801	801	801	0.31	0.23	0.31	0.31	0.23
107	4,242	4,242	1,781	1,462	1,781	7,579	7,579	5,051	4,752	5,051	4.46	2.91	3.72	3.87	2.40
108	261	261	0	0	0	3,400	3,400	3,141	3,141	3,141	3.53	2.15	3.51	3.54	2.14
109	262	262	262	262	262	1,065	1,065	1,065	1,065	1,065	0.32	0.24	0.32	0.32	0.24
110	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.67	0.46	0.66	0.66	0.46
111	534	534	225	225	225	2,333	2,333	2,011	2,012	2,011	1.29	0.85	1.25	1.26	0.82
112	1,118	1,118	920	946	920	4,311	4,311	4,107	4,137	4,107	3.95	2.54	3.80	3.87	2.46
113	139	139	139	139	139	1,381	1,381	1,381	1,381	1,381	0.65	0.45	0.65	0.66	0.45
114	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.27	0.84	1.27	1.28	0.84

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
115	306	306	309	309	309	1,556	1,556	1,560	1,560	1,560	0.68	0.48	0.68	0.69	0.48
116	4,747	4,747	5,306	5,306	5,306	7,999	7,999	8,611	8,611	8,611	4.54	3.08	4.22	4.53	3.05
117	1,889	1,889	1,527	1,592	1,527	3,271	3,271	2,910	2,972	2,910	0.93	0.68	0.78	0.84	0.63
118	19,965	19,965	7,250	8,294	7,250	22,707	22,707	9,398	10,464	9,398	3.17	2.49	1.73	2.03	1.45
119	803	803	974	974	974	3,969	3,969	4,155	4,155	4,155	3.70	2.31	3.63	3.67	2.28
120	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.45	1.04	1.46	1.46	1.05
121	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	4.29	2.97	4.28	4.30	2.98
122	79	79	79	79	79	3,222	3,222	3,222	3,222	3,222	3.93	2.59	3.92	3.95	2.58
123	2,298	2,298	1,211	1,211	1,211	3,691	3,691	2,550	2,550	2,550	0.83	0.58	0.71	0.75	0.53
124	86,620	86,620	30,799	33,560	30,799	92,540	92,540	33,970	36,817	33,970	10.14	8.46	2.72	3.54	2.71
125	13,693	13,693	6,694	20,340	6,694	17,357	17,357	10,150	24,414	10,150	4.73	3.14	3.86	4.92	2.59
126	148	148	0	0	0	967	967	801	801	801	0.33	0.24	0.31	0.31	0.23
128	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3.75	2.37	3.73	3.75	2.37
129	706	706	705	705	705	2,554	2,554	2,538	2,538	2,538	1.39	0.93	1.31	1.36	0.91

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
130	758	758	32	32	32	3,944	3,944	3,175	3,175	3,175	3.63	2.22	3.51	3.54	2.15
131	127	127	85	85	85	1,907	1,907	1,865	1,865	1,865	1.23	0.79	1.21	1.22	0.78
132	9	9	9	9	9	1,251	1,251	1,251	1,251	1,251	0.64	0.43	0.63	0.63	0.43
133	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1.21	0.77	1.21	1.21	0.77
134	661	661	1,051	1,051	1,051	2,484	2,484	2,864	2,864	2,864	1.27	0.82	1.22	1.25	0.80
135	10,640	10,640	4,218	6,100	4,218	12,964	12,964	6,463	8,430	6,463	2.40	1.75	1.82	2.09	1.33
136	1,226	1,226	387	387	387	3,050	3,050	2,198	2,199	2,198	1.31	0.86	1.24	1.26	0.81
137	26,086	26,086	101,933	30,924	101,933	30,780	30,780	108,024	35,034	108,024	8.66	6.51	9.69	6.10	9.83
138	0	0	0	0	0	1,777	1,777	1,777	1,777	1,777	1.27	0.84	1.26	1.27	0.84
139	39	39	205	205	205	3,186	3,186	3,354	3,354	3,354	3.68	2.29	3.64	3.67	2.29
140	1,492	1,492	2,866	1,892	2,866	2,804	2,804	4,190	3,189	4,190	0.74	0.52	0.73	0.71	0.54
141	717,142	717,142	266,910	1,134,425	266,910	746,943	746,943	277,151	1,173,414	277,151	78.91	66.98	15.76	91.03	16.65
142	5,882	5,882	3,093	3,093	3,093	7,928	7,928	5,017	5,017	5,017	1.77	1.27	1.34	1.39	0.93
143	23,269	23,269	7,722	9,161	7,722	27,511	27,511	11,133	12,665	11,133	6.26	4.51	3.91	4.22	2.59

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
144	0	0	0	0	0	800	800	800	800	800	0.32	0.24	0.32	0.32	0.24
146	1,638	1,638	1,681	1,681	1,681	2,934	2,934	3,009	3,009	3,009	0.74	0.53	0.73	0.76	0.55
147	165	165	631	631	631	1,414	1,414	1,897	1,897	1,897	0.68	0.47	0.68	0.69	0.48
148	1,777	1,777	1,111	1,230	1,111	3,655	3,655	2,960	3,082	2,960	1.39	0.93	1.28	1.32	0.86
149	0	0	0	0	0	800	800	800	800	800	0.30	0.22	0.30	0.30	0.22
150	230	230	87	87	87	2,035	2,035	1,881	1,881	1,881	1.32	0.88	1.28	1.29	0.85
151	244	244	39	39	39	3,413	3,413	3,185	3,185	3,185	3.67	2.27	3.58	3.59	2.20
152	13,251	13,251	1,603	1,603	1,603	15,206	15,206	2,935	2,935	2,935	1.46	1.12	0.68	0.71	0.49
153	54,195	54,195	275,315	89,282	275,315	59,473	59,473	288,926	95,998	288,926	10.13	7.73	16.45	9.98	18.04
154	2,487,785	2,487,785	15,310	10,804	15,310	2,630,495	2,630,495	18,901	14,191	18,901	364.95	305.40	4.18	4.08	2.96
155	0	0	0	0	0	805	805	802	802	802	0.29	0.21	0.29	0.29	0.20
157	2,241	2,241	977	977	977	3,586	3,586	2,287	2,287	2,287	0.85	0.63	0.74	0.78	0.57
158	586	586	1,136	1,136	1,136	2,384	2,384	2,942	2,942	2,942	1.31	0.87	1.31	1.33	0.89
160	10	10	325	93	325	1,789	1,789	2,127	1,888	2,127	1.23	0.79	1.26	1.27	0.84

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
161	1,015	1,015	659	848	659	1,896	1,896	1,509	1,716	1,509	0.38	0.28	0.33	0.36	0.25
162	8,137	8,137	4,185	4,598	4,185	10,152	10,152	6,121	6,554	6,121	1.91	1.39	1.42	1.57	1.04
163	3,021	3,021	3,152	2,600	3,152	6,282	6,282	6,390	5,820	6,390	4.02	2.56	3.71	3.80	2.42
164	0	0	0	0	0	802	802	802	802	802	0.31	0.22	0.31	0.31	0.22
165	1,571	1,571	36	36	36	3,477	3,477	1,819	1,819	1,819	1.57	1.10	1.31	1.32	0.89
166	1,290	1,290	2,085	1,904	2,085	3,102	3,102	3,903	3,724	3,903	1.36	0.92	1.35	1.39	0.95
167	37,424	37,424	9,404	6,711	9,404	40,929	40,929	11,719	8,828	11,719	5.07	4.13	1.89	1.91	1.65
168	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	0.68	0.47	0.67	0.68	0.47
169	0	0	29	29	29	1,777	1,777	1,810	1,810	1,810	1.25	0.82	1.24	1.25	0.82
170	5,723	5,723	4,330	5,447	4,330	9,106	9,106	7,715	8,837	7,715	4.25	2.80	3.93	4.11	2.64
171	1,479	1,479	1,204	1,194	1,204	2,353	2,353	2,083	2,071	2,083	0.39	0.29	0.34	0.36	0.27
172	0	0	0	0	0	1,241	1,241	1,241	1,241	1,241	0.67	0.47	0.67	0.67	0.47
173	92	92	405	410	405	1,878	1,878	2,210	2,214	2,210	1.36	0.93	1.36	1.39	0.96
174	9	9	492	492	492	3,151	3,151	3,638	3,638	3,638	3.84	2.47	3.83	3.85	2.48

Table B.2: GAC algorithms on ternary-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
175	68	68	68	68	68	1,851	1,851	1,850	1,850	1,850	1.25	0.81	1.24	1.25	0.81
176	135	135	43	43	43	939	939	846	846	846	0.32	0.24	0.32	0.32	0.23
177	0	0	0	0	0	3,138	3,138	3,137	3,137	3,137	4.31	3.00	4.31	4.31	3.00
178	0	0	0	0	0	2,362	2,362	2,363	2,363	2,363	2.15	1.39	2.12	2.13	1.38
179	409	409	140	140	140	2,224	2,224	1,927	1,927	1,927	1.30	0.85	1.22	1.23	0.79
180	342	342	0	0	0	3,490	3,490	3,139	3,139	3,139	4.31	2.99	4.29	4.30	2.99

B.1.2 Global-Tables Model

Similarly, Table B.3 and Table B.4 report the performance of three GAC-based algorithms with dom/deg [Bessière and Régin, 1996] and dom/wdeg [Boussemart *et al.*, 2004] variable ordering heuristic, respectively, on global-tables model.

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
001	7,552	7,552	7,552	7,552	7,552	9,693	9,693	9,693	9,693	9,693	856.31	866.21	853.79	855.65	861.72
002	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
003	31	31	31	31	31	443	443	443	443	443	25.21	25.28	25.18	25.21	25.23
004	15	15	15	15	15	604	604	604	604	604	1,399.49	1,400.95	1,399.51	1,399.55	1,399.74
005	32	32	32	32	32	630	630	630	630	630	10,874.12	10,854.53	10,854.46	10,852.12	10,854.30
006	6,072	6,072	6,072	6,072	6,072	8,680	8,680	8,680	8,680	8,680	223.18	229.79	220.50	223.54	231.60
007	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
008	6,651	6,651	6,651	6,651	6,651	8,638	8,638	8,638	8,638	8,638	1,815.45	1,843.16	1,811.60	1,815.24	1,831.46
009	0	0	0	0	0	1,015	1,015	1,015	1,015	1,015	480.82	481.14	480.98	480.83	481.17
010	0	0	0	0	0	401	401	401	401	401	5.30	5.32	5.30	5.29	5.32
011	254	254	254	254	254	871	871	871	871	871	80.98	81.27	80.96	80.96	81.27
012	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
013	0	0	0	0	0	577	577	577	577	577	54.06	54.12	54.06	54.05	54.11
014	160,756	160,756	160,756	160,756	160,756	182,614	182,614	182,614	182,614	182,614	2,544.90	2,838.34	2,478.19	2,536.23	2,764.76
015	197	197	197	197	197	1,256	1,256	1,256	1,256	1,256	555.76	556.48	555.81	555.72	556.35
016	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
017	7,290	7,290	7,290	7,290	7,290	10,419	10,419	10,419	10,419	10,419	72.70	83.29	69.20	70.99	72.86
018	0	0	0	0	0	576	576	576	576	576	5,777.47	5,778.58	5,778.53	5,777.51	5,780.53

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
019	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
020	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
021	0	0	0	0	0	576	576	576	576	576	230.17	230.27	230.19	230.14	230.26
022	470	470	470	470	470	1,558	1,558	1,558	1,558	1,558	5,421.61	5,423.49	5,420.70	5,421.22	5,422.94
023	4,122	4,122	4,122	4,122	4,122	5,590	5,590	5,590	5,590	5,590	21.61	25.95	19.76	21.28	24.99
024	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
025	27,040	27,040	27,040	27,040	27,040	30,939	30,939	30,939	30,939	30,939	5,436.66	5,563.89	5,413.43	5,421.54	5,475.38
026	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
027	0	0	0	0	0	576	576	576	576	576	78.53	78.65	78.56	78.52	78.65
028	480,607	480,607	480,607	480,607	480,607	551,685	551,685	551,685	551,685	551,685	4,395.32	6,098.42	3,665.84	4,224.91	6,734.20
029	0	0	0	0	0	576	576	576	576	576	1,919.61	1,915.23	1,914.82	1,914.75	1,915.25
030	0	0	0	0	0	400	400	400	400	400	3.05	3.07	3.06	3.05	3.06
031	0	0	0	0	0	578	578	578	578	578	11.16	11.19	11.16	11.15	11.19
032	342	342	342	342	342	901	901	901	901	901	17.34	17.68	17.07	17.21	17.44
033	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
034	23	23	23	23	23	610	610	610	610	610	173.83	174.13	173.83	173.78	174.05
035	1,028	1,028	1,028	1,028	1,028	1,729	1,729	1,729	1,729	1,729	5.11	5.34	5.01	5.08	5.38
036	175,176	175,176	175,176	175,176	175,176	211,161	211,161	211,161	211,161	211,161	892.45	1,711.39	744.39	876.56	1,647.28

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
037	0	0	0	0	0	406	406	406	406	406	9.27	9.31	9.27	9.26	9.30
038	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
039	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
040	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
041	22	22	22	22	22	286	286	286	286	286	0.13	0.13	0.13	0.13	0.13
042	5	5	5	5	5	264	264	264	264	264	0.58	0.59	0.59	0.58	0.59
043	0	0	0	0	0	258	258	258	258	258	2.02	2.03	2.02	2.02	2.03
044	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
045	0	0	0	0	0	257	257	257	257	257	3.10	3.12	3.10	3.10	3.12
047	0	0	0	0	0	401	401	401	401	401	11.84	11.87	11.85	11.84	11.87
048	0	0	0	0	0	402	402	402	402	402	5.85	5.88	5.87	5.85	5.88
049	720	720	720	720	720	1,135	1,135	1,135	1,135	1,135	2.74	2.83	2.67	2.74	2.81
050	6	6	6	6	6	268	268	268	268	268	1.06	1.07	1.06	1.06	1.07
051	-	-	12,247,721	12,247,721	-	-	-	15,438,189	15,438,189	-	-	-	1,841.73	8,405.78	-
052	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
053	171	171	171	171	171	771	771	771	771	771	1,083.97	1,084.74	1,084.41	1,083.98	1,084.81
054	150	150	150	150	150	738	738	738	738	738	64.20	64.34	64.21	64.19	64.31
055	648	648	648	648	648	1,702	1,702	1,702	1,702	1,702	3,747.01	3,748.22	3,747.31	3,747.26	3,748.48

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
056	0	0	0	0	0	257	257	257	257	257	0.66	0.66	0.66	0.66	0.66
057	30,817	30,817	30,817	30,817	30,817	39,211	39,211	39,211	39,211	39,211	188.80	224.67	182.83	187.99	225.36
058	6,852	6,852	6,852	6,852	6,852	9,739	9,739	9,739	9,739	9,739	12,812.17	13,083.35	12,776.74	12,799.83	12,843.66
059	0	0	0	0	0	578	578	578	578	578	133.18	133.28	133.22	133.16	133.28
060	0	0	0	0	0	577	577	577	577	577	10,274.61	10,288.14	10,276.14	10,274.35	10,275.81
062	3	3	3	3	3	262	262	262	262	262	1.05	1.05	1.05	1.04	1.05
064	85	85	85	85	85	494	494	494	494	494	6.14	6.19	6.15	6.14	6.19
066	720,483	720,483	720,483	720,483	720,483	806,684	806,684	806,684	806,684	806,684	86.40	199.08	27.83	83.62	217.29
067	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
068	0	0	0	0	0	256	256	256	256	256	0.29	0.29	0.29	0.29	0.29
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	0	0	0	0	0	576	576	576	576	576	673.13	673.40	673.24	673.10	673.34
071	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
072	0	0	0	0	0	576	576	576	576	576	1,057.23	1,057.72	1,057.38	1,057.24	1,057.66
073	57	57	57	57	57	478	478	478	478	478	8.54	8.59	8.53	8.53	8.58
074	98	98	98	98	98	512	512	512	512	512	26.98	27.06	26.97	26.97	27.04
075	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
076	252	252	252	252	252	680	680	680	680	680	58.81	59.04	58.75	58.78	58.97

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
077	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
078	116	116	116	116	116	719	719	719	719	719	500.49	500.83	500.43	500.41	500.79
079	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
080	117	117	117	117	117	550	550	550	550	550	3.56	3.61	3.55	3.56	3.62
081	66	66	66	66	66	660	660	660	660	660	40.79	40.92	40.79	40.78	40.92
082	366	366	366	366	366	1,207	1,207	1,207	1,207	1,207	233.48	234.16	233.32	233.45	234.11
083	0	0	0	0	0	256	256	256	256	256	0.10	0.10	0.10	0.10	0.10
085	13,418	13,418	13,418	13,418	13,418	17,289	17,289	17,289	17,289	17,289	17.34	25.59	15.04	16.87	22.95
087	115	115	115	115	115	708	708	708	708	708	160.97	161.20	160.95	160.93	161.14
088	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
089	13,598	13,598	13,598	13,598	13,598	16,003	16,003	16,003	16,003	16,003	57.06	61.71	55.83	56.86	60.70
090	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
091	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
092	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
093	14	14	14	14	14	276	276	276	276	276	0.20	0.20	0.20	0.20	0.20
094	71	71	71	71	71	662	662	662	662	662	1,618.93	1,619.35	1,618.25	1,617.25	1,617.88
095	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
096	0	0	0	0	0	576	576	576	576	576	3,738.09	3,735.19	3,738.55	3,734.62	3,738.73

