2005

Forward checking in the primal and dual constraint graphs.

Robert George Price
University of Windsor

Forward Checking in the Primal and Dual Constraint Graphs

by

Robert George Price

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the School of Computer Science
In Partial Fulfillment of the Requirements for
The Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada

2005

Your file Votre référence
ISBN: 0-494-09811-2
Our file Notre référence
ISBN: 0-494-09811-2

Abstract

Constraint Satisfaction Problems (CSPs) have been a subject of research in Artificial Intelligence for many years. CSPs are a general way of describing problems that can be used to represent many different types of real-world problems, including scheduling, planning, timetabling, and other combinatorial problems. The primal and dual constraint graphs are two ways of representing a CSP. Some CSPs have features that can be exploited by algorithms trying to find solutions. In this work, results from solving CSPs using forward-checking algorithms that use the primal- and dual-graph representations will be presented, and regions where one representation performs better than the other will be identified. It will be shown that the dual representation performs better than the primal representation on CSPs with tight constraints.

Dedication

To my Dad

Acknowledgements

I would like to thank my supervisor Dr. Scott Goodwin, for his guidance, and for making this process an enjoyable experience. I am also grateful for his support, and for the scholarships provided by the University of Windsor.

I also extend my appreciation to the members of my committee - Dr. Myron Hlynka and Dr. Joan Morrissey, for their feedback.

I am also grateful for the support from my family and friends during this process. I want to especially thank my wife Divina, for her encouragement and patience during this long process.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Constraint Satisfaction Problems (CSPs) have been a subject of research in Artificial Intelligence for many years [SmB95a]. CSPs are a general way of describing problems that can be used to represent many different types of real-world problems, including scheduling, planning, timetabling, and other combinatorial problems. Some CSPs have features that can be exploited by algorithms trying to find solutions. In the following sections two different algorithms that solve CSPs will be explained, and the experiments I performed are described.

1.1 Problem Statement

A Constraint Satisfaction Problem is a problem with a finite set of variables (each with a finite domain of values) and a set of constraints that restrict the possible assignments of values to variables. The primal- and dual-graph methods are two different ways of modelling a CSP. It has been recently shown in [Hua04] that for some problems, it is quicker to solve the CSP using the primal representation, and on other problems, the dual representation is faster. I have solved several different CSPs using the basic Forward Checking algorithm (FC) on the primal representation and the dual representation. I have varied the constraint tightness of the problems to be solved, to identify regions where one representation is better than another for solving CSPs.

1.2 Outline

Chapter 2 provides a detailed background of CSPs, and provides definitions and gives examples of the different representations. Chapter 3 describes two forward-checking algorithms, FC and Constraint-Directed Forward Checking (CDFC), which are extensions of two backtracking algorithms, Backtracking (BT) and Constraint-Directed Backtracking (CDBT). Chapter 4 outlines the experiments that I performed, Chapter 5 analyses the results, and Chapter 6 provides conclusions. The complete results are listed, and a sample calculation is included in the Appendix.

Chapter 2: Background

2.1 Constraint-Satisfaction Problems

A Constraint-Satisfaction Problem (CSP) is a problem with a finite set of variables, with each variable having a finite domain of values that it can take its value from, and a set of constraints that can restrict the values that the variables can simultaneously take. A compound label is the simultaneous assignment of values to a set of variables. Formally, Tsang [Tsa93] (Definition 1-12, p.9) defines a constraint-satisfaction problem as a triple:

(Z, D, C)
where
Z = a finite set of variables $\{x_1, x_2, \ldots, x_n\}$;
D = a function which maps every variable in Z to a set of objects of arbitrary type:
  $D : Z \rightarrow$ finite set of objects (of any type)
  We shall take D_{xi} as the set of objects mapped from x_i by D. We call these objects possible values of x_i and the set D_{xi} the domain of x_i;
C = a finite (possibly empty) set of constraints on an arbitrary subset of variables in Z. In other words, C is a set of sets of compound labels.

We use csp(P) to denote that P is a constraint satisfaction problem.

Solving a CSP means to assign a value to each variable without violating any constraints. Formally, Tsang [Tsa93] (Definition 1-13,p.10) defines a solution tuple of a CSP as a compound label for all those variables that satisfy all the constraints:

$$\forall\ csp((Z,D,C)) : \forall\ x_1,x_2, \ldots ,x_n \in Z : (\forall\ v_1 \in D_{x1},\ v_2 \in D_{x2}, \ldots, v_n \in D_{xn}:$$
$$solution\_tuple((\langle x_1,v_1\rangle\langle x_2,v_2\rangle\ldots\langle x_n,v_n\rangle, (Z,D,C)) \equiv ((Z=\{x_1,x_2,\ldots,x_n\}) \wedge (\forall c \in C: satisfies(\langle x_1,v_1\rangle\langle x_2,v_2\rangle\ldots\langle x_n,v_n\rangle), c)))$$

Depending on the application, the goal might be to find any solution tuple, all solution tuples, or optimal solutions, where the optimal is defined depending on the domain.

2.2 Example: The n-Queens Problem

The n-Queens problem requires n queens to be placed on an n-by-n sized chessboard so that no two queens are attacking each other. It can be modelled as a CSP several different ways, but a common way is by having n variables that represent the n ranks on the chessboard. The domain of each variable is {1… n}, representing the file that the queen on that rank is placed. For example, assigning the first variable the value 3 means a queen is placed on the third file in the first rank. The constraints in this CSP ensure no two queens attack one another. One example of a constraint might be "if there is a queen in rank I, file J, the queen in rank I+1 cannot be in file J, J-1, or J+1". One solution to the 4-queens problem is to place the queen from rank 1 on file 2, the queen from rank 2 on file 4, the queen from rank 3 on file 1, and the queen from rank 4 on file 3 (<V1, 2>, <V2, 4>, <V3, 1>, <V4, 3>). Figure 1 shows this solution.

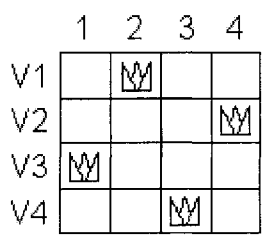


Figure 1: A Solution to the 4-Queens Problem

2.3 Characteristics of CSPs

A **Unary Constraint** is a constraint on one variable, for example: X is an even number. A **Binary Constraint** is a constraint over two variables, for example: $X + Y < 10$. Any constraint that uses more than two variables is considered a **Non-Binary Constraint.** The **Arity** of a constraint is the number of variables involved in a constraint. A CSP can be binary or non-binary. A **Binary CSP** is a CSP that contains only unary and binary constraints. A **Non-Binary CSP** has one or more non-binary constraints. A non-binary CSP is often called a **General CSP**. The arity of a CSP is the maximum arity of the constraints in the CSP.

Given a set of variables, if you assign each variable a value from its domain, it is called an **instantiation**. As explained before, a solution to a CSP is an instantiation of all the variables in the CSP, such that none of the constraints are violated. A CSP that has at least one solution is **solvable** or **consistent**, and is **unsolvable** or **over-constrained** otherwise.

The **Tightness** of a constraint is defined as 1 minus (the number of consistent instantiations of the variables involved in that constraint divided by the total possible number of instantiations of those variables). For example, let A and B be variables, each with domains {1,2,3}. If constraint C on variables A and B stated that A was less than B, there would be three tuples that would satisfy that constraint {(<A,1>,<B,2>), (<A,1>,<B,3>), and (<A,2>,<B,3>)} out of a possible nine tuples, so the tightness of C would be 1-(3/9), or 66.6%. Other things being equal, the tighter a constraint is, the fewer tuples there are that will satisfy it. A similar definition exists for the tightness of a CSP, which is a measure of the number of solution tuples over the total number of distinct-value tuples in all the variables in the problem. For example, if a CSP has two variables, each with a domain size of 10, and only 3 solutions, the tightness of the CSP would be 1 - (3/(10*10)), or 97%.

The **Constraint Density** of a CSP is the measure of constraints in the CSP compared to the total possible number of constraints that could be in that CSP. For example, let P be a binary CSP with variables A, B, and C, and constraints between variables A and B, and between variables B and C. Since there are two constraints out of a possible three constraints (those two, plus one on variables A and C) the constraint density would be 2/3, or 66.6%. A lesser-used definition of constraint density is the fraction of possible constraints beyond the minimum that the problem has. A binary CSP with n variables needs at least n-1 constraints for its graph to be connected. If the graph is not connected, the subgraphs can be treated as independent problems and solved separately.

Constraints in a CSP can be given implicitly or explicitly. If a constraint is given implicitly, for example A < B, it is an **intensional** constraint. If the constraint is given explicitly, for example {(<A,1>,<B,2>), (<A,1>,<B,3>), ...}, it is an **extensional** constraint. Section 2.7 mentions how some algorithms can take advantage of intensional constraints to save on the total space required.

A **Graph** is a structure <V, E>, where V is a finite set of vertices (sometimes referred to as nodes), and E is a finite set of edges. Every edge in E contains two vertices from V, which means that these two vertices are connected. Section 2.5 explains how a CSP can be represented as graph. A **Hypergraph** is a similar structure, however, the edges in E can contain more than two vertices. Any such edge that contains three or more vertices is called a hyperedge, and connects all those vertices together in the hypergraph.

2.4 Global and Local Search Methods

CSP-solving methods can be broken up into two categories: global and local search methods. A global search method is one that takes a CSP, and systematically goes through the entire search space. A search tree (or search graph), which is generated from an initial state by finding the possible successor states to that state, is used to do this. The non-leaf nodes of the search tree represent partial solutions to the problem. For some problems, the path to the goal is irrelevant. Only the final solution is important (such as in the n-queens problem).

If the path to the goal does not matter, as in vehicle routing, or job-shop scheduling, a local search method may be more suitable. Local search algorithms operate using a current state, and generally move to neighbours of that state. Two advantages that they have over global search algorithms are: they use little memory, and they can usually find reasonable solutions in large state spaces where global search algorithms are unsuitable. A disadvantage is that a local search method cannot guarantee that a solution it returns is an optimal one, since it does not search through the entire search space. Also, a global method is needed to show that a CSP is over-constrained. A local search method that does not find a solution for a certain CSP might not return a solution because there is none, or because it just cannot find a solution.

2.5 Primal and Dual Representations

There are different ways of representing CSPs. Nagarajan [Nag00] gives the following definitions for the primal-constraint graph and the dual-constraint graph (Definitions 2.6 and 2.7, p. 10): "For a binary CSP, the primal-constraint graph associated with it is a labeled constraint graph, where the nodes are the variables, and there is an edge between two nodes if there is a constraint between those variables. For a CSP, the dual-constraint graph associated with it is a labeled graph where the nodes are the constraints, and for every two constraints that have variables in common, there is an edge in the dual-graph connecting the two constraints."

Nagarajan then gives the following example to illustrate the differences. The CSP has four variables, variables 1 and 2 with domains {0...2}, and variables 3 and 4 with domains {1...4}. There is a constraint between variables 1, 2, and 3 stating that variable 1 + variable 2 < variable 3. There is a constraint between variables 1, 2, 3, and 4 stating that all the variables have to have different values. There is a constraint between variables 1 and 4 stating that variable 1 < variable 4. Finally, there is a constraint between variables 2 and 3 stating that their values are not equal. The following figure illustrates the different representations.

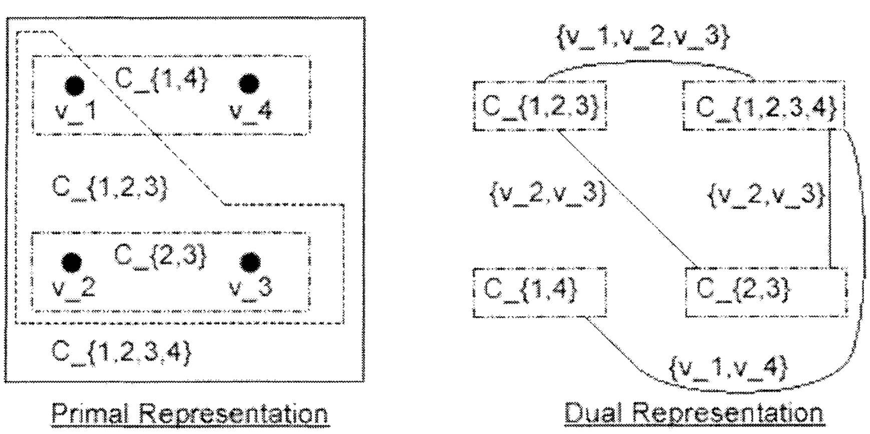
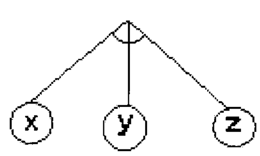


Figure 2: Primal and Dual Representations of a CSP (from [Nag00], fig 2.1, p.11)

Note that the dual representation is a binary CSP (all of the dual constraints specify the values that at most two dual variables can take at the same time), even though the primal representation is a general CSP (there are some constraints that specify the values that more than two variables can take at the same time, like the constraint between variables 1, 2, and 3).

Tsang [Tsa93] points out that several CSP solving techniques are only applicable to binary CSPs, and that although every general CSP can be converted to a binary CSP with the same solutions, he points out that it might not be beneficial to convert it. For example, Figure 3 shows two ways of expressing the same constraint. The left hand side shows a general constraint between three variables, each having a domain {1,2,3}, where each variable has to have a different value. The right hand side shows the same problem, but a new variable has been added, and the general constraint has been replaced with three binary constraints.

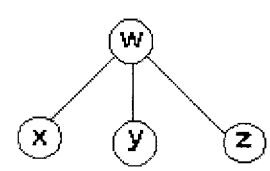
General constraint:
valid combinations are:
{(<x,1><y,2><Z,3>), (<x,1><y,3><z,2>),
(<x,2><y,1><z,3>), (<x,2><y,3><z,1>),
(<x,3><y,1><z,2>), (<x,3><y,2><z,1>)}



The domains of x, y, and z are all {1,2,3}

General Constraint

New variable, domain is:
{(<x,1><y,2><Z,3>), (<x,1><y,3><z,2>),
(<x,2><y,1><z,3>), (<x,2><y,3><z,1>),
(<x,3><y,1><z,2>), (<x,3><y,2><z,1>)}



The constraint between x and w is a binary constraint, requiring the value for x to be a projection of the value of w. Similar constraints exist between y and w, and z and w.

Binary Constraints

Figure 3: A General Constraint and Three Binary Constraints

Nagarajan [Nag00] points out that even though it is always possible to construct a binary representation of a CSP that is equivalent to the general one, the binary one does not always use the same set of variables. This binary representation sometimes loses some information relating to the real-life problem, since many constraints are naturally formulated with more than two variables. For example, the case where n variables each have to be given a different value (the all-different constraint) is discussed. The all-different constraint can be specified using binary constraints specifying that every two variables in the set need to be assigned different values, but Nagarajan states that the general constraint is a more natural way of expressing the constraint.

2.6: Consistency Techniques and Heuristics

There are many techniques that can be employed to make a CSP easier to solve. [Tsa93] and [Rus03] describe Forward Checking (FC), Maintaining Arc-Consistency (MAC), and Backjumping (BJ). Some of these techniques propagate information through constraints, removing values from domains of variables that cannot take part in any solution. This reduces the search space of the problem, making it easier to solve. There are variable and value-ordering heuristics that can be used with CSP solving algorithms that will help find a solution more quickly than just using the given ordering. Bacchus and van Run [Bac95] experiment with the dynamic variable-ordering heuristic with twelve different algorithms on several different problems, and provide results that show that with all of the problems, the three best algorithms used the Dynamic Variable Ordering (DVO) heuristic. Kwan and Tsang point out in [Kwa95] that it is important to use variable-ordering heuristics when comparing different CSP algorithms, especially since it is likely that they will only be used with heuristics in practice. They run experiments where random CSPs were solved both with and without variable-ordering heuristics. The DVO heuristic picks the variable that has the smallest domain when deciding which unassigned variable to try to assign a value to next.

2.7: Constraint Covers

Nagarajan [Nag00] states, "the set of all the solutions of a constraint-satisfaction problem is equal to the join of the relational instances corresponding to all the constraints". The author then goes on to define a constraint cover as (Definition 3.16, p.51)

> Given $C=\{C_1, C_2, \ldots, C_m\}$, and a subset of C, C_{cover} Each $C_i \in C_{cover}$ is given as $<V_i, S_i>$, where $V_i \subseteq V$. C_{cover} covers V iff $\cup^m_{i=1} V_i = V$. C_{cover} is a constraint cover of V. As well, C_{cover} is a minimal constraint cover of V if it is a constraint cover of V and no proper subset of C_{cover} is a constraint cover of V.

The author uses the notation $<V_i, S_i>$ to state that constraint C_i constrains the variables in V_i, and that S_i is the set of compound labels that are allowed by the constraint. The author then shows that a search procedure that searches through the dual encoding of a CSP, based on the constraints in a constraint cover of a CSP, is sound and complete. For the constraints that are not in the cover, an intensional representation can be used, which makes the total space required for the dual encoding based on the constraint covering less than a standard dual encoding.

2.8: Hard and Exceptionally Hard CSPs

Cheeseman, Kanefsky, and Taylor describe a region where CSPs are harder to solve [Che91]. This region occurs between CSPs that are easy to solve (under-constrained), and those that are unlikely to have a solution (over-constrained). They discuss this phenomenon using Hamiltonian circuits and graph-colouring problems, and then give experimental results backing up the theoretical ones. Gent and Walsh explore this phenomenon further in [Gen94]. They generated and solved several random CSPs in an area where most CSPs were easy to solve, and realized that there were some CSPs that took much longer to solve than other CSPs in that area.

Smith and Grant [SmB95b] note that there are CSPs that can be found in the region where almost all problems are soluble, but are more difficult by at least an order of magnitude than almost all other problems with the same parameter values, and are more difficult than almost all problems that occur in the region between the easily solved and the "unlikely to have a solution" region. A solution to one of these "exceptionally hard problems" is examined in detail in [SmB95b], and it is shown that the thrashing behaviour that is experienced is caused by a poor choice of the first variable, which causes an insolvable subproblem that is difficult for Forward Checking to recognize. The authors show that the maintaining arc-consistency algorithm (MAC) is also susceptible to this behaviour, and point out that a problem that is exceptionally difficult for one algorithm to solve may be easy for another one to solve.

Chapter 3: Backtracking Algorithms

3.1 Backtracking Algorithm (BT)

The Backtracking (BT) algorithm is a simple global search algorithm. The BT algorithm picks a variable, and then assigns it a value from its domain. If there are no constraints violated by this partial solution, then the process is repeated until all of the variables are assigned values. If assigning the current variable the current value violates a constraint, a different value is assigned. If no value can be assigned to the current variable without violating a constraint, the algorithm "backtracks" to the last assigned variable, and assigns it a different value. Figure 4 shows the search tree that BT goes through for the 4-queens problem when trying to assign the variables in the following order: V1, V2, V3, V4. The number in the top-right corner of each node gives the order that the backtracking algorithm searches through the space.

