Breaking Symmetries in Graph Representation

Michael Codish
Dept. of Computer Science
Ben-Gurion Univ. of the Negev
Beer-Sheva, Israel

Alice Miller and Patrick Prosser
School of Computing Science
University of Glasgow
Scotland

Peter J. Stuckey
Dept. of Comp. Sci. and SE
and NICTA Victoria Laboratory
University of Melbourne, Australia

Abstract

There are many complex combinatorial problems which involve searching for an undirected graph satisfying a certain property. These problems are often highly challenging because of the large number of isomorphic representations of a possible solution. In this paper we introduce novel, effective and compact, symmetry breaking constraints for undirected graph search. While incomplete, these prove highly beneficial in pruning the search for a graph. We illustrate the application of symmetry breaking in graph representation to resolve several open instances in extremal graph theory.

1 Introduction

The canonical graph representation problem is pertinent to a wide range of scientific applications. It is closely related to the graph isomorphism problem, as two graphs are isomorphic if and only if they have the same canonical representation. Examples of applications include data-mining [Washio and Motoda, 2003], mathematical chemistry [Faulon, 1998] and bio-informatics [Gardiner, 2011]. The two problems are poly-time equivalent, and are among the few that are known to be in NP but not known either to be solvable in polynomial time, nor to be NP-complete.

There are a variety of software tools devoted to solving the two problems "in practice", one of which is `nauty`, due to McKay [1990]. `Nauty` is sometimes referred to as the world's fastest isomorphism testing program. It is also able to produce a canonically-labeled isomorph of a graph to assist in isomorphism testing.

This paper is about constraint problems which involve the search for a graph that satisfies certain properties. For example, consider the problem to determine if there exists an undirected graph with 31 nodes, 81 edges, and which does not contain cycles of length 4 or less. This question arises in "extremal graph theory" [Bollobás, 1978], and its answer is unknown [Garnick *et al.*, 1993]. The search space for problems of this type is enormous, and search may be optimized by restricting it to focus on canonical representations, or to avoid as often as possible isomorphic graphs. The general idea is to "break" symmetries in the search space. However, it is not clear how to apply this idea when searching for a graph. In this type of problem the graph is a variable, so graph algorithms for canonical representation and isomorphism, as well as tools such as `nauty`, all of which operate on given graphs, do not apply. This paper provides a solution to this problem.

We assume a setting where testing for the existence of a graph G satisfying a property P is posed as a Boolean constraint $P(A_G)$ on the variables of the Boolean adjacency matrix A_G of G. We follow the approach advocated by Crawford *et al.* [1996], where a predicate, $\mathsf{sb}(A_G)$, is introduced to break symmetries in the search space. In this way the satisfiability of $P(A_G)$ is equivalent to that of $P(A_G) \wedge \mathsf{sb}(A_G)$. Ideally, $\mathsf{sb}(A_G)$ is satisfied by a single member of each equivalence class of A_G under graph isomorphism, thus drastically restricting the search space for $P(A_G) \wedge \mathsf{sb}(A_G)$. However, this is not realistically possible as such a predicate also determines a canonical representation. In practice, it is sufficient that $\mathsf{sb}(A_G)$ is satisfied by at least one member of the equivalence class of A_G under isomorphism (typically by more than one) and in this case we say that sb is a *symmetry breaking predicate*. Shlyakhter [2007] notes that the difficulty is to identify a symmetry-breaking predicate which is both *effective* (rules out a large portion of the search space) and *compact* (so that checking the additional constraints does not slow down the search).

The presentations in [Crawford *et al.*, 1996; Shlyakhter, 2007] consider symmetry breaking in terms of isomorphism, but focus on different structures such as acyclic digraphs, relations, permutations and functions. We introduce a novel, effective and compact predicate to break symmetries on graph representation. We consider simple graphs (undirected, with no multiple, nor self edges). We demonstrate the effectiveness of our approach through experimentation and resolve several open instances in extremal graph theory.

2 Graphs and their Canonical Representation

Throughout this paper we consider undirected simple graphs without loops or multiple edges. We focus on finite graphs and typically name the n nodes of a graph in the set $\{1,\ldots,n\}$. We denote the Boolean values *true* and *false* by 1 and 0 respectively.