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
097	163	163	163	163	163	443	443	443	443	443	0.07	0.07	0.06	0.07	0.07
098	103	103	103	103	103	522	522	522	522	522	1.29	1.32	1.29	1.29	1.32
099	8	8	8	8	8	598	598	598	598	598	216.43	216.75	216.40	216.56	216.73
100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
101	0	0	0	0	0	435	435	435	435	435	2,913.63	2,914.19	2,913.69	2,916.44	2,917.03
102	0	0	0	0	0	577	577	577	577	577	134.68	134.83	134.72	134.68	134.83
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	0	0	0	0	0	401	401	401	401	401	0.84	0.84	0.84	0.84	0.84
105	763	763	763	763	763	1,917	1,917	1,917	1,917	1,917	33.49	34.35	33.19	33.34	34.23
106	0	0	0	0	0	257	257	257	257	257	1.18	1.18	1.18	1.18	1.18
107	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
108	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
109	441	441	441	441	441	766	766	766	766	766	1.06	1.11	1.04	1.05	1.08
110	0	0	0	0	0	400	400	400	400	400	1.35	1.36	1.35	1.35	1.36
111	7,825	7,825	7,825	7,825	7,825	9,882	9,882	9,882	9,882	9,882	26.19	29.95	25.70	26.04	28.42
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	3	3	3	3	3	406	406	406	406	406	32.66	32.71	32.66	32.66	32.71
114	0	0	0	0	0	576	576	576	576	576	490.91	491.03	490.93	491.46	491.05

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
115	65	65	65	65	65	477	477	477	477	477	0.29	0.30	0.29	0.29	0.30
116	3,946	3,946	3,946	3,946	3,946	5,171	5,171	5,171	5,171	5,171	1,108.19	1,111.95	1,107.12	1,108.07	1,114.49
117	9,315	9,315	9,315	9,315	9,315	10,821	10,821	10,821	10,821	10,821	26.19	27.40	25.13	26.50	28.91
118	4,900,001	4,900,001	4,900,001	4,900,001	4,900,001	5,915,710	5,915,710	5,915,710	5,915,710	5,915,710	2,760.43	6,731.84	783.65	2,552.15	8,393.72
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	0	0	0	0	0	577	577	577	577	577	209.24	208.75	208.71	208.90	209.13
121	0	0	0	0	0	1,024	1,024	1,024	1,024	1,024	4,748.70	4,758.92	4,749.15	4,752.53	4,749.82
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	5,782	5,782	5,782	5,782	5,782	7,217	7,217	7,217	7,217	7,217	8.21	10.13	7.74	8.09	9.41
124	17,206	17,206	17,206	17,206	17,206	19,685	19,685	19,685	19,685	19,685	49.97	62.11	47.58	49.62	59.32
125	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
126	0	0	0	0	0	256	256	256	256	256	0.23	0.23	0.23	0.23	0.23
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	18,265	18,265	18,265	18,265	18,265	21,468	21,468	21,468	21,468	21,468	1,470.36	1,491.99	1,468.60	1,470.66	1,498.24
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
131	123	123	123	123	123	730	730	730	730	730	115.85	115.99	115.80	115.78	116.03
132	0	0	0	0	0	405	405	405	405	405	0.61	0.62	0.61	0.61	0.62
133	0	0	0	0	0	576	576	576	576	576	2.36	2.36	2.36	2.35	2.36

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
134	54,468	54,468	54,468	54,468	54,468	62,306	62,306	62,306	62,306	62,306	25.72	47.27	19.83	24.48	46.36
135	21,132	21,132	21,132	21,132	21,132	26,981	26,981	26,981	26,981	26,981	34.01	46.24	28.09	33.25	50.85
136	785	785	785	785	785	1,541	1,541	1,541	1,541	1,541	550.12	552.41	549.54	549.68	551.88
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	2	2	2	2	2	582	582	582	582	582	1,075.33	1,075.55	1,075.42	1,075.29	1,075.57
139	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	36,712	36,712	36,712	36,712	36,712	44,010	44,010	44,010	44,010	44,010	9.05	18.07	4.71	8.31	15.73
141	7,007,678	7,007,678	7,007,678	7,007,678	7,007,678	7,676,807	7,676,807	7,676,807	7,676,807	7,676,807	2,002.18	9,382.50	1,628.74	2,003.40	9,633.76
142	51	51	51	51	51	638	638	638	638	638	298.88	299.11	298.88	299.09	299.10
143	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
144	3	3	3	3	3	262	262	262	262	262	1.33	1.34	1.33	1.33	1.34
146	11,398	11,398	11,398	11,398	11,398	13,224	13,224	13,224	13,224	13,224	31.47	35.26	30.51	31.29	34.34
147	770	770	770	770	770	1,237	1,237	1,237	1,237	1,237	20.60	20.82	20.54	20.59	20.78
148	325,811	325,811	325,811	325,811	325,811	424,028	424,028	424,028	424,028	424,028	2,042.06	2,928.94	1,909.93	2,031.50	2,851.48
149	0	0	0	0	0	256	256	256	256	256	0.15	0.15	0.15	0.15	0.15
150	0	0	0	0	0	576	576	576	576	576	247.00	247.40	247.07	246.98	247.24
151	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
152	21,499	21,499	21,499	21,499	21,499	25,974	25,974	25,974	25,974	25,974	14.16	25.57	12.55	13.53	20.50

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
153	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
154	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
155	0	0	0	0	0	257	257	257	257	257	0.05	0.05	0.05	0.05	0.05
157	29	29	29	29	29	439	439	439	439	439	11.15	11.21	11.15	11.15	11.20
158	48	48	48	48	48	632	632	632	632	632	151.85	152.11	151.85	151.84	152.10
160	0	0	0	0	0	578	578	578	578	578	4.59	4.62	4.59	4.59	4.62
161	90	90	90	90	90	360	360	360	360	360	0.40	0.41	0.40	0.40	0.41
162	165	165	165	165	165	771	771	771	771	771	18.76	18.93	18.74	18.74	18.89
163	5,124	5,124	5,124	5,124	5,124	6,463	6,463	6,463	6,463	6,463	702.82	711.73	700.74	701.93	707.36
164	0	0	0	0	0	257	257	257	257	257	0.07	0.07	0.07	0.07	0.07
165	276	276	276	276	276	886	886	886	886	886	358.03	358.70	357.91	357.96	358.50
166	1,342	1,342	1,342	1,342	1,342	2,068	2,068	2,068	2,068	2,068	462.75	465.09	461.85	462.78	463.41
167	1,674,813	1,674,813	1,674,813	1,674,813	1,674,813	1,891,821	1,891,821	1,891,821	1,891,821	1,891,821	700.26	2,115.55	226.20	649.69	1,894.66
168	0	0	0	0	0	400	400	400	400	400	56.43	56.52	56.47	56.43	56.52
169	580	580	580	580	580	1,258	1,258	1,258	1,258	1,258	399.99	401.21	399.61	399.64	400.80
170	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
171	772	772	772	772	772	1,181	1,181	1,181	1,181	1,181	0.81	0.89	0.77	0.80	0.88
172	0	0	0	0	0	403	403	403	403	403	47.25	47.28	47.24	47.25	47.27

Table B.3: GAC algorithms on global-tables with dom/deg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
173	2,268	2,268	2,268	2,268	2,268	3,107	3,107	3,107	3,107	3,107	2,680.43	2,683.86	2,678.54	2,679.87	2,683.09
174	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
175	252	252	252	252	252	851	851	851	851	851	244.75	245.01	244.90	244.69	244.96
176	0	0	0	0	0	258	258	258	258	258	0.28	0.28	0.28	0.28	0.28
177	61	61	61	61	61	1,090	1,090	1,090	1,090	1,090	720.03	720.25	720.07	720.61	720.28
178	8	8	8	8	8	778	778	778	778	778	222.08	222.23	222.28	222.08	222.24
179	0	0	0	0	0	576	576	576	576	576	4.14	4.16	4.15	4.14	4.16
180	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
001	6,291	6,291	19	19	19	8,399	8,399	608	608	608	855.59	864.00	853.46	856.75	853.75
002	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
003	31	31	31	31	31	443	443	443	443	443	25.24	25.26	25.23	25.21	25.25
004	15	15	15	15	15	604	604	604	604	604	1,399.47	1,401.84	1,401.65	1,400.38	1,403.92
005	-	45	50	50	50	-	630	638	638	638	10,852.40	10,854.30	10,854.81	-	10,881.17
006	1,050	1,050	808	786	808	1,788	1,788	1,572	1,519	1,572	220.33	221.09	220.11	220.35	221.32
007	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
008	7,460	7,460	3,995	4,291	3,995	9,865	9,865	4,974	5,213	4,974	1,811.42	1,830.63	1,810.04	1,814.59	1,816.44
009	0	0	0	0	0	1,015	1,015	1,015	1,015	1,015	480.83	481.17	480.90	481.38	481.19
010	0	0	0	0	0	402	402	403	403	403	5.30	5.33	5.31	5.30	5.33
011	83	83	129	135	129	674	674	736	737	736	80.95	81.16	80.96	80.95	81.20
012	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
013	0	0	0	0	0	577	577	577	577	577	54.05	54.12	54.06	54.06	54.12
014	58,140	58,140	22,692	31,265	22,692	67,619	67,619	25,203	34,995	25,203	2,480.34	2,650.75	2,474.56	2,496.01	2,497.92

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
015	1,077	1,077	75	184	75	2,260	2,260	1,112	1,237	1,112	555.72	557.84	555.82	555.86	556.12
016	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
017	104	104	51	51	51	538	538	472	472	472	68.91	69.14	68.90	68.96	69.01
018	0	0	0	0	0	576	576	576	576	576	5,777.45	5,778.74	5,778.48	5,778.41	5,778.10
019	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
020	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
021	0	0	0	0	0	576	576	576	576	576	230.15	230.27	230.19	230.17	230.27
022	447	447	340	325	340	1,523	1,523	1,392	1,372	1,392	5,420.62	5,423.11	5,420.88	5,421.83	5,422.12
023	910	910	1,540	869	1,540	1,669	1,669	2,267	1,535	2,267	19.71	20.88	19.59	19.87	20.68
024	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
025	13,096	13,096	12,813	10,715	12,813	14,752	14,752	14,368	12,156	14,368	5,414.11	5,436.05	5,412.68	5,415.54	5,428.84
026	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
027	0	0	0	0	0	576	576	576	576	576	78.53	78.65	78.57	78.58	78.66
028	4,240	4,240	8,586	5,574	8,586	6,053	6,053	10,171	6,986	10,171	3,631.99	3,646.06	3,631.02	3,636.24	3,651.57

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
029	0	0	0	0	0	576	576	576	576	576	1,914.81	1,915.25	1,914.84	1,914.79	1,917.68
030	0	0	0	0	0	400	400	400	400	400	3.05	3.07	3.06	3.06	3.07
031	9	9	0	0	0	590	590	579	579	579	11.16	11.20	11.17	11.16	11.21
032	3,183	3,183	92	90	92	4,452	4,452	510	507	510	17.06	19.65	17.05	18.11	17.14
033	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
034	23	23	23	23	23	610	610	611	610	611	173.79	174.14	173.82	173.82	174.07
035	2,111	2,111	778	978	778	3,015	3,015	1,446	1,661	1,446	5.12	5.75	5.01	5.27	5.33
036	193,384	193,384	789	693	789	215,398	215,398	1,888	1,793	1,888	731.11	998.20	730.78	778.68	734.00
037	0	0	0	0	0	403	403	402	402	402	9.28	9.31	9.27	9.27	9.30
038	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
039	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
040	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
041	22	22	22	22	22	286	286	286	286	286	0.13	0.13	0.13	0.13	0.13
042	5	5	5	5	5	265	265	264	264	264	0.58	0.59	0.59	0.59	0.59

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
043	0	0	0	0	0	259	259	259	259	259	2.02	2.04	2.03	2.02	2.04
044	455	455	413	411	413	1,207	1,207	1,157	1,153	1,157	1,414.67	1,416.53	1,415.10	1,415.98	1,416.64
045	0	0	0	0	0	258	258	258	258	258	3.10	3.12	3.11	3.10	3.12
047	0	0	0	0	0	401	401	401	401	401	11.84	11.88	11.85	11.84	11.88
048	0	0	0	0	0	402	402	402	402	402	5.86	5.89	5.87	5.86	5.89
049	167	167	109	162	109	467	467	403	452	403	2.68	2.71	2.66	2.68	2.71
050	6	6	15	15	15	269	269	282	282	282	1.06	1.07	1.06	1.06	1.08
051	39,340	39,340	5,007	4,775	5,007	45,219	45,219	6,077	5,845	6,077	1,105.74	1,160.57	1,102.54	1,116.75	1,108.89
052	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
053	90	90	305	227	305	687	687	930	840	930	1,083.96	1,084.60	1,084.41	1,083.97	1,084.94
054	92	92	74	74	74	690	690	663	663	663	64.20	64.40	64.22	64.22	64.32
055	254	254	685	628	685	1,326	1,326	1,753	1,713	1,753	3,747.04	3,748.74	3,747.29	3,747.18	3,749.17
056	0	0	0	0	0	258	258	258	258	258	0.66	0.67	0.66	0.66	0.67
057	7,386	7,386	8,411	12,798	8,411	8,934	8,934	9,762	14,538	9,762	182.88	188.61	181.43	182.35	187.61

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
058	-	4,667	470	717	470	-	6,153	1,569	1,898	1,569	12,777.50	12,814.50	12,776.23	-	12,796.74
059	0	0	0	0	0	577	577	578	578	578	133.17	133.29	133.22	133.18	133.29
060	0	-	0	0	0	577	-	577	577	577	10,274.46	-	10,277.61	10,282.45	10,275.81
062	3	3	3	3	3	262	262	262	262	262	1.05	1.06	1.05	1.05	1.06
064	281	281	54	107	54	706	706	458	520	458	6.15	6.23	6.15	6.16	6.18
066	1,367	1,367	1,596	943	1,596	2,058	2,058	2,288	1,554	2,288	10.62	10.82	10.45	10.63	10.95
067	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
068	0	0	0	0	0	256	256	256	256	256	0.29	0.29	0.29	0.29	0.29
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	0	0	0	0	0	576	576	576	576	576	673.09	674.75	673.25	674.29	673.82
071	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
072	0	0	0	0	0	576	576	576	576	576	1,057.23	1,057.65	1,057.38	1,057.36	1,057.67
073	904	904	443	134	443	1,420	1,420	896	553	896	8.53	8.72	8.54	8.57	8.64
074	147	147	2	2	2	534	534	392	392	392	26.95	27.00	26.96	26.95	26.99

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
075	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
076	381	381	62	61	62	837	837	474	472	474	58.74	59.07	58.75	58.79	58.88
077	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
078	2,924	2,924	84	84	84	4,102	4,102	674	674	674	500.38	504.54	500.43	501.17	500.72
079	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
080	59	59	59	59	59	469	469	468	468	468	3.54	3.57	3.55	3.54	3.57
081	179	179	212	59	212	776	776	807	651	807	40.78	41.00	40.80	40.81	40.95
082	355	355	9	9	9	1,196	1,196	802	802	802	233.30	234.41	233.31	233.44	233.56
083	0	0	0	0	0	256	256	256	256	256	0.10	0.10	0.10	0.10	0.10
085	1,122	1,122	2,729	1,914	2,729	1,849	1,849	3,679	2,813	3,679	14.62	14.99	14.47	14.56	15.41
087	59	59	105	112	105	651	651	693	703	693	160.93	161.22	160.95	160.97	161.16
088	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
089	4,608	4,608	335	356	335	5,554	5,554	789	813	789	55.58	57.63	55.55	56.05	55.76
090	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
091	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
092	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
093	10	10	10	10	10	270	270	270	270	270	0.20	0.20	0.20	0.20	0.20
094	55	55	91	81	91	642	642	690	681	690	1,618.38	1,618.92	1,617.23	1,617.27	1,617.89
095	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
096	0	0	0	0	0	576	576	576	576	576	3,737.12	3,737.88	3,738.55	3,734.60	3,738.14
097	16	16	16	16	16	282	282	282	282	282	0.06	0.07	0.06	0.07	0.07
098	255	255	158	133	158	706	706	598	565	598	1.30	1.35	1.29	1.31	1.35
099	36	36	8	8	8	630	630	595	595	595	216.57	216.98	216.41	216.46	216.73
100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
101	0	0	0	0	0	435	435	435	435	435	2,913.62	2,914.20	2,917.02	2,913.69	2,914.22
102	0	0	0	0	0	577	577	577	577	577	134.69	134.83	134.72	134.68	134.84
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	0	0	0	0	0	401	401	401	401	401	0.84	0.84	0.84	0.84	0.84