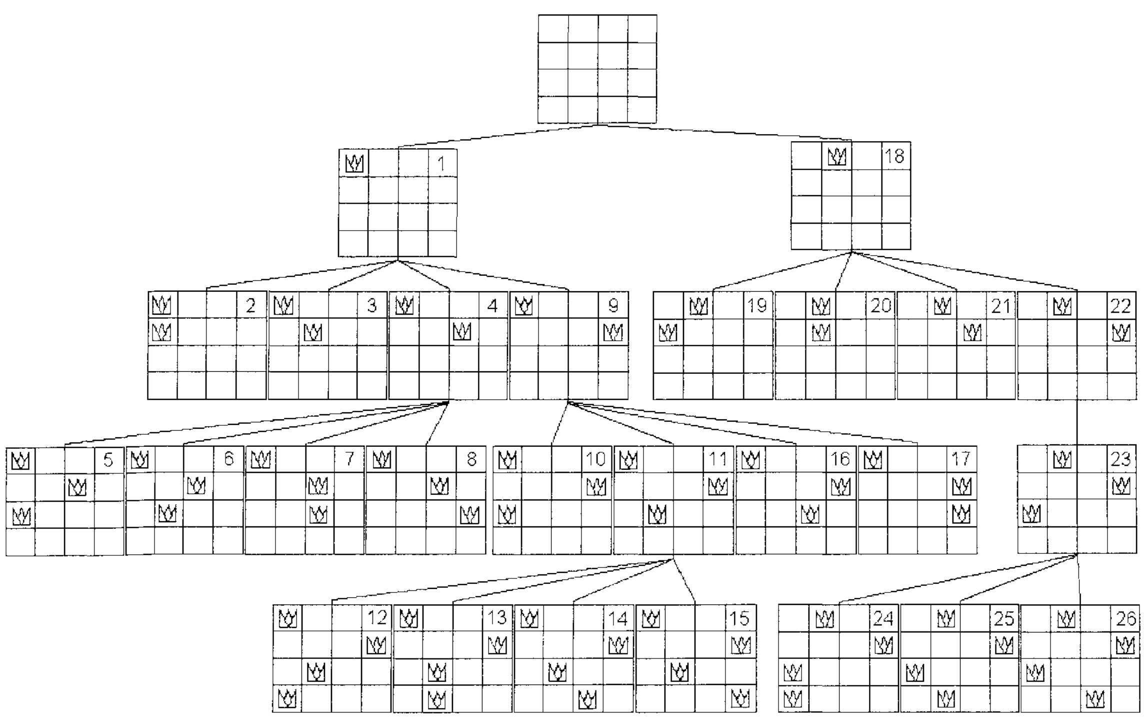


Figure 4: The Search Tree Created by BT on the 4-Queens Problem

3.2 Constraint-Directed Backtracking (CDBT)

The basic idea of the Constraint-Directed Backtracking (CDBT) algorithm is to search instantiations of variables in a variable set from a given constraint imposed on that variable set, and append it to a partial solution. When a partial solution cannot be extended, CDBT backtracks to a previously instantiated variable set, re-instantiates variables in that set, and continues from there. This depth-first algorithm is similar to the basic Back-Tracking algorithm (BT), except that instead of looking through the domain of a variable to pick the next value, the constraints are used to pick the next variable. A complete description of the CDBT algorithm is given in [Pan96].

In [Pan97], the authors compare CDBT to the basic BT algorithm, prove that it is sound and complete, and give some experimental results showing that CDBT performs better than BT on the n-queens problem. The authors note that advanced backtracking techniques such as Backjumping, Conflict-Directed Backjumping, and Forward Checking, that can be applied to BT, can also be applied to CDBT. The authors claim that CDBT's advantages are easier to see when solving general CSPs, but no experimental results were provided for BT on general CSPs. The authors compare CDBT to two other decomposition schemes: the Tree Clustering Scheme (TC) and the Hinge Decomposition Scheme (HD). The authors state, "on CSPs with only one maximal clique or only one minimal hinge, both TC and HD degenerate…and lose all advantage. On the one hand, CDBT is a general method that can be applied to any kind of problems without losing its advantage" [Pan97] (p. 9). A clique is a complete graph (all vertices are adjacent), or a fully connected sub-graph of a graph. A hinge is a vertex that when deleted, along with the incident edges, breaks a connected graph into two or more disconnected pieces. More information on the TC method, cliques, hinges, and other problem-specific features that can be used for solving CSPs can be found in [Tsa93].

If the binary 4-queens CSP problem is converted into a dual representation, the dual variables will be C_V1_V2, C_V1_V3, C_V1_V4, C_V2_V3, C_V2_V4, and C_V3_V4. Every pair of variables in the primal representation is a variable in the dual representation because there was a constraint between every pair of queens. The dual constraints in this problem state that the value assigned to the queen V1 in C_V1_V2 is the same as the value assigned to the queen V1 in C_V1_V3. Similar constraints exist between every pair of dual variables that have a queen in common.

Figure 5 shows the search tree created by CDBT when the variables are assigned in this order: C_V1_V2, C_V2_V3, C_V3_V4. These dual variables form a cover over all the variables in the primal constraint graph. The remaining dual variables will be used to check to make sure the current partial solution is valid. The domains of C_V1_V2, C_V2_V3, C_V3_V4 are all {(1, 3), (1, 4), (2, 4), (3, 1), (4, 1), (4, 2)}.

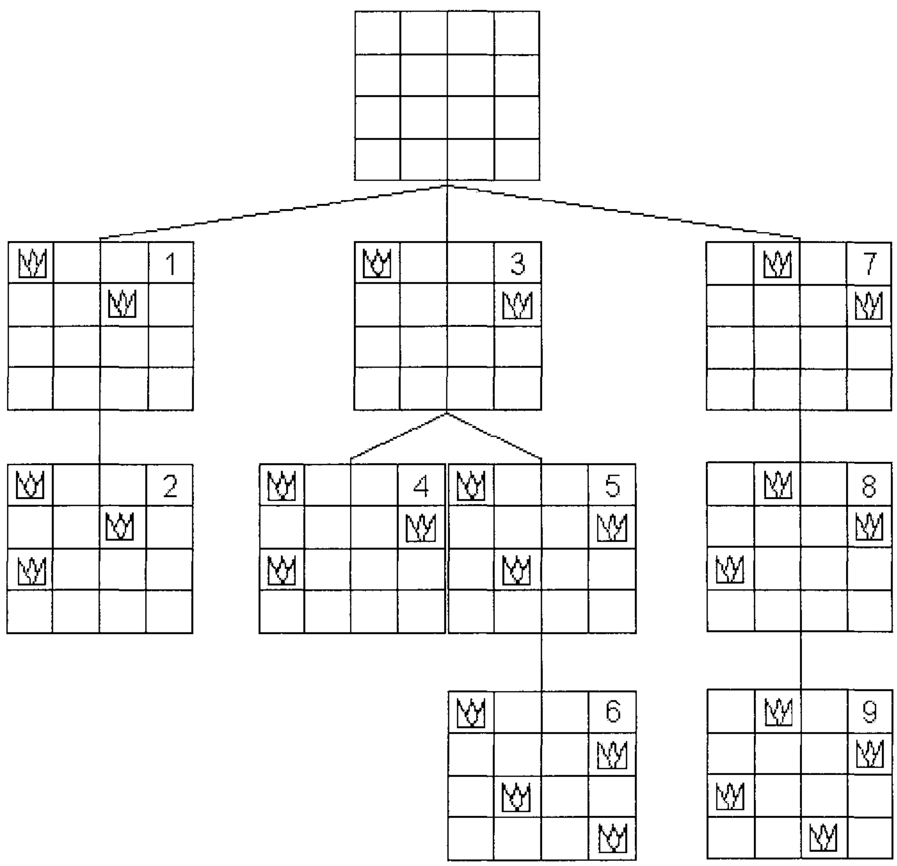


Figure 5: The Search Tree Created by CDBT on the 4-Queens Problem

First, CDBT assigns the value (1, 3) to variable C_V1_V2. Next, the variable C_V2_V3 is assigned the value (3, 1) (since it is the first tuple that assigns 3 to V2). This assignment violates the constraint on C_V1_V3 because (1, 1) is not in the domain for that variable. CDBT tries to assign another value to C_V2_V3, but cannot, since there are no more values in the domain that have a 3 for V2. The algorithm backtracks and assigns the next value to C_V1_V2, which is (1, 4). Next, the variable C_V2_V3 is assigned the value (4, 1) (since it is the first tuple that assigns 4 to V2). This assignment also violates the constraint on C_V1_V4 because (1, 1) is not in the domain for that variable. Next, CDBT assigns the variable C_V2_V3 the value (4, 2) (since it is the next tuple that assigns 4 to V2). This partial assignment is ok, since (1, 3) is in the domain for C_V1_V3. Next, CDBT assigns the variable C_V3_V4 the value (2, 4) (since it is the next tuple that assigns 2 to V3). This assignment violates the constraint on C_V2_V4 because (4, 4) is not in the domain for that variable. CDBT backtracks out of C_V3_V4, and then out of C_V2_V3 since there are no more values in the domains to try. CDBT assigns the next value to C_V1_V2, which is (2, 4). Next, the variable C_V2_V3 is assigned the value (4, 1) (since it is the first tuple that assigns 4 to V2). This partial assignment is ok, since (2, 1) is in the domain for C_V1_V3. Next, the variable C_V3_V4 is assigned the value (1, 3) (since it is the first tuple that assigns 4 to V2). This assignment is ok, since (2, 3) is in the domain for C_V1_V4 and (4, 3) is in the domain for C_V2_V4. The algorithm stops here and returns the solution (<V1, 2>, <V2, 4>, <V3, 1>, <V4, 3>).

3.3 Forward Checking (FC)

The Forward Checking algorithm is similar to the Backtracking algorithm, except after each instantiation of a variable, the domains of the remaining unassigned variables are checked. During this checking, elements that are not consistent with the instantiation of the current variable are removed. For example, in the 4-queens problem, after a queen is placed in the first column of the first row, the domain of the second queen will be modified to remove the first and second column, because it will conflict with the queen that is in the first column in row 1. If a point is reached where the domain of a variable is reduced to nothing, it is referred to as Domain Wipe-Out (DWO), and the algorithm can backtrack immediately, since the partial solution cannot be extended to a solution. Figure 6 shows the search tree for the 4-queens problem using Forward Checking.

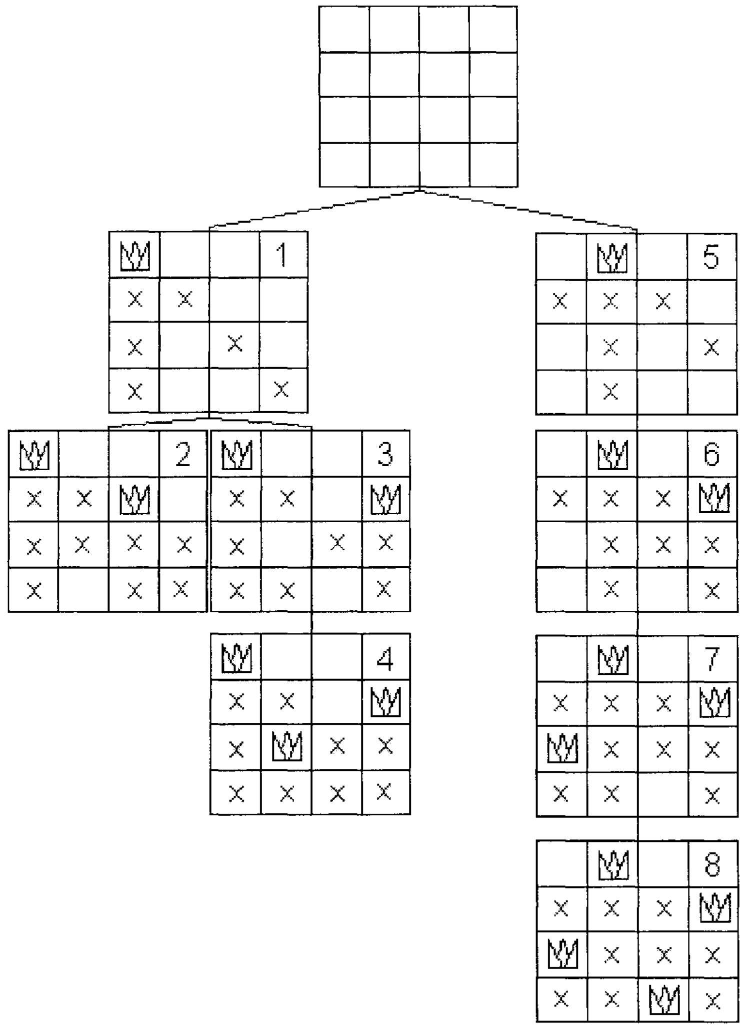


Figure 6: The Search Tree Created by FC on the 4-Queens Problem

3.4 Constraint-Directed Forward Checking

Constraint-Directed Forward Checking is similar to Forward Checking in the dual-graph, except some constraints are used as a cover, and others are used for testing. For example, in the 4-queens problem, the constraints between row 1 and row 2 (C_V1_V2), row 2 and row 3(C_V2_V3), and row 3 and row 4 (C_V3_V4) can be a cover, and the remaining constraints (C_V1_V3, C_V2_V4, C_V1_V4) can be used to test partial solutions for consistency. Unlike using FC with Dynamic Variable Ordering (DVO), CDFC will always use the same constraints when picking variables. It is possible that the variable ordering will change using DVO when finding all solutions to a CSP. For example, after picking a value for variable 1, the variable with the smallest domain might be variable 2, but after backtracking and trying a different value for variable one, the variable with the smallest domain might be variable 3.

Using a minimal cover will ensure a minimal depth of the search tree, however, it might not reduce the overall size of the search tree. This is because using DVO will pick the variable with the smallest domain size (which will translate into less branching at the current node).

Here is a concrete example. This problem has three constraints each with an arity of three. Constraint 1 is on variables 1,2, and 3; Constraint 2 is on variables 2,3, and 4; and Constraint 3 is on variables 3,4, and 5.

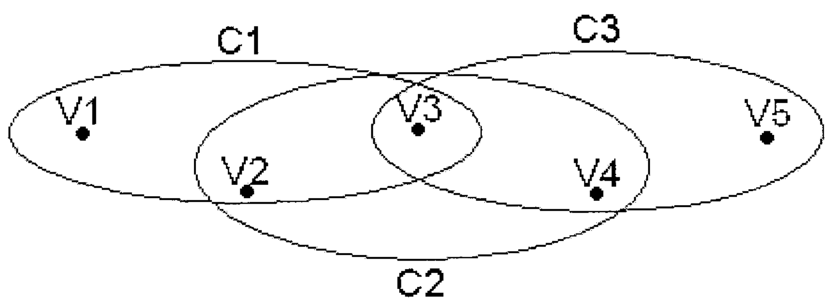


Figure 7: A Sample CSP Graph

Here are the tuples that satisfy the constraints:

C1		
V1	V2	V3
1	2	3
1	1	3
1	3	3

C2		
V2	V3	V4
2	3	4
2	2	4
2	4	4

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

First, all solutions will be found using FC in the dual-graph using DVO, then all solutions will be found using CDFC with a tight-cover.

To find all solutions: Using DVO

1) pick 1,2,3 from C1 (nodes=1)

2) use FC to prune the domains of C2,C3

C1		
V1	V2	V3
1	2	3

C2		
V2	V3	V4
2	3	4

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

3) pick 2,3,4 from C2 (nodes=2)

4) use FC to prune the domain of C3

C1		
V1	V2	V3
1	2	3

C2		
V2	V3	V4
2	3	4

C3		
V3	V4	V5
3	4	5

5) pick 3,4,5 from C3 (Solution found)(nodes=3)

6) backtrack to step 1 to pick another tuple (there are no other choices for C2)

7) pick 1,1,3 from C1 (nodes=4)

8) use FC to prune the domains of C2,C3

C1		
V1	V2	V3
1	1	3

C2		
V2	V3	V4

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2 = empty, backtrack

9) pick 1,3,3 from C1 (nodes=5)

10) use FC to prune the domains of C2,C3

C1		
V1	V2	V3
1	3	3

C2		
V2	V3	V4

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2 = empty, backtrack

11) no more tuples to try for C1.

Results:

1 solution found, 5 nodes in the search tree

Now all solutions will be found for the same problem using CDFC. C1 and C3 form a minimal cover for this problem.

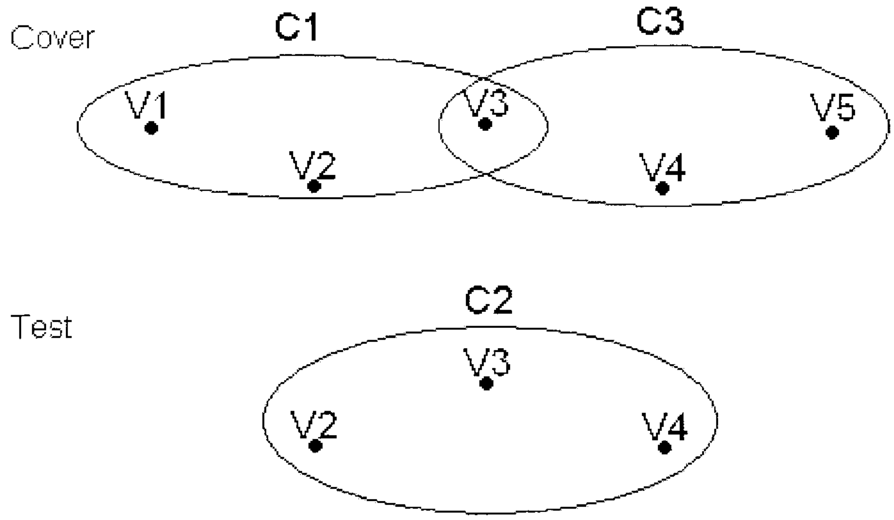


Figure 8: The Sample CSP with a Minimal Cover, and a Constraint for Testing.

Original CSP

C1		
V1	V2	V3
1	2	3
1	1	3
1	3	3

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2(test only)		
V2	V3	V4
2	3	4
2	2	4
2	4	4

1) pick 1,2,3 from C1 (nodes=1)

2) FC doesn't prune anything from C3

C1		
V1	V2	V3
1	2	3

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2(test only)		
V2	V3	V4
2	3	4
2	2	4
2	4	4

3) pick 3,4,5 from C3 (nodes=2)

4) test projection on V2,V3,V4 on C2. 2,3,4 is in C2, so (Solution found)

5) backtrack and pick next tuple from C3

6) pick 3,3,5 from C3 (nodes=3)

7) test projection on V2,V3,V4 on C2. 2,3,3 is not in C2, so backtrack

8) pick 3,5,5 from C3 (nodes=4)

9) test projection on V2,V3,V4 on C2. 2,3,5 is not in C2, so backtrack

10) no more tuples to try in C3, so backtrack to C1

11) pick 1,1,3 from C1 (nodes=5)

12) FC doesn't prune anything from C3

C1		
V1	V2	V3
1	1	3

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2(test only)		
V2	V3	V4
2	3	4
2	2	4
2	4	4

13) pick 3,4,5 from C3 (nodes=6)

14) test projection on V2,V3,V4 on C2. 1,3,4 is not in C2, so backtrack

15) pick 3,3,5 from C3 (nodes=7)

16) test projection on V2,V3,V4 on C2. 1,3,3 is not in C2, so backtrack

17) pick 3,5,5 from C3 (nodes=8)

18) test projection on V2,V3,V4 on C2. 1,3,5 is not in C2, so backtrack

19) no more tuples to try in C3, so backtrack to C1

20) pick 1,3,3 from C1 (nodes=9)

21) FC doesn't prune anything from C3

C1		
V1	V2	V3
1	3	3

C3		
V3	V4	V5
3	4	5
3	3	5
3	5	5

C2(test only)		
V2	V3	V4
2	3	4
2	2	4
2	4	4

22) pick 3,4,5 from C3 (nodes=10)

23) test projection on V2,V3,V4 on C2. 3,3,4 is not in C2, so backtrack

24) pick 3,3,5 from C3 (nodes=11)

25) test projection on V2,V3,V4 on C2. 3,3,3 is not in C2, so backtrack

26) pick 3,5,5 from C3 (nodes=12)

27) test projection on V2,V3,V4 on C2. 3,3,5 is not in C2, so backtrack

28) no more tuples to try for C1.

Results:

1 solution found, 12 nodes in the search tree

So, although using a minimal cover results in a search tree that is more shallow, the overall result is that more nodes have to be searched.

Chapter 4: Methodology and Results

4.1 Methodology

I have solved different random CSPs using forward checking in the primal-graph and using constraint-directed forward checking in the dual-graph. Before solving the CSPs, I made sure that the primal-graphs of the CSPs were connected. If they were not connected, the sub-graphs could have been solved separately. I performed several sets of experiments, where I made several CSPs that have similar characteristics, and varied the constraint tightness (the number of tuples that satisfy each constraint). The problems solved were modelled in both the primal and dual-graph representations. The number of nodes visited, and the run time to find the first solution was recorded, and for all solutions. There were ten problems solved in each problem class. The experiments were carried out on a computer with a 3.0 GHz processor with 512 MB of RAM.

For each problem, the CSP is first solved with FC in the dual-graph using the DVO heuristic. The order that the dual variables were picked to find that solution is then used to create an ordering of primal variables for the FC algorithm to use in the primal-graph. Then the CSP is solved with FC in the primal-graph using the DVO heuristic. Finally, a tight cover is calculated over the dual variables, and that is used to solve the CSP in the dual-graph picking variables only from the cover.