Definition 1 (Graph) A *graph* $G = (V, E)$ has nodes $V = \{1,\ldots,n\}$ and edges $E \subseteq V \times V$ where $(x,y) \in E \Rightarrow (y,x) \in E$. The Boolean *adjacency matrix*, A_G of G, is the $n \times n$ symmetric matrix where $A_G[x,y] \leftrightarrow (x,y) \in E$. The i^{th} row of matrix A is denoted by $A[i]$, and $A[i,j]$ denotes the j^{th} element of $A[i]$. The degree of node $u \in V$ is $degree(u) = |\{(u,v) \mid (u,v) \in E\}|$. We denote the minimum and maximum degrees of the nodes in G as $\delta(G)$ and $\Delta(G)$, or δ and Δ when the context is clear.

Example 1 Figure 1 illustrates three graphs with corresponding adjacency matrices.

We use cycle notation to represent permutations. For example, the permutation (1,2,6)(3,4) maps 1 to 2, 2 to 6, 6 to 1, and 3 to 4, 4 to 3, and 5 to 5.

Definition 2 (permuting nodes) Let $G = (V, E)$ be a graph with n nodes, A_G the adjacency matrix for G, and π a permutation on $\langle 1, \ldots, n\rangle$. Then $\pi(G)$ is the graph represented by permuting the nodes of G using π. Formally, $\pi(G) = (V, E')$ where $E' = \{(\pi(x), \pi(y)) \mid (x,y) \in E\}$ and $\pi(A_G)$ is the adjacency matrix of $\pi(G)$.

Definition 3 (graph isomorphism) G and G' are isomorphic if there exists a permutation π such that $A_G = A_{\pi(G')}$.

Example 2 The graphs in Figure 1 are isomorphic. We can permute G_1 to G_2 using $\pi_1 = (2,8,5,9,4,7,3)$ and G_1 to G_3 using $\pi_2 = (2,9,4,8,6,7,3)$.

Definition 4 (sequences, lexicographic order) Let A a be matrix and $A[i]A[j]$ the concatenation of rows i and j (viewed as sequences). The length of a sequence s is denoted $|s|$. We use $\preceq$ to denote the usual lexicographic order on sequences. We extend this notation in the obvious way: for matrices, with n and m rows respectively, $A \preceq B$, if and only if $A[1]A[2] \cdots A[n] \preceq B[1]B[2] \ldots B[m]$; and for graphs, $G \preceq G'$ if and only if $A_G \preceq A_{G'}$.

One way to define a canonical representation of a graph is to take the smallest graph (i.e. in the lexicographic order) which is isomorphic to G [Read, 1978]. This is the definition which we adopt throughout the paper.

Definition 5 (canonical form of a graph) The canonical form of a graph G is the graph with $can(G) = \min_{\preceq}\{\pi(G) \mid \pi \text{ is a permutation}\}$. We say that G is canonical if $G = can(G)$.

Example 3 Consider the graphs of Figure 1. The graph G_3 is the canonical representation of G_1, G_2 and G_3.

Note that the canonical representation of a graph does not necessarily order the nodes by degree. In Figure 1, the nodes of G_2 are ordered by degree: nodes $\{1,2,3,4\}$ are of degree 2, nodes $\{5,6,7,8\}$ are of degree 3 and node 9 is degree 4. But this is not the case for the canonical form, G_3.

3 Symmetry breaking on Representation

We first consider a symmetry breaking predicate, introduced without proof in [Miller and Prosser, 2012], which constrains the rows of the adjacency matrix to be sorted lexicographically in non-decreasing order.

Definition 6 (lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix. We define

$$\mathsf{sb}_\ell(A) = \bigwedge_{i=1}^{n-1} A[i] \preceq A[i+1]$$

Observe the graphs in Figure 1. We have $\mathsf{sb}_\ell(A_{G_1}) = \mathit{false}$, $\mathsf{sb}_\ell(A_{G_2}) = \mathit{false}$, and $\mathsf{sb}_\ell(A_{G_3}) = \mathit{true}$.

Definition 6 is more subtle than might first appear. It defines a symmetry breaking predicate only because for every adjacency matrix A, $\mathsf{sb}_\ell(A')$ is true for at least one of the matrices A' isomorphic to A. No such proof is provided in [Miller and Prosser, 2012]. In fact, were we to reverse the order, taking $A[i] \succeq A[i+1]$ instead, it would not define a symmetry breaking constraint. Consider for example any representation of the graph G with 2 nodes and a single edge. Then $A_G[1] \not\succeq A_G[2]$. The subtlety arises because, in contrast to the case of breaking symmetries in matrix problems where rows and columns can be reordered, such as in [Gent *et al.*, 2002; Flener *et al.*, 2002; Frisch *et al.*, 2003], here we need to reorder rows and columns both in the same way. To prove the correctness of Definition 6 it is sufficient to show that $\mathsf{sb}_\ell(can(A))$ holds.