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
105	512	512	464	502	464	1,578	1,578	1,529	1,572	1,529	33.23	33.54	33.17	33.26	33.55
106	0	0	0	0	0	257	257	257	257	257	1.18	1.19	1.18	1.18	1.19
107	4,849	4,849	-	-	-	6,782	6,782	-	-	-	-	1,102.36	-	1,094.97	-
108	23,924	-	11,756	-	11,756	26,341	-	13,604	-	13,604	-	-	11,676.94	11,694.03	11,705.12
109	24	24	92	92	92	288	288	356	356	356	1.03	1.05	1.04	1.04	1.05
110	0	0	0	0	0	400	400	400	400	400	1.35	1.36	1.35	1.35	1.36
111	2,386	2,386	611	1,334	611	3,353	3,353	1,266	2,094	1,266	25.58	26.53	25.51	25.69	25.77
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	1	1	0	0	0	405	405	402	402	402	32.66	32.71	32.66	32.66	32.71
114	0	0	0	0	0	576	576	576	576	576	490.84	491.05	490.89	490.90	491.05
115	61	61	65	65	65	472	472	477	477	477	0.29	0.30	0.29	0.29	0.30
116	13,508	13,508	5,183	11,533	5,183	15,389	15,389	6,614	13,160	6,614	1,110.36	1,120.06	1,107.32	1,110.47	1,121.16
117	251	251	365	381	365	693	693	812	825	812	24.91	24.96	24.86	24.87	25.05
118	2,149	2,149	2,522	5,867	2,522	2,984	2,984	3,336	7,466	3,336	512.68	510.60	509.04	509.82	510.53

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	0	0	0	0	0	577	577	577	577	577	208.60	208.76	208.89	208.61	209.02
121	0	0	0	0	0	1,024	1,024	1,024	1,024	1,024	4,748.59	4,749.83	4,754.89	4,748.70	4,749.83
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	1,683	1,683	1,418	1,039	1,418	2,272	2,272	2,027	1,590	2,027	7.71	7.98	7.63	7.69	8.10
124	1,908	1,908	1,632	7,301	1,632	2,709	2,709	2,457	8,877	2,457	48.15	48.32	47.13	47.30	48.29
125	38,597	38,597	44,875	50,560	44,875	44,574	44,574	50,455	56,901	50,455	6,569.03	6,695.98	6,550.76	6,564.83	6,666.73
126	0	0	0	0	0	256	256	256	256	256	0.23	0.23	0.23	0.23	0.23
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	546	546	752	466	752	1,244	1,244	1,447	1,093	1,447	1,467.82	1,468.98	1,467.54	1,467.51	1,470.23
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
131	89	89	12	12	12	674	674	594	594	594	115.76	115.91	115.80	115.79	115.88
132	0	0	0	0	0	401	401	402	402	402	0.61	0.62	0.62	0.62	0.62
133	0	0	0	0	0	576	576	576	576	576	2.36	2.37	2.37	2.36	2.36

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
134	12,539	12,539	3,027	3,873	3,027	14,307	14,307	3,792	4,718	3,792	18.45	22.16	18.18	19.38	19.01
135	8,707	8,707	597	525	597	10,225	10,225	1,363	1,287	1,363	26.96	30.12	26.83	28.13	27.37
136	279	279	320	265	320	902	902	946	903	946	549.54	550.48	549.53	549.79	550.49
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	0	0	0	0	0	578	578	578	578	578	1,075.31	1,075.55	1,075.36	1,075.29	1,075.56
139	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	3,018	3,018	308	299	308	3,926	3,926	779	773	779	3.56	4.30	3.54	3.83	3.65
141	175,904	175,904	73,024	34,472	73,024	187,841	187,841	79,379	38,299	79,379	1,479.53	1,602.28	1,478.43	1,492.83	1,521.24
142	1,223	1,223	53	93	53	1,953	1,953	637	682	637	298.86	299.75	298.89	298.98	299.10
143	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
144	3	3	3	3	3	262	262	262	262	262	1.33	1.34	1.33	1.33	1.34
146	1,173	1,173	534	534	534	1,673	1,673	1,000	978	1,000	30.29	30.61	30.28	30.34	30.51
147	928	928	642	500	642	1,417	1,417	1,102	950	1,102	20.56	20.81	20.54	20.60	20.74
148	269	269	674	664	674	938	938	1,331	1,452	1,331	1,893.97	1,894.80	1,893.60	1,893.78	1,895.37

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
149	0	0	0	0	0	256	256	256	256	256	0.15	0.16	0.16	0.15	0.16
150	0	0	0	0	0	576	576	576	576	576	246.99	247.24	247.07	246.99	247.41
151	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
152	6,136	6,136	1,173	1,878	1,173	7,545	7,545	1,784	2,569	1,784	12.23	14.69	12.13	12.63	12.64
153	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
154	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
155	0	0	0	0	0	257	257	257	257	257	0.05	0.05	0.05	0.05	0.05
157	0	0	63	9	63	401	401	470	412	470	11.14	11.18	11.15	11.14	11.19
158	51	51	375	305	375	638	638	996	924	996	151.88	152.19	151.87	151.87	152.35
160	4,387	4,387	0	0	0	5,673	5,673	577	577	577	4.59	6.61	4.60	5.46	4.62
161	45	45	39	39	39	313	313	300	300	300	0.40	0.41	0.40	0.40	0.40
162	120	120	248	132	248	715	715	859	728	859	18.74	18.86	18.75	18.75	18.89
163	494	494	937	1,549	937	1,656	1,656	2,111	2,788	2,111	701.39	704.97	700.57	701.43	703.04
164	0	0	0	0	0	257	257	257	257	257	0.07	0.07	0.07	0.07	0.07

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
165	456	456	557	215	557	1,102	1,102	1,175	817	1,175	358.02	358.95	357.92	358.07	358.68
166	1,592	1,592	1,880	1,128	1,880	2,347	2,347	2,643	1,848	2,643	462.16	465.74	461.87	462.86	463.51
167	25,348	25,348	17,773	17,021	17,773	28,912	28,912	20,594	19,769	20,594	181.74	187.61	174.56	181.93	190.80
168	0	0	0	0	0	400	400	400	400	400	56.47	56.55	56.44	56.43	56.52
169	326	326	242	245	242	967	967	844	852	844	399.82	400.69	399.60	399.65	400.19
170	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
171	274	274	49	49	49	561	561	311	311	311	0.75	0.78	0.75	0.76	0.76
172	0	0	0	0	0	402	402	402	403	402	47.22	47.28	47.24	47.22	47.27
173	49	49	310	348	310	633	633	936	989	936	2,678.58	2,678.89	2,678.24	2,678.20	2,679.76
174	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
175	317	317	321	321	321	938	938	928	928	928	244.71	245.12	244.72	244.93	245.27
176	0	0	0	0	0	257	257	258	258	258	0.28	0.28	0.28	0.28	0.29
177	96	96	61	61	61	1,131	1,131	1,091	1,091	1,091	720.03	720.26	720.07	720.00	720.25
178	8	8	8	8	8	778	778	778	778	778	222.13	222.26	222.15	222.09	222.25

Table B.4: GAC algorithms on global-tables with dom/wdeg

Instance #	# BT					# NV					CPU Time				
	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO	STR2-lex	GAC2001-lex	CT-FIFO	STR2-FIFO	GAC2001-FIFO
179	0	0	0	0	0	576	576	576	576	576	4.14	4.16	4.15	4.15	4.16
180	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

B.2 Comparing All Consistency Algorithms

In this section, we report the performance of all consistency algorithms with dom/wdeg [Boussemart *et al.*, 2004] ordering heuristic and FIFO propagation queue on two models of the Nonogram puzzle, namely, the ternary-tables and the global-tables models.

B.2.1 Ternary-Tables Model

Tables B.5 and B.6 report the performance of all algorithms with # BT and # NV, and CPU Time, respectively, on ternary-tables model.

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak$^+$-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC**$_{around\cup}$	**PrePeak$^+$-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
001	0	2,752	0	2,752	0	0	0	2,752	2,752	2,752	1,776	4,552	1,776	4,552	1,776	1,776	1,776	4,552	4,552	4,552
002	0	0	0	0	0	0	0	0	0	0	3,136	3,139	3,136	3,139	3,136	3,136	3,136	3,139	3,139	3,139
003	0	11	0	11	0	0	0	11	11	11	1,240	1,254	1,240	1,254	1,240	1,240	1,240	1,254	1,254	1,254
004	0	0	0	0	0	0	0	0	0	0	1,776	1,778	1,776	1,778	1,776	1,776	1,776	1,778	1,778	1,778
005	0	56	26	56	0	0	0	56	56	56	1,776	1,843	1,811	1,843	1,776	1,776	1,776	1,843	1,843	1,843
006	0	272	29	272	0	0	0	272	272	272	1,776	2,076	1,814	2,076	1,776	1,776	1,776	2,076	2,076	2,076
007	0	0	0	0	0	0	0	0	0	0	3,233	3,236	3,234	3,236	3,233	3,233	3,233	3,236	3,236	3,236
008	0	5,336	0	5,336	0	0	0	6,846	6,846	5,336	1,776	7,360	1,776	7,360	1,776	1,776	1,776	8,945	8,945	7,360
009	0	0	0	0	0	0	0	0	0	0	3,109	3,109	3,109	3,109	3,109	3,109	3,109	3,109	3,109	3,109
010	0	0	0	0	0	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240
011	0	923	160	923	0	0	0	923	923	923	1,776	2,785	1,944	2,785	1,776	1,776	1,776	2,785	2,785	2,785
012	0	367	2	367	0	0	0	367	367	367	3,136	3,527	3,145	3,527	3,136	3,136	3,136	3,527	3,527	3,527
013	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,777	1,777	1,776	1,776	1,776	1,777	1,777	1,777
014	0	8,667	4,928	8,667	0	0	0	10,999	10,999	8,667	1,776	10,916	6,906	10,916	1,776	1,776	1,776	13,288	13,288	10,916
015	0	75	82	75	0	0	0	102	102	75	3,136	3,225	3,232	3,225	3,136	3,136	3,136	3,258	3,258	3,225
016	0	3	16	3	0	0	0	3	3	3	3,136	3,143	3,154	3,143	3,136	3,136	3,136	3,143	3,143	3,143

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
017	0	0	0	0	0	0	0	0	0	0	1,240	1,245	1,240	1,245	1,240	1,240	1,240	1,245	1,245	1,245
018	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,776	1,777	1,776	1,776	1,776	1,777	1,777	1,777
019	0	0	0	0	0	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136
020	0	566	24	566	0	0	0	566	566	566	3,136	3,729	3,166	3,729	3,136	3,136	3,136	3,729	3,729	3,729
021	0	19	19	19	0	0	0	19	19	19	1,776	1,798	1,798	1,798	1,776	1,776	1,776	1,798	1,798	1,798
022	0	516	0	516	0	0	0	516	516	516	3,136	3,678	3,136	3,678	3,136	3,136	3,136	3,678	3,678	3,678
023	0	543	129	543	0	0	0	543	543	543	1,776	2,348	1,920	2,348	1,776	1,776	1,776	2,348	2,348	2,348
024	0	4	0	4	0	0	0	4	4	4	3,136	3,143	3,136	3,143	3,136	3,136	3,136	3,143	3,143	3,143
025	0	4,498	1,110	4,498	0	0	0	5,857	5,857	4,498	3,136	7,715	4,271	7,715	3,136	3,136	3,136	9,060	9,060	7,715
026	0	0	0	0	0	0	0	0	0	0	3,136	3,137	3,136	3,137	3,136	3,136	3,136	3,137	3,137	3,137
027	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,776	1,777	1,776	1,776	1,776	1,777	1,777	1,777
028	0	43,636	7,479	43,636	0	0	0	56,645	56,645	43,636	3,136	48,617	10,771	48,617	3,136	3,136	3,136	62,243	62,243	48,617
029	0	107	0	107	0	0	0	107	107	107	1,776	1,893	1,776	1,893	1,776	1,776	1,776	1,893	1,893	1,893
030	0	0	0	0	0	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240
031	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,776	1,777	1,776	1,776	1,776	1,777	1,777	1,777
032	0	0	0	0	0	0	0	0	0	0	1,240	1,241	1,240	1,241	1,240	1,240	1,240	1,241	1,241	1,241
033	0	0	0	0	0	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
034	0	462	240	462	0	0	0	462	462	462	1,776	2,279	2,043	2,279	1,776	1,776	1,776	2,279	2,279	2,279
035	0	768	145	768	0	0	0	768	768	768	1,776	2,604	1,933	2,604	1,776	1,776	1,776	2,604	2,604	2,604
036	0	154	51	154	0	0	0	154	154	154	3,136	3,297	3,193	3,297	3,136	3,136	3,136	3,297	3,297	3,297
037	0	0	0	0	0	0	0	0	0	0	1,240	1,242	1,242	1,242	1,240	1,240	1,240	1,242	1,242	1,242
038	0	0	0	0	0	0	0	0	0	0	3,003	3,005	3,005	3,005	3,003	3,003	3,003	3,005	3,005	3,005
039	0	1,225	0	1,225	0	0	0	1,225	1,225	1,225	3,136	4,411	3,136	4,411	3,136	3,136	3,136	4,410	4,410	4,411
040	0	1,301	268	1,301	0	0	0	1,301	1,301	1,301	1,972	3,326	2,246	3,326	1,972	1,972	1,972	3,326	3,326	3,326
041	0	75	0	75	0	0	0	75	75	75	800	879	800	879	800	800	800	879	879	879
042	0	0	0	0	0	0	0	0	0	0	800	802	800	802	800	800	800	802	802	802
043	0	200	0	200	0	0	0	200	200	200	800	1,032	801	1,032	800	800	800	1,032	1,032	1,032
044	0	206	0	206	0	0	0	206	206	206	1,776	2,009	1,776	2,009	1,776	1,776	1,776	2,009	2,009	2,009
045	0	151	0	151	0	0	0	151	151	151	800	962	800	962	800	800	800	962	962	962
046	0	7	0	7	0	0	0	7	7	7	1,179	1,190	1,179	1,190	1,179	1,179	1,179	1,190	1,190	1,190
047	0	0	0	0	0	0	0	0	0	0	1,240	1,242	1,240	1,242	1,240	1,240	1,240	1,242	1,242	1,242
048	0	0	0	0	0	0	0	0	0	0	1,240	1,241	1,240	1,241	1,240	1,240	1,240	1,241	1,241	1,241
049	0	238	1	238	0	0	0	226	226	238	800	1,069	808	1,069	800	800	800	1,056	1,056	1,069
050	0	0	0	0	0	0	0	0	0	0	800	801	800	801	800	800	800	801	801	801

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
051	0	468	126	468	0	0	0	468	468	468	1,776	2,266	1,915	2,266	1,776	1,776	1,776	2,266	2,266	2,266
052	0	626	127	626	0	0	0	626	626	626	3,136	3,808	3,274	3,808	3,136	3,136	3,136	3,808	3,808	3,808
053	0	161	0	161	0	0	0	161	161	161	1,776	1,954	1,776	1,954	1,776	1,776	1,776	1,954	1,954	1,954
054	0	732	76	732	0	0	0	732	732	732	1,776	2,534	1,856	2,534	1,776	1,776	1,776	2,534	2,534	2,534
055	0	0	0	0	0	0	0	0	0	0	3,136	3,137	3,138	3,137	3,136	3,136	3,136	3,137	3,137	3,137
056	0	342	0	342	0	0	0	334	334	342	800	1,162	802	1,162	800	800	800	1,152	1,152	1,162
057	0	2,047	3,227	2,047	0	0	0	2,047	2,047	2,047	1,776	3,974	5,190	3,974	1,776	1,776	1,776	3,974	3,974	3,974
058	0	257	0	257	0	0	0	257	257	257	3,136	3,416	3,137	3,416	3,136	3,136	3,136	3,416	3,416	3,416
059	0	19	14	19	0	0	0	19	19	19	1,776	1,799	1,792	1,799	1,776	1,776	1,776	1,799	1,799	1,799
060	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,776	1,777	1,776	1,776	1,776	1,777	1,777	1,777
061	0	11,631	8,344	11,631	0	0	0	8,092	8,092	11,631	3,233	15,106	11,740	15,106	3,233	3,233	3,233	11,509	11,509	15,106
062	0	16	0	16	0	0	0	16	16	16	800	819	800	819	800	800	800	819	819	819
064	0	891	350	891	0	0	0	700	700	891	1,240	2,198	1,607	2,198	1,240	1,240	1,240	1,988	1,988	2,198
065	0	1,732	7,283	1,732	0	0	0	1,735	1,735	1,732	1,972	3,808	9,427	3,808	1,972	1,972	1,972	3,812	3,812	3,808
066	0	0	0	0	0	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240
067	0	1,879	0	1,879	0	0	0	1,879	1,879	1,879	3,136	5,080	3,138	5,080	3,136	3,136	3,136	5,080	5,080	5,080
068	0	123	0	123	0	0	0	123	123	123	800	925	800	925	800	800	800	925	925	925