4.2 Results

The complete results are listed in the Appendix. The following are the averages for each class. The domain size for each primal variable in the CSP was 20.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	5268.3	117651.2	31299.6	76430.8
10%	265.4	1921.4	5532.4	1881
15%	307.4	1430.8	1315.2	178115.3
20%	468.5	1441.7	2411.9	685.6
25%	619.6	1179.3	1349.6	14305.5
30%	765.3	1263.4	1441.7	1062.1
35%	998.3	1441.8	1395	1609
40%	1284	1588.5	1516.8	1382.4
45%	1604.1	1685.4	1652.7	1724.6
50%	1941.7	1911.9	1713.5	2904.3

Table 1: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 8 constraints.

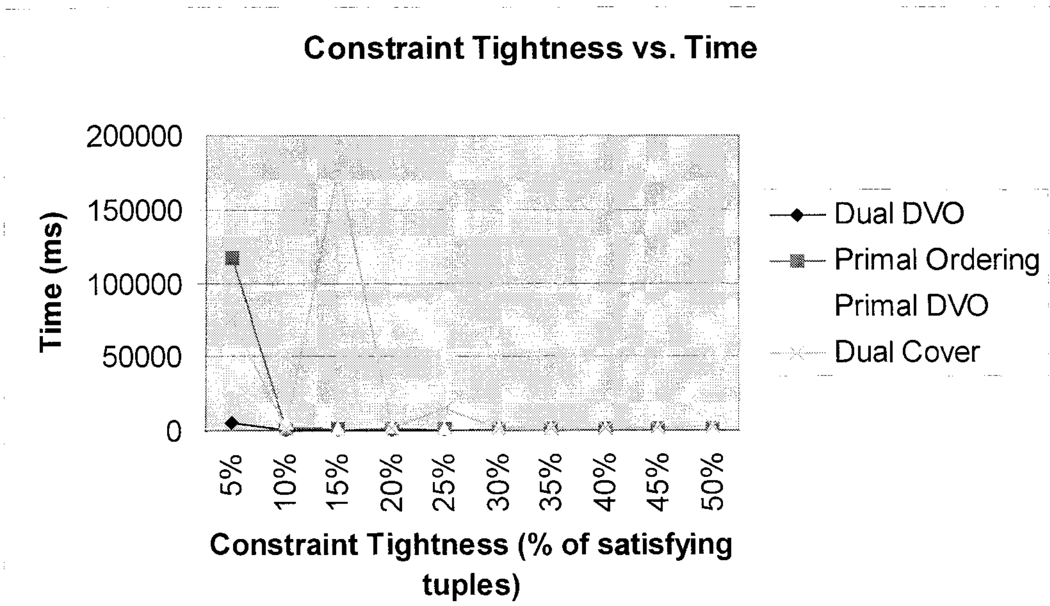


Figure 9: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 8 constraints.

Figure 9 shows that when the constraints are tightest, the FC using the dual-graph with the DVO heuristic performs better than the others. Note the large increase in time for the dual-graph method using the tight cover when 15% of the possible tuples satisfied the constraint. Figure 10 takes a closer view of the same data for constraints that are looser.

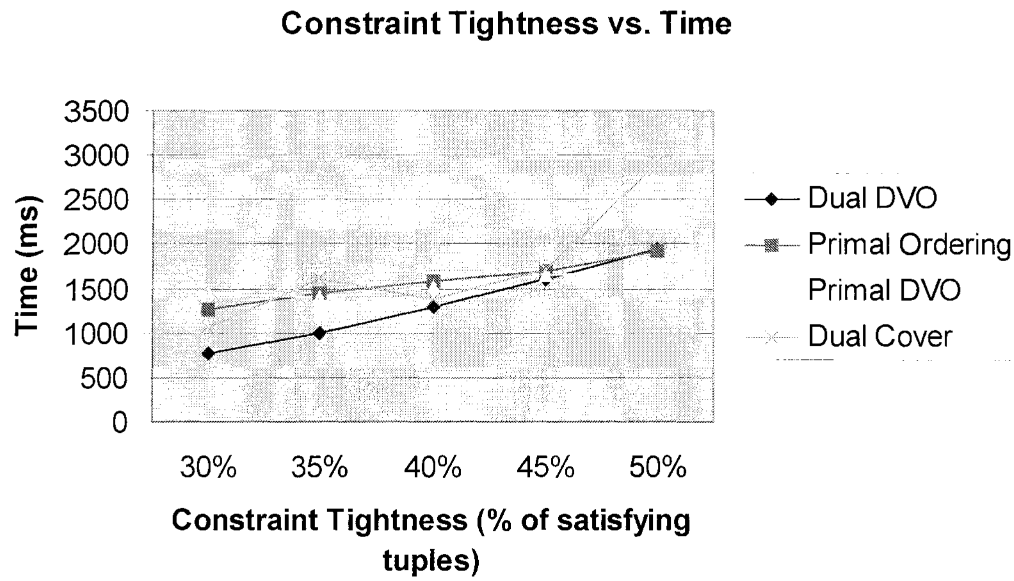


Figure 10: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 8 constraints.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	215680.7	336637.1	1985112	491860.5
10%	5935.6	124698.1	245637.1	47779.1
15%	1179.1	5360.4	12323	22676.2
20%	573	1899.5	3491.7	3637.1
25%	847.8	2273.1	3421.4	2546.4
30%	990.1	1763.5	2141.6	1637.1
35%	1273.1	1710.5	2193.3	1412.1
40%	1591.9	1915.1	2001	1776.2
45%	2034.2	1910.6	1899.4	2115.2
50%	2566.6	1984	2026.2	2727.5

Table 2: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 10 constraints.

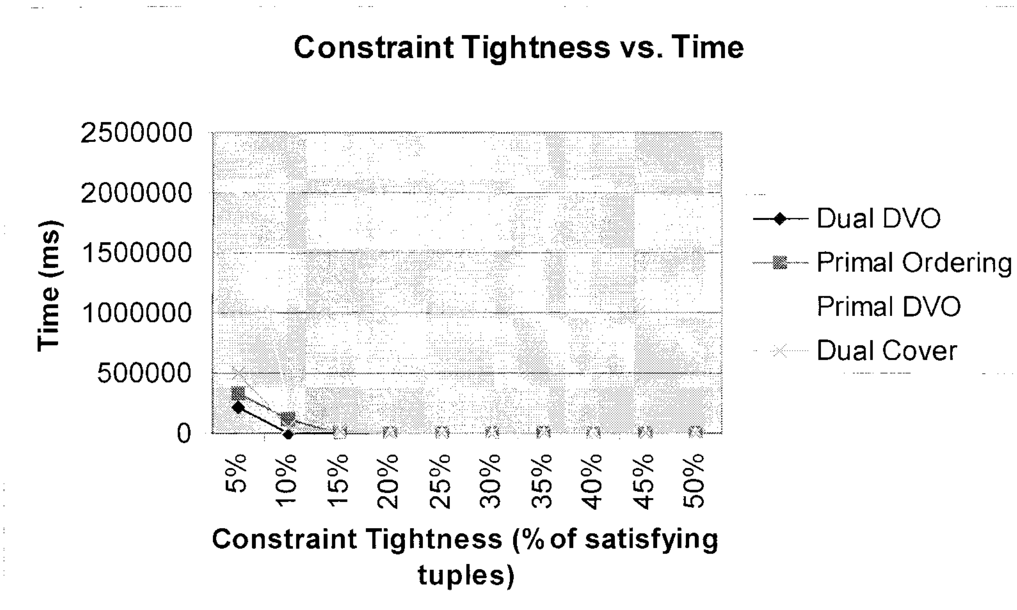


Figure 11: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 10 constraints.

Figures 11 through 14 show similar trends in that the dual-graph method using DVO performs the best when the constraints are tight.

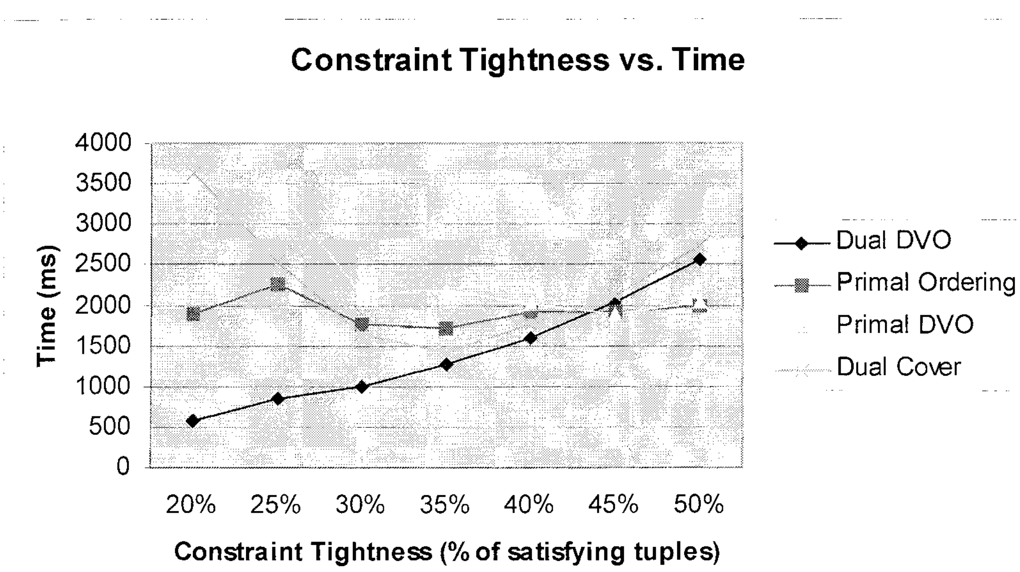


Figure 12: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 10 constraints.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
10%	636704.4	2208462	1365048	18484725
15%	14157.3	101683.9	195285.5	235230.8
20%	2098	5901.2	26930.7	142262.1
25%	1187.1	6363.6	7619.7	146135.5
30%	1263.8	2501.2	6573	6788.7
35%	1629.4	3729.3	5660.5	11385.4
40%	1924.2	2910.5	2123.1	4947.8
45%	2413.6	2318.5	2280.9	4743.2
50%	3072.9	2446.3	2295	3568.3

Table 3: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 12 constraints.

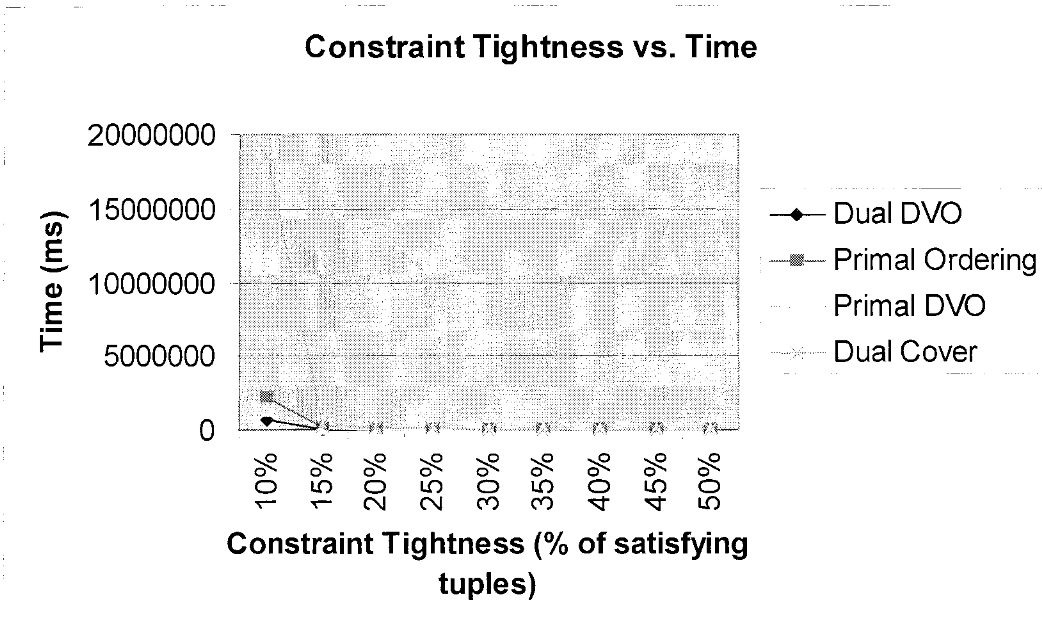


Figure 13: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 12 constraints.

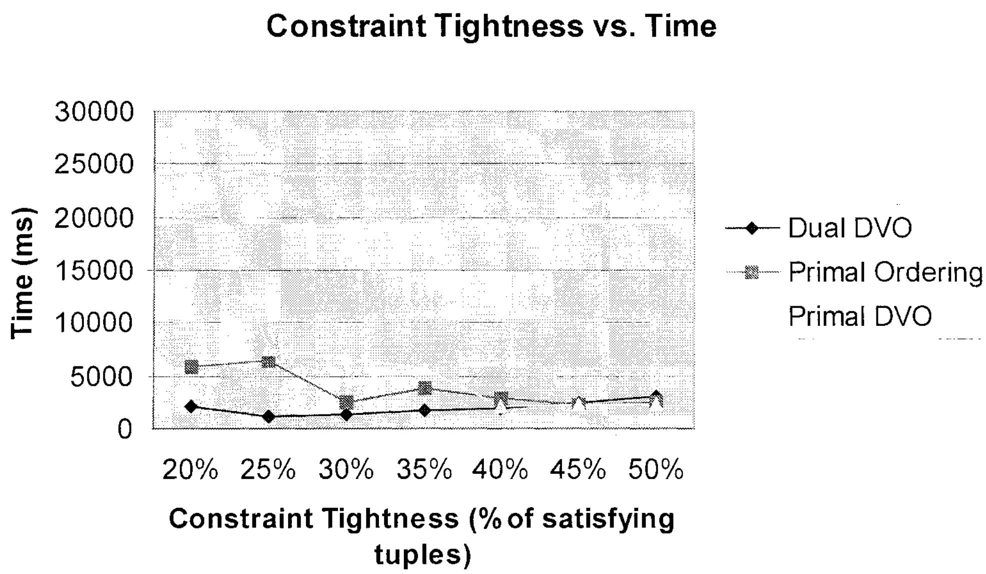


Figure 14: Average time (in ms) to find the first solution on problems with arity = 4, 10 variables, 12 constraints.

Notice that when there are more constraints, the primal method using DVO performs worse in CSPs that are tight.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	1.5	16.6	16.6	17.1
10%	1.5	24.6	26.2	3
15%	6	31	37	10.5
20%	4.5	44.5	37	10.5
25%	9	49.2	49.2	4.5
30%	9	54	50.8	6
35%	6	60.4	58.8	6
40%	10.5	62	65.2	9
45%	12	66.8	65.2	13.5
50%	10.5	73.2	73.2	12

Table 4: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 6 constraints.

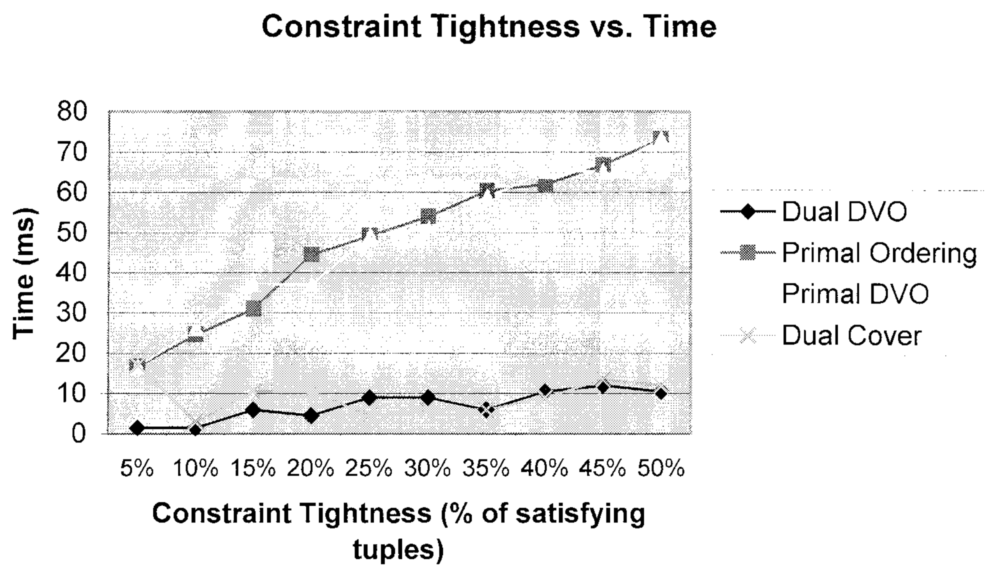


Figure 15: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 6 constraints.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	5268.3	117651.2	31299.6	76430.8
10%	265.4	1921.4	5532.4	1881
15%	307.4	1430.8	1315.2	178115.3
20%	468.5	1441.7	2411.9	685.6
25%	619.6	1179.3	1349.6	14305.5
30%	765.3	1263.4	1441.7	1062.1
35%	998.3	1441.8	1395	1609
40%	1284	1588.5	1516.8	1382.4
45%	1604.1	1685.4	1652.7	1724.6
50%	1941.7	1911.9	1713.5	2904.3

Table 5: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 8 constraints.

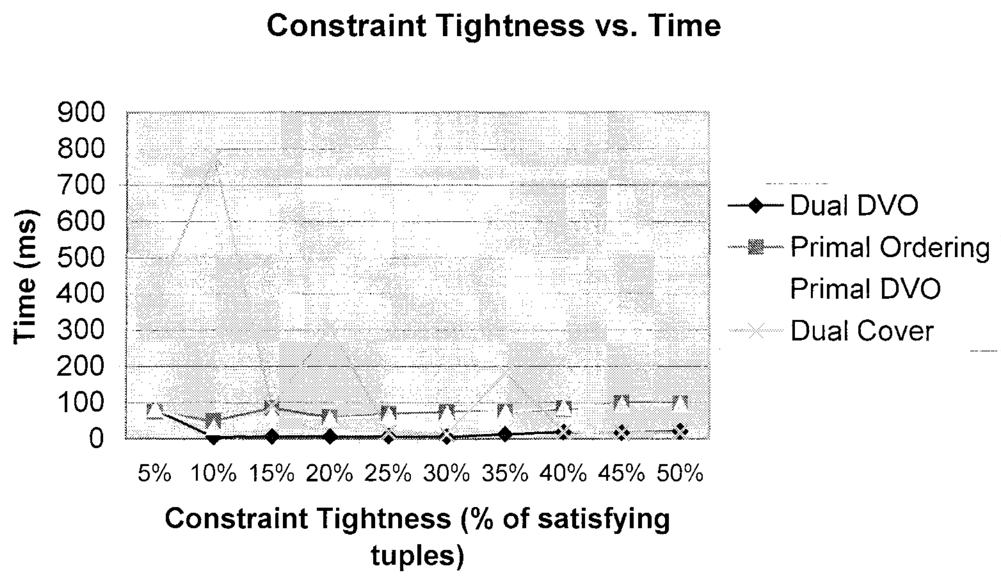


Figure 16: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 8 constraints.

On these small problems, the dual-graph method using DVO, and the two primal methods performed well, but the dual-graph method using the tight cover often takes considerably longer to find the first solution.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	576.1	1326	784.1	2505.7
10%	40.2	144.8	148	5829.3
15%	13.5	97.9	305.7	1266.8
20%	9.1	135.4	105.9	151.1
25%	16.6	101.2	77.6	57.3
30%	10.5	80.9	82.5	82.3
35%	16.6	93.3	91.7	193.3
40%	15	90	94.7	66.7
45%	18.2	99.4	105.8	1796.4
50%	19.8	109	105.8	24.6

Table 6: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 10 constraints.