Theorem 1 *Let G be a graph. Then* $\mathsf{sb}_\ell(can(A_G))$.

Proof: Let A be canonical and assume to the contrary that A does not satisfy $\mathsf{sb}_\ell(A)$. Let i be such that $A[i] \not\preceq A[i+1]$. It follows that there is a j such that for every $1 \leq j' < j$, $A[i,j'] = A[i+1,j']$ and $A[i,j] = 1$ and $A[i+1,j] = 0$. Let A' be the matrix obtained by swapping rows $i, i+1$ as well as columns $i, i+1$. We show that $A' \prec A$ in contradiction to A being canonical. Considering that $A[i,j] = 1$, so $i \neq j$ and there are two cases. We detail the case for $i < j$. The other case is similar.

If $i < j$, note that because the $j-1$ length prefixes of $A[i]$ and $A[i+1]$ are equal, hence $A[i,1] \cdots A[i,j-1] = A'[i,1] \cdots A'[i,j-1]$. Note also that $A[i'] = A'[i']$ for $i' < i$ (the i and $i+1$ elements in $A[i']$ are equal because A is symmetric and $A[i,i'] = A[i+1,i']$). It follows that the first cell to differ in A and A' is $A[i,j] = 1$ and $A'[i,j] = 0$. So $A' \prec A$. Contradiction. □

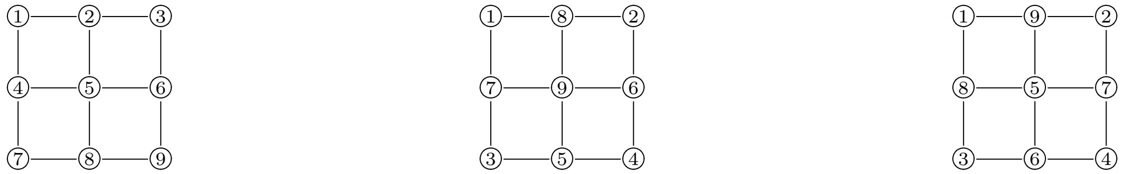
We now proceed to strengthen this notion of symmetry breaking. The following example illustrates a symmetry not captured by $\mathsf{sb}_\ell(A)$.

Example 4 Consider the adjacency matrix A_1 depicted in Figure 2 for which $\mathsf{sb}_\ell(A_1) = \mathit{true}$ as the rows are ordered lexicographically. Observe that $A[2] \preceq A[3]$ independent of whether we swap the nodes (rows and columns) 2 and 3, or not. Adjacency matrix A_2 depicted in Figure 2 is the result of this swap and it too satisfies $\mathsf{sb}_\ell(A_2) = \mathit{true}$. However, it is "closer" to canonical as $A_2 \preceq A_1$. Indeed A_2 is the canonical representative of this graph. Figure 2 highlights that the first 3 elements of rows 2 and 3 are invariant under node swap.

In view of Example 4 we introduce the following definition and then introduce a stronger symmetry breaking constraint.

Definition 7 (extended lexicographic order)

Let s be a sequence and $I \subseteq \{1, \ldots, |s|\}$. We denote by $(s \restriction I)$ the sequence obtained from s by simultaneously omitting the elements at positions I. For a set of natural numbers I we denote by $\preceq_I$ the order on sequences of length at least $\max(I)$ defined by: $s_1 \preceq_I s_2 \Leftrightarrow (s_1 \restriction I) \preceq (s_2 \restriction I)$.



G_1 G_2 G_3

$$\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
2 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
5 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
6 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
8 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
9 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}$$

A_{G_1}

$$\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
6 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
7 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
8 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
9 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0
\end{array}$$

A_{G_2}

$$\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
6 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
7 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
8 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
9 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}$$

A_{G_3}

Figure 1: Three example graphs and their adjacency matrices

$$\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 0 & 0 & 1 \\
2 & 0 & 0 & 1 & 1 \\
3 & 0 & 1 & 0 & 0 \\
4 & 1 & 1 & 0 & 0
\end{array}$$

A_1

$$\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 0 & 0 & 1 \\
2 & 0 & 0 & 1 & 0 \\
3 & 0 & 1 & 0 & 1 \\
4 & 1 & 0 & 1 & 0
\end{array}$$

A_2

Figure 2: Graphs and adjacency matrices for Example 4.