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
069	-	-	-	-	-	-	-	80,029,057	-	81,776,189	-	-	-	-	-	-	-	82,592,361	-	84,361,593
070	0	0	0	0	0	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776
071	0	0	0	0	0	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136
072	0	0	0	0	0	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776
073	0	99	0	99	0	0	0	99	99	99	1,240	1,349	1,244	1,349	1,240	1,240	1,240	1,349	1,349	1,349
074	0	133	126	133	0	0	0	133	133	133	1,192	1,333	1,326	1,333	1,192	1,192	1,192	1,333	1,333	1,333
075	0	2,229	1,657	2,229	0	0	0	2,229	2,229	2,229	3,136	5,394	4,812	5,394	3,136	3,136	3,136	5,394	5,394	5,394
076	0	48	0	48	0	0	0	48	48	48	1,240	1,300	1,240	1,300	1,240	1,240	1,240	1,300	1,300	1,300
077	0	360	0	360	0	0	0	360	360	360	3,136	3,528	3,137	3,528	3,136	3,136	3,136	3,528	3,528	3,528
078	0	462	0	462	0	0	0	462	462	462	1,776	2,249	1,776	2,249	1,776	1,776	1,776	2,249	2,249	2,249
079	0	332	0	332	0	0	0	332	332	332	3,136	3,515	3,137	3,515	3,136	3,136	3,136	3,515	3,515	3,515
080	0	0	0	0	0	0	0	0	0	0	1,240	1,241	1,240	1,241	1,240	1,240	1,240	1,241	1,241	1,241
081	0	220	0	220	0	0	0	220	220	220	1,776	2,007	1,778	2,007	1,776	1,776	1,776	2,007	2,007	2,007
082	0	0	0	0	0	0	0	0	0	0	2,408	2,408	2,408	2,408	2,408	2,408	2,408	2,408	2,408	2,408
083	0	0	0	0	0	0	0	0	0	0	800	801	800	801	800	800	800	801	801	801
085	0	1,059	882	1,059	0	0	0	1,059	1,059	1,059	1,776	2,911	2,730	2,911	1,776	1,776	1,776	2,911	2,911	2,911
086	0	21	0	21	0	0	0	21	21	21	2,748	2,776	2,748	2,776	2,748	2,748	2,748	2,776	2,776	2,776

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
087	0	308	157	308	0	0	0	308	308	308	1,776	2,103	1,941	2,103	1,776	1,776	1,776	2,103	2,103	2,103
088	0	1,235	0	1,235	0	0	0	1,238	1,238	1,235	3,136	4,401	3,137	4,401	3,136	3,136	3,136	4,404	4,404	4,401
089	0	0	0	0	0	0	0	0	0	0	1,240	1,241	1,241	1,241	1,240	1,240	1,240	1,241	1,241	1,241
090	0	0	0	0	0	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136
091	0	0	0	0	0	0	0	0	0	0	3,136	3,140	3,140	3,140	3,136	3,136	3,136	3,140	3,140	3,140
092	0	357	0	357	0	0	0	357	357	357	3,136	3,514	3,137	3,514	3,136	3,136	3,136	3,514	3,514	3,514
093	0	0	0	0	0	0	0	0	0	0	800	803	800	803	800	800	800	803	803	803
094	0	78	71	78	0	0	0	78	78	78	1,776	1,865	1,849	1,865	1,776	1,776	1,776	1,865	1,865	1,865
095	0	2,625	1,474	2,625	0	0	0	2,625	2,625	2,625	3,136	5,875	4,666	5,875	3,136	3,136	3,136	5,875	5,875	5,875
096	0	0	0	0	-	-	-	0	0	0	1,776	1,776	1,776	1,776	-	-	-	1,776	1,776	1,776
097	0	308	154	308	0	0	0	308	308	308	800	1,125	962	1,125	800	800	800	1,125	1,125	1,125
098	0	294	0	294	0	0	0	294	294	294	1,240	1,546	1,240	1,546	1,240	1,240	1,240	1,546	1,546	1,546
099	0	74	74	74	0	0	0	74	74	74	1,776	1,857	1,857	1,857	1,776	1,776	1,776	1,857	1,857	1,857
100	0	348	297	348	0	0	0	184	184	348	3,136	3,509	3,454	3,509	3,136	3,136	3,136	3,334	3,334	3,509
101	0	0	0	0	-	-	-	0	0	0	1,349	1,349	1,349	1,349	-	-	-	1,349	1,349	1,349
102	0	4,359	4,359	4,359	0	0	0	5,130	5,130	4,359	1,776	6,276	6,276	6,276	1,776	1,776	1,776	7,087	7,087	6,276
103	0	11	0	11	0	0	0	11	11	11	3,136	3,153	3,138	3,153	3,136	3,136	3,136	3,153	3,153	3,153

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
104	0	0	0	0	0	0	0	2	2	0	1,240	1,242	1,240	1,242	1,240	1,240	1,240	1,248	1,248	1,242
105	0	1,195	1,195	1,195	0	0	0	1,375	1,375	1,195	3,136	4,374	4,374	4,374	3,136	3,136	3,136	4,558	4,558	4,374
106	0	0	0	0	-	-	-	0	0	0	800	801	800	801	-	-	-	801	801	801
107	0	1,462	1,462	1,462	0	0	0	1,781	1,781	1,462	3,136	4,752	4,751	4,752	3,136	3,136	3,136	5,051	5,051	4,752
108	0	0	0	0	0	0	0	0	0	0	3,136	3,141	3,141	3,141	3,136	3,136	3,136	3,141	3,141	3,141
109	0	262	0	262	0	0	0	262	262	262	800	1,065	800	1,065	800	800	800	1,065	1,065	1,065
110	0	0	0	0	0	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240
111	0	225	245	225	0	0	0	225	225	225	1,776	2,012	2,039	2,012	1,776	1,776	1,776	2,011	2,011	2,012
112	0	946	0	946	0	0	0	920	920	946	3,136	4,137	3,136	4,137	3,136	3,136	3,136	4,107	4,107	4,137
113	0	139	138	139	0	0	0	139	139	139	1,240	1,381	1,380	1,381	1,240	1,240	1,240	1,381	1,381	1,381
114	0	0	0	0	0	0	0	0	0	0	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776	1,776
115	0	309	0	309	0	0	0	309	309	309	1,240	1,560	1,244	1,560	1,240	1,240	1,240	1,560	1,560	1,560
116	0	5,306	1,847	5,306	0	0	0	5,306	5,306	5,306	3,136	8,611	5,023	8,611	3,136	3,136	3,136	8,611	8,611	8,611
117	0	1,592	94	1,592	0	0	0	1,527	1,527	1,592	1,240	2,972	1,346	2,972	1,240	1,240	1,240	2,910	2,910	2,972
118	0	8,294	6,380	8,294	0	0	0	7,250	7,250	8,294	1,776	10,464	8,429	10,464	1,776	1,776	1,776	9,398	9,398	10,464
119	0	974	395	974	0	0	0	974	974	974	3,136	4,155	3,564	4,155	3,136	3,136	3,136	4,155	4,155	4,155
120	0	0	0	0	-	-	-	0	0	0	1,776	1,776	1,776	1,776	-	-	-	1,776	1,776	1,776

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak$^+$-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC**$_{around\cup}$	**PrePeak$^+$-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
121	0	0	0	0	0	0	0	0	0	0	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136	3,136
122	0	79	30	79	0	0	0	79	79	79	3,136	3,222	3,170	3,222	3,136	3,136	3,136	3,222	3,222	3,222
123	0	1,211	1,862	1,211	0	0	0	1,211	1,211	1,211	1,240	2,550	3,221	2,550	1,240	1,240	1,240	2,550	2,550	2,550
124	10,015	33,560	29,023	33,560	0	0	0	30,799	30,799	33,560	12,172	36,817	32,106	36,817	1,776	1,776	1,776	33,970	33,970	36,817
125	0	20,340	9,510	20,340	0	0	0	6,694	6,694	20,340	3,136	24,414	13,125	24,414	3,136	3,136	3,136	10,150	10,150	24,414
126	0	0	0	0	0	0	0	0	0	0	800	801	801	801	800	800	800	801	801	801
128	0	0	0	0	-	-	-	0	0	0	3,136	3,136	3,136	3,136	-	-	-	3,136	3,136	3,136
129	0	705	36	705	0	0	0	705	705	705	1,776	2,538	1,817	2,538	1,776	1,776	1,776	2,538	2,538	2,538
130	0	32	0	32	0	0	0	32	32	32	3,136	3,175	3,138	3,175	3,136	3,136	3,136	3,175	3,175	3,175
131	0	85	0	85	0	0	0	85	85	85	1,776	1,865	1,776	1,865	1,776	1,776	1,776	1,865	1,865	1,865
132	0	9	0	9	0	0	0	9	9	9	1,240	1,251	1,241	1,251	1,240	1,240	1,240	1,251	1,251	1,251
133	0	0	0	0	-	-	-	0	0	0	1,776	1,776	1,776	1,776	-	-	-	1,776	1,776	1,776
134	0	1,051	1,190	1,051	0	0	0	1,051	1,051	1,051	1,776	2,864	3,017	2,864	1,776	1,776	1,776	2,864	2,864	2,864
135	0	6,100	14,524	6,100	0	0	0	4,218	4,218	6,100	2,068	8,430	17,105	8,430	2,068	2,068	2,068	6,463	6,463	8,430
136	0	387	0	387	0	0	0	387	387	387	1,776	2,199	1,777	2,199	1,776	1,776	1,776	2,198	2,198	2,199
137	0	30,924	16,870	30,924	0	0	0	101,933	101,933	30,924	3,136	35,034	20,494	35,034	3,136	3,136	3,136	108,024	108,024	35,034
138	0	0	0	0	0	0	0	0	0	0	1,776	1,777	1,776	1,777	1,776	1,776	1,776	1,777	1,777	1,777

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
139	0	205	0	205	0	0	0	205	205	205	3,136	3,354	3,136	3,354	3,136	3,136	3,136	3,354	3,354	3,354
140	0	1,892	510	1,892	0	0	0	2,866	2,866	1,892	1,240	3,189	1,758	3,189	1,240	1,240	1,240	4,190	4,190	3,189
141	106,826	1,134,425	565,816	1,134,425	36	51,851	36	266,910	266,910	1,134,425	111,715	1,173,414	586,654	1,173,414	3,175	55,725	3,176	277,151	277,151	1,173,414
142	0	3,093	0	3,093	0	0	0	3,093	3,093	3,093	1,776	5,017	1,779	5,017	1,776	1,776	1,776	5,017	5,017	5,017
143	0	9,161	5,081	9,161	0	0	0	7,722	7,722	9,161	3,136	12,665	8,380	12,665	3,136	3,136	3,136	11,133	11,133	12,665
144	0	0	0	0	0	0	0	0	0	0	800	800	800	800	800	800	800	800	800	800
146	0	1,681	0	1,681	0	0	0	1,681	1,681	1,681	1,240	3,009	1,240	3,009	1,240	1,240	1,240	3,009	3,009	3,009
147	0	631	537	631	0	0	0	631	631	631	1,240	1,897	1,801	1,897	1,240	1,240	1,240	1,897	1,897	1,897
148	0	1,230	0	1,230	0	0	0	1,111	1,111	1,230	1,776	3,082	1,777	3,082	1,776	1,776	1,776	2,960	2,960	3,082
149	0	0	0	0	0	0	0	0	0	0	800	800	800	800	800	800	800	800	800	800
150	0	87	0	87	0	0	0	87	87	87	1,776	1,881	1,776	1,881	1,776	1,776	1,776	1,881	1,881	1,881
151	0	39	0	39	0	0	0	39	39	39	3,136	3,185	3,136	3,185	3,136	3,136	3,136	3,185	3,185	3,185
152	0	1,603	1,135	1,603	0	0	0	1,603	1,603	1,603	1,240	2,935	2,445	2,935	1,240	1,240	1,240	2,935	2,935	2,935
153	0	89,282	29,130	89,282	0	0	0	275,315	275,315	89,282	3,136	95,998	33,234	95,998	3,136	3,136	3,136	288,926	288,926	95,998
154	0	10,804	12,105	10,804	0	0	0	15,310	15,310	10,804	3,136	14,191	15,539	14,191	3,136	3,136	3,136	18,901	18,901	14,191
155	0	0	0	0	0	0	0	0	0	0	800	802	801	802	800	800	800	802	802	802
157	0	977	739	977	0	0	0	977	977	977	1,240	2,287	2,032	2,287	1,240	1,240	1,240	2,287	2,287	2,287

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
158	0	1,136	0	1,136	0	0	0	1,136	1,136	1,136	1,776	2,942	1,782	2,942	1,776	1,776	1,776	2,942	2,942	2,942
160	0	93	94	93	0	0	0	325	325	93	1,776	1,888	1,889	1,888	1,776	1,776	1,776	2,127	2,127	1,888
161	0	848	443	848	0	0	0	659	659	848	800	1,716	1,282	1,716	800	800	800	1,509	1,509	1,716
162	0	4,598	2,537	4,598	0	0	0	4,185	4,185	4,598	1,776	6,554	4,427	6,554	1,776	1,776	1,776	6,121	6,121	6,554
163	0	2,600	574	2,600	0	0	0	3,152	3,152	2,600	3,136	5,820	3,739	5,820	3,136	3,136	3,136	6,390	6,390	5,820
164	0	0	0	0	0	0	0	0	0	0	800	802	801	802	800	800	800	802	802	802
165	0	36	11	36	0	0	0	36	36	36	1,776	1,819	1,791	1,819	1,776	1,776	1,776	1,819	1,819	1,819
166	0	1,904	0	1,904	0	0	0	2,085	2,085	1,904	1,776	3,724	1,776	3,724	1,776	1,776	1,776	3,903	3,903	3,724
167	999	6,711	4,238	6,711	0	0	0	9,404	9,404	6,711	2,808	8,828	6,204	8,828	1,776	1,779	1,776	11,719	11,719	8,828
168	0	0	0	0	0	0	0	0	0	0	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240	1,240
169	0	29	27	29	0	0	0	29	29	29	1,776	1,810	1,810	1,810	1,776	1,776	1,776	1,810	1,810	1,810
170	0	5,447	1,317	5,447	0	0	0	4,330	4,330	5,447	3,136	8,837	4,514	8,837	3,136	3,136	3,136	7,715	7,715	8,837
171	0	1,194	0	1,194	0	0	0	1,204	1,204	1,194	800	2,071	800	2,071	800	800	800	2,083	2,083	2,071
172	0	0	0	0	-	-	-	0	0	0	1,240	1,241	1,241	1,241	-	-	-	1,241	1,241	1,241
173	0	410	0	410	0	0	0	405	405	410	1,776	2,214	1,776	2,214	1,776	1,776	1,776	2,210	2,210	2,214
174	0	492	0	492	0	0	0	492	492	492	3,136	3,638	3,136	3,638	3,136	3,136	3,136	3,638	3,638	3,638
175	0	68	71	68	0	0	0	68	68	68	1,776	1,850	1,850	1,850	1,776	1,776	1,776	1,850	1,850	1,850

Table B.5: All algorithms # BT and # NV on ternary-tables

Instance #	# BT										# NV									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**APOAC$_{around\cup}$**	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
176	0	43	0	43	0	0	0	43	43	43	800	846	800	846	800	800	800	846	846	846
177	0	0	0	0	0	0	0	0	0	0	3,136	3,137	3,137	3,137	3,136	3,136	3,136	3,137	3,137	3,137
178	0	0	0	0	0	0	0	0	0	0	2,360	2,363	2,360	2,363	2,360	2,360	2,360	2,363	2,363	2,363
179	0	140	0	140	0	0	0	140	140	140	1,776	1,927	1,777	1,927	1,776	1,776	1,776	1,927	1,927	1,927
180	0	0	0	0	0	0	0	0	0	0	3,136	3,139	3,137	3,139	3,136	3,136	3,136	3,139	3,139	3,139

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak**$^{+}$**-POAC1**	**A**$\cup_{cjc}^{bjsc}$**POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
001	1.49	1.85	1.88	1.91	1.42	0.98	0.97	1.28	0.85	1.30
002	4.42	5.18	5.16	5.22	4.26	2.87	2.85	3.56	2.21	3.59
003	0.90	0.96	1.17	1.05	0.85	0.65	0.64	0.67	0.48	0.68
004	1.76	1.84	2.07	2.00	1.69	1.25	1.23	1.30	0.87	1.31
005	1.58	1.85	2.35	1.85	1.49	1.05	1.04	1.30	0.88	1.31
006	1.70	1.90	2.42	1.87	1.63	1.19	1.18	1.34	0.93	1.36
007	5.19	5.92	6.56	6.26	5.00	3.56	3.54	4.20	2.78	4.23
008	1.68	2.31	2.05	2.41	1.63	1.16	1.14	1.63	1.30	1.75
009	4.17	5.31	5.18	4.58	3.82	2.47	2.45	3.72	2.39	3.66
010	0.80	0.93	1.26	0.95	0.75	0.54	0.53	0.64	0.44	0.65
011	1.57	1.85	2.61	1.86	1.49	1.04	1.02	1.27	0.85	1.30
012	4.35	5.34	5.87	5.34	4.12	2.73	2.71	3.72	2.37	3.75
013	1.56	1.91	2.07	1.76	1.42	0.99	0.99	1.36	0.94	1.36
014	1.73	2.41	2.82	2.55	1.67	1.22	1.18	1.74	1.42	1.85
015	5.20	6.17	7.27	6.14	4.84	3.55	3.52	4.57	3.28	4.58
016	4.71	5.73	6.46	5.48	4.39	3.04	3.01	4.13	2.80	4.14
017	0.79	0.92	1.05	1.02	0.73	0.52	0.52	0.64	0.43	0.64
018	1.45	1.76	1.86	1.77	1.35	0.91	0.90	1.21	0.78	1.22
019	4.45	5.47	6.24	5.33	4.18	2.81	2.78	3.83	2.48	3.85
020	4.15	5.28	6.10	5.13	3.89	2.51	2.49	3.65	2.29	3.68