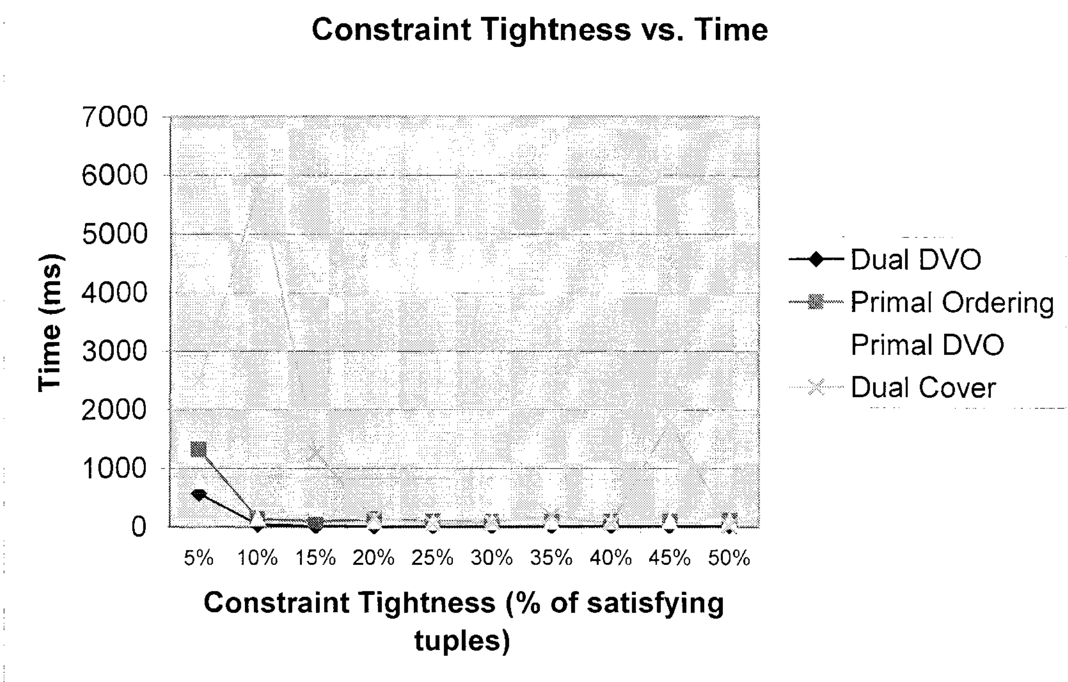


Figure 17: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 10 constraints.

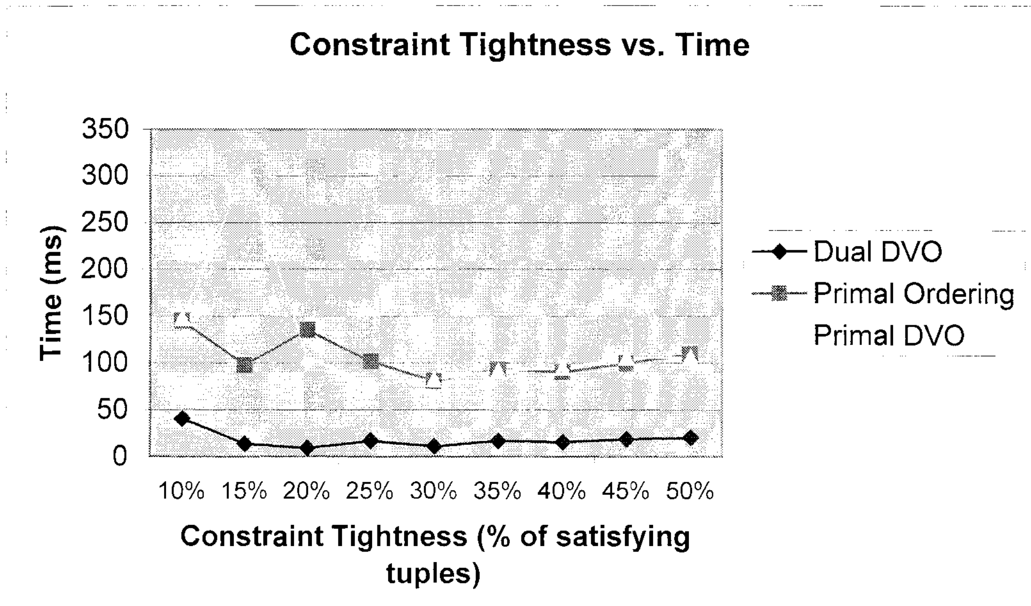


Figure 18: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 10 constraints.

Figure 18 shows a closer look of the data from Figure 17, with the dual-graph using a cover omitted. Once the constraint tightness reaches about 30%, the primal-graph method using DVO finds the first solution almost without backtracking.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
10%	823	2802.7	3041.7	17921.5
15%	185.5	729.1	6048	10713.7
20%	47.9	173.2	209	5415.2
25%	21.4	91.8	107.5	3030.8
30%	15.1	102.8	99.5	99.5
35%	22.9	100.9	101	79.2
40%	19.8	116.8	107.4	34
45%	32.4	115.4	113.8	85.5
50%	26.2	120.2	123.3	37.1

Table 7: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 12 constraints.

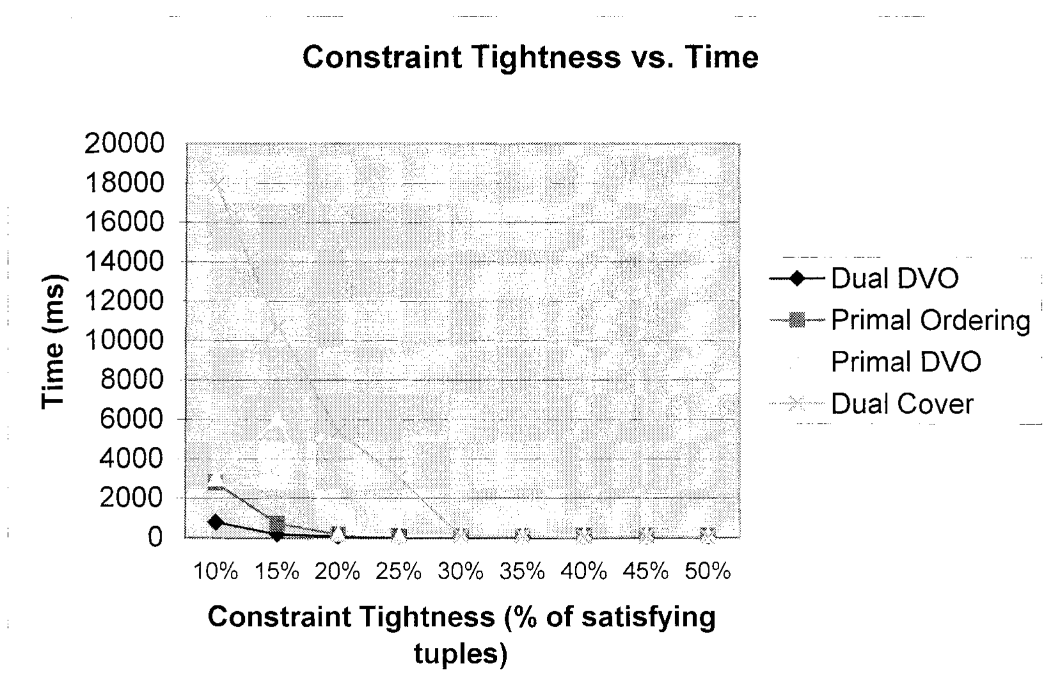


Figure 19: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 12 constraints.

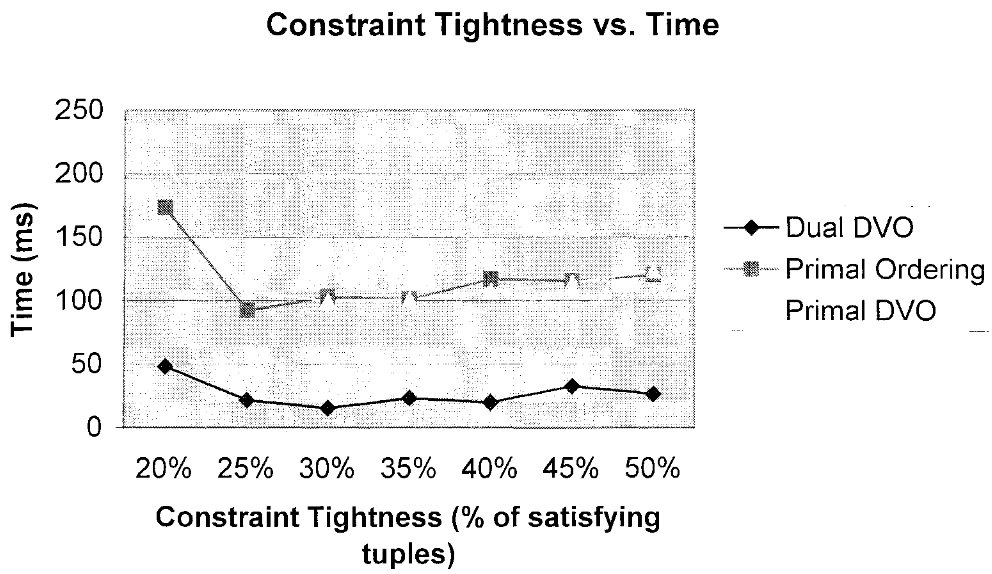


Figure 20: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 12 constraints.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
10%	3857.4	12060.7	10241.6	41062.1
15%	1669.8	3962	9149.6	91233.9
20%	230.6	1541.7	1677.7	25101.1
25%	150.9	773	488.7	10977.7
30%	68.2	3429.3	141.7	122316.8
35%	26	120	137	5071.5
40%	35.5	138.6	127.9	538.6
45%	27.8	123.3	124.8	205.8
50%	31	132.6	132.5	57.4

Table 8: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 14 constraints.

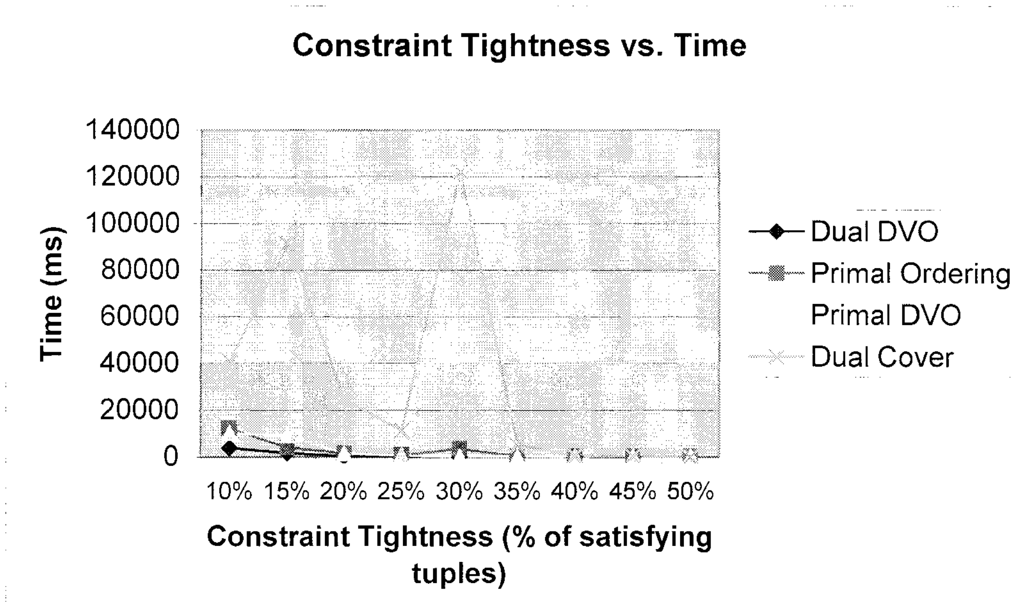


Figure 21: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 14 constraints.

In these larger problems with arity = 3, the dual-graph method with the cover often performs considerably worse than the other methods.

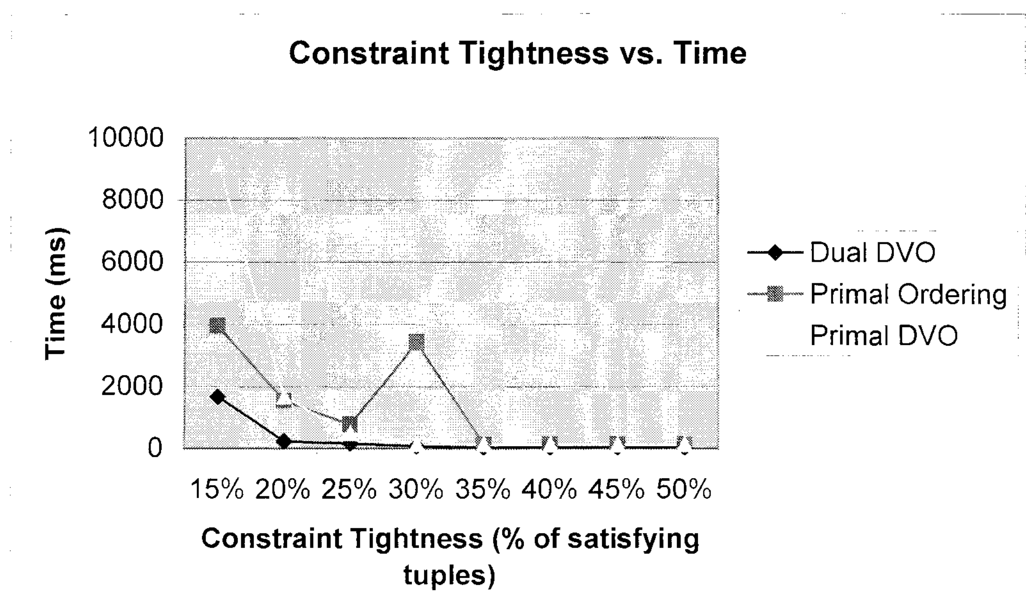


Figure 22: Average time (in ms) to find the first solution on problems with arity = 3, 10 variables, 14 constraints.

Satisfying Tuples	Dual DVO	Primal Ordering	Primal DVO	Dual Cover
5%	42632.4	521090.2	255746.6	109643.4
10%	1336688.778	13351881.44	15785593.22	2948470

Table 9: Average time (in ms) to find all solutions on problems with arity = 3, 10 variables, 6 constraints.

When finding all solutions on tight problems, the dual-graph methods perform better than the primal-graph methods. The better performance of the dual-graph method with the cover over the primal-graph methods (Figure 23) exists even though it visits considerably more nodes in its search-tree (Figure 24).

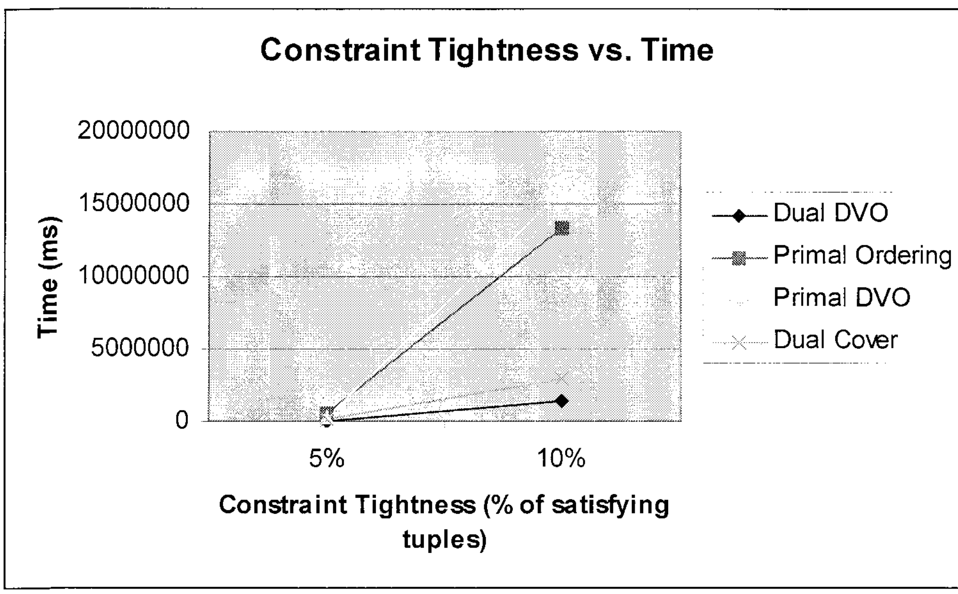


Figure 23: Average time (in ms) to find all solutions on problems with arity = 3, 10 variables, 6 constraints.

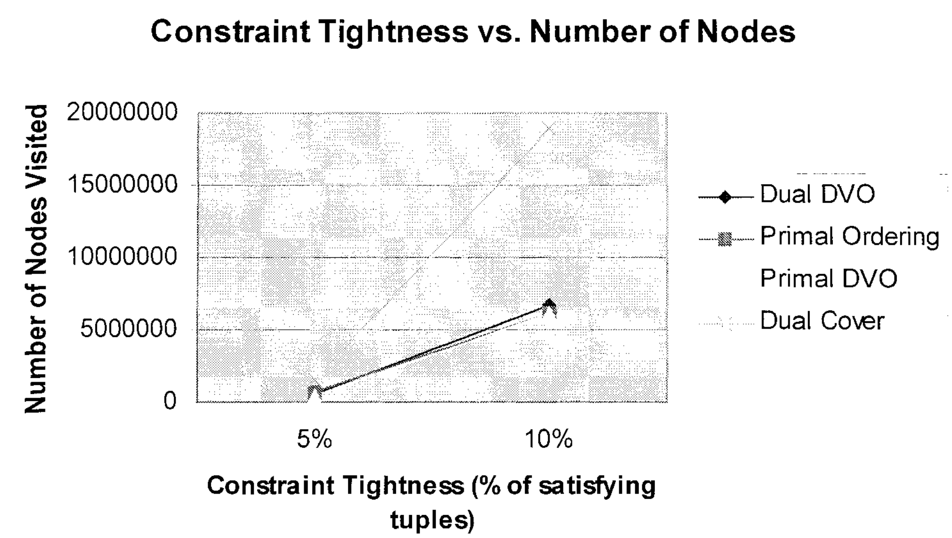


Figure 24: Average number of nodes visited when finding all solutions on problems with arity = 3, 10 variables, 6 constraints.

Chapter 5: Analysis

The results show that on CSPs with tight constraints, FC in the dual-graph often performs better than FC in the primal-graph. Tests were conducted using the t statistic to verify the statistical significance of the results reported – see the Appendix for details. Also, as the constraints became less tight, the algorithms were able to find a solution with less backtracking, even though the time to find the first solution took longer. For example, on problems with arity = 4, domain size = 20, number of variables = 10, number of constraints = 8 using FC in the primal-graph with DVO, we can see that the problems are solved with little or no backtracking (10 nodes is the minimum amount of nodes FC will visit on a CSP with 10 variables), but the time to find the first solution continues to increase. This is because it is taking longer to go through the constraint tables and remove tuples that are not consistent with the partial instantiation of variables at each step.

% of satisfying tuples	Average time (ms)	Average number of nodes visited
35%	1395	10.3
40%	1516.8	10.3
45%	1652.7	10.1
50%	1713.5	10.1

Table 10: Average time (in ms) and number of nodes visited for FC in the primal-graph to find the first solution on problems with arity = 4, 10 variables, 8 constraints.

If only one solution is required for a CSP with constraints that are so loose, perhaps a local search algorithm would perform better.

It was apparent that some of the CSPs were particularly difficult for FC in the dual-graph using a cover. For example, the following tables are from the Appendix from set 3: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 24000 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5586	4791	218	2906	812	781
2	9972	5156	921	1609	1531	1769734
3	2611	7702	234	1328	796	234
4	238	1454	250	890	1000	281
5	1432	5390	218	1187	1750	250
6	9461	2867	234	1812	1437	265
7	4893	2081	250	796	1218	578
8	9202	7944	234	2125	2109	1968
9	3429	7731	234	859	1781	1687
10	6593	7720	281	796	718	5375
		Avg	307.4	1430.8	1315.2	178115.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	24	35	10	5101
2	17	19	14	2870636
3	28	12	10	93
4	74	13	12	973
5	7	13	18	169
6	44	25	17	388
7	8	10	15	4430
8	12	38	27	2946
9	21	11	23	18208
10	14	10	10	2118
Avg	24.9	18.6	15.6	290506.2

Table 11: Set 3 from the Appendix

Problem number 2 fits the description of an exceptionally hard problem (EHP) for FC in the dual-graph using a cover. (See section 2.8 or [SmB95b] for more on EHPs.) This problem took orders of magnitude longer to solve than the next longest problem with similar characteristics, and it was relatively easy for other algorithms to solve. In [SmB95b], it was noted that EHPs occur when an algorithm's first choices for solving a problem cannot lead to a solution, but the algorithm cannot determine that, so it searches through a large area of the tree before backtracking back near the beginning to make different choices. To test if this was what was happening here, I changed the orders of the tuples in the constraint table (specifically, removing the first 50 tuples from each constraint, and appending them to the end). After running 'problem number 2' from set 3 again, I got the following results:

Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
296	1734	1531	328	7	19	14	308

Table 12: Problem 2 from Set 3 run with a different ordering in the constraints

There were other possible EHPs for the FC in the dual-graph with a cover, so I changed the order of the tuples for the following problems and obtained the following results:

Set 46: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 2800 (35%)

Problem number: 6

Time: 1687 ms, next longest time: 31 ms.