Definition 8 (improved lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix. We define

$$\mathsf{sb}^*_\ell(A) = \bigwedge_{i<j} A[i] \preceq_{\{i,j\}} A[j]$$

Observe that Definition 8 introduces $O(n^2)$ constraints on lexicographic order whereas Definition 6 introduces only $O(n)$. This is needed because we lack a "transitivity" like property stating that if $s_1 \preceq_{\{i,j\}} s_2$ and $s_2 \preceq_{\{j,k\}} s_3$ then also $s_1 \preceq_{\{i,k\}} s_3$. But this does not hold as illustrated by the following example.

Example 5 Consider the adjacency matrix A shown in Figure 3. While clearly $A[1] \preceq_{\{1,2\}} A[2]$ and $A[2] \preceq_{\{2,4\}} A[4]$, it is not the case that $A[1] \preceq_{\{1,4\}} A[4]$.

Interesting, transitivity does hold for rows two apart.

Theorem 2 $A[i] \preceq_{\{i,i+1\}} A[i+1] \wedge A[i+1] \preceq_{\{i+1,i+2\}} A[i+2] \Rightarrow A[i] \preceq_{\{i,i+2\}} A[i+2]$

Proof: Assume the premise and adopt the following representation where the boxed elements are at positions i, $i+1$ and $i+2$.

$$\begin{aligned}
A[i] &= S_1\boxed{0\ x\ y}T_1 \\
A[i+1] &= S_2\boxed{x\ 0\ z}T_2 \\
A[i+2] &= S_3\boxed{y\ z\ 0}T_3
\end{aligned}$$

From the premise by definition of $\preceq_I$, we have $S_1yT_1 \preceq S_2zT_2$ and $S_2xT_2 \preceq S_3yT_3$. We prove that $S_1xT_1 \preceq S_3zT_3$ which gives the result. There are two cases: either **(a)** $S_1 \prec S_2$, and since $S_2 \preceq S_3$ we have $S_1 \prec S_3$ and the result holds; or **(b)** if $S_1 = S_2$, then either $S_2 \prec S_3$, and the result follows, or $S_2 = S_3$, and it remains to show that $xT_1 \preceq zT_3$. So we have $S_1 = S_2 = S_3$. Suppose $y \prec z$ then since $x \preceq y$ we have that $x \prec z$ and the result holds. Otherwise, $y = z$. Suppose $x \prec y$ then clearly $x \prec z$ and the result holds. So assume $x = y = z$. Then we have that $T_1 \preceq T_2$ and $T_2 \preceq T_3$ and hence the result holds. ☐

So we can refine Definition 8.

Corollary 1

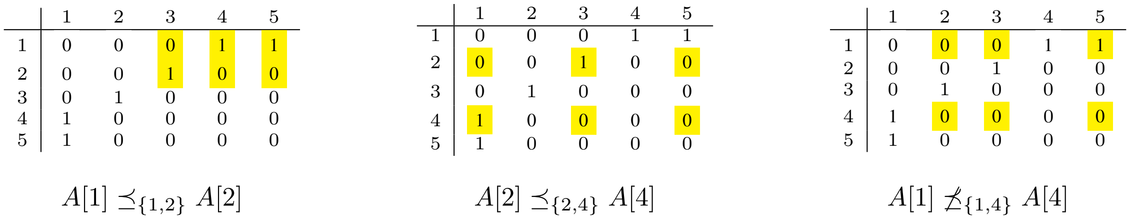
$$\mathsf{sb}^*_\ell(A) = \bigwedge_{\substack{i<j \\ j-i\neq 2}} A[i] \preceq_{\{i,j\}} A[j]$$

The following proves that sb^*_ℓ is a symmetry-breaking predicate.

Theorem 3 *Let A be a canonical adjacency matrix. Then $\mathsf{sb}^*_\ell(A)$ holds.*

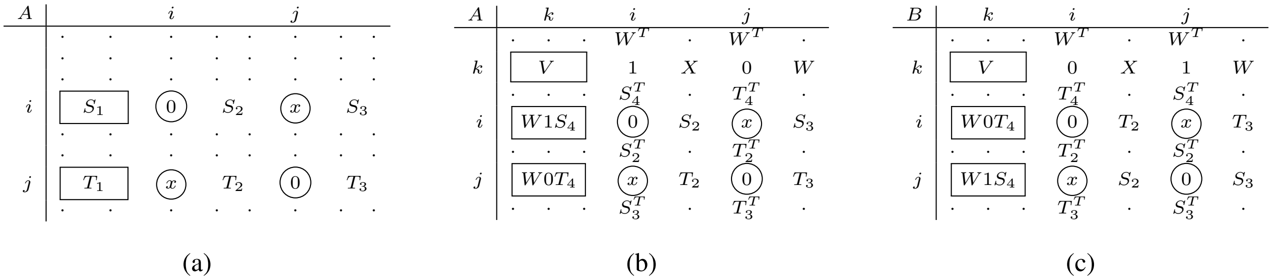
Proof: Let A be the canonical adjacency matrix for a graph G and assume to the contrary that A does not satisfy $\mathsf{sb}^*_\ell(A)$. That is, there exist i and j such that $i < j$ and $A[i] \npreceq_{\{i,j\}} A[j]$. Let π denote the permutation which swaps nodes i and j in G. We show that $B = A_{\pi(G)} \prec A$.