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak**$^{+}$**-POAC1**	**A**$\cup_{cjc}^{bfsc}$**POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
021	1.41	1.73	2.07	1.67	1.32	0.87	0.86	1.18	0.74	1.19
022	4.01	5.09	5.31	5.14	3.80	2.40	2.38	3.46	2.09	3.50
023	1.60	1.83	2.62	1.88	1.55	1.09	1.07	1.26	0.84	1.29
024	4.55	5.56	5.65	5.97	4.33	2.93	2.91	3.95	2.62	3.97
025	5.02	5.41	6.74	6.13	4.97	3.47	3.35	3.72	2.39	3.81
026	4.95	6.00	6.31	5.89	4.60	3.28	3.26	4.40	3.09	4.41
027	1.62	1.97	2.06	1.78	1.47	1.05	1.05	1.42	1.01	1.42
028	4.99	11.10	7.42	12.10	4.85	3.44	3.38	8.31	8.40	9.36
029	1.70	1.94	2.04	1.96	1.62	1.19	1.18	1.38	0.97	1.39
030	0.75	0.92	1.01	0.83	0.67	0.47	0.46	0.64	0.44	0.65
031	1.48	1.75	2.01	1.91	1.40	0.96	0.95	1.21	0.77	1.21
032	0.86	0.93	1.15	1.06	0.81	0.60	0.59	0.64	0.44	0.65
033	4.22	5.38	5.25	4.63	3.86	2.49	2.48	3.77	2.41	3.79
034	1.76	2.01	2.58	2.05	1.67	1.25	1.24	1.45	1.04	1.47
035	1.57	1.82	2.31	2.03	1.51	1.06	1.03	1.24	0.82	1.28
036	4.30	5.14	6.51	5.35	4.40	2.73	2.70	3.52	2.16	3.56
037	0.80	0.93	1.23	1.04	0.75	0.54	0.53	0.65	0.45	0.65
038	4.45	4.86	5.93	4.60	4.32	3.05	3.02	3.37	2.13	3.40
039	4.38	5.25	5.53	5.09	4.20	2.81	2.78	3.60	2.25	3.65
040	1.93	2.26	3.00	2.55	2.03	1.30	1.28	1.56	1.05	1.60

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak**$^{+}$**-POAC1**	**A**$\cup_{cjc}^{bjsc}$**POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
041	0.36	0.42	0.50	0.46	0.33	0.24	0.24	0.29	0.20	0.29
042	0.37	0.44	0.52	0.43	0.33	0.25	0.25	0.31	0.23	0.31
043	0.41	0.47	0.74	0.50	0.39	0.30	0.30	0.33	0.25	0.34
044	1.52	1.81	1.94	1.86	1.44	1.00	0.99	1.25	0.82	1.27
045	0.41	0.45	0.61	0.49	0.38	0.29	0.29	0.32	0.23	0.32
046	0.76	0.87	0.99	0.89	0.71	0.53	0.52	0.61	0.43	0.61
047	0.83	0.99	1.12	0.96	0.75	0.55	0.55	0.71	0.51	0.71
048	0.83	0.99	1.10	0.92	0.74	0.55	0.54	0.71	0.51	0.71
049	0.40	0.45	0.73	0.48	0.37	0.28	0.28	0.32	0.24	0.33
050	0.38	0.43	0.51	0.48	0.35	0.26	0.26	0.30	0.22	0.30
051	1.60	1.81	2.16	1.76	1.53	1.08	1.07	1.25	0.82	1.26
052	5.20	5.68	7.17	5.70	5.70	3.67	3.63	4.05	2.72	4.09
053	1.50	1.82	1.92	1.87	1.41	0.97	0.96	1.27	0.84	1.28
054	1.57	1.83	2.53	1.86	1.49	1.05	1.04	1.27	0.85	1.29
055	4.46	5.21	6.66	5.60	4.29	2.90	2.87	3.60	2.24	3.62
056	0.41	0.46	0.72	0.52	0.38	0.29	0.29	0.32	0.24	0.33
057	1.60	1.95	2.56	2.04	1.53	1.09	1.07	1.33	0.94	1.40
058	4.36	5.12	5.98	4.88	4.21	2.81	2.78	3.49	2.13	3.53
059	1.48	1.82	2.07	1.75	1.37	0.93	0.92	1.28	0.85	1.28
060	1.65	1.91	2.04	2.03	1.55	1.13	1.12	1.36	0.94	1.36

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cjc}^{bjsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
061	6.94	6.72	7.75	6.70	7.17	5.30	5.07	4.44	3.19	5.01
062	0.39	0.44	0.56	0.47	0.36	0.27	0.27	0.30	0.22	0.31
064	0.83	1.01	1.46	1.12	0.78	0.57	0.57	0.69	0.50	0.73
065	2.07	2.36	3.37	2.36	2.00	1.45	1.41	1.60	1.13	1.70
066	0.90	0.94	1.20	1.02	0.87	0.65	0.64	0.65	0.45	0.66
067	5.23	5.20	6.09	5.61	5.06	3.69	3.45	3.56	2.20	3.61
068	0.38	0.43	0.52	0.52	0.35	0.27	0.26	0.30	0.22	0.30
069	-	-	-	-	-	-	-	3,877.14	-	5,606.73
070	1.52	1.88	1.95	1.66	1.38	0.95	0.94	1.33	0.91	1.34
071	4.56	5.69	5.58	4.97	4.17	2.82	2.81	4.08	2.75	4.10
072	1.65	1.99	2.15	1.85	1.50	1.09	1.08	1.44	1.03	1.44
073	0.77	0.91	1.24	0.93	0.71	0.50	0.49	0.62	0.42	0.63
074	0.78	0.88	1.21	0.90	0.73	0.54	0.53	0.61	0.43	0.62
075	4.38	5.39	6.10	5.24	4.12	2.74	2.70	3.75	2.41	3.79
076	0.86	0.99	1.23	1.02	0.81	0.60	0.59	0.70	0.51	0.71
077	4.17	5.22	6.53	5.45	3.93	2.55	2.53	3.60	2.24	3.64
078	1.47	1.79	1.93	1.84	1.39	0.95	0.94	1.23	0.80	1.25
079	4.13	5.09	6.16	5.11	3.94	2.54	2.52	3.47	2.10	3.51
080	0.81	0.90	1.29	1.01	0.76	0.55	0.54	0.62	0.42	0.62
081	1.51	1.75	2.47	1.94	1.44	0.99	0.98	1.20	0.77	1.21

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak**$^{+}$**-POAC1**	**A**$\cup_{cjc}^{bfsc}$**POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
082	2.76	3.11	4.05	3.21	2.65	1.85	1.84	2.13	1.35	2.15
083	0.38	0.43	0.52	0.46	0.36	0.27	0.27	0.30	0.21	0.30
085	1.52	1.84	2.42	1.82	1.44	1.00	0.99	1.26	0.84	1.30
086	3.40	3.99	4.43	3.79	3.28	2.22	2.20	2.73	1.70	2.76
087	1.47	1.78	2.35	1.90	1.38	0.94	0.93	1.22	0.80	1.24
088	4.27	5.34	6.57	5.46	4.02	2.64	2.62	3.70	2.36	3.75
089	0.85	0.95	1.28	1.09	0.80	0.59	0.59	0.66	0.46	0.67
090	4.23	5.38	5.25	4.64	3.87	2.49	2.48	3.77	2.42	3.79
091	4.81	5.39	6.00	5.08	4.70	3.27	3.24	3.77	2.43	3.80
092	4.39	5.44	6.29	5.36	4.63	2.76	2.74	3.82	2.48	3.85
093	0.36	0.42	0.53	0.45	0.33	0.25	0.24	0.29	0.21	0.29
094	1.74	1.91	2.70	2.13	1.71	1.23	1.22	1.36	0.94	1.37
095	4.65	5.57	7.28	5.78	4.67	3.08	3.01	3.88	2.57	3.98
096	1.56	1.83	1.91	1.88	-	-	-	1.28	0.85	1.29
097	0.38	0.43	0.60	0.44	0.36	0.26	0.26	0.29	0.21	0.30
098	0.82	0.95	1.10	0.96	0.77	0.57	0.56	0.66	0.46	0.67
099	1.71	1.89	2.42	1.92	1.63	1.20	1.18	1.34	0.92	1.35
100	4.81	5.56	6.62	5.42	4.63	3.27	3.24	3.89	2.56	3.96
101	0.97	1.17	1.28	1.11	-	-	-	0.83	0.60	0.84
102	1.75	2.15	2.76	2.20	1.70	1.26	1.22	1.53	1.19	1.60

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cjc}^{bjsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
103	4.36	5.28	6.26	5.71	4.16	2.78	2.76	3.66	2.31	3.69
104	0.78	0.91	1.03	0.93	0.72	0.51	0.51	0.63	0.43	0.63
105	4.33	5.29	6.31	5.49	4.22	2.76	2.73	3.65	2.29	3.72
106	0.38	0.44	0.53	0.47	-	-	-	0.31	0.23	0.31
107	4.77	5.48	7.01	5.27	4.60	3.18	3.06	3.69	2.40	3.87
108	5.26	5.12	6.24	5.42	5.25	3.71	3.61	3.53	2.16	3.53
109	0.41	0.45	0.54	0.53	0.38	0.30	0.29	0.32	0.24	0.32
110	0.77	0.94	1.04	0.84	0.69	0.49	0.48	0.66	0.46	0.66
111	1.52	1.80	2.34	1.90	1.43	1.00	0.98	1.25	0.82	1.26
112	4.39	5.47	5.26	5.69	4.21	2.84	2.80	3.78	2.46	3.88
113	0.80	0.93	1.22	1.00	0.75	0.54	0.54	0.65	0.45	0.65
114	1.45	1.81	1.88	1.59	1.31	0.88	0.87	1.27	0.84	1.27
115	0.84	0.97	1.30	1.07	0.78	0.58	0.57	0.68	0.48	0.69
116	5.13	6.14	7.41	6.82	4.95	3.58	3.53	4.25	3.04	4.50
117	0.87	1.13	1.52	1.18	0.82	0.62	0.61	0.78	0.63	0.85
118	1.65	2.59	3.34	2.82	1.58	1.15	1.13	1.73	1.45	2.02
119	4.83	5.28	7.12	5.00	4.79	3.31	3.26	3.64	2.28	3.68
120	1.66	2.00	2.09	1.81	-	-	-	1.46	1.04	1.46
121	4.80	5.89	5.84	5.28	4.39	3.07	3.06	4.29	2.98	4.29
122	4.73	5.54	7.55	5.93	4.51	3.15	3.12	3.92	2.59	3.94

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cjc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
123	0.86	1.03	1.46	1.11	0.86	0.60	0.59	0.71	0.53	0.75
124	3.88	4.19	5.39	4.91	2.99	2.60	2.06	2.70	2.71	3.54
125	5.64	6.57	7.12	6.98	5.73	4.05	3.73	3.86	2.59	4.92
126	0.41	0.44	0.62	0.47	0.38	0.30	0.29	0.31	0.23	0.31
128	4.19	5.35	5.22	4.66	-	-	-	3.74	2.38	3.76
129	1.58	1.90	2.34	1.92	1.50	1.06	1.05	1.31	0.91	1.36
130	4.35	5.13	6.15	5.53	4.19	2.78	2.76	3.51	2.15	3.54
131	1.46	1.76	1.90	1.92	1.38	0.94	0.93	1.21	0.78	1.22
132	0.79	0.92	1.20	0.98	0.74	0.53	0.53	0.63	0.43	0.64
133	1.47	1.75	1.89	1.83	-	-	-	1.21	0.77	1.21
134	1.82	1.80	2.43	1.90	1.78	1.30	1.24	1.23	0.80	1.25
135	2.80	2.83	5.07	3.21	2.73	2.15	2.04	1.82	1.33	2.09
136	1.50	1.80	2.60	2.02	1.42	0.98	0.97	1.24	0.81	1.26
137	4.75	7.79	7.89	8.24	4.80	3.19	3.11	9.68	9.83	6.10
138	1.63	1.81	2.24	2.06	1.55	1.11	1.09	1.27	0.84	1.27
139	4.63	5.26	5.30	5.36	4.48	3.09	3.06	3.65	2.29	3.67
140	0.95	1.00	1.51	1.09	0.91	0.70	0.67	0.73	0.54	0.71
141	12.76	95.72	68.20	118.10	93.59	73.82	70.50	15.67	17.05	90.51
142	1.51	1.95	2.30	2.13	1.43	0.99	0.98	1.34	0.93	1.39
143	4.51	5.84	7.00	6.03	6.95	2.92	2.83	3.88	2.59	4.22

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cjc}^{bjsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
144	0.40	0.45	0.55	0.49	0.37	0.28	0.28	0.32	0.24	0.32
146	0.85	1.05	1.14	1.10	0.81	0.60	0.59	0.73	0.54	0.76
147	0.84	0.97	1.27	1.00	0.78	0.58	0.57	0.68	0.48	0.69
148	1.51	1.85	2.61	1.89	1.43	0.99	0.98	1.28	0.86	1.31
149	0.36	0.43	0.50	0.39	0.32	0.23	0.23	0.30	0.22	0.30
150	1.57	1.83	1.93	1.83	1.48	1.05	1.03	1.28	0.85	1.29
151	4.27	5.17	5.13	5.52	4.10	2.69	2.68	3.56	2.20	3.59
152	0.86	0.99	1.36	1.08	0.84	0.60	0.58	0.68	0.49	0.71
153	5.59	11.85	9.28	13.66	5.84	3.99	3.72	16.44	18.04	9.98
154	4.79	5.69	10.07	6.22	4.62	3.21	3.07	4.18	2.96	4.08
155	0.37	0.42	0.64	0.48	0.35	0.26	0.25	0.29	0.20	0.29
157	0.88	1.07	1.34	1.09	0.82	0.62	0.61	0.75	0.57	0.78
158	1.65	1.87	2.34	2.05	1.56	1.13	1.10	1.31	0.89	1.33
160	1.57	1.81	2.34	1.76	1.50	1.06	1.05	1.26	0.84	1.27
161	0.43	0.49	0.72	0.53	0.40	0.32	0.31	0.33	0.25	0.36
162	1.71	2.13	2.49	2.17	1.65	1.19	1.16	1.42	1.04	1.57
163	4.35	5.40	5.99	5.29	4.18	2.79	2.75	3.71	2.42	3.80
164	0.39	0.43	0.53	0.47	0.36	0.27	0.27	0.30	0.22	0.31
165	1.72	1.86	2.67	2.01	1.65	1.21	1.19	1.31	0.89	1.32
166	1.62	1.94	2.27	2.03	1.55	1.11	1.10	1.35	0.95	1.39

Table B.6: All algorithms CPU time on ternary-tables

Instance #	CPU Time									
	APOAC$_{around\cup}$	**PrePeak**$^{+}$**-POAC1**	**A**$\cup_{cjc}^{bfsc}$**POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**
167	2.81	2.48	3.03	2.74	4.97	5.91	3.81	1.89	1.65	1.91
168	0.79	0.96	1.05	0.86	0.71	0.50	0.49	0.67	0.47	0.68
169	1.51	1.79	2.21	1.97	1.42	0.99	0.98	1.24	0.82	1.25
170	4.72	5.72	7.25	6.13	4.55	3.17	3.12	3.94	2.64	4.11
171	0.39	0.49	0.54	0.54	0.37	0.28	0.27	0.34	0.27	0.36
172	0.81	0.95	1.23	1.01	-	-	-	0.67	0.47	0.67
173	1.65	1.96	2.07	1.94	1.56	1.12	1.11	1.36	0.96	1.40
174	4.33	5.49	5.44	5.18	4.07	2.67	2.68	3.83	2.51	3.89
175	1.45	1.79	2.01	1.81	1.35	0.92	0.90	1.24	0.81	1.25
176	0.40	0.45	0.60	0.49	0.37	0.29	0.28	0.32	0.24	0.32
177	4.86	5.91	6.14	5.50	4.51	3.16	3.13	4.31	3.00	4.35
178	2.62	3.06	3.56	3.32	2.48	1.75	1.71	2.12	1.38	2.13
179	1.46	1.78	2.35	1.83	1.38	0.93	0.93	1.22	0.79	1.23
180	4.88	5.90	6.41	5.72	4.57	3.21	3.18	4.29	2.99	4.30

B.2.2 Global-Tables Model

Similarly, Tables B.7, B.8, and B.9 report the performance of all algorithms with # BT, # NV, and CPU Time, respectively, on global-tables model.