	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
Original	15	78	93	1687	7	10	10	17544
After changing order	31	93	78	31	11	10	10	11

Table 13: Problem 6 from Set 46 run with two different orderings in the constraints

Set 58: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 3600 (45%)

Problem number: 6

Time: 17734 ms, next longest time: 62 ms.

	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
Original	15	93	109	17734	14	10	10	128703
After changing order	31	109	109	296	8	10	10	2330

Table 14: Problem 6 from Set 58 run with two different orderings in the constraints

Since there were many of these hard problems occuring with the FC algorithm in the dual-graph with a cover, it appears that the FC algorithm in the dual-graph using dynamic variable ordering should be preferred over the cover since it is less likely to encounter an EHP. This is not that much of a surprise, since in [Kwa95] it was shown that using DVO heuristics in algorithms that solve CSPs using the primal-graph perform better than algorithms that do not use such heuristics.

It also appears that using FC in the dual-graph with a cover would be a poor choice if all of the solutions of a CSP were required. If conditions existed where a certain value picked for one of the first variables would lead to a large sub-tree in the search space to be explored without a solution being found, such as in an EHP, the algorithm would eventually encounter it when looking for all solutions, and would waste time in a sub-tree that it cannot recognise has no solutions.

Finally, although the dual-graph representation appears to perform better than the primal-graph representation, the advantage seems more apparent on problems with higher constraint densities. For example: the following figure shows the average time to find a solution on problems with arity = 4, 10 variables, and each constraint containing 15% of the possible tuples. As more constraints are added to the CSP, the dual representation continues to perform better than the primal representation. Similar results are shown on Figure 25.

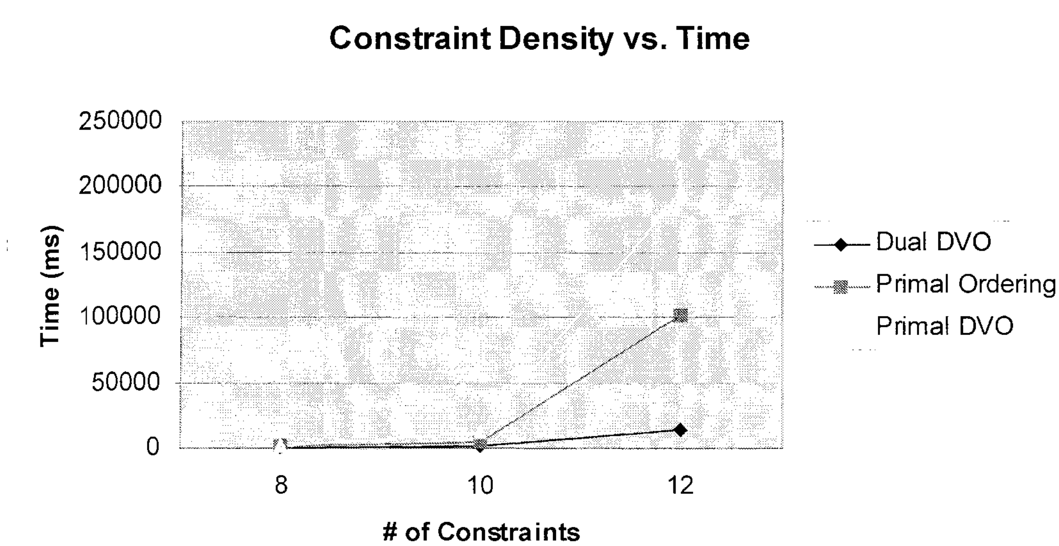


Figure 25: Average time to find the first solution on problems with arity = 4, 10 variables, and each constraint containing 15% of the possible tuples.

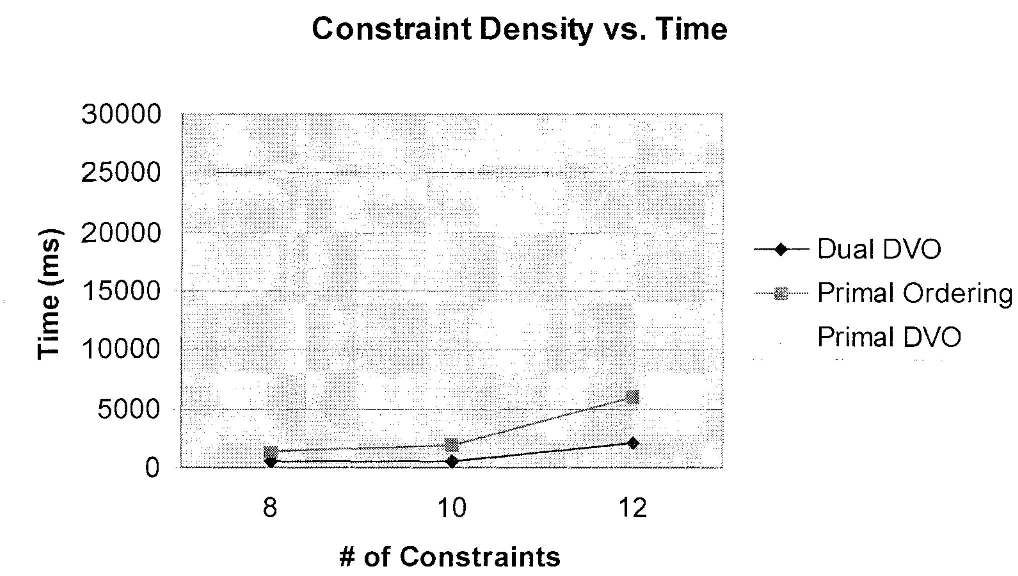


Figure 26: Average time to find the first solution on problems with arity = 4, 10 variables, and each constraint containing 20% of the possible tuples.

The advantage that the dual-graph representation has over the primal-graph representation can be attributed to two things. Some of the advantage that the dual representation has over the primal one comes from the different order that the primal variables are assigned values (the difference between Primal DVO and Primal Ordering). FC in the primal-graph with DVO assigns the variable with the smallest domain next, whereas FC with the ordering will assign all of the primal variables in the same order that they were assigned values from using FC with DVO in the dual-graph representation. The rest of the advantage comes from the different order that values are assigned to the variables (the difference between Primal Ordering and Dual DVO). The dual-graph method assigns the dual variable (or primal constraint) a value from its domain, which amounts to each primal variable associated with that constraint being assigned a value that will satisfy that constraint, whereas the primal-graph method tries to assign each variable the next available value.

Chapter 6: Conclusions

The results appear to show that on CSPs that are very tight, FC using the dual-graph representation performs better than FC using the primal-graph representation. Since there are regions where forward checking using the dual-graph representation performs better than forward checking in the primal-graph representation, this research can be extended to comparing more advanced and state-of-the-art algorithms that are used for solving real-life problems. If we can determine beforehand what algorithm will perform better on a certain CSP by looking at its various properties such as constraint tightness or constraint density, we can pick the algorithm that will perform best for that problem. It is also possible that one algorithm could be used on a subset of the constraints in a problem, and that another algorithm would perform better on the rest of the problem. It has also been shown that using FC in the dual-graph using a dynamic variable ordering heuristic is more preferable than using it with a cover.

Appendix: Experimental Results

The table headings for the following results are:

Problem Number – The problem number of the CSP in this set.

Seed – The seed used to generate the constraint graph.

Tuple Seed – The seed used to generate the tuples for each constraint

Time Dual DVO – The time (in milliseconds) needed to solve the problem using forward checking in the dual-graph using the DVO heuristic to pick the next dual variable to try.

Time Primal Ordering – The time (in milliseconds) needed to solve the CSP using forward checking in the primal-graph using the same variable ordering that was used to find the solution using forward checking in the dual-graph with the DVO heuristic.

Time Primal DVO – The time (in milliseconds) needed to solve the problem using forward checking in the primal-graph using the DVO heuristic to pick the next primal variable to try.

Time Dual Cover – The time (in milliseconds) needed to solve the problem using forward checking in the dual-graph using a tight constraint cover.

Nodes Dual DVO – The number of nodes in the search tree to solve the problem using forward checking in the dual-graph using the DVO heuristic to pick the next dual variable to try.

Nodes Primal Ordering – The number of nodes in the search tree to solve the CSP using forward checking in the primal-graph using the same variable ordering that was used to find the solution using forward checking in the dual-graph with the DVO heuristic.

Nodes Primal DVO – The number of nodes in the search tree to solve the problem using forward checking in the primal-graph using the DVO heuristic to pick the next primal variable to try.

Nodes Dual Cover – The number of nodes in the search tree to solve the problem using forward checking in the dual-graph using a tight constraint cover.

Note: The nodes in the primal representation are not the same as the nodes in the dual representation

First Solution:

Set 1: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 8000 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4807	8228	1562	14953	656	1812
2	2062	2419	6375	917734	31968	86906
3	512	7998	3750	8250	115093	38390
4	5045	6163	4718	85437	58359	478265
5	598	5894	4062	18671	13250	5156
6	1020	3653	2171	2328	4578	2546
7	9442	650	7468	80140	58234	11109
8	724	5214	781	4515	781	3375
9	649	6312	2921	37234	19234	43906
10	1717	800	18875	7250	10843	92843
		Avg	5268.3	117651.2	31299.6	76430.8

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	552	964	19	4010
2	1022	17878	767	64431
3	228	227	3578	146722
4	2765	2721	1467	3324278
5	1735	514	412	14928
6	196	63	136	7243
7	6594	1352	1271	13278
8	697	147	22	18069
9	1197	1146	436	50895
10	4526	253	355	316060
Avg	1951.2	2526.5	846.3	395991.4

Calculations for set 1:

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)

$\bar{x}_1$ = 5268.3 ms = 5.2583 s

$\bar{x}_2$ = 31299.6 ms = 31.299.6 s

$$S_p^2 = \frac{\sum(x_1-\bar{x}_1)^2 + \sum(x_2-\bar{x}_2)^2}{n_1+n_2-2}$$

$$= \frac{\begin{gathered}(1.562-5.2683)^2+(6.375-5.2683)^2+(3.750-5.2683)^2+(4.718-5.2683)^2+(4.062-5.2683)^2\\ +(2.171-5.2683)^2+(7.468-5.2683)^2+(0.781-5.2683)^2+(2.921-5.2683)^2+(18.875-5.2683)^2\\ +(0.656-31.2996)^2+(31.968-31.2996)^2+(115.093-31.2996)^2+(58.359-31.2996)^2+(13.250-31.2996)^2\\ +(4.578-31.2996)^2+(58.234-31.2996)^2+(0.781-31.2996)^2+(19.234-31.2996)^2+(10.843-31.2996)^2\end{gathered}}{10+10-2}$$

=(13.73665969+1.22478489+2.30523489+0.30283009+1.45515969
+9.59326729+4.83868009+20.13586129+5.50981729+185.14228489
+939.03022096+0.44675856+7021.33388356+732.21112836+325.78806016
+714.04390656+725.46190336+931.38494596+145.57870336+418.47248356)/18

$s_p^2 \cong 677.6665$

For the small-sample test statistic testing the null hypothesis $H_0:(\mu_1-\mu_2)=0$
(That there will be no difference in the mean run times for these two algorithms)

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{s_p^2\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}}$$

$$= \frac{(5.2683 - 31.2996)}{\sqrt{677.6665\left(\frac{1}{10}+\frac{1}{10}\right)}}$$

$= -2.2360$

The rejection region will be two-tailed and based on a t-distribution with 18 degrees of freedom (10+10-2). For α=0.05, the rejection region for the test is $t < -t_{\alpha/2}$ or $t > t_{\alpha/2}$ $t<-2.101$ or $t>2.101$. Since the observed value of t falls in the rejection region, the test results are statistically significant at the α=0.05 level of significance. Since the rejection is in the negative tail of the t-distribution, it appears that FC in the dual-graph performs better than FC in the primal-graph in for these problems.

Similar calculations were performed on the other sets of data. The calculations were not performed when one or more problems in a set were solved in <15 milliseconds.

Other t-values that were used:
For α=0.1, the rejection region for the test is $t < -t_{\alpha/2}$ or $t > t_{\alpha/2}$: $t<-1.734$ or $t>1.734$.
For α=0.01, the rejection region for the test is $t < -t_{\alpha/2}$ or $t > t_{\alpha/2}$: $t<-2.878$ or $t>2.878$.
For α=0.001, the rejection region for the test is $t < -t_{\alpha/2}$ or $t > t_{\alpha/2}$: $t<-3.922$ or $t>3.922$.

Set 2: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 16000 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	2376	8801	312	1171	2703	1453
2	5796	4595	125	1593	578	453
3	5490	395	234	609	10265	3125
4	8431	9663	484	734	1359	5500
5	5825	1151	125	1765	2703	109
6	3096	1205	328	2687	1687	312

7	5871	7714	109	812	14687	2281
8	8893	9909	484	906	703	5093
9	5440	28	125	6984	1421	156
10	9446	6143	328	1953	19218	328
		Avg	265.4	1921.4	5532.4	1881

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	349	26	52	10844
2	9	32	10	940
3	21	10	150	33944
4	313	20	24	59693
5	8	46	26	11
6	200	52	25	1252
7	8	12	229	1630
8	69	15	12	14820
9	99	207	20	968
10	89	39	315	585
Avg	116.5	45.9	86.3	12468.7

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.47776
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 3: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 24000 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5586	4791	218	2906	812	781
2	9972	5156	921	1609	1531	1769734
3	2611	7702	234	1328	796	234
4	238	1454	250	890	1000	281
5	1432	5390	218	1187	1750	250
6	9461	2867	234	1812	1437	265
7	4893	2081	250	796	1218	578
8	9202	7944	234	2125	2109	1968
9	3429	7731	234	859	1781	1687
10	6593	7720	281	796	718	5375
		Avg	307.4	1430.8	1315.2	178115.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	24	35	10	5101
2	17	19	14	2870636
3	28	12	10	93

4	74	13	12	973
5	7	13	18	169
6	44	25	17	388
7	8	10	15	4430
8	12	38	27	2946
9	21	11	23	18208
10	14	10	10	2118
Avg	24.9	18.6	15.6	290506.2

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -6.03824
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 4: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 32000 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	239	8841	375	1265	2468	375
2	2299	7220	375	937	968	437
3	306	8732	375	1312	1343	421
4	8179	7322	359	1015	1312	375
5	4854	2523	421	1968	1546	625
6	9075	2037	359	1593	11187	375
7	7452	5623	375	1250	953	1562
8	9528	4898	375	1421	1156	843
9	4894	8572	1312	2625	2171	1312
10	2203	2322	359	1031	1015	531
		Avg	468.5	1441.7	2411.9	685.6

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	10	12	31	58
2	24	10	11	1876
3	23	15	15	284
4	9	10	13	18
5	115	16	27	2065
6	15	22	107	96
7	19	15	10	11534
8	7	13	11	5180
9	199	26	18	174
10	29	12	10	283
Avg	45	15.1	25.3	2156.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)

t= -1.95758

FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 5: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 40000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	333	1287	546	1234	1750	546
2	2724	3206	546	1125	1140	546
3	1734	7797	546	1218	1390	546
4	8715	2536	531	1218	1125	546
5	6692	9408	546	1125	1109	546
6	431	9633	531	1109	1625	546
7	8186	4197	546	1187	1171	546
8	1206	1717	546	1250	1140	578
9	4558	1969	546	1156	1859	718
10	4421	5359	1312	1171	1187	137937
		Avg	619.6	1179.3	1349.6	14305.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	11	14	8
2	10	16	10	34
3	15	10	11	10
4	7	10	10	40
5	13	10	10	59
6	7	10	13	63
7	8	13	11	53
8	12	10	10	35
9	9	10	15	2252
10	22	11	11	2039
Avg	11	11.1	11.5	459.3

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)

t= -6.10966

FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 6: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 48000 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7082	3256	781	1218	1515	765
2	683	8966	765	1437	1296	765

3	4205	5958	781	1343	1281	781
4	3643	9418	781	1187	1281	765
5	4416	8389	750	1296	1203	781
6	4405	8432	765	1218	1687	859
7	9117	2088	750	1203	2421	765
8	8906	6981	750	1296	1156	750
9	9808	3198	765	1140	1312	765
10	8355	2099	765	1296	1265	3625
		Avg	765.3	1263.4	1441.7	1062.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	15	11	12	10
2	7	14	10	115
3	14	10	12	53
4	9	10	10	36
5	8	10	10	24
6	11	10	11	68
7	7	10	14	13
8	7	12	10	16
9	8	10	14	6
10	6	10	11	1443
Avg	9.2	10.7	11.4	178.4

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -5.65955
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 7: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 56000 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	2770	3885	1000	1281	1484	1296
2	225	4892	1000	1953	1156	1000
3	2408	3312	984	1515	1328	1250
4	6226	1004	1000	1406	1656	1156
5	208	4685	984	1437	1250	1484
6	3652	8580	1000	1343	1390	5671
7	6941	3014	1000	1359	1562	1140
8	7840	466	1031	1390	1406	1031
9	7186	32	984	1375	1328	1078
10	5465	4079	1000	1359	1390	984
		Avg	998.3	1441.8	1395	1609

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	15	10	11	14
2	16	13	10	104
3	9	10	10	8
4	8	10	11	14
5	6	10	10	9
6	8	10	10	1049
7	9	10	11	31
8	6	10	10	20
9	7	10	10	94
10	16	10	10	26
Avg	10	10.3	10.3	136.9

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -8.56817
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 8: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 64000 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7101	9375	1281	1515	1515	1296
2	3001	4684	1281	1765	1812	1281
3	7597	6869	1265	1640	1500	1296
4	1156	7962	1281	1593	1453	1265
5	4201	5846	1296	1796	1656	1312
6	6285	2986	1281	1531	1703	2250
7	6566	7504	1281	1500	1343	1281
8	2659	3148	1312	1468	1531	1281
9	3530	4794	1281	1484	1390	1281
10	1505	616	1281	1593	1265	1281
		Avg	1284	1588.5	1516.8	1382.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	10
2	7	10	11	11
3	11	10	10	37
4	6	10	10	71
5	7	11	11	9
6	9	10	11	53
7	7	10	10	6
8	14	10	10	14
9	8	10	10	7

10	9	10	10	8
Avg	8.5	10.1	10.3	22.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -4.3583
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 9: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 72000 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5110	4517	1593	1546	1656	1609
2	2342	4149	1593	1750	1687	2546
3	7814	3507	1593	1687	1687	1593
4	4302	2702	1609	1812	1703	1625
5	7349	3554	1640	1671	1546	1859
6	6029	7989	1593	1671	1562	1578
7	1288	8018	1609	1687	1609	1609
8	2497	4118	1578	1671	1703	1578
9	5996	4097	1640	1609	1640	1671
10	5472	3629	1593	1750	1734	1578
		Avg	1604.1	1685.4	1652.7	1724.6

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	13
2	13	10	10	13
3	7	10	10	9
4	8	10	11	10
5	7	10	10	89
6	6	10	10	71
7	7	10	10	19
8	6	10	10	14
9	8	10	10	11
10	7	10	10	9
Avg	7.5	10	10.1	25.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.31577
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 10: arity = 4, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 80000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9610	1509	1953	1953	1718	8703
2	3628	7155	1953	1796	1531	1968
3	9301	7491	1968	1796	1515	2000
4	1375	7997	1921	1984	1609	1968
5	7680	3451	1937	1796	1796	3468
6	6042	71	1953	2078	1718	1953
7	8394	3639	1921	1843	1859	2906
8	5469	1912	1937	1968	1890	1937
9	7687	5796	1937	1968	1593	2125
10	6902	361	1937	1937	1906	2015
		Avg	1941.7	1911.9	1713.5	2904.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	35
2	6	10	10	217
3	7	10	10	12
4	7	10	10	11
5	8	10	10	13
6	7	10	10	14
7	9	10	10	7
8	8	10	10	15
9	7	10	10	50
10	8	10	11	25
Avg	7.4	10	10.1	39.9

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= 4.885085
FC in the dual-graph performs worse than FC in the primal-graph in for these problems. (α=0.001)

Set 11: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 8000 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1406	573	333484	444500	248687	342250
2	7847	1810	229093	446515	5269359	542187
3	2346	1349	64234	306062	27703	755203
4	569	4228	278093	352437	10399937	404937
5	6053	2633	19468	115531	144593	20046
6	3616	2052	190171	118140	375328	241078
7	7225	7078	335343	6031	23796	357187
8	8006	9474	28625	298375	312890	29453

9	9955	3660	214312	609390	205468	240593
10	4780	5971	463984	669390	2843359	1985671
		Avg	215680.7	336637.1	1985112	491860.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12734	7617	5491	87844
2	13987	12075	114325	858950
3	25181	12527	533	5167338
4	16089	7971	216664	490328
5	1079	3266	3516	6673
6	8055	2234	11920	28098
7	21691	131	446	307056
8	1815	6007	8853	25828
9	91336	14319	3862	494378
10	162139	15347	58562	3073041
Avg	35410.6	8149.4	42417.2	1053953.4

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.63694
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 12: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 16000 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3885	1642	2750	28000	32671	20718
2	5022	3206	13375	4453	104328	25140
3	7190	1211	3546	3781	18281	232296
4	8624	2244	687	681687	40671	3156
5	9792	7174	8875	53218	1610500	9843
6	7012	6934	16843	56125	4531	23687
7	9457	712	3328	242515	13453	11390
8	84	8820	4625	166562	534890	140093
9	8003	6344	1734	4234	14000	3125
10	3602	5492	3593	6406	83046	8343
		Avg	5935.6	124698.1	245637.1	47779.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	182	271	341	9057
2	1158	51	905	43440
3	66	73	167	73842
4	377	20204	351	34574
5	370	711	17674	10803

6	12821	898	62	163498
7	486	3561	176	3795
8	2476	2064	7797	345390
9	911	62	167	22766
10	204	86	1127	14253
Avg	1905.1	2798.1	2876.7	72141.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.50011
FC in the dual-graph performs about as well as than FC in the primal-graph in for these problems. (α=0.1)

Set 13: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 24000 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4203	6741	312	1265	1765	328
2	6580	2844	2171	1390	1359	16000
3	4348	3857	593	9921	7828	968
4	8794	1897	3843	3046	9734	1125
5	4504	2661	2234	6062	39359	17828
6	72	8575	281	4000	7796	2171
7	3859	8262	265	7796	27546	12359
8	523	640	812	2859	16609	2578
9	2124	8458	671	6000	10234	45968
10	8554	275	609	11265	1000	127437
		Avg	1179.1	5360.4	12323	22676.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	23	15	20	56
2	111	24	15	96318
3	56	77	70	1479
4	718	60	86	5359
5	127	65	762	16728
6	11	40	83	7698
7	19	162	200	1048
8	71	33	202	2465
9	151	50	120	65663
10	54	97	12	87932
Avg	134.1	62.3	157	28474.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.8201
FC in the dual-graph performs better than FC in the primal-graph in for these problems.