We denote the i and j rows, $A[i] = S_1\textcircled{0}S_2\textcircled{x}S_3$ and $A[j] = T_1\textcircled{x}T_2\textcircled{0}T_3$ such that the circled 0 and x are at positions i and j in $A[i]$ and at positions j and i in $A[j]$. They are circled to correspond to Figure 4(a). Since $A[i] \npreceq_{\{i,j\}} A[j]$, hence $S = S_1S_2S_3 \succ T_1T_2T_3 = T$. Since $S \succ T$, hence S and T must have prefixes of the form $W1$ and $W0$, respectively. Namely, W is a common prefix followed by a 1 in S and by a 0 in T. Let k be the column where this first



$A[1] \preceq_{\{1,2\}} A[2]$ $A[2] \preceq_{\{2,4\}} A[4]$ $A[1] \npreceq_{\{1,4\}} A[4]$

Figure 3: Extended lexicographic comparisons for Example 5 where only the highlighted entries are compared.



(a) (b) (c)

Figure 4: Diagram illustrating Theorem 3. (a) shows the basic form of rows i and j; (b) illustrates the original adjacency matrix for the first case of the proof where we assume $S_1 = W1S_4$ and $T_1 = W0T_4$; and (c) represents the adjacency matrix after swapping i with j.

difference occurs (so $|W| = k - 1$). It is not possible that $k = i$ since this column does not occur in S and T.

Consider the case where $k < i$. The first k-1 rows in A and B are identical. This is clear except for columns i and j. But these are the same as rows i and j (because of symmetry) and hence identical in the first $k - 1$ positions. We show that $B[k] \prec A[k]$. Let $A[k] = V1X0W$ where $|V| = i - 1$ and $|V1X| = j - 1$, then by construction $B[k] = V0X1W$ and $B \prec A$. The proof for the second case, $k > i$, is similar. $\square$

Note that often we may wish to separate nodes of the graph into equivalence classes a priori, and generate a graph that satisfies those equivalence classes. We can still use (extended) lexicographic ordering to help constrain the resulting adjacency matrices, since we can extend Theorem 3 to this case.

Definition 9 (ordered partition)] Let G be a graph. Then $P = \{P_1, \dots, P_p\}$ is an *ordered partition* of the nodes of G if $\forall 1 \le i < j \le p$ then $v_i \in P_i \wedge v_j \in P_j \Rightarrow v_i < v_j$.

Definition 10 (partition preserving permutation) Let $P = \{P_1, \dots, P_p\}$ be an ordered partition on the nodes of G. A permutation π on the nodes of G is *partition preserving* for P if $\forall 1 \le i \le p, \forall v_i \in P_i, \pi(v_i) \in P_i$.

Example 6 Consider the graph G_2 from Figure 1 and the ordered partition $P = \{\{1,2,3,4\},\{5,6,7,8\},\{9\}\}$, which partitions vertices by degree. Then the permutation $\pi = (2,3,4)$ is partition preserving for P. It only maps elements in P_1 to other elements in P_1 and is otherwise the identity.

Definition 11 (canonical partitioned adjacency matrix) The canonical form of a graph G with respect to an ordered partition P is the graph $can(G,P) = \min_{\preceq}\{\pi(G) \mid \pi \text{ is a partition preserving permutation for } P\}$. We say that G is canonical for P if $G = can(G,P)$.