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
001	0	19	19	19	0	0	0	19	19	19	-	-	-	-
002	-	-	-	-	-	-	-	-	-	-	-	-	-	-
003	0	31	0	31	0	0	0	31	31	31	31	-	31	-
004	0	15	0	15	0	0	0	15	15	15	-	-	-	-
005	0	50	50	50	0	0	0	50	50	50	-	-	-	-
006	0	786	408	786	0	0	0	808	808	786	-	-	-	-
007	0	0	0	0	-	-	0	0	0	0	-	-	-	-
008	0	4,291	387	4,291	0	0	0	3,995	3,995	4,291	-	-	-	-
009	0	0	0	0	0	0	0	0	0	0	0	0	-	0
010	0	0	0	0	0	0	0	0	0	0	0	0	0	0
011	0	135	135	135	0	0	0	129	129	135	254	-	254	-
012	0	105	15	105	0	0	0	116	116	105	-	-	-	-
013	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014	0	31,265	10,234	31,265	0	0	0	22,692	22,692	31,265	-	-	-	-
015	0	184	0	184	0	0	0	75	75	184	197	-	197	-
016	-	-	-	-	-	-	-	-	-	-	-	-	-	-
017	0	51	0	51	0	0	0	51	51	51	-	-	-	-
018	0	0	0	0	-	-	-	0	0	0	-	-	-	-
019	-	-	-	-	-	-	-	-	-	-	-	-	-	-
020	-	-	-	-	-	-	-	-	-	-	-	-	-	-
021	0	0	0	0	0	0	0	0	0	0	0	0	0	0
022	0	325	0	325	0	0	0	340	340	325	-	-	-	-
023	0	869	869	869	0	0	0	1,540	1,540	869	-	-	-	-

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
024	-	-	-	-	-	-	-	-	-	-	-	-	-	-
025	0	10,715	12,893	10,715	0	0	0	12,813	12,813	10,715	-	-	-	-
026	-	-	-	-	-	-	-	-	-	-	-	-	-	-
027	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028	0	5,574	0	5,574	0	0	0	8,586	8,586	5,574	-	-	-	-
029	0	0	0	0	0	0	0	0	0	0	-	-	-	-
030	0	0	0	0	0	0	0	0	0	0	0	0	0	0
031	0	0	0	0	0	0	0	0	0	0	0	0	0	0
032	0	90	0	90	0	0	0	92	92	90	342	0	342	0
033	-	-	-	-	-	-	-	-	-	-	-	-	-	-
034	0	23	0	23	0	0	0	23	23	23	-	-	-	-
035	0	978	978	978	0	0	0	778	778	978	1,028	-	1,028	-
036	0	693	0	693	0	0	0	789	789	693	-	-	-	-
037	0	0	0	0	0	0	0	0	0	0	0	0	0	0
038	-	-	-	-	-	-	-	-	-	-	-	-	-	-
039	-	-	-	-	-	-	-	-	-	-	-	-	-	-
040	0	2,000	-	2,000	0	0	0	1,692	1,692	2,000	-	-	-	-
041	0	22	9	22	0	0	0	22	22	22	22	0	22	0
042	0	5	0	5	0	0	0	5	5	5	5	0	5	0
043	0	0	0	0	0	0	0	0	0	0	0	0	0	0
044	0	411	1	411	0	0	0	413	413	411	-	-	-	-
045	0	0	0	0	0	0	0	0	0	0	0	0	0	0
047	0	0	0	0	0	0	0	0	0	0	0	0	0	0
048	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
049	0	162	0	162	0	0	0	109	109	162	720	0	720	0
050	0	15	15	15	0	0	0	15	15	15	6	0	6	0
051	0	4,775	3,179	4,775	0	0	0	5,007	5,007	4,775	-	-	-	-
052	-	-	-	-	-	-	-	-	-	-	-	-	-	-
053	0	227	227	227	0	0	0	305	305	227	171	-	-	-
054	0	74	0	74	0	0	0	74	74	74	150	-	150	-
055	0	628	0	628	0	0	0	685	685	628	-	-	-	-
056	0	0	0	0	0	0	0	0	0	0	0	0	0	0
057	0	12,798	10,111	12,798	0	0	0	8,411	8,411	12,798	-	-	-	-
058	0	717	757	717	0	0	0	470	470	717	-	-	-	-
059	0	0	0	0	0	0	0	0	0	0	0	0	0	0
060	0	0	0	0	0	0	0	0	0	0	-	-	-	-
062	0	3	0	3	0	0	0	3	3	3	3	0	3	0
064	0	107	0	107	0	0	0	54	54	107	85	-	85	-
066	0	943	524	943	0	0	0	1,596	1,596	943	-	-	-	-
067	-	-	-	-	-	-	-	-	-	-	-	-	-	-
068	0	0	0	0	0	0	0	0	0	0	0	0	0	0
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	0	0	0	0	0	0	0	0	0	0	-	-	-	-
071	-	-	-	-	-	-	-	-	-	-	-	-	-	-
072	0	0	0	0	0	0	0	0	0	0	0	0	0	0
073	0	134	134	134	0	0	0	443	443	134	57	0	57	0
074	0	2	2	2	0	0	0	2	2	2	98	-	98	-
075	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
076	0	61	49	61	0	0	0	62	62	61	252	0	252	0
077	-	-	-	-	-	-	-	-	-	-	-	-	-	-
078	0	84	79	84	0	0	0	84	84	84	-	-	-	-
079	-	-	-	-	-	-	-	-	-	-	-	-	-	-
080	0	59	59	59	0	0	0	59	59	59	117	0	117	0
081	0	59	59	59	0	0	0	212	212	59	66	-	66	-
082	0	9	9	9	0	0	0	9	9	9	-	-	-	-
083	0	0	0	0	0	0	0	0	0	0	0	0	0	0
085	0	1,914	1,914	1,914	0	0	0	2,729	2,729	1,914	-	-	-	-
087	0	112	112	112	0	0	0	105	105	112	115	0	115	0
088	-	-	-	-	-	-	-	-	-	-	-	-	-	-
089	0	356	241	356	0	0	0	335	335	356	-	0	-	0
090	-	-	-	-	-	-	-	-	-	-	-	-	-	-
091	-	-	-	-	-	-	-	-	-	-	-	-	-	-
092	-	-	-	-	-	-	-	-	-	-	-	-	-	-
093	0	10	10	10	0	0	0	10	10	10	14	0	14	0
094	0	81	45	81	0	0	0	91	91	81	-	-	-	-
095	-	-	-	-	-	-	-	-	-	-	-	-	-	-
096	0	0	0	0	-	-	-	0	0	0	-	-	-	-
097	0	16	0	16	0	0	0	16	16	16	163	0	163	0
098	0	133	95	133	0	0	0	158	158	133	103	0	103	0
099	0	8	0	8	0	0	0	8	8	8	-	-	-	-
100	0	17	0	17	0	0	0	17	17	17	-	-	-	-
101	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
102	0	0	0	0	0	0	0	0	0	0	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	0	0	0	0	0	0	0	0	0	0	0	0	0	0
105	0	502	0	502	0	0	0	464	464	502	763	-	763	-
106	0	0	0	0	0	0	0	0	0	0	0	0	0	0
107	2,638	1,048,890	-	-	0	2,933	0	-	-	-	-	-	-	-
108	0	17,868	17,868	17,868	0	0	0	11,756	11,756	17,868	-	-	-	-
109	0	92	0	92	0	0	0	92	92	92	441	0	441	0
110	0	0	0	0	0	0	0	0	0	0	0	0	0	0
111	0	1,334	640	1,334	0	0	0	611	611	1,334	7,825	-	7,825	-
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	0	0	0	0	0	0	0	0	0	0	3	0	3	0
114	0	0	0	0	0	0	0	0	0	0	0	0	0	0
115	0	65	10	65	0	0	0	65	65	65	65	0	65	0
116	0	11,533	0	11,533	0	0	0	5,183	5,183	11,533	-	-	-	-
117	0	381	877	381	0	0	0	365	365	381	-	-	-	-
118	0	5,867	524	5,867	0	0	0	2,522	2,522	5,867	-	-	-	-
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	0	0	0	0	-	-	-	0	0	0	0	0	0	0
121	0	0	0	0	-	-	-	0	0	0	0	0	0	0
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	51	1,039	711	1,039	0	0	0	1,418	1,418	1,039	5,782	-	5,782	-
124	1,067	7,301	7,301	7,301	1	-	1	1,632	1,632	7,301	-	-	-	-
125	25,515	50,560	20,621	50,560	0	18,101	0	44,875	44,875	50,560	-	-	-	-

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
126	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	0	466	466	466	0	0	0	752	752	466	-	-	-	-
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-
131	0	12	12	12	0	0	0	12	12	12	123	0	123	0
132	0	0	0	0	0	0	0	0	0	0	0	0	0	0
133	0	0	0	0	0	0	0	0	0	0	0	0	0	0
134	5	3,873	3,873	3,873	0	224	0	3,027	3,027	3,873	-	-	-	-
135	0	525	485	525	0	423	0	597	597	525	-	-	-	-
136	0	265	265	265	0	0	0	320	320	265	-	-	-	-
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	0	0	0	0	0	0	0	0	0	0	2	-	2	-
139	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	0	299	299	299	0	0	0	308	308	299	-	-	-	-
141	21,767	34,472	96,210	34,472	2,303	23,448	-	73,024	73,024	34,472	-	-	-	-
142	0	93	93	93	0	0	0	53	53	93	51	-	51	-
143	-	-	-	-	-	-	-	-	-	-	-	-	-	-
144	0	3	0	3	0	0	0	3	3	3	3	0	3	0
146	0	534	507	534	0	0	0	534	534	534	-	-	-	-
147	0	500	500	500	0	0	0	642	642	500	770	-	770	-
148	0	664	664	664	0	0	0	674	674	664	-	-	-	-
149	0	0	0	0	0	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0	0	0	0	0	0
151	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
152	0	1,878	1,106	1,878	0	0	0	1,173	1,173	1,878	21,499	105	21,499	105
153	-	-	-	-	-	-	-	-	-	-	-	-	-	-
154	-	-	-	-	-	-	-	-	-	-	-	-	-	-
155	0	0	0	0	0	0	0	0	0	0	0	0	0	0
157	0	9	45	9	0	0	0	63	63	9	29	-	29	-
158	0	305	378	305	0	0	0	375	375	305	-	-	-	-
160	0	0	0	0	0	0	0	0	0	0	0	-	0	-
161	0	39	39	39	0	0	0	39	39	39	90	0	90	0
162	0	132	132	132	0	0	0	248	248	132	165	-	165	-
163	0	1,549	1,549	1,549	0	0	0	937	937	1,549	-	-	-	-
164	0	0	0	0	0	0	0	0	0	0	0	0	0	0
165	0	215	215	215	0	0	0	557	557	215	-	-	-	-
166	0	1,128	445	1,128	0	0	0	1,880	1,880	1,128	-	-	-	-
167	7,133	17,021	11,420	17,021	10	5,206	10	17,773	17,773	17,021	-	-	-	-
168	0	0	0	0	0	0	0	0	0	0	0	0	0	0
169	0	245	245	245	0	0	0	242	242	245	-	-	-	-
170	-	-	-	-	-	-	-	-	-	-	-	-	-	-
171	0	49	53	49	0	0	0	49	49	49	772	0	772	0
172	0	0	0	0	0	0	0	0	0	0	0	0	0	0
173	0	348	144	348	0	0	0	310	310	348	-	-	-	-
174	-	-	-	-	-	-	-	-	-	-	-	-	-	-
175	0	321	202	321	0	0	0	321	321	321	252	-	252	-
176	0	0	0	0	0	0	0	0	0	0	0	0	0	0
177	0	61	0	61	0	0	0	61	61	61	61	0	61	0

Table B.7: All algorithms # BT on global-tables

Instance #	# BT													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
178	0	8	0	8	0	0	0	8	8	8	8	-	8	-
179	0	0	0	0	0	0	0	0	0	0	0	0	0	0
180	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
001	576	608	608	608	576	576	576	608	608	608	-	-	-	-
002	-	-	-	-	-	-	-	-	-	-	-	-	-	-
003	400	443	400	443	400	400	400	443	443	443	443	-	443	-
004	576	604	576	604	576	576	576	604	604	604	-	-	-	-
005	576	638	638	638	576	576	576	638	638	638	-	-	-	-
006	576	1,519	1,040	1,519	576	576	576	1,572	1,572	1,519	-	-	-	-
007	1,056	1,056	1,056	1,056	-	-	1,056	1,056	1,056	1,056	-	-	-	-
008	576	5,213	1,057	5,213	576	576	576	4,974	4,974	5,213	-	-	-	-
009	1,015	1,015	1,015	1,015	1,015	1,015	1,015	1,015	1,015	1,015	1,015	1,015	-	1,015
010	400	403	403	403	400	400	400	403	403	403	401	400	401	400
011	576	737	737	737	576	576	576	736	736	737	871	-	871	-
012	1,024	1,158	1,044	1,158	1,024	1,024	1,024	1,166	1,166	1,158	-	-	-	-
013	576	577	577	577	576	576	576	577	577	577	577	576	577	576

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around^{\cup}}$	**PrePeak^{+}-POAC1**	**A$^{\cup}$$_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
014	577	34,995	12,224	34,995	576	577	576	25,203	25,203	34,995	-	-	-	-
015	1,024	1,237	1,024	1,237	1,024	1,024	1,024	1,112	1,112	1,237	1,256	-	1,256	-
016	-	-	-	-	-	-	-	-	-	-	-	-	-	-
017	400	472	401	472	400	400	400	472	472	472	-	-	-	-
018	576	576	576	576	-	-	-	576	576	576	-	-	-	-
019	-	-	-	-	-	-	-	-	-	-	-	-	-	-
020	-	-	-	-	-	-	-	-	-	-	-	-	-	-
021	576	576	576	576	576	576	576	576	576	576	576	576	576	576
022	1,024	1,372	1,024	1,372	1,024	1,024	1,024	1,392	1,392	1,372	-	-	-	-
023	576	1,535	1,535	1,535	576	576	576	2,267	2,267	1,535	-	-	-	-
024	-	-	-	-	-	-	-	-	-	-	-	-	-	-
025	1,024	12,156	14,484	12,156	1,024	1,024	1,024	14,368	14,368	12,156	-	-	-	-
026	-	-	-	-	-	-	-	-	-	-	-	-	-	-
027	576	576	576	576	576	576	576	576	576	576	576	576	576	576
028	1,026	6,986	1,024	6,986	1,024	1,024	1,024	10,171	10,171	6,986	-	-	-	-
029	576	576	576	576	576	576	576	576	576	576	-	-	-	-
030	400	400	400	400	400	400	400	400	400	400	400	400	400	400
031	576	579	576	579	576	576	576	579	579	579	578	576	578	576
032	400	507	401	507	400	400	400	510	510	507	901	400	901	400
033	-	-	-	-	-	-	-	-	-	-	-	-	-	-
034	576	610	577	610	576	576	576	611	611	610	-	-	-	-
035	576	1,661	1,661	1,661	576	576	576	1,446	1,446	1,661	1,729	-	1,729	-
036	1,024	1,793	1,024	1,793	1,024	1,024	1,024	1,888	1,888	1,793	-	-	-	-
037	400	402	400	402	400	400	400	402	402	402	406	400	406	400

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
038	-	-	-	-	-	-	-	-	-	-	-	-	-	-
039	-	-	-	-	-	-	-	-	-	-	-	-	-	-
040	640	2,793	-	2,793	640	640	640	2,456	2,456	2,793	-	-	-	-
041	256	286	268	286	256	256	256	286	286	286	286	256	286	256
042	256	264	256	264	256	256	256	264	264	264	264	256	264	256
043	256	259	257	259	256	256	256	259	259	259	258	256	258	256
044	576	1,153	580	1,153	576	576	576	1,157	1,157	1,153	-	-	-	-
045	256	258	256	258	256	256	256	258	258	258	257	256	257	256
047	400	401	401	401	400	400	400	401	401	401	401	400	401	400
048	400	402	402	402	400	400	400	402	402	402	402	400	402	400
049	256	452	258	452	256	256	256	403	403	452	1,135	256	1,135	256
050	256	282	282	282	256	256	256	282	282	282	268	256	268	256
051	576	5,845	4,000	5,845	576	576	576	6,077	6,077	5,845	-	-	-	-
052	-	-	-	-	-	-	-	-	-	-	-	-	-	-
053	576	840	840	840	576	576	576	930	930	840	771	-	-	-
054	576	663	576	663	576	576	576	663	663	663	738	-	738	-
055	1,024	1,713	1,024	1,713	1,024	1,024	1,024	1,753	1,753	1,713	-	-	-	-
056	256	258	256	258	256	256	256	258	258	258	257	256	257	256
057	576	14,538	11,512	14,538	576	576	576	9,762	9,762	14,538	-	-	-	-
058	1,024	1,898	1,977	1,898	1,024	1,024	1,024	1,569	1,569	1,898	-	-	-	-
059	576	578	577	578	576	576	576	578	578	578	578	576	578	576
060	576	577	576	577	576	576	576	577	577	577	-	-	-	-
062	256	262	256	262	256	256	256	262	262	262	262	256	262	256
064	400	520	400	520	400	400	400	458	458	520	494	-	494	-