(α=0.05)

Set 14: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 32000 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1148	3848	593	3906	1625	2312
2	8053	250	921	1062	2484	1000
3	7590	5437	500	1156	8125	4625
4	9456	6856	578	2796	3718	1421
5	3548	6896	593	2125	5562	16265
6	7251	9323	453	1218	3671	968
7	900	1700	468	1843	1687	3062
8	1014	2129	453	1312	4312	609
9	4441	9994	703	2390	2515	5328
10	5522	5291	468	1187	1218	781
		Avg	573	1899.5	3491.7	3637.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	17	38	14	18592
2	14	10	16	315
3	48	11	52	1899
4	67	29	27	4437
5	142	20	46	10901
6	13	13	30	4521
7	34	27	14	24747
8	34	12	31	1473
9	32	20	22	255
10	24	12	10	2647
Avg	42.5	19.2	26.2	6978.7

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -4.34636
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.001)

Set 15: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 40000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1532	9193	687	1421	1359	750
2	605	120	687	2609	1656	1171
3	1914	5767	1296	1531	19171	1750

4	7802	7379	671	2125	2265	687
5	8224	9503	1437	1203	1515	968
6	3890	8924	687	4593	1250	890
7	3948	7137	984	2625	1312	1546
8	3473	7727	671	2640	2234	12296
9	8931	7604	671	1203	1281	781
10	6060	1712	687	2781	2171	4625
		Avg	847.8	2273.1	3421.4	2546.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12	12	13	962
2	110	24	14	4261
3	11	14	178	89
4	13	23	16	120
5	152	11	11	248
6	38	29	10	1211
7	44	19	10	970
8	8	18	17	44465
9	13	10	11	1466
10	48	19	15	19784
Avg	44.9	17.9	29.5	7357.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.46466
FC in the dual-graph performs about as well as than FC in the primal-graph in for these problems. (α=0.1)

Set 16: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 48000 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1524	4230	1125	1468	3296	953
2	9935	7076	953	1468	1296	1390
3	2853	783	968	2390	2234	953
4	5708	9573	937	1531	1515	953
5	4895	4741	1046	2593	3921	2656
6	5314	1827	921	1468	1437	1296
7	8329	4192	937	1453	1671	1109
8	7316	3763	937	1921	2343	2468
9	9454	8641	1140	1859	1625	3265
10	1008	2271	937	1484	2078	1328
		Avg	990.1	1763.5	2141.6	1637.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover

1	11	10	16	53
2	16	11	11	241
3	37	15	16	39
4	11	11	11	65
5	21	20	29	2829
6	10	10	10	107
7	10	10	11	1804
8	9	12	14	548
9	8	18	12	68
10	10	10	16	237
Avg	14.3	12.7	14.6	599.1

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -4.21838
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 17: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 56000 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7083	3948	1265	1734	1625	1328
2	538	9657	1250	1609	1593	1953
3	993	9016	1296	1734	3953	1296
4	4284	986	1328	1484	1437	1500
5	215	6307	1328	1546	1593	1296
6	7696	8976	1312	2109	4515	1500
7	7091	8511	1265	1578	1437	1265
8	1656	1110	1234	2031	2093	1281
9	9569	194	1203	1640	2109	1218
10	183	5283	1250	1640	1578	1484
		Avg	1273.1	1710.5	2193.3	1412.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	55	15	10	598
2	20	10	10	1215
3	8	10	19	218
4	34	10	10	2711
5	6	10	11	29
6	9	13	21	23
7	12	11	10	21
8	9	14	14	66
9	27	10	12	24
10	11	10	10	59
Avg	19.1	11.3	12.7	496.4

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.6219
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 18: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 64000 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1827	8635	1578	1984	1843	1625
2	8681	336	1578	1765	1859	1640
3	9851	4891	1578	2796	1500	1609
4	4675	7771	1625	1718	2687	3250
5	2844	6139	1593	1781	1890	1703
6	9922	7256	1609	1765	1640	1578
7	6146	5413	1609	1890	1671	1593
8	5777	2298	1578	1890	1640	1578
9	6529	2168	1578	1812	1718	1593
10	1242	4016	1593	1750	3562	1593
		Avg	1591.9	1915.1	2001	1776.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12	12	10	325
2	22	13	10	505
3	10	13	10	66
4	12	10	13	57
5	7	10	11	490
6	8	10	10	75
7	11	10	10	131
8	11	10	10	19
9	16	10	10	39
10	8	10	16	9
Avg	11.7	10.8	11	171.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.02644
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 19: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 72000 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4983	1953	2390	1656	2031	2062
2	5211	3385	2000	1843	2046	1968
3	8948	735	2000	2125	1843	2000
4	5504	8809	2000	1859	1718	1984
5	6151	1927	2000	2093	1921	1984
6	435	2908	2000	2000	2109	2062
7	8615	4312	1984	1859	1890	3156
8	3772	7409	2000	1734	1812	1984
9	2070	2221	1984	2000	1734	1984
10	1822	544	1984	1937	1890	1968
		Avg	2034.2	1910.6	1899.4	2115.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	15	10	10	28
2	15	10	10	5
3	13	11	10	52
4	32	10	10	35
5	16	12	10	17
6	11	10	10	41
7	8	10	10	395
8	17	10	10	21
9	9	10	10	96
10	8	10	10	38
Avg	14.4	10.3	10	72.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= 2.351958
FC in the dual-graph performs worse than FC in the primal-graph in for these problems. (α=0.05)

Set 20: arity = 4, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 80000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	6280	347	2468	1796	2015	2437
2	504	6884	2437	2000	2000	2421
3	7972	5716	2421	2031	2062	2421
4	3809	4111	2421	2250	2062	2437
5	3151	8580	2421	1781	2203	2437
6	9131	2229	3718	2109	2078	3390
7	5149	8033	2437	2031	1796	2468
8	4341	7123	2453	1984	1968	4406
9	8406	1977	2437	1937	2000	2421

10	4468	4667	2453	1921	2078	2437
		Avg	2566.6	1984	2026.2	2727.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	23	10	10	9
2	10	10	10	39
3	7	10	10	6
4	7	10	10	14
5	9	10	11	54
6	13	10	10	283
7	7	10	10	77
8	10	10	10	18
9	10	10	10	23
10	8	10	10	24
Avg	10.4	10	10.1	54.7

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= 4.088538
FC in the dual-graph performs worse than FC in the primal-graph in for these problems. (α=0.001)

Set 21: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 16000 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5486	8665	17875	481703	868859	20000
2	8907	7639	181437	92765	1213437	1424968
3	3309	2240	67453	9696187	569953	158250
4	9682	5245	2653078	1304187	548468	84970046
5	7265	9724	32375	739265	2909796	3726546
6	7319	8849	565156	538406	955062	2068328
7	7961	3079	1559671	1253906	4561250	1695531
8	2770	9382	653281	1419250	204578	64399468
9	1221	6213	90203	6248718	1673937	293406
10	7495	8925	546515	310234	145140	26090703
		Avg	636704.4	2208462	1365048	18484725

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	704	7141	9513	20114
2	15764	1145	11909	938164
3	29178	175424	7929	677559
4	212244	16368	4780	12335820
5	817	13383	36702	251709
6	7770	4145	9615	2265296

7	23888	13381	38849	551208
8	190693	21420	2832	66763630
9	10900	88995	18816	108194
10	52478	3082	1448	4053289
Avg	54443.6	34448.4	14239.3	8796498

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-1.41701
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 22: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 24000 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	136	3388	23906	43937	4281	36093
2	9621	8315	8859	251843	32796	241718
3	5433	6704	21031	16437	16109	40765
4	5458	5620	20093	28109	10156	176296
5	3383	3329	8468	2828	11109	16000
6	1618	3795	32296	419187	334703	226906
7	9797	3275	2828	202187	199234	33265
8	821	3184	8968	1515	292765	289250
9	3881	2185	6281	19421	8921	57609
10	726	5147	8843	31375	1042781	1234406
		Avg	14157.3	101683.9	195285.5	235230.8

Problem Number		Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
	1	10706	384	33	12681
	2	4148	2625	303	2242786
	3	2377	135	146	220171
	4	645	262	86	244536
	5	377	43	102	4916
	6	5874	5004	2483	329234
	7	974	1802	1732	72378
	8	2821	16	2160	161776
	9	280	206	92	77139
	10	750	323	7945	40209
Avg		2895.2	1080	1508.2	340582.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-1.76676
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 23: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 32000 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1461	5550	3296	10875	167687	10390
2	5228	8889	3656	5796	9734	57703
3	3243	5624	1062	6906	10359	259734
4	1228	8713	1593	6500	1187	3781
5	8297	7127	703	1656	34796	46375
6	639	5391	4625	3812	11968	136546
7	8024	6794	609	5796	7828	3656
8	2516	4429	562	4609	6171	781
9	3352	6087	1156	9375	6468	4859
10	2017	406	3718	3687	13109	898796
		Avg	2098	5901.2	26930.7	142262.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	335	157	1354	18112
2	726	119	68	16675
3	587	88	113	174638
4	1188	40	12	12307
5	61	14	249	122315
6	1010	27	92	162720
7	104	72	43	893
8	18	54	56	2332
9	118	78	69	5189
10	1087	33	104	760951
Avg	523.4	68.2	216	127613.2

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-1.56152
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=01.)

Set 24: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 40000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	8083	2878	843	12031	3171	10640
2	9156	9676	890	11390	2015	220531
3	9574	8180	1468	4375	3546	3203
4	6183	7903	2390	23546	14687	10390
5	8873	695	828	1531	10531	890

6	9205	2389	828	1843	1812	9312
7	5858	5016	1703	4218	1671	2140
8	2795	3074	1265	1359	2078	597578
9	268	963	828	2015	23468	605437
10	8906	8310	828	1328	13218	1234
		Avg	1187.1	6363.6	7619.7	146135.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	74	75	21	5014
2	164	123	19	450200
3	51	21	27	7348
4	852	289	77	11848
5	10	10	58	427
6	15	15	13	21155
7	105	29	12	712
8	172	10	15	825088
9	61	17	113	428602
10	94	10	115	6101
Avg	159.8	59.9	47	175649.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-2.69938
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 25: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 48000 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	8733	3330	1156	2328	3250	8765
2	8802	9336	1250	1437	2812	1375
3	3606	4576	1156	1625	9843	2031
4	7732	4402	2078	1546	15109	2359
5	5633	868	1187	1703	23296	1359
6	5446	4034	1125	1781	1593	26343
7	2558	6098	1140	8062	3437	1406
8	7016	1844	1140	1718	1953	1703
9	5401	7733	1156	2109	1750	1578
10	373	8127	1250	2703	2687	20968
		Avg	1263.8	2501.2	6573	6788.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	21	12	19	111
2	47	11	21	1557

3	31	10	65	6539
4	40	11	86	321
5	212	11	90	1004
6	15	11	10	9671
7	11	41	18	2791
8	19	11	12	3615
9	134	16	11	231
10	44	18	18	345
Avg	57.4	15.2	35	2618.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-2.29062
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 26: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 56000 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9876	4823	1515	11562	22531	6953
2	56	8838	1796	1890	2468	2218
3	8249	882	2281	3718	3859	11406
4	3854	9962	1500	1593	2750	1671
5	4198	4226	1500	3703	9359	42171
6	4506	4606	1500	1781	2062	1515
7	9743	3596	1703	3750	3015	3546
8	6529	1380	1500	2125	1937	2453
9	9186	81	1484	3609	5078	40296
10	3294	8527	1515	3562	3546	1625
		Avg	1629.4	3729.3	5660.5	11385.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	16	74	158	2195
2	338	11	14	1910
3	29	18	18	3087
4	31	11	17	1049
5	18	30	27	17062
6	20	10	11	260
7	340	34	20	1203
8	21	11	11	8175
9	9	42	22	36884
10	30	17	17	23
Avg	85.2	25.8	31.5	7184.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-2.01724
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 27: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 64000 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	6648	589	1921	1968	2328	13656
2	9125	4949	1921	7531	2109	1968
3	7085	360	1921	2515	1765	4875
4	9453	4133	1921	2046	1781	2046
5	2768	9774	1921	2156	2531	2187
6	4399	3457	1937	2281	1781	1921
7	3940	7606	1921	1828	2859	14218
8	3028	8190	1921	2375	2140	2218
9	56	2697	1937	2015	1984	2593
10	1529	3154	1921	4390	1953	3796
		Avg	1924.2	2910.5	2123.1	4947.8

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	9	10	12	3822
2	51	35	11	178
3	10	15	11	70
4	18	16	10	245
5	9	12	13	3869
6	16	11	10	127
7	15	12	17	5774
8	20	12	11	1922
9	102	15	11	33
10	20	19	10	249
Avg	27	15.7	11.6	1628.9

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=-1.75092
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 28: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 72000 (45%)

Problem	Seed	Tuple Seed	Time Dual DVO	Time Primal	Time Primal	Time Dual

Number				Ordering	DVO	Cover
1	769	7985	2390	2843	2109	2468
2	5500	6241	2375	2375	2375	6000
3	3981	9656	2390	2781	2781	2484
4	7649	5580	2593	2109	2078	19484
5	7868	5578	2390	2250	2218	2421
6	691	7590	2375	2125	2250	2421
7	3561	7295	2390	2296	2046	2375
8	4953	4730	2437	2078	1906	2718
9	3069	6507	2390	2078	2328	4671
10	3944	2427	2406	2250	2718	2390
		Avg	2413.6	2318.5	2280.9	4743.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12	14	10	484
2	24	10	10	1236
3	16	17	12	1802
4	82	10	10	3917
5	9	11	11	217
6	14	12	10	52
7	11	10	10	52
8	24	10	10	561
9	40	11	11	596
10	8	10	13	14
Avg	24	11.5	10.7	893.1

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=1.44203
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 29: arity = 4, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 80000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7256	2967	2953	2625	2187	3015
2	6647	2196	2921	2593	2578	2937
3	9570	1101	2968	2281	2265	3125
4	1031	160	2921	2296	2125	2921
5	342	1700	2906	3296	2078	3000
6	981	2033	2937	2296	2375	2921
7	1436	4330	3125	2296	2484	3984
8	2401	8912	3437	2328	2437	6812
9	5126	7178	3640	2187	2234	3406
10	7148	9352	2921	2265	2187	3562

		Avg	3072.9	2446.3	2295	3568.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	14	13	10	230
2	15	10	12	236
3	30	12	10	81
4	14	10	10	106
5	24	14	11	221
6	15	10	10	11
7	25	11	10	91
8	11	10	10	910
9	31	10	10	173
10	14	10	10	41
Avg	19.3	11	10.3	210

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t=8.032342
FC in the dual-graph performs worse than FC in the primal-graph in for these problems.
(α=0.001)

Set 30: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 400 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3054	5029	<15	15	15	<15
2	2287	4452	15	15	15	<15
3	9697	5873	<15	15	15	31
4	4907	6691	<15	15	15	140
5	2687	3139	<15	31	31	<15
6	2018	5900	<15	15	15	<15
7	9611	7157	<15	15	15	<15
8	7100	2355	<15	15	15	<15
9	7607	8340	<15	15	15	<15
10	8980	668	<15	15	15	<15
		Avg		16.6	16.6	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	16
2	230	12	12	264
3	21	12	11	99
4	6	18	10	791
5	14	15	15	31
6	7	10	14	17
7	7	10	14	10

8	7	13	10	6
9	6	10	11	28
10	6	11	10	9
Avg	31	12.1	11.7	127.1

Set 31: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 800 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4419	7569	<15	31	31	<15
2	2868	5841	<15	15	31	<15
3	3234	6203	<15	15	31	<15
4	720	8309	<15	31	15	15
5	6349	5435	<15	15	31	<15
6	6969	9583	<15	31	15	<15
7	1725	3790	<15	31	31	15
8	7928	2574	<15	31	31	<15
9	3282	929	15	15	31	<15
10	7245	6010	<15	31	15	<15
		Avg		24.6	26.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	6	10	10	20
3	9	10	10	31
4	6	10	10	6
5	6	10	10	15
6	6	10	10	11
7	9	10	12	51
8	7	10	10	7
9	6	10	10	10
10	6	10	10	10
Avg	6.7	10	10.2	16.7

Set 32: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 1200 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7999	750	15	31	46	<15
2	634	8105	15	31	46	<15
3	7133	2150	<15	31	31	15
4	2909	7232	15	31	31	15
5	9433	3508	<15	31	31	15

6	7338	3578	<15	31	31	15
7	5851	9791	<15	31	31	15
8	2137	8423	<15	31	46	<15
9	491	2102	15	31	31	15
10	6006	1759	<15	31	46	15
		Avg		31	37	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	11
2	5	12	13	5
3	6	10	10	8
4	6	10	10	10
5	6	10	10	7
6	6	10	10	37
7	13	10	13	13
8	6	10	10	6
9	6	10	10	13
10	6	10	10	53
Avg	6.7	10.2	10.6	16.3

Set 33: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 1600 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9269	963	<15	46	46	<15
2	1204	7832	15	46	31	15
3	5645	6434	<15	46	31	15
4	9264	3073	<15	31	46	15
5	2151	6139	<15	46	31	15
6	4500	2839	15	46	31	15
7	9526	7588	<15	46	31	15
8	7091	2379	15	46	31	15
9	9719	8908	<15	46	46	<15
10	1641	83	<15	46	46	<15
		Avg		44.5	37	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	6	10	10	5
3	15	10	10	15
4	6	10	10	10
5	6	10	10	5
6	6	10	10	44
7	6	10	10	6

8	6	10	10	6
9	6	10	10	20
10	6	10	10	6
Avg	6.9	10	10	12.3

Set 34: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 2000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	6585	1433	<15	62	46	<15
2	1420	7398	15	46	62	<15
3	3022	5343	<15	46	46	15
4	9185	3565	15	46	46	<15
5	6436	5185	15	46	46	<15
6	1367	416	15	46	62	<15
7	2417	1205	<15	62	46	<15
8	2133	2673	<15	46	46	<15
9	9153	3801	15	46	46	15
10	5730	8828	15	46	46	15
		Avg		49.2	49.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	10	10	11	10
3	6	10	10	11
4	6	10	10	8
5	6	10	10	13
6	6	10	11	5
7	6	10	10	8
8	6	10	10	6
9	6	10	10	5
10	6	10	10	6
Avg	6.4	10	10.2	7.8