We can define a symmetry breaking predicate for partitioned graphs as follows:

Definition 12 (partitioned lexicographic symmetry break) Let A be an $n \times n$ adjacency matrix and $P = \{P_1, P_2, \dots, P_p\}$ be an ordered partition. We define

$$\mathsf{sb}^*_\ell(A,P) = \bigwedge_{k=1}^{p} \bigwedge_{\{i,j\}\subseteq P_k, i<j} A[i] \preceq_{\{i,j\}} A[j]$$

Theorem 4 *Let G be a canonical partitioned graph for an ordered partition P. Then $\mathsf{sb}^*_\ell(A_G,P)$ holds.*

Proof: Let A be the canonical adjacency matrix for graph G and assume to the contrary that A does not satisfy $\mathsf{sb}^*_\ell(A,P)$. That is, there exists a partition P_k and $\{i,j\} \subseteq P_k$ with $i < j$ where $A[i] \npreceq_{\{i,j\}} A[j]$. We show that $B = A_{\pi(G)} \prec A$ where π swaps i and j. Note that π is a partition preserving permutation for P. The remainder of the proof is essentially identical to that of Theorem 3. $\square$

4 Extremal Graph Problems

We apply a constraint-based approach to extremal graph problems and illustrate the advantage of symmetry breaking on the graph representation.

The girth of a graph is the size of the smallest cycle contained in it. Let $\mathcal{F}_k(v)$ denote the set of graphs with v vertices and girth at least $k+1$. Let $f_k(v)$ denote the maximum number of edges in a graph in $\mathcal{F}_k(v)$. A graph in $\mathcal{F}_k(v)$ with $f_k(v)$ edges is called *extremal*. The number of non-isomorphic extremal graphs in $\mathcal{F}_k(v)$ is denoted $F_k(v)$. Extremal graph problems are about discovering values of $f_k(v)$ and $F_k(v)$ and about finding witnesses. In [Abajo and Diánez, 2010] the

$$\forall_{1\le i<j\le v}.\ (A[i,j] \equiv A[j,i] \quad \text{and} \quad A[i,i] \equiv \mathit{false}) \tag{1}$$

$$\forall_{i,j,k}.\ A[i,j] + A[j,k] + A[k,i] < 3 \tag{2}$$

$$\forall_{i,j,k,l}.\ A[i,j] + A[j,k] + A[k,l] + A[l,i] < 4 \tag{3}$$

$$\sum_{1\le i<j\le v} A[i,j] = e \tag{4}$$

$$\forall_{1\le i\le v}.\ \left(\begin{array}{l} \delta \le \sum\limits_{1\le j\le v} A[i,j] \le \Delta,\ \min\limits_{i}(\sum\limits_{1\le j\le v} A[i,j]) = \delta \\ \text{and } \ \max_i(\Sigma A[i]) = \Delta \end{array}\right) \tag{5}$$

Figure 5: Basic constraint model for extremal graph problems (no cycles of length 4 or less)

authors attribute the discovery of values $f_4(v)$ for $v \le 24$ to [Garnick *et al.*, 1993] and for $25 \le v \le 30$ to [Garnick and Nieuwejaar, 1992]. In [Garnick *et al.*, 1993] the authors report values of $F_4(v)$ for $v \le 21$. In [Garnick *et al.*, 1993] and [Wang *et al.*, 2001] the authors apply algorithms to compute lower bounds on $f_4(v)$ for $31 \le v \le 200$. Some of these lower bounds are improved in [Abajo *et al.*, 2010]. Values of $f_4(v)$ for $v \le 30$, and of $F_4(v)$ for $v \le 21$ are available as sequences `A006856` and `A159847` of the On-Line Encyclopedia of Integer Sequences [OEIS, 2010].

Our basic constraint model is depicted as Figure 5 where we assume that A is a Boolean $v \times v$ matrix. Constraint (1) states that the graph is simple (symmetric with no self loops), Constraints (2) and (3) express that there are no cycles of length 3 and 4, and Constraint (4), that the number of edges is e. Constraints (2) and (3) are implemented more efficiently. We introduce additional Boolean variables for each triplet of (distinct) vertices i, j, k with $i < k$: $x_{i,j,k} \leftrightarrow A[i,j] \wedge A[j,k]$ represents a length 2 path between i and k via j; and $x_{i,k} \leftrightarrow \vee\{x_{i,j,k} \mid j \ne i, j \ne k\}$ represents the existence of any length 2 path between i and k. We then express Constraints (2) and (3) as $\forall_{i,k}.\ A[i,k] + x_{i,k} < 2$ and $\forall_{i,k}.\ \sum_j x_{i,j,k} < 2$.