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
066	401	1,554	1,056	1,554	400	401	400	2,288	2,288	1,554	-	-	-	-
067	-	-	-	-	-	-	-	-	-	-	-	-	-	-
068	256	256	256	256	256	256	256	256	256	256	256	256	256	256
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	576	576	576	576	576	576	576	576	576	576	-	-	-	-
071	-	-	-	-	-	-	-	-	-	-	-	-	-	-
072	576	576	576	576	576	576	576	576	576	576	576	576	576	576
073	400	553	553	553	400	400	400	896	896	553	478	400	478	400
074	384	392	392	392	384	384	384	392	392	392	512	-	512	-
075	-	-	-	-	-	-	-	-	-	-	-	-	-	-
076	400	472	460	472	400	400	400	474	474	472	680	400	680	400
077	-	-	-	-	-	-	-	-	-	-	-	-	-	-
078	576	674	663	674	576	576	576	674	674	674	-	-	-	-
079	-	-	-	-	-	-	-	-	-	-	-	-	-	-
080	400	468	468	468	400	400	400	468	468	468	550	400	550	400
081	576	651	651	651	576	576	576	807	807	651	660	-	660	-
082	784	802	802	802	784	784	784	802	802	802	-	-	-	-
083	256	256	256	256	256	256	256	256	256	256	256	256	256	256
085	576	2,813	2,813	2,813	576	576	576	3,679	3,679	2,813	-	-	-	-
087	576	703	703	703	576	576	576	693	693	703	708	576	708	576
088	-	-	-	-	-	-	-	-	-	-	-	-	-	-
089	400	813	698	813	400	400	400	789	789	813	-	400	-	400
090	-	-	-	-	-	-	-	-	-	-	-	-	-	-
091	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
092	-	-	-	-	-	-	-	-	-	-	-	-	-	-
093	256	270	270	270	256	256	256	270	270	270	276	256	276	256
094	576	681	630	681	576	576	576	690	690	681	-	-	-	-
095	-	-	-	-	-	-	-	-	-	-	-	-	-	-
096	576	576	576	576	-	-	-	576	576	576	-	-	-	-
097	256	282	256	282	256	256	256	282	282	282	443	256	443	256
098	400	565	505	565	400	400	400	598	598	565	522	400	522	400
099	576	595	576	595	576	576	576	595	595	595	-	-	-	-
100	1,024	1,055	1,024	1,055	1,024	1,024	1,024	1,055	1,055	1,055	-	-	-	-
101	435	435	435	435	435	435	435	435	435	435	435	435	435	435
102	576	577	576	577	576	576	576	577	577	577	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	400	401	401	401	400	400	400	401	401	401	401	400	401	400
105	1,024	1,572	1,024	1,572	1,024	1,024	1,024	1,529	1,529	1,572	1,917	-	1,917	-
106	256	257	256	257	256	256	256	257	257	257	257	256	257	256
107	3,906	1,153,563	-	-	1,024	4,264	1,024	-	-	-	-	-	-	-
108	1,024	20,235	20,235	20,235	1,024	1,024	1,024	13,604	13,604	20,235	-	-	-	-
109	256	356	256	356	256	256	256	356	356	356	766	256	766	256
110	400	400	400	400	400	400	400	400	400	400	400	400	400	400
111	576	2,094	1,286	2,094	576	576	576	1,266	1,266	2,094	9,882	-	9,882	-
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	400	402	400	402	400	400	400	402	402	402	406	400	406	400
114	576	576	576	576	576	576	576	576	576	576	576	576	576	576
115	400	477	412	477	400	400	400	477	477	477	477	400	477	400

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
116	1,024	13,160	1,024	13,160	1,024	1,024	1,024	6,614	6,614	13,160	-	-	-	-
117	400	825	1,422	825	400	400	400	812	812	825	-	-	-	-
118	576	7,466	1,229	7,466	576	576	576	3,336	3,336	7,466	-	-	-	-
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	576	577	577	577	-	-	-	577	577	577	577	576	577	576
121	1,024	1,024	1,024	1,024	-	-	-	1,024	1,024	1,024	1,024	1,024	1,024	1,024
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	456	1,590	1,198	1,590	400	401	400	2,027	2,027	1,590	7,217	-	7,217	-
124	1,764	8,877	8,877	8,877	579	-	579	2,457	2,457	8,877	-	-	-	-
125	28,407	56,901	22,892	56,901	1,024	20,201	1,024	50,455	50,455	56,901	-	-	-	-
126	256	256	256	256	256	256	256	256	256	256	256	256	256	256
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	576	1,093	1,093	1,093	576	576	576	1,447	1,447	1,093	-	-	-	-
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-
131	576	594	594	594	576	576	576	594	594	594	730	576	730	576
132	400	402	402	402	400	400	400	402	402	402	405	400	405	400
133	576	576	576	576	576	576	576	576	576	576	576	576	576	576
134	583	4,718	4,718	4,718	576	824	576	3,792	3,792	4,718	-	-	-	-
135	673	1,287	1,230	1,287	672	1,159	672	1,363	1,363	1,287	-	-	-	-
136	576	903	903	903	576	576	576	946	946	903	-	-	-	-
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	576	578	577	578	576	576	576	578	578	578	582	-	582	-
139	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	400	773	773	773	400	400	400	779	779	773	-	-	-	-

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
141	23,774	38,299	104,493	38,299	3,459	25,817	-	79,379	79,379	38,299	-	-	-	-
142	576	682	680	682	576	576	576	637	637	682	638	-	638	-
143	-	-	-	-	-	-	-	-	-	-	-	-	-	-
144	256	262	256	262	256	256	256	262	262	262	262	256	262	256
146	400	978	946	978	400	400	400	1,000	1,000	978	-	-	-	-
147	400	950	950	950	400	400	400	1,102	1,102	950	1,237	-	1,237	-
148	576	1,452	1,452	1,452	576	576	576	1,331	1,331	1,452	-	-	-	-
149	256	256	256	256	256	256	256	256	256	256	256	256	256	256
150	576	576	576	576	576	576	576	576	576	576	576	576	576	576
151	-	-	-	-	-	-	-	-	-	-	-	-	-	-
152	400	2,569	1,708	2,569	400	400	400	1,784	1,784	2,569	25,974	529	25,974	529
153	-	-	-	-	-	-	-	-	-	-	-	-	-	-
154	-	-	-	-	-	-	-	-	-	-	-	-	-	-
155	256	257	256	257	256	256	256	257	257	257	257	256	257	256
157	400	412	448	412	400	400	400	470	470	412	439	-	439	-
158	576	924	1,011	924	576	576	576	996	996	924	-	-	-	-
160	576	577	577	577	576	576	576	577	577	577	578	-	578	-
161	256	300	300	300	256	256	256	300	300	300	360	256	360	256
162	576	728	728	728	576	576	576	859	859	728	771	-	771	-
163	1,024	2,788	2,788	2,788	1,024	1,024	1,024	2,111	2,111	2,788	-	-	-	-
164	256	257	256	257	256	256	256	257	257	257	257	256	257	256
165	576	817	817	817	576	576	576	1,175	1,175	817	-	-	-	-
166	576	1,848	1,060	1,848	576	576	576	2,643	2,643	1,848	-	-	-	-
167	8,460	19,769	13,520	19,769	589	6,167	589	20,594	20,594	19,769	-	-	-	-

Table B.8: All algorithms # NV on global-tables

Instance #	# NV													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
168	400	400	400	400	400	400	400	400	400	400	400	400	400	400
169	576	852	852	852	576	576	576	844	844	852	-	-	-	-
170	-	-	-	-	-	-	-	-	-	-	-	-	-	-
171	256	311	312	311	256	256	256	311	311	311	1,181	256	1,181	256
172	400	403	400	403	400	400	400	402	402	403	403	400	403	400
173	576	989	752	989	576	576	576	936	936	989	-	-	-	-
174	-	-	-	-	-	-	-	-	-	-	-	-	-	-
175	576	928	795	928	576	576	576	928	928	928	851	-	851	-
176	256	258	256	258	256	256	256	258	258	258	258	256	258	256
177	1,024	1,091	1,024	1,091	1,024	1,024	1,024	1,091	1,091	1,091	1,090	1,024	1,090	1,024
178	768	778	768	778	768	768	768	778	778	778	778	-	778	-
179	576	576	576	576	576	576	576	576	576	576	576	576	576	576
180	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
001	854.13	853.51	854.49	854.23	853.72	854.14	854.13	853.45	853.73	858.30	-	-	-	-
002	-	-	-	-	-	-	-	-	-	-	-	-	-	-
003	25.32	25.26	25.51	25.38	25.26	25.25	25.26	25.15	25.20	25.17	1,459.28	-	1,459.27	-
004	1,402.53	1,399.65	1,402.79	1,400.39	1,399.79	1,400.04	1,401.97	1,399.51	1,402.27	1,400.49	-	-	-	-
005	10,855.25	10,853.94	10,861.28	10,862.34	10,854.21	10,877.85	10,854.98	10,853.92	10,853.83	10,858.15	-	-	-	-
006	220.61	220.44	221.28	221.00	220.28	220.68	220.67	220.12	221.32	220.32	-	-	-	-
007	22,194.73	22,194.04	22,197.38	22,197.78	-	-	22,195.87	22,192.48	22,193.27	22,191.74	-	-	-	-
008	1,812.61	1,811.73	1,811.85	1,813.07	1,811.06	1,813.40	1,813.32	1,811.08	1,816.38	1,811.40	-	-	-	-
009	481.26	481.23	482.24	481.52	480.76	481.13	481.22	480.92	481.18	480.84	13,711.85	9,371.06	-	9,371.13
010	5.40	5.36	5.61	5.48	5.33	5.39	5.39	5.31	5.33	5.30	113.89	1,104.21	113.90	1,104.21
011	81.19	81.06	81.67	81.47	80.99	81.18	81.18	80.96	81.20	80.96	4,578.04	-	4,580.31	-
012	14,654.89	14,652.46	14,655.27	14,657.02	14,658.11	14,654.31	14,655.33	14,652.78	14,653.80	14,652.77	-	-	-	-
013	54.18	54.16	54.50	54.26	54.08	54.16	54.15	54.06	54.12	54.05	111.20	131.89	111.21	131.90
014	2,476.62	2,480.59	2,479.10	2,484.65	2,475.05	2,478.12	2,478.26	2,474.99	2,497.91	2,480.36	-	-	-	-
015	556.36	555.91	557.85	556.78	555.81	556.42	556.68	555.82	556.17	555.70	9,738.90	-	9,740.43	-
016	-	-	-	-	-	-	-	-	-	-	-	-	-	-
017	69.08	68.97	69.34	69.25	68.95	69.19	69.15	68.99	69.12	69.01	-	-	-	-
018	5,778.55	5,778.62	5,779.82	5,779.65	-	-	-	5,777.84	5,778.21	5,777.49	-	-	-	-

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfs}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
019	-	-	-	-	-	-	-	-	-	-	-	-	-	-
020	-	-	-	-	-	-	-	-	-	-	-	-	-	-
021	230.48	230.24	230.79	230.53	230.33	230.58	230.57	230.18	230.26	230.15	1,423.15	3,078.53	1,423.08	3,078.60
022	5,436.45	5,424.39	5,425.87	5,423.80	5,421.87	5,425.39	5,437.04	5,424.07	5,422.06	5,421.36	-	-	-	-
023	19.99	19.83	20.40	20.16	19.80	19.97	19.96	19.59	20.68	19.71	-	-	-	-
024	-	-	-	-	-	-	-	-	-	-	-	-	-	-
025	5,417.29	5,414.57	5,420.82	5,417.68	5,413.48	5,419.66	5,419.86	5,412.79	5,428.14	5,414.35	-	-	-	-
026	-	-	-	-	-	-	-	-	-	-	-	-	-	-
027	78.72	78.91	79.03	78.78	78.82	78.68	78.68	78.57	78.66	78.53	85.55	96.86	85.54	96.48
028	3,634.63	3,632.50	3,636.24	3,635.43	3,632.26	3,638.18	3,650.12	3,628.62	3,651.70	3,632.10	-	-	-	-
029	1,915.57	1,914.91	1,915.99	1,922.28	1,914.87	1,915.63	1,919.08	1,916.07	1,915.22	1,914.82	-	-	-	-
030	3.11	3.11	3.27	3.14	3.07	3.08	3.08	3.06	3.07	3.05	10.44	67.10	10.44	67.11
031	11.29	11.27	11.80	11.49	11.21	11.27	11.26	11.17	11.20	11.16	195.58	1,884.22	195.51	1,882.92
032	17.24	17.18	17.40	17.30	17.16	17.17	17.16	17.05	17.14	17.06	12,747.84	4,627.33	12,727.81	4,627.47
033	-	-	-	-	-	-	-	-	-	-	-	-	-	-
034	174.56	173.90	174.73	174.50	174.12	174.87	174.73	173.82	174.17	173.78	-	-	-	-
035	5.31	5.23	5.77	5.53	5.15	5.29	5.28	5.02	5.33	5.12	2,519.85	-	2,519.88	-
036	732.05	731.35	733.48	733.63	731.37	732.04	732.01	730.75	733.97	731.17	-	-	-	-

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
037	9.35	9.33	9.52	9.45	9.29	9.33	9.33	9.27	9.30	9.26	202.48	292.01	202.48	292.01
038	-	-	-	-	-	-	-	-	-	-	-	-	-	-
039	-	-	-	-	-	-	-	-	-	-	-	-	-	-
040	20,875.89	20,819.64	-	20,821.37	20,817.69	20,826.70	20,826.13	20,817.44	20,893.29	20,817.23	-	-	-	-
041	0.16	0.16	0.25	0.20	0.14	0.15	0.15	0.13	0.13	0.13	3.20	13.66	3.20	13.67
042	0.61	0.62	0.69	0.64	0.59	0.60	0.60	0.59	0.59	0.58	4.82	10.42	4.81	10.43
043	2.07	2.05	2.15	2.13	2.04	2.05	2.05	2.02	2.04	2.02	153.73	2,174.32	153.81	2,174.32
044	1,416.77	1,414.67	1,415.05	1,415.81	1,413.23	1,414.15	1,414.97	1,414.06	1,417.53	1,415.62	-	-	-	-
045	3.15	3.13	3.22	3.20	3.12	3.14	3.14	3.11	3.12	3.10	142.82	990.73	143.10	990.72
047	11.93	11.90	12.09	11.97	11.86	11.93	11.92	11.85	11.88	11.84	13.96	24.78	13.96	24.79
048	5.94	5.93	6.09	5.96	5.87	5.90	5.90	5.87	5.89	5.86	7.12	13.21	7.12	13.14
049	2.72	2.71	2.83	2.78	2.68	2.70	2.71	2.66	2.72	2.69	3,098.80	3,704.04	3,098.70	3,695.46
050	1.09	1.09	1.20	1.14	1.07	1.08	1.08	1.06	1.08	1.06	74.72	292.82	74.68	292.69
051	1,103.89	1,104.68	1,105.34	1,105.96	1,102.83	1,107.32	1,104.23	1,103.26	1,108.90	1,104.81	-	-	-	-
052	-	-	-	-	-	-	-	-	-	-	-	-	-	-
053	1,085.85	1,084.27	1,088.73	1,085.19	1,084.26	1,085.35	1,085.60	1,084.52	1,085.07	1,087.35	17,681.18	-	-	-
054	64.39	64.30	64.80	64.60	64.23	64.36	64.36	64.22	64.32	64.20	909.70	-	909.68	-
055	3,749.37	3,751.25	3,750.63	3,757.48	3,747.30	3,749.75	3,749.74	3,747.47	3,749.14	3,747.74	-	-	-	-

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
056	0.69	0.69	0.77	0.75	0.67	0.68	0.68	0.66	0.67	0.66	31.63	410.00	31.63	410.00
057	181.93	183.03	184.06	183.99	181.61	181.96	181.95	181.47	187.95	183.08	-	-	-	-
058	12,795.16	12,778.34	12,785.71	12,782.08	12,779.33	12,787.69	12,787.28	12,776.34	12,779.89	12,777.94	-	-	-	-
059	133.40	133.30	133.81	133.54	133.22	133.37	133.34	133.31	133.29	133.17	542.39	590.40	542.80	590.43
060	10,275.59	10,275.31	10,310.34	10,277.22	10,277.95	10,283.53	10,276.99	10,277.19	10,276.34	10,274.55	-	-	-	-
062	1.08	1.08	1.16	1.12	1.06	1.08	1.08	1.05	1.06	1.05	22.68	80.62	22.68	80.62
064	6.42	6.21	6.49	6.38	6.30	6.51	6.57	6.15	6.18	6.15	404.97	-	404.97	-
066	10.69	10.68	11.08	10.89	10.59	10.68	10.69	10.45	10.95	10.62	-	-	-	-
067	-	-	-	-	-	-	-	-	-	-	-	-	-	-
068	0.32	0.32	0.43	0.35	0.30	0.31	0.31	0.29	0.29	0.29	5.75	18.07	5.75	18.07
069	-	-	-	-	-	-	-	-	-	-	-	-	-	-
070	673.42	673.23	673.75	673.63	673.12	673.36	673.37	673.40	673.35	673.11	-	-	-	-
071	-	-	-	-	-	-	-	-	-	-	-	-	-	-
072	1,057.75	1,057.39	1,058.08	1,057.80	1,058.32	1,057.73	1,057.73	1,057.37	1,057.72	1,057.23	1,153.76	1,134.62	1,153.78	1,134.63
073	8.64	8.59	8.84	8.71	8.56	8.64	8.63	8.54	8.64	8.53	77.61	666.32	77.62	666.31
074	27.12	27.01	27.28	27.16	27.00	27.12	27.12	26.96	26.99	26.95	1,277.44	-	1,277.54	-
075	-	-	-	-	-	-	-	-	-	-	-	-	-	-
076	58.94	58.80	59.22	59.04	58.78	59.22	58.99	58.76	58.88	58.75	6,538.21	7,160.18	6,538.27	7,198.66