Set 35: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 2400 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4213	501	<15	62	46	15
2	7683	6323	15	46	62	<15
3	887	5002	15	62	46	15
4	4207	3683	15	62	46	<15
5	6849	7844	15	46	46	15

6	7657	8025	15	46	46	15
7	4664	9044	<15	46	46	<15
8	3847	1827	<15	46	62	<15
9	4116	4969	<15	62	62	<15
10	4618	27	15	62	46	<15
		Avg		54	50.8	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	9
2	6	10	10	6
3	6	10	10	6
4	6	10	10	5
5	11	10	10	11
6	6	10	10	10
7	6	10	10	6
8	6	10	10	5
9	6	10	10	15
10	11	10	10	11
Avg	7	10	10	8.4

Set 36: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 2800 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1971	3983	<15	62	46	15
2	833	8607	15	62	46	15
3	1454	3374	15	62	62	<15
4	9520	3479	<15	62	62	<15
5	8091	4257	15	46	62	15
6	622	7721	<15	62	62	<15
7	7209	1253	<15	62	62	<15
8	2556	5727	<15	62	62	<15
9	5971	2170	15	62	62	15
10	136	2857	<15	62	62	<15
		Avg		60.4	58.8	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	5
2	6	10	10	12
3	6	10	10	10
4	6	10	10	5
5	6	10	10	7
6	6	10	10	6

7	6	11	10	6
8	6	10	10	5
9	6	10	10	5
10	6	10	10	5
Avg	6	10.1	10	6.6

Set 37: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 3200 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7838	7230	15	62	78	<15
2	8519	7444	15	62	62	15
3	6382	8499	15	62	62	15
4	7116	6164	<15	62	62	15
5	5451	5452	15	62	78	<15
6	720	6165	<15	62	46	<15
7	7470	9313	15	62	62	<15
8	7076	281	15	62	62	15
9	8020	7357	<15	62	78	15
10	3985	3933	15	62	62	15
		Avg		62	65.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	6	10	10	6
3	6	10	10	12
4	6	10	10	5
5	6	10	10	6
6	6	10	10	6
7	6	10	10	5
8	6	10	10	10
9	6	10	10	5
10	6	10	11	6
Avg	6	10	10.1	6.7

Set 38: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 3600 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3886	7407	15	62	78	15
2	7231	3909	15	78	62	15
3	1885	9092	15	62	62	15
4	6065	8840	15	62	62	15

5	5855	4430	15	62	62	15
6	6469	2019	15	62	62	15
7	8918	8540	<15	78	62	15
8	9874	6507	15	62	62	15
9	9304	7833	<15	78	62	15
10	1195	953	15	62	78	<15
		Avg		66.8	65.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	6	10	10	5
3	6	10	10	6
4	6	10	10	6
5	6	10	10	6
6	5	10	10	5
7	6	10	10	8
8	6	10	10	5
9	6	10	10	5
10	6	10	10	7
Avg	5.9	10	10	5.9

Set 39: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 4000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	8626	9967	15	62	78	15
2	1132	698	<15	78	78	15
3	9226	2013	15	78	62	15
4	468	6567	15	78	62	15
5	1491	2443	15	62	78	15
6	7845	7992	15	62	78	15
7	829	6535	15	78	78	<15
8	627	5770	15	78	62	15
9	3478	6272	<15	78	78	<15
10	3549	6718	<15	78	78	15
		Avg		73.2	73.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	6	10	10	6
2	6	10	10	11
3	6	10	10	6
4	6	10	10	6
5	6	10	10	6
6	6	10	10	10

7	6	10	10	10
8	6	10	10	5
9	6	10	10	5
10	6	10	10	6
Avg	6	10	10	7.1

Set 40: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 400 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3754	5974	15	31	46	15
2	3414	4359	<15	46	46	31
3	5215	9427	15	31	62	<15
4	8734	647	<15	171	93	31
5	2328	9237	31	328	421	359
6	913	9261	46	46	15	468
7	9242	1685	671	15	15	2859
8	544	9005	15	15	31	15
9	9557	4709	15	31	62	15
10	8849	3616	<15	31	31	<15
		Avg		74.5	82.2	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	30	17	38	73
2	10	37	28	291
3	34	23	35	114
4	35	149	44	242
5	43	155	321	5213
6	154	19	13	7257
7	10044	12	11	27666
8	16	10	19	210
9	30	25	33	119
10	8	21	14	27
Avg	1040.4	46.8	55.6	4121.2

Set 41: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 800 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3176	184	<15	31	31	<15
2	7281	9276	<15	31	46	<15
3	4011	4495	15	46	31	46

4	898	6081	<15	62	31	125
5	4540	3755	<15	46	46	<15
6	4942	757	15	31	31	7484
7	8808	3938	15	46	46	<15
8	2096	9187	<15	46	15	31
9	435	7937	<15	31	31	<15
10	9135	3568	<15	125	15	93
		Avg		49.5	32.3	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	10	10	10	59
2	8	11	11	59
3	20	14	12	295
4	24	17	11	1956
5	12	17	11	33
6	11	10	10	128323
7	11	14	10	22
8	8	12	10	604
9	8	10	10	39
10	21	27	10	952
Avg	13.3	14.2	10.5	13234.2

Set 42: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 1200 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1587	271	<15	46	46	<15
2	9421	5927	<15	62	78	15
3	7249	2111	<15	46	46	15
4	1658	7071	<15	46	109	656
5	6658	506	15	78	46	78
6	8308	3481	15	62	62	15
7	7500	3119	15	78	62	31
8	8591	3505	15	93	62	78
9	5651	1265	<15	46	109	15
10	96	5113	15	296	46	15
		Avg		85.3	66.6	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	8	11	10	19
2	8	14	16	145
3	8	10	12	13
4	12	12	19	11413
5	21	12	13	136

6	10	10	10	49
7	22	11	10	34
8	16	22	10	754
9	8	10	20	195
10	48	90	10	315
Avg	16.1	20.2	13	1307.3

Set 43: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 1600 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7576	1355	<15	78	78	281
2	5838	1296	15	62	62	<15
3	6125	5277	<15	62	46	15
4	6711	7529	<15	62	46	15
5	8988	9104	15	46	46	62
6	4352	1116	15	62	46	2671
7	2414	8499	<15	46	46	<15
8	5420	83	<15	62	46	15
9	1451	5541	<15	46	78	15
10	669	5974	15	62	62	15
		Avg		58.8	55.6	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	13	10	11	2644
2	10	11	11	11
3	11	11	10	14
4	8	10	10	17
5	12	10	11	87
6	8	10	10	33742
7	7	10	10	13
8	12	16	10	18
9	13	12	17	84
10	7	12	12	9
Avg	10.1	11.2	11.2	3663.9

Set 44: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 2000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4239	8791	<15	109	109	15
2	1547	3540	15	62	78	31
3	8777	8792	15	78	62	46

4	501	7196	<15	62	62	15
5	3270	6028	15	62	62	<15
6	204	5171	<15	62	62	31
7	2233	9548	<15	62	62	15
8	7011	7648	<15	62	62	<15
9	4272	5877	15	62	31	15
10	5378	3490	15	62	46	15
		Avg		68.3	63.6	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	9	19	19	9
2	14	10	11	782
3	8	10	10	316
4	10	12	10	5
5	13	10	10	8
6	7	11	11	20
7	7	10	10	6
8	15	11	10	38
9	9	10	10	13
10	8	10	10	15
Avg	10	11.3	11.1	121.2

Set 45: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 2400 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3034	5492	<15	78	62	15
2	3644	9520	<15	78	62	15
3	6844	7523	<15	62	78	15
4	6803	1061	<15	78	62	15
5	1140	7901	<15	78	62	15
6	4650	8168	31	78	62	15
7	1691	6061	15	62	78	<15
8	1436	7469	<15	78	62	15
9	8946	5764	<15	62	46	15
10	323	5158	15	78	109	15
		Avg		73.2	68.3	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	8	10	10	23
2	12	10	10	20
3	7	10	10	39
4	7	10	10	8
5	8	10	10	14

6	7	10	10	7
7	15	10	10	16
8	10	10	10	10
9	8	13	10	8
10	8	10	15	20
Avg	9	10.3	10.5	16.5

Set 46: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 2800 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5163	6041	<15	78	78	<15
2	5306	3488	15	78	62	15
3	4874	7661	15	62	78	15
4	5090	7137	15	62	78	15
5	842	927	15	78	78	15
6	658	8576	15	78	93	1687
7	5415	4597	15	78	78	15
8	5744	4489	15	78	46	31
9	1283	8738	15	78	78	31
10	8864	6419	15	78	78	<15
		Avg		74.8	74.7	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	16
2	8	10	10	7
3	8	10	10	30
4	7	10	10	18
5	8	10	11	13
6	7	10	10	17544
7	8	10	10	8
8	8	10	10	46
9	7	10	10	186
10	8	10	10	20
Avg	7.6	10	10.1	1788.8

Set 47: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 3200 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3436	9969	15	93	78	15
2	1848	717	15	93	78	62
3	9760	6162	15	78	78	15

4	2458	4506	15	78	93	15
5	5688	3624	15	78	109	15
6	3164	2297	31	93	78	15
7	8166	3939	31	78	78	15
8	8441	9433	15	78	93	15
9	4472	6617	15	78	93	15
10	1363	7606	15	78	78	15
		Avg	18.2	82.5	85.6	19.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	15
2	8	10	10	714
3	10	10	10	22
4	9	10	10	23
5	8	10	10	8
6	8	10	10	7
7	8	10	10	8
8	7	10	10	42
9	7	10	10	8
10	7	10	10	7
Avg	7.9	10	10	85.4

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -16.6885
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 48: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 3600 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	2345	7132	15	93	78	31
2	2368	2340	15	93	93	15
3	1024	3263	15	93	93	15
4	6698	929	15	93	93	15
5	7008	9686	15	78	93	15
6	7463	5316	15	109	93	15
7	5416	5566	15	109	93	15
8	586	8524	31	156	125	15
9	1793	9613	15	78	93	15
10	2025	4784	15	93	93	15
		Avg	16.6	99.5	94.7	16.6

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover

1	8	10	10	8
2	8	10	10	8
3	8	10	10	21
4	9	10	10	7
5	9	10	10	6
6	8	10	10	5
7	7	10	10	10
8	8	10	10	8
9	7	10	10	9
10	10	10	10	8
Avg	8.2	10	10	9

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -19.4542
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 49: arity = 3, domain size = 20, number of variables = 10, number of constraints = 8, number of tuples satisfying each constraint: 4000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7404	7473	15	93	93	15
2	9886	8854	31	93	109	31
3	1427	2078	15	93	93	15
4	3462	5179	31	93	93	15
5	8358	1898	15	93	93	15
6	9796	5543	31	93	93	15
7	2494	2486	31	109	93	31
8	4853	7648	15	93	109	15
9	4566	3384	15	125	78	15
10	9918	1594	15	93	93	15
		Avg	21.4	97.8	94.7	18.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	7	10	10	7
2	8	10	10	67
3	9	10	10	11
4	7	10	10	11
5	8	10	10	6
6	8	10	10	5
7	8	10	10	22
8	8	10	10	11
9	9	11	10	5
10	7	10	10	10
Avg	7.9	10.1	10	15.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -19.1217
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 50: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 400 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4106	402	203	281	328	421
2	866	7198	468	265	125	437
3	4272	7283	125	78	31	906
4	8128	6620	859	1703	1703	3281
5	230	6805	187	718	703	937
6	8333	8790	2421	2171	703	8593
7	6039	1739	468	2593	2078	2468
8	6849	3316	703	5046	1812	2843
9	4937	8289	187	140	140	4906
10	1386	870	140	265	218	265
		Avg	576.1	1326	784.1	2505.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	372	144	144	4113
2	2440	265	69	3119
3	460	47	14	12994
4	3825	1602	908	21338
5	357	342	362	12532
6	7908	824	346	74971
7	1924	1433	1097	29340
8	1840	3756	922	40322
9	248	69	62	68711
10	439	224	137	1565
Avg	1981.3	870.6	406.1	26900.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -.6274
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 51: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 800 (10%)

Problem	Seed	Tuple Seed	Time Dual DVO	Time Primal	Time Primal	Time Dual

Number				Ordering	DVO	Cover
1	9420	8066	31	125	78	156
2	5019	3331	62	62	62	734
3	855	5349	<15	93	46	31
4	9248	2237	93	140	62	265
5	9201	2427	31	46	546	2750
6	9097	5966	31	343	125	93
7	5466	5999	62	140	125	2062
8	2048	4678	31	125	125	2015
9	7901	137	15	171	265	234
10	2591	8477	46	203	46	49953
		Avg		144.8	148	5829.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	106	48	23	937
2	75	17	17	2590
3	10	45	16	259
4	358	65	17	2819
5	103	24	122	7349
6	94	95	34	1373
7	73	37	31	32232
8	153	55	54	9071
9	24	76	94	2335
10	182	59	12	638975
Avg	117.8	52.1	42	69794

Set 52: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 1200 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1551	2479	15	62	46	1500
2	536	2118	15	93	93	46
3	9486	6261	15	46	46	46
4	1085	9058	15	109	2453	<15
5	9028	5443	15	296	125	500
6	8313	7638	15	46	46	6656
7	4645	4986	15	78	62	1593
8	875	7230	<15	109	62	2281
9	9518	7187	15	78	62	15
10	6651	8	15	62	62	31
		Avg		97.9	305.7	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	31	15	10	23861

2	23	25	18	153
3	24	10	11	137
4	46	19	472	15
5	35	97	26	2852
6	57	13	11	111329
7	24	26	12	11491
8	8	39	13	34825
9	9	16	12	107
10	25	15	14	164
Avg	28.2	27.5	59.9	18493.4

Set 53: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 1600 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	2689	4513	31	187	62	218
2	5044	136	<15	62	62	953
3	2670	1434	15	93	78	125
4	8393	6363	15	62	359	46
5	2020	3021	<15	62	62	15
6	782	9561	<15	109	78	15
7	1810	242	<15	62	78	31
8	9073	1574	15	109	62	15
9	2982	613	15	515	156	15
10	2961	8610	<15	93	62	78
		Avg		135.4	105.9	151.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	58	36	12	3749
2	10	10	11	16701
3	18	22	17	1045
4	17	13	53	171
5	14	10	10	82
6	14	17	13	57
7	15	13	11	172
8	8	13	11	24
9	19	72	21	21
10	24	35	10	1147
Avg	19.7	24.1	16.9	2316.9

Set 54: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 2000 (25%)

Problem	Seed	Tuple Seed	Time Dual DVO	Time Primal	Time Primal	Time Dual

Number				Ordering	DVO	Cover
1	2329	4717	15	109	62	46
2	2070	3865	31	281	93	296
3	9184	8523	15	62	93	15
4	7547	3444	15	93	93	<15
5	3670	1913	15	93	78	62
6	2121	5659	15	78	62	31
7	7805	1151	15	62	93	15
8	674	4	15	78	62	15
9	789	9418	15	78	62	78
10	5728	590	15	78	78	15
		Avg	16.6	101.2	77.6	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	21	17	10	335
2	152	34	12	3554
3	11	11	11	22
4	8	21	13	24
5	21	13	11	76
6	13	10	11	65
7	26	11	12	139
8	12	10	10	39
9	13	11	10	992
10	39	12	10	35
Avg	31.6	15	11	528.1

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -12.4724
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 55: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 2400 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	6057	5155	15	93	78	15
2	7090	3308	15	78	93	187
3	9516	232	<15	93	78	31
4	2039	8996	15	62	93	15
5	9825	6013	15	93	62	15
6	5302	5435	15	78	62	62
7	6372	7710	<15	78	78	437
8	1402	1733	<15	78	125	15
9	5179	6453	15	78	78	15
10	570	7734	15	78	78	31

		Avg		80.9	82.5	82.3

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	11	11	10	20
2	41	11	13	157
3	10	10	10	147
4	56	12	10	62
5	9	11	10	44
6	17	10	10	50
7	8	10	10	2991
8	11	10	14	43
9	17	12	11	11
10	11	10	10	160
Avg	19.1	10.7	10.8	368.5

Set 56: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 2800 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5904	6074	15	93	93	15
2	9301	6555	15	78	93	15
3	8376	4874	31	156	78	31
4	7706	7055	15	78	93	15
5	976	2496	15	93	93	625
6	5953	3090	15	93	78	15
7	3077	3364	15	93	109	<15
8	732	7263	15	78	109	1046
9	2153	7570	15	78	93	15
10	2316	5697	15	93	78	156
		Avg	16.6	93.3	91.7	

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	10	10	10	14
2	13	10	10	5
3	61	15	12	161
4	7	12	11	10
5	9	10	10	428
6	9	10	10	46
7	10	11	11	8
8	9	10	11	18990
9	10	10	11	21
10	8	10	10	925
Avg	14.6	10.8	10.6	2060.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -19.0379
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 57: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 3200 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7315	1906	15	78	93	15
2	8622	7658	15	93	93	500
3	5562	3394	15	93	93	31
4	7014	1968	15	78	109	15
5	6297	9342	15	93	109	15
6	9835	6584	15	93	93	15
7	5696	8785	15	93	93	15
8	724	3238	15	93	78	31
9	4983	7445	15	93	93	15
10	4137	9080	15	93	93	15
		Avg	15	90	94.7	66.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	10	10	10	18
2	11	10	10	5929
3	10	10	10	66
4	32	10	10	16
5	9	11	10	21
6	9	11	10	25
7	10	10	10	24
8	12	12	10	51
9	8	10	10	29
10	11	10	10	48
Avg	12.2	10.4	10	622.7

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -28.414
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 58: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 3600 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover

1	1713	4972	31	109	109	15
2	1586	7108	15	93	109	31
3	5991	5773	15	93	109	15
4	4500	1642	15	93	93	31
5	3772	9361	15	93	109	15
6	6964	8820	15	93	109	17734
7	1177	2500	15	125	109	62
8	8084	2202	15	93	109	15
9	5991	6105	15	93	109	15
10	6502	9766	31	109	93	31
		Avg	18.2	99.4	105.8	1796.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12	10	10	18
2	8	10	10	13
3	14	10	10	11
4	12	10	10	12
5	9	10	10	9
6	14	10	10	128703
7	9	10	10	48
8	12	10	10	13
9	8	10	10	10
10	10	10	10	63
Avg	10.8	10	10	12890

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -29.0356
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.001)

Set 59: arity = 3, domain size = 20, number of variables = 10, number of constraints = 10, number of tuples satisfying each constraint: 4000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9378	5330	15	109	93	31
2	6324	9327	15	93	109	15
3	6839	1835	31	109	125	31
4	4762	2162	15	109	109	31
5	6381	4087	15	109	109	31
6	6643	537	15	93	109	15
7	2338	6271	15	125	109	15
8	2924	7952	15	125	109	15
9	3243	19	31	109	93	31
10	84	7937	31	109	93	31
		Avg	19.8	109	105.8	24.6

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	8	10	10	7
2	8	10	10	7
3	10	10	10	12
4	14	10	10	15
5	8	10	10	7
6	9	10	10	5
7	10	10	10	5
8	8	10	10	8
9	16	10	10	11
10	9	10	10	8
Avg	10	10	10	8.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -21.3581
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 60: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 800 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3947	1836	1062	4515	4921	13109
2	2901	8873	140	906	640	859
3	8686	1299	1484	93	109	1687
4	8441	7480	1406	203	171	10328
5	5137	186	2062	9968	9328	9609
6	5907	5874	359	250	296	115718
7	5303	7481	281	984	4875	5609
8	7242	6061	156	1000	1156	1562
9	6541	2725	187	6171	5000	8734
10	4564	7606	1093	3937	3921	12000
		Avg	823	2802.7	3041.7	17921.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	1518	916	1352	166570
2	172	230	165	7133
3	2261	30	32	5237
4	3909	84	56	66202
5	6899	3209	2619	109384
6	870	63	106	1387941
7	851	511	1096	24676
8	163	273	345	21993