To explain Constraint (5) we recall Propositions 2.6 and 2.7 from [Garnick *et al.*, 1993] which state that for every graph in $\mathcal{F}_4(v)$ with e edges the minimum and maximum vertex degrees, denoted δ and Δ, satisfy the following equations (assuming $v \ge 1$):

$$\begin{array}{l} v \ge 1 + \Delta\delta \ge 1 + \delta^2, \text{ and} \\ \delta \ge e - f_4(v-1), \text{ and } \Delta \ge \lceil 2e/v \rceil \end{array} \tag{$\star$}$$

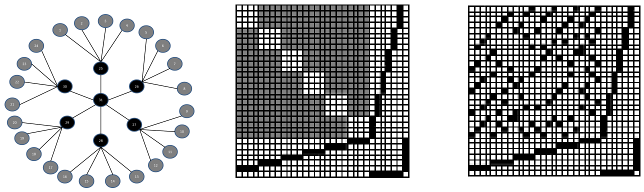
Given values for v and e we model the problem separately for each potential pair (δ, Δ) introducing Constraint (5). In addition to the above constraints we introduce symmetry breaking constraints sb_ℓ or sb^*_ℓ.

Example 7 For $v = 31$ and $e = 80$ the possible (δ, Δ) pairs satisfying Equation ($\star$) are $\{(4,6), (4,7), (5,6)\}$. Similarly, for $v = 31$ and $e = 81$ the single pair is $(5,6)$.

We describe three experiments to evaluate the impact of different symmetry breaking strategies. Experiments were run using two different constraint solvers, BEE [Metodi and Codish, 2012] and Choco [Laburthe and Jussien, 2004]. We present the results obtained using BEE which compiles finite domain constraints to CNF and solves them using an underlying SAT solver. Our configuration uses CryptoMiniSat v2.5.1 [Soos, 2010]. BEE performs CNF simplification by applying a constraint-driven technique called equi-propagation [Metodi *et al.*, 2011] and partial evaluation. All experiments are performed on a single core of an Intel(R) Core(TM) i5-2400 3.10GHz CPU with 4GB memory under Linux (Ubuntu lucid, kernel 2.6.32-24-generic). BEE is written in Prolog and run using SWI Prolog v6.0.2 64-bits. All experiments were replicated and verified using the Choco constraint programming toolkit.

		no sym break		sb_ℓ		sb^*_ℓ	
v	f$_4$(v)	e = f$_4$(v)	e = f$_4$(v)+1	e = f$_4$(v)	e = f$_4$(v)+1	e = f$_4$(v)	e = f$_4$(v)+1
11	16	0.08	∞	0.07	0.37	0.12	0.22
12	18	0.06	0.00	0.06	0.00	0.07	0.00
13	21	0.09	0.00	0.09	0.00	0.11	0.00
14	23	0.14	0.23	0.12	0.87	0.15	0.45
15	26	0.14	0.00	0.15	0.00	0.16	0.00
16	28	0.17	213.56	0.23	4.45	0.23	3.45
17	31	0.10	0.01	0.13	0.01	0.10	0.01
18	34	0.14	0.01	0.22	0.01	0.16	0.01
19	38	0.11	0.00	0.13	0.00	0.11	0.00
20	41	9.46	0.00	6.17	0.00	0.20	0.00
21	44	5.37	1796.56	39.94	0.62	4.53	0.49
22	47	2.17	∞	19.16	32.43	228.81	28.66
23	50	0.51	∞	26.47	2566.82	69.27	2391.52
24	54	9.86	∞	1.04	0.00	40.89	0.00
31	80	15.28	∞	1.42	2408.63	1.65	2373.49
32	85	588.78	0.00	125.07	0.00	100.35	0.00

Table 1: Computing $f_4(v)$ (time in seconds; timeout 4hrs)



(a) (b) (c)

Figure 6: (a) the star $S_{6,4}$, (b) its adjacency matrix, and (c) a member of $\mathcal{F}_4(31)$ with 80 edges (0=white, 1=black, not determined = gray)

4.1 Experiment 1: computing $f_4(v)$

Table 1 summarizes the results for a constraint-based approach to compute values of $f_4(v)$. We compare the computation time for three configurations specified as columns for: no symmetry, and breaking symmetries using sb_ℓ and sb^*_ℓ.