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
077	-	-	-	-	-	-	-	-	-	-	-	-	-	-
078	502.94	500.51	501.28	501.03	500.42	500.93	500.86	500.50	500.73	500.40	-	-	-	-
079	-	-	-	-	-	-	-	-	-	-	-	-	-	-
080	3.65	3.60	3.84	3.71	3.58	3.65	3.65	3.55	3.57	3.54	310.71	1,076.15	310.71	1,076.16
081	41.06	40.89	41.36	41.11	40.85	41.10	41.09	40.80	40.96	40.78	2,178.57	-	2,178.46	-
082	234.26	233.47	234.52	234.07	233.53	234.26	234.27	233.34	233.57	233.36	-	-	-	-
083	0.13	0.13	0.20	0.16	0.11	0.12	0.11	0.10	0.10	0.10	2.58	4.76	2.58	4.76
085	14.65	14.74	15.32	15.07	14.48	14.64	14.64	14.47	15.47	14.67	-	-	-	-
087	161.30	161.06	161.68	161.44	160.97	161.26	161.24	160.99	161.16	160.94	2,122.10	1,997.89	2,122.08	1,998.12
088	-	-	-	-	-	-	-	-	-	-	-	-	-	-
089	55.77	55.66	56.00	55.84	55.61	55.75	55.75	55.55	55.76	55.58	-	5,106.86	-	5,106.86
090	-	-	-	-	-	-	-	-	-	-	-	-	-	-
091	-	-	-	-	-	-	-	-	-	-	-	-	-	-
092	-	-	-	-	-	-	-	-	-	-	-	-	-	-
093	0.22	0.23	0.32	0.26	0.21	0.21	0.21	0.20	0.20	0.20	3.66	14.73	3.67	14.74
094	1,618.06	1,617.17	1,618.27	1,618.39	1,617.01	1,617.66	1,618.06	1,617.22	1,617.99	1,617.32	-	-	-	-
095	-	-	-	-	-	-	-	-	-	-	-	-	-	-
096	3,735.72	3,735.14	3,751.50	3,736.11	-	-	-	3,735.65	3,735.24	3,734.63	-	-	-	-

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
097	0.09	0.10	0.20	0.13	0.08	0.08	0.08	0.06	0.07	0.06	4.90	18.82	4.91	18.82
098	1.37	1.36	1.68	1.47	1.32	1.35	1.35	1.29	1.35	1.30	159.71	499.85	159.71	499.85
099	216.82	216.48	217.12	217.05	216.43	216.80	216.82	216.42	216.73	216.39	-	-	-	-
100	16,810.74	16,805.87	16,813.05	16,810.21	16,807.30	16,813.13	16,813.23	16,806.34	16,807.83	16,805.24	-	-	-	-
101	2,914.25	2,913.69	2,914.51	2,914.38	2,913.54	2,914.37	2,914.74	2,913.69	2,914.43	2,913.73	3,654.66	4,176.05	3,654.76	4,176.25
102	135.69	134.88	135.62	135.31	135.79	135.71	135.70	134.72	134.84	134.68	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	0.89	0.90	1.11	0.98	0.86	0.87	0.87	0.84	0.84	0.84	12.57	62.29	12.57	62.29
105	33.90	33.46	35.46	34.15	33.39	33.96	33.95	33.17	33.55	33.23	16,875.93	-	16,876.36	-
106	1.21	1.21	1.29	1.24	1.19	1.20	1.20	1.18	1.19	1.18	13.28	49.52	13.28	49.52
107	1,097.32	1,271.90	-	-	1,093.30	1,117.43	1,096.33	-	-	-	-	-	-	-
108	11,680.82	11,685.46	11,689.63	11,690.04	11,677.90	11,683.45	11,682.02	11,676.90	11,705.13	11,685.05	-	-	-	-
109	1.10	1.07	1.16	1.14	1.07	1.09	1.09	1.04	1.05	1.03	230.25	683.77	230.25	683.77
110	1.40	1.41	1.56	1.44	1.37	1.37	1.37	1.35	1.36	1.35	18.69	17.42	18.69	17.42
111	25.79	25.70	26.27	26.01	25.59	25.83	25.82	25.51	25.77	25.58	2,794.22	-	2,796.30	-
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	32.80	32.72	33.02	32.90	32.69	32.81	32.81	32.68	32.73	32.69	827.71	1,543.89	829.46	1,543.66
114	491.59	491.01	491.60	491.21	490.89	491.12	491.06	491.37	491.04	490.92	4,889.35	3,846.69	4,889.60	3,846.95

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup_{cyc}^{bfsc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
115	0.35	0.36	0.60	0.44	0.32	0.33	0.33	0.29	0.30	0.29	17.85	42.80	17.85	42.80
116	1,109.84	1,110.62	1,111.42	1,116.11	1,107.80	1,110.43	1,110.52	1,107.31	1,121.44	1,110.31	-	-	-	-
117	25.10	24.97	25.73	25.17	24.98	25.06	25.05	24.86	25.05	24.91	-	-	-	-
118	509.93	512.69	513.24	513.79	512.77	509.96	510.03	509.04	511.06	512.55	-	-	-	-
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	208.88	208.77	209.18	208.96	-	-	-	208.72	208.76	208.59	256.63	316.04	256.64	316.06
121	4,749.72	4,749.01	4,751.29	4,750.15	-	-	-	4,749.66	4,749.67	4,770.47	4,991.51	5,088.52	4,990.91	5,088.29
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	8.07	7.78	8.10	7.97	7.79	8.05	7.94	7.63	8.10	7.72	9,663.49	-	9,680.18	-
124	48.17	48.29	49.21	48.93	50.85	-	56.50	47.13	48.30	48.15	-	-	-	-
125	6,587.55	6,569.65	6,603.51	6,584.00	6,554.05	6,694.93	6,567.08	6,558.57	6,666.80	6,569.07	-	-	-	-
126	0.26	0.26	0.33	0.30	0.25	0.25	0.25	0.23	0.23	0.23	13.27	128.61	13.27	128.61
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	1,468.40	1,467.59	1,469.35	1,468.52	1,467.61	1,468.79	1,468.70	1,467.49	1,469.73	1,467.41	-	-	-	-
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-
131	115.99	115.86	116.44	116.19	115.80	116.03	116.05	115.80	115.88	115.76	575.75	683.08	575.76	683.10
132	0.67	0.67	0.88	0.76	0.64	0.65	0.64	0.62	0.62	0.61	26.31	45.92	26.31	45.93
133	2.47	2.46	2.84	2.60	2.41	2.44	2.44	2.36	2.36	2.36	103.74	160.14	103.74	160.15

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
134	19.36	18.56	19.19	18.94	18.48	21.84	19.09	18.17	19.00	18.43	-	-	-	-
135	28.69	27.10	28.37	27.50	27.63	31.98	28.62	26.83	27.37	26.95	-	-	-	-
136	550.39	549.70	550.61	550.71	550.06	550.81	550.75	549.53	550.65	549.53	-	-	-	-
137	-	-	-	-	-	-	-	-	-	-	-	-	-	-
138	1,075.85	1,075.43	1,076.32	1,076.08	1,075.37	1,075.82	1,075.77	1,075.35	1,075.77	1,075.33	17,283.06	-	17,280.05	-
139	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	3.82	3.64	4.00	3.77	3.67	3.91	3.78	3.54	3.65	3.56	-	-	-	-
141	1,493.68	1,480.77	1,494.81	1,483.89	1,562.07	1,524.88	-	1,478.40	1,525.84	1,484.24	-	-	-	-
142	299.52	299.16	300.08	299.56	299.14	299.42	299.65	298.90	299.35	299.01	18,757.79	-	18,755.78	-
143	-	-	-	-	-	-	-	-	-	-	-	-	-	-
144	1.36	1.36	1.44	1.40	1.34	1.35	1.35	1.33	1.34	1.33	14.86	25.20	14.86	25.20
146	30.54	30.35	30.81	30.58	30.36	30.58	30.58	30.28	30.51	30.29	-	-	-	-
147	20.72	20.64	20.92	20.79	20.59	20.72	20.72	20.54	20.74	20.57	4,318.21	-	4,318.27	-
148	1,894.70	1,894.02	1,896.31	1,895.05	1,895.83	1,894.98	1,895.05	1,893.67	1,896.31	1,893.93	-	-	-	-
149	0.18	0.19	0.25	0.19	0.16	0.16	0.16	0.16	0.16	0.15	1.03	4.52	1.04	4.52
150	247.50	247.25	247.65	247.57	247.03	247.36	247.34	247.07	247.26	246.99	2,025.68	2,789.52	2,025.57	2,789.48
151	-	-	-	-	-	-	-	-	-	-	-	-	-	-
152	12.35	12.32	12.64	12.48	12.22	12.43	12.42	12.13	12.64	12.23	11,474.07	14,007.11	11,473.61	14,006.78

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
153	-	-	-	-	-	-	-	-	-	-	-	-	-	-
154	-	-	-	-	-	-	-	-	-	-	-	-	-	-
155	0.07	0.08	0.18	0.12	0.06	0.06	0.06	0.05	0.05	0.05	1.24	8.22	1.24	8.22
157	11.26	11.21	11.61	11.36	11.18	11.25	11.25	11.15	11.19	11.14	1,498.81	-	1,498.82	-
158	152.28	152.11	152.87	152.41	152.03	152.39	152.41	151.87	152.35	151.88	-	-	-	-
160	4.75	4.70	5.40	4.90	4.66	4.71	4.71	4.60	4.62	4.59	742.57	-	741.61	-
161	0.45	0.43	0.57	0.47	0.42	0.44	0.44	0.40	0.40	0.40	22.15	208.09	22.17	207.99
162	19.33	18.85	19.40	19.16	19.08	19.45	19.67	18.76	18.89	18.74	2,748.99	-	2,748.78	-
163	704.18	701.65	704.10	703.19	701.81	705.45	706.83	700.81	703.09	701.42	-	-	-	-
164	0.10	0.10	0.17	0.14	0.09	0.09	0.09	0.07	0.07	0.07	1.95	5.89	1.95	5.89
165	358.61	358.23	359.11	358.78	358.17	358.52	358.54	358.02	358.70	357.97	-	-	-	-
166	462.80	462.29	463.47	462.97	462.15	463.02	462.84	461.87	463.73	462.35	-	-	-	-
167	182.21	182.05	181.96	183.55	185.42	194.80	186.08	174.57	190.80	181.74	-	-	-	-
168	56.62	56.49	56.77	56.61	56.44	56.58	56.53	56.45	56.52	56.43	272.29	211.19	272.37	211.19
169	400.53	399.66	400.60	400.60	399.68	400.59	400.56	399.61	400.72	400.40	-	-	-	-
170	-	-	-	-	-	-	-	-	-	-	-	-	-	-
171	0.81	0.78	0.92	0.84	0.78	0.80	0.80	0.75	0.76	0.75	250.91	198.72	250.91	198.73
172	47.32	47.34	47.54	47.45	47.23	47.32	47.35	47.24	47.27	47.25	451.17	582.87	451.17	582.89

Table B.9: All algorithms CPU Time on global-tables

Instance #	# CPU Time													
	APOAC$_{around\cup}$	**PrePeak^{+}-POAC1**	**A$\cup^{bfsc}_{cyc}$POAC**	**ANPOAC**	**SAC1**	**APOAC**	**POAC1**	**CT**	**GAC2001**	**STR2**	**R(*,3)C**	**R(*,4)C**	**wR(*,3)C**	**wR(*,4)C**
173	2,679.13	2,678.81	2,680.75	2,679.96	2,678.52	2,685.83	2,679.56	2,678.25	2,679.80	2,678.69	-	-	-	-
174	-	-	-	-	-	-	-	-	-	-	-	-	-	-
175	245.00	244.82	245.61	245.23	244.72	245.07	245.07	244.75	245.04	244.73	3,402.74	-	3,402.69	-
176	0.31	0.31	0.39	0.36	0.30	0.30	0.30	0.28	0.29	0.28	8.23	14.58	8.23	14.58
177	720.50	720.28	721.98	720.64	720.08	720.44	720.38	720.08	720.28	720.00	758.16	935.74	758.49	935.99
178	222.63	222.41	223.05	222.86	222.36	222.48	222.48	222.18	222.25	222.11	2,483.59	-	2,483.30	-
179	4.25	4.25	4.58	4.45	4.19	4.23	4.22	4.16	4.16	4.15	173.00	343.96	172.89	344.22
180	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bibliography

[Balafrej *et al.*, 2014] Amine Balafrej, Christian Bessière, El-Houssine Bouyakhf, and Gilles Trombettoni. Adaptive Singleton-Based Consistencies. In *Proceedings of AAAI-2014*, pages 2601–2607, 2014.

[Bayer *et al.*, 2006] Ken Bayer, Josh Snyder, and Berthe Y. Choueiry. An Interactive Constraint-Based Approach to Minesweeper. In *Proceedings of AAAI-2006*, pages 1933–1934, Boston, MA, 2006.

[Bessière and Régin, 1996] Christian Bessière and Jean-Charles Régin. MAC and Combined Heuristics: Two Reasons to Forsake FC (and CBJ?) on Hard Problems. In *Proceedings of* 2 nd *International Conference on Principle and Practice of Constraint Programming (CP'96)*, volume 1118 of *LNCS*, pages 61–75. Springer, 1996.

[Bessière *et al.*, 2005] Christian Bessière, Jean-Charles Régin, Roland H.C. Yap, and Yuanlin Zhang. An Optimal Coarse-Grained Arc Consistency Algorithm. *Artificial Intelligence*, 165(2):165–185, 2005.

[Bessière *et al.*, 2008] Christian Bessière, Emmanuel Hebrard, Brahim Hnich, Zeynep Kiziltan, and Toby Walsh. SLIDE: A useful special case of the CARDPATH constraint. In *Proceedings of* 18 th *European Conference on Artificial Intelligence (ECAI 2008)*, pages 475–479, 2008.

[Boussemart *et al.*, 2004] Frédéric Boussemart, Fred Hemery, Christophe Lecoutre, and Lakhdar Sais. Boosting Systematic Search by Weighting Constraints. In *Proceedings of the 16 th European Conference on Artificial Intelligence*, pages 146–150, 2004.

[Briggs and Torczon, 1993] Preston Briggs and Linda Torczon. An Efficient Representation for Sparse Sets. *ACM Lett. Program. Lang. Syst.*, 2(1-4):59–69, March 1993.

[Debruyne and Bessière, 1997] Romuald Debruyne and Christian Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *Proceedings of the 15 th International Joint Conference on Artificial Intelligence*, pages 412–417, 1997.

[Demeulenaere *et al.*, 2016] Jordan Demeulenaere, Renaud Hartert, Christophe Lecoutre, Guillaume Perez, Laurent Perron, Jean-Charles Régin, and Pierre Schaus. Compact-Table: Efficiently Filtering Table Constraints with Reversible Sparse Bit-Sets. In *Proceedings of* 22 *nd International Conference on Principle and Practice of Constraint Programming (CP 2016)*, volume 9892 of *LNCS*, pages 207–223. Springer, 2016.

[Gyssens, 1986] M. Gyssens. On the Complexity of Join Dependencies. *ACM Trans. Database Systems*, 11(1):81–108, 1986.

[Howell *et al.*, 2018] Ian Howell, Robert J. Woodward, Berthe Y. Choueiry, and Christian Bessière. Solving Sudoku with Consistency: A Visual and Interactive Approach. In *Proceedings of the 27 th International Joint Conference on Artificial Intelligence*, pages 5829–5831, Stockholm, Sweden, 2018.

[Janssen *et al.*, 1989] P. Janssen, Philippe Jégou, B. Nougier, and M.C. Vilarem. A Filtering Process for General Constraint-Satisfaction Problems: Achieving Pairwise-Consistency Using an Associated Binary Representation. In *IEEE Workshop on Tools for AI*, pages 420–427, 1989.

[Karakashian *et al.*, 2010] Shant Karakashian, Robert Woodward, Christopher Reeson, Berthe Y. Choueiry, and Christian Bessière. A First Practical Algorithm for High Levels of Relational Consistency. In *Proceedings of AAAI-2010*, pages 101–107, 2010.

[le Clément *et al.*, 2013] Vianney le Clément, Pierre Schaus, Christine Solnon, and Christophe Lecoutre. Sparse-Sets for Domain Implementation. In *Proc. of the CP Workshop on TRICS 2013*, 2013.

[Lecoutre, 2011] Christophe Lecoutre. STR2: Optimized Simple Tabular Reduction for Table Constraints. *Constraints*, 16(4):341–371, 2011.

[Mackworth, 1977] Alan K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118, 1977.

[Pesant, 2004] Gilles Pesant. A regular language membership constraint for finite sequences of variables. In *Proceedings of* 10 th *International Conference on Principles and Practice of Constraint Programming (CP 2004)*, volume 3258 of *LNCS*, pages 482–495. Springer, 2004.

[Reeson *et al.*, 2007] Christopher G. Reeson, Kai-Chen Huang, Kenneth M. Bayer, and Berthe Y. Choueiry. An Interactive Constraint-Based Approach to Sudoku. In *Proceedings of AAAI-2007*, pages 1976–1977, Vancouver, British Columbia, 2007.

[Swearingn *et al.*, 2011] Amanda Swearingn, Berthe Y. Choueiry, and Eugene C. Freuder. A Reformulation Strategy for Multi-Dimensional CSPs: The Case Study of the Set Game. In *Ninth International Symposium on Abstraction, Reformulation and Approximation (SARA 2011)*, pages 107–116. AAAI Press, 2011.

[Woodward, 2018] Robert J. Woodward. *Higher-Level Consistencies: Where, When, and How Much.* PhD thesis, Department of Computer Science and Engineering, University of Nebraska-Lincoln, 2018.