9	567	1471	1345	20893
10	861	987	826	192943
Avg	1807.1	777.4	794.2	200297.2

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.23126
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 61: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 1200 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9629	260	62	890	1468	78
2	3134	2152	109	812	390	515
3	926	2351	125	171	4515	4421
4	1515	4875	15	93	859	250
5	3935	9205	109	3812	45828	3859
6	8040	8644	203	406	265	19625
7	5013	9346	1046	687	6250	59968
8	8264	7922	15	93	62	78
9	8340	8917	109	109	109	17843
10	6898	6645	62	218	734	500
		Avg	185.5	729.1	6048	10713.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	227	252	339	279
2	1091	293	118	6363
3	266	41	535	49663
4	95	23	177	3509
5	176	944	7947	57758
6	449	150	48	56165
7	4140	145	1622	339780
8	22	22	13	1098
9	145	24	20	223222
10	271	46	135	1969
Avg	688.2	194	1095.4	73980.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.31148
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 62: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 1600 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3888	6149	187	156	234	45015
2	3053	462	15	78	187	15
3	7506	813	15	156	203	4750
4	5575	4369	15	156	171	93
5	2419	694	93	390	234	15
6	9688	945	15	453	109	78
7	520	1199	15	62	62	1031
8	5438	9996	78	78	250	93
9	4238	4393	31	78	78	2359
10	2495	5326	15	125	562	703
		Avg	47.9	173.2	209	5415.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	243	26	36	944266
2	16	12	24	44
3	54	22	29	77941
4	24	23	23	386
5	654	141	42	33
6	148	191	19	499
7	19	14	11	13867
8	149	15	44	693
9	216	14	12	26960
10	81	21	75	9839
Avg	160.4	47.9	31.5	107452.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -3.35639
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 63: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 2000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1781	1506	31	78	93	25375
2	748	650	15	78	109	125
3	6853	1616	15	78	78	62
4	8584	3983	15	109	78	343
5	3740	7414	15	140	78	31
6	674	4549	93	93	109	3906

7	4337	5852	15	93	203	171
8	3644	8163	<15	93	171	93
9	6892	754	<15	78	78	140
10	4161	3486	15	78	78	62
		Avg		91.8	107.5	3030.8

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	18	11	12	313818
2	60	10	12	252
3	14	10	11	539
4	25	21	10	739
5	26	26	11	43
6	1305	12	13	2398
7	48	12	47	1598
8	16	12	26	81
9	32	20	11	564
10	135	10	12	320
Avg	167.9	14.4	16.5	32035.2

Set 64: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 2400 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7700	9131	15	78	93	250
2	824	5087	15	187	93	31
3	1050	5898	31	78	109	281
4	7259	7557	15	140	93	15
5	999	4876	15	125	125	15
6	8883	8783	15	93	93	171
7	8856	1249	<15	78	78	93
8	2641	696	15	78	93	31
9	2827	4067	15	93	109	93
10	7543	7875	15	78	109	15
		Avg		102.8	99.5	99.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	33	10	10	5200
2	10	26	15	93
3	115	10	12	253
4	28	61	10	210
5	13	17	12	37
6	89	18	12	167
7	19	10	10	288
8	14	10	10	30

9	51	13	11	469
10	9	12	12	124
Avg	38.1	18.7	11.4	687.1

Set 65: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 2800 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	2651	8048	31	93	93	31
2	2224	5807	15	93	93	15
3	3251	5555	15	93	125	31
4	7146	5291	15	93	109	15
5	3514	9304	15	93	93	46
6	3944	386	62	140	109	78
7	9229	248	15	109	93	46
8	3346	9029	31	93	109	15
9	7173	3786	15	93	93	15
10	7000	2767	15	109	93	500
		Avg	22.9	100.9	101	79.2

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	9	10	10	173
2	11	10	11	71
3	15	12	18	53
4	9	11	10	115
5	15	10	15	213
6	230	12	11	192
7	11	11	10	214
8	10	12	11	24
9	12	10	11	34
10	9	11	10	6274
Avg	33.1	10.9	11.7	736.3

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -12.9988
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 66: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 3200 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	8260	7545	15	109	109	46

2	9230	132	31	109	125	31
3	5736	1965	15	140	93	15
4	1517	9930	31	109	109	31
5	2064	5812	15	109	109	78
6	5688	4990	15	140	93	31
7	8368	1920	15	109	109	31
8	1845	579	31	109	109	31
9	1856	5167	15	109	93	31
10	9199	8714	15	125	125	15
		Avg	19.8	116.8	107.4	34

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	26	10	10	192
2	10	11	12	107
3	13	12	10	13
4	15	12	10	23
5	22	10	10	646
6	16	11	10	27
7	16	11	10	29
8	14	10	11	24
9	12	12	10	8
10	12	15	11	13
Avg	15.6	11.4	10.4	108.2

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -19.6316
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 67: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 3600 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	3407	3281	31	125	109	31
2	1098	722	31	93	125	15
3	8305	1540	31	125	109	31
4	9881	1965	15	109	125	62
5	4500	9732	31	125	109	31
6	8537	3371	15	109	109	62
7	5918	25	31	109	109	546
8	8239	6787	31	125	109	31
9	7968	425	93	125	125	31
10	7992	3289	15	109	109	15
		Avg	32.4	115.4	113.8	85.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	11	10	10	20
2	9	11	10	13
3	11	10	10	24
4	12	10	10	46
5	12	10	10	15
6	8	11	10	27
7	13	10	10	4341
8	9	10	10	22
9	20	11	10	59
10	12	10	10	42
Avg	11.7	10.3	10	460.9

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -10.7817
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.001)

Set 68: arity = 3, domain size = 20, number of variables = 10, number of constraints = 12, number of tuples satisfying each constraint: 4000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7592	9903	31	109	125	31
2	3956	4778	31	109	140	15
3	5781	5278	31	125	125	31
4	2262	7185	31	125	109	46
5	8306	9436	15	125	125	31
6	8099	7633	31	125	125	62
7	1808	1450	15	125	125	31
8	7570	3415	31	109	109	78
9	9862	9467	31	125	125	15
10	3955	5748	15	125	125	31
		Avg	26.2	120.2	123.3	37.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	11	10	10	14
2	13	10	10	9
3	11	10	10	58
4	15	10	10	16
5	10	10	10	48
6	8	10	10	77
7	13	11	10	44
8	8	10	10	83
9	12	10	10	12

10	13	10	10	127
Avg	11.4	10.1	10	48.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -26.0996
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 69: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 800 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9203	7860	1265	703	718	34437
2	4065	1432	3203	3250	1531	20187
3	9742	4097	3593	21531	15765	41671
4	7609	9660	1859	12109	43046	4531
5	252	4271	2546	5578	6531	22015
6	2418	135	13078	29531	3359	82953
7	3309	1329	1656	6734	3343	16562
8	8491	5483	312	1140	921	1531
9	5321	1121	1281	15953	6656	15203
10	8954	8684	9781	24078	20546	171531
		Avg	3857.4	12060.7	10241.6	41062.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	1504	181	140	160002
2	8052	730	380	62732
3	6519	9986	4299	251427
4	2761	3409	9126	32550
5	4193	1346	1570	200299
6	25045	7091	858	412974
7	2959	2552	826	66614
8	428	340	233	8914
9	2869	4891	1718	129712
10	16650	4786	4434	463221
Avg	7098	3531.2	2358.4	178844.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.4499
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 70: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 1200 (15%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	6920	6399	218	390	234	1359
2	4169	5404	7890	17531	17937	326328
3	9548	9639	1734	921	171	184218
4	5227	1767	1562	156	1375	34203
5	7877	1003	2015	11218	4203	286921
6	6364	593	1718	3531	93	34890
7	7216	3196	250	3328	65656	2328
8	3456	4070	859	1281	453	39171
9	6714	3118	406	921	937	2500
10	8064	5589	46	343	437	421
		Avg	1669.8	3962	9149.6	91233.9

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	475	156	60	6318
2	10082	2621	2535	3864564
3	3347	210	32	1094587
4	4362	34	324	297145
5	2983	2487	839	1544399
6	3397	603	19	55916
7	530	784	10318	7529
8	1910	392	62	205197
9	860	162	128	2846
10	87	81	72	2727
Avg	2803.3	753	1438.9	708122.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.14154
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 71: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 1600 (20%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	8931	2942	359	93	125	88484
2	8119	3978	421	468	625	3968
3	1603	3165	828	10984	421	1062
4	3816	9676	46	93	2609	26359
5	1287	3448	171	234	2718	42531
6	7404	9439	140	203	9531	12187

7	9755	2608	15	328	234	5406
8	6768	7143	93	2140	93	937
9	6298	4335	15	531	109	859
10	4452	6112	218	343	312	69218
		Avg	230.6	1541.7	1677.7	25101.1

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	705	20	20	233846
2	2119	208	114	29530
3	1223	3321	55	3255
4	244	12	453	39141
5	402	49	321	173928
6	610	36	1172	21695
7	60	103	42	77319
8	325	590	14	6898
9	50	64	16	5606
10	214	95	55	381958
Avg	595.2	449.8	226.2	97317.6

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.55233
FC in the dual-graph performs about as well as FC in the primal-graph in for these problems. (α=0.1)

Set 72: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 2000 (25%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5709	5491	46	343	218	78
2	8078	2894	781	125	203	17140
3	5328	1125	359	687	93	70984
4	9903	7125	31	93	109	1687
5	9798	8146	46	484	1484	593
6	877	7169	46	171	1375	46
7	1041	441	46	5234	234	2359
8	8222	836	15	78	187	8875
9	1891	7519	93	359	125	203
10	6319	2933	46	156	859	7812
		Avg	150.9	773	488.7	10977.7

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	105	207	42	421
2	3013	16	23	19024
3	247	88	11	54440

4	69	14	15	3119
5	149	83	134	3849
6	30	25	78	344
7	341	1332	23	28191
8	40	10	26	9689
9	526	102	17	2441
10	48	22	96	69770
Avg	456.8	189.9	46.5	19128.8

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -1.79464
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.1)

Set 73: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 2400 (30%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	4699	3781	15	156	171	328
2	3107	8580	31	203	93	31
3	3304	2983	46	156	93	1656
4	1224	5922	31	156	140	250
5	2323	9064	15	93	109	140
6	6261	4430	359	32859	312	595781
7	8928	2295	109	234	78	434468
8	6175	367	15	93	203	484
9	7322	4793	46	187	125	189609
10	3713	940	15	156	93	421
		Avg	68.2	3429.3	141.7	122316.8

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	12	19	24	1428
2	172	92	11	56
3	163	40	10	9040
4	60	15	15	671
5	21	14	10	1612
6	2402	6682	43	10525817
7	302	58	10	5271660
8	19	12	20	2213
9	216	20	15	303452
10	14	17	10	3782
Avg	338.1	696.9	16.8	1611973

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)

t= -1.8159
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.1)

Set 74: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 2800 (35%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9954	2298	46	109	125	49500
2	827	9721	15	140	156	31
3	6770	8247	15	125	140	140
4	8952	2274	46	109	93	15
5	8762	4477	31	125	125	312
6	329	1512	31	109	171	375
7	4236	9758	15	140	140	140
8	6295	8496	15	109	156	46
9	5435	1883	31	125	171	31
10	1884	1196	15	109	93	125
		Avg	26	120	137	5071.5

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	15	10	11	432905
2	26	27	16	58
3	16	12	10	322
4	81	12	10	9
5	37	23	11	2659
6	10	11	16	2430
7	56	12	13	912
8	18	11	17	161
9	11	12	16	139
10	35	17	10	638
Avg	30.5	14.7	13	44023.3

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -11.295
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 75: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 3200 (40%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	711	5490	31	93	93	125

2	7268	5881	62	93	125	31
3	1626	922	15	140	125	31
4	830	5613	31	218	125	3734
5	870	5803	31	109	125	46
6	6646	4277	31	156	125	93
7	6119	8242	31	125	125	31
8	880	3163	62	125	140	46
9	5529	7110	15	140	125	406
10	7683	4250	46	187	171	843
		Avg	35.5	138.6	127.9	538.6

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	60	10	10	81
2	152	11	14	142
3	10	14	11	117
4	10	16	10	13336
5	24	10	10	195
6	40	14	11	260
7	25	10	10	29
8	55	12	13	36
9	14	14	10	3104
10	19	10	10	8321
Avg	40.9	12.1	10.9	2562.1

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -11.5727
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

Set 76: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 3600 (45%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	5635	3238	31	125	125	609
2	958	3246	15	125	125	31
3	3963	2941	31	109	125	46
4	9585	6972	31	125	140	140
5	137	6525	31	125	125	46
6	4164	6687	15	125	125	171
7	9488	5081	31	140	140	31
8	4057	2801	31	109	109	31
9	7353	5119	31	125	109	875
10	9518	7301	31	125	125	78
		Avg	27.8	123.3	124.8	205.8

Problem Number	Nodes Dual DVO	Nodes Primal	Nodes Primal DVO	Nodes Dual Cover

		Ordering		
1	12	10	10	1117
2	13	10	10	29
3	11	11	10	19
4	16	11	10	563
5	14	12	11	144
6	14	10	12	1242
7	11	11	12	46
8	52	10	10	29
9	14	11	10	326
10	13	10	10	250
Avg	17	10.6	10.5	376.5

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -24.8509
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.001)

Set 77: arity = 3, domain size = 20, number of variables = 10, number of constraints = 14, number of tuples satisfying each constraint: 4000 (50%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	7487	3588	31	125	125	31
2	3304	8219	31	156	140	93
3	2952	6560	31	140	125	31
4	5005	6702	31	125	140	31
5	3260	5765	31	125	140	93
6	9076	7530	31	125	125	31
7	124	836	31	140	140	46
8	3491	3032	31	140	125	31
9	7824	4067	31	125	140	125
10	4996	114	31	125	125	62
		Avg	31	132.6	132.5	57.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover
1	8	10	10	13
2	9	11	10	289
3	16	10	10	17
4	13	10	10	30
5	8	10	10	99
6	7	12	10	17
7	12	10	10	58
8	11	11	10	41
9	8	10	10	592
10	17	10	10	16
Avg	10.9	10.4	10	117.2

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -40.6
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.001)

All Solutions:
Set 78: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 400 (5%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	9900	2074	162968	2871187	930531	487500
2	9533	9647	14906	108578	103406	68343
3	4210	6879	14656	134843	130921	25343
4	5309	1536	21546	124109	117937	65078
5	4620	4195	13203	119140	112703	43125
6	4632	4577	14343	122812	119500	55312
7	6442	9096	52203	457937	348828	103906
8	4553	6628	12843	117500	126390	59703
9	4648	2367	106781	1033796	449500	132359
10	5433	3244	12875	121000	117750	55765
		Avg	42632.4	521090.2	255746.6	109643.4

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover	Number of Solutions
1	1213512	2752546	925197	3467652	157876
2	250145	299611	287093	1424316	170193
3	208520	323900	316952	338714	184917
4	306461	334801	326440	1537898	192178
5	184749	287041	279309	829082	163757
6	173678	266968	261283	1292187	147284
7	435164	497390	431090	1987800	161311
8	205460	322705	319013	1004234	182665
9	1702623	1842711	724810	2407615	167897
10	195484	306242	293440	1086869	170987
	487579.6	723391.5	416462.7	1537636.7	169907

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -2.49196
FC in the dual-graph performs better than FC in the primal-graph in for these problems. (α=0.05)

Set 79: arity = 3, domain size = 20, number of variables = 10, number of constraints = 6, number of tuples satisfying each constraint: 800 (10%)

Problem Number	Seed	Tuple Seed	Time Dual DVO	Time Primal Ordering	Time Primal DVO	Time Dual Cover
1	1316	6737	758484	9867140	10038781	1764609
2	3709	1451	853109	13736265	9145703	871015
3	2442	6425	1365546	17985984	11108046	4244140
4	4969	1487	2142750	9204109	16809421	4074484
5	2990	1265	2279000	15580984	32484546	5304796
6	1163	5199	693437	10627046	10440718	1919406
7	564	3332	2472812	18237187	30077500	3972093
8	658	1625	729093	14539484	11160171	1297500
9	4311	9010	735968	10388734	10805453	3088187
10	6266	1492	740031	10892218	11510328	704906
		Avg	1336689	13351881	15785593	2948470

Problem Number	Nodes Dual DVO	Nodes Primal Ordering	Nodes Primal DVO	Nodes Dual Cover	Number of Solutions
1	11976650	14804358	14868760	32389268	10015636
2	14832816	16984212	15393436	14832891	10098845
3	14267063	18943473	15825043	71410705	10707041
4	31551322	16366977	18118219	66534414	9848396
5	28887672	17972044	22144480	90798258	9891500
6	10581947	15218159	15064466	36552096	10162251
7	35869462	18280691	25372013	93902002	9963117
8	11537092	17239666	15557197	19487102	10127770
9	11794939	15356091	15130753	56314225	10281545
10	10500035	15069536	15087008	10541465	10250525
Avg	19033218.11	16796186	17497152	53580107	10121789

Comparing FC in the Dual-graph with DVO(x_1) to FC in the Primal-graph with DVO(x_2)
t= -5.25234
FC in the dual-graph performs better than FC in the primal-graph in for these problems.
(α=0.001)

References

[Bac02] Bacchus, F., X. Chen, P. van Beek, and T. Walsh. "Binary vs. Non-Binary Constraints", Artificial Intelligence vol. 140, pp. 1-37, 2002

[Bac98] Bacchus, F., and van Beek, P. "On the Conversion between Non-Binary and Binary Constraint Satisfaction Problems", Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), pp. 326-333, 1998.

[Bac95] Bacchus, F., and van Run, P. "Dynamic Variable Ordering in CSPs", Proceedings of the First International Conference on Constraint Programming, Eds. Ugo Montanari, and Francesca Rossi, pp. 258-275, 1995.

[Bec97] Beck, J. C., and Jackson, W. K. "Constrainedness and the Phase Transition in Job Shop Scheduling", Technical Report TR 97-21, School of Computer Science, Simon Fraser University, 1997.

[Che91] Cheeseman, P., Kanefsky, B., and Taylor, W. M. "Where the Really Hard Problems Are", Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, pp. 331-337, 1991.

[Gen94] Gent, I. P., and Walsh, T. "Easy Problems are Sometimes Hard", Artificial Intelligence, volume 70 number 1, pp. 335-345, 1994.

[Hua04] Huang, Mingyan. "Duelling CSP Representations: Local Search in the Primal versus Dual Constraint Graph", MSc. Thesis, School of Computer Science, University of Windsor, 2004.

[Jai96] Jain, A. S., and Meeran, S. "Scheduling a Job-Shop Using a Modified Back-Error Propagation Neural Network", Proceedings of the IMS'96 First Turkish Symposium on Intelligent Manufacturing Systems, pp. 462-474, 1996.

[Kwa95] Kwan, A. C. M., and Tsang, E. P. K. "Comparing CSP Algorithms Without Considering Variable Ordering Heuristics can be Misleading", Technical Report CSM-262, 1995.

[Nag00] Nagarajan, S. "On Dual Encodings for Constraint Satisfaction", PhD thesis, University of Regina, Canada, 2000.

[Pan97] Pang, W., and Goodwin, S. D. "Constraint-Directed Backtracking", The 10th Australian Joint Conference on AI, pp. 47-56, 1997.

[Pan96] Pang, W., and Goodwin, S. D. "Application of CSP Algorithms to Job Shop Scheduling Problems", The 2nd International Symposium on Operations Research and Its Applications (ISORA '96), 1996.

[Rus03] Russell, S. J., and Norvig, P. "Artificial Intelligence: A Modern Approach", 2nd edition, Pearson Education Inc, pp.137-158, 2003.

[SmB95a] Smith, B. M. "A Tutorial on Constraint Programming", Research Report 95.14, School of Computer Studies, University of Leeds, 1995.

[SmB95b] Smith, B. M., and Grant, S. A. "Where the Exceptionally Hard Problems Are", Proceedings of the CP-95 workshop on Really Hard Problems, 1995.

[SmD95] Smith, D., and Goodwin, S. D. "Constraint-Based Intelligent Scheduling", Technical Report CS 95-02, Dept. of Computer Science, University of Regina, 1995.

[Tsa93] Tsang, E. "Foundations of Constraint Satisfaction", Academic Press Inc, San Diego, 1993.

Vita Auctoris

NAME:	Robert George Price
PLACE OF BIRTH:	Windsor, Ontario.
YEAR OF BIRTH:	1975
EDUCATION:	General Amherst High School, Amherstburg, Ontario
	1989-1994
	University of Windsor, Windsor, Ontario
	1994-1998 B.Sc.
	University of Windsor, Windsor, Ontario
	2002-2005 M.Sc.