For smaller instances, $v \le 15$, we apply the constraint model from Figure 5. For larger instances, $16 \le v \le 24$, we add an additional constraint to the model. To this end we follow [Garnick *et al.*, 1993] where it is noted that every graph in $\mathcal{F}(v)$ with at least 5 vertices contains a $(\Delta, \delta - 1)$-star. In general, an (m, n)-star is a rooted tree, denoted $S_{m,n}$, with m children, each of which has $n \ge 1$ children, all of which are leaves. So, we add constraints to explicitly embed $S_{\Delta,\delta-1}$ in the adjacency matrix. In this setting, based on Theorem 4 we impose symmetry breaking on the m clusters of n leaves of $S_{m,n}$ as well as to the cluster of nodes not in $S_{m,n}$. Figure 6(a) illustrates the star $S_{6,4}$. A 31×31 adjacency matrix with an embedded $S_{6,4}$ is depicted as Figure 6(b). Black and white cells indicate values 1 and 0 respectively, and gray cells indicate unassigned Boolean variables. The last row of the matrix corresponds to the root. Then moving up we find the 6 children of the root, and then its 24 grandchildren. We will later return to explain Figure 6(c). Finally, in the third part of the table, we consider two open instances. Here we also consider a model with the constraints that embed the star.

For each value of v and each type of symmetry break, we search for a graph with $f_4(v)$ edges (columns $\mathrm{e}=\mathrm{f}_4(\mathrm{v})$), and show the non-existence of a graph with $f_4(v) + 1$ edges (columns $\mathrm{e}=\mathrm{f}_4(\mathrm{v})+1$). Examining the three $\mathrm{e}=\mathrm{f}_4(\mathrm{v})$ columns, it appears that there is no significant gain in symmetry breaking when the instance is satisfiable and we need only find a single witness. When instances are unsatisfiable (the three $\mathrm{e}=\mathrm{f}_4(\mathrm{v})+1$ columns) we encounter two types of instances: those which involve search and those which do not. For the later type, unsatisfiability derives from the propagation of the constraints in Equation ($\star$) and the computation is fast for all three configurations. For the other instances, the solver must explore the entire search space and symmetry breaking is then useful.

The bottom two rows in Table 1 describe our results for two open instances, computing $f_4(31)$ and $f_4(32)$. A lower bound of $f_4(31) \geq 80$ is given in [Garnick *et al.*, 1993] and a witness (discovered using our model in less than 2 seconds) is depicted as Figure 6(c). It is canonical with respect to a partitioning where the first 24 rows form 6 clusters of size 4 each (the grandchildren), the next 6 rows form a cluster (the children), and the last row is a singleton cluster (the root). With the proof that there is no witness with 81 edges (determined using our model in 40 minutes of CPU time) we conclude that $f_4(31) = 80$. Given that $f_4(31) = 80$, Equation ($\star$) implies that $f_4(32) \leq 85$ and hence that the lower bound $f_4(32) \geq 85$ reported in [Garnick *et al.*, 1993] is the precise value, consequently $f_4(32) = 85$. These are both new results.

4.2 Experiment 2: computing $F_4(v)$

In this experiment we apply a constraint-based approach to compute the number of non-isomorphic extremal graphs with v vertices. We apply a constraint solver to generate all graphs satisfying the constraint model for v vertices and $e = f_4(v)$ edges with corresponding symmetry breaking constraints. We then apply `nauty` to determine the number of non-isomorphic graphs within this set. The time required to run `nauty` is negligible and not detailed in our results.

For smaller values, $v \leq 15$, we consider the constraint model of Figure 5. Table 2 shows for each value of v the maximum number of edges $f_4(v)$, the number of non-isomorphs $F_4(v)$, and the number of graphs generated (columns $sols$) and computation time ($time$), for each of the three configurations. Our results are as expected: improving symmetry breaking makes a significant difference. The bottom two rows in Table 2 describe our results for two open instances, computing $F_4(24)$ and $F_4(32)$. To obtain these results we first extend, in the following lemma, the observation of Garnik regarding the embedding of a star for the two cases. We then explicitly embed the extended structures in the encoding.

v	$\mathrm{f}_4(\mathrm{v})$	$\mathrm{F}_4(\mathrm{v})$	no sym break		sb_ℓ		sb_ℓ^*	
			# sols	time	# sols	time	# sols	time
4	3	2	9	0.01	3	0.03	2	0.01
5	5	1	12	0.01	1	0.01	1	0.01
6	6	2	120	0.05	4	0.03	2	0.03
7	8	1	900	0.25	6	0.01	3	0.01
8	10	1	2520	1.95	4	0.02	1	0.02
9	12	1	10080	16.48	6	0.06	3	0.06
10	15	1	30240	48.96	2	0.03	1	0.04
11	16	3	—	∞	48	0.91	16	0.74
12	18	7	—	∞	469	1.93	192	0.75
13	21	1	—	∞	66	0.21	27	0.15
14	23	4	—	∞	2888	81.67	1021	16.02
15	26	1	—	∞	812	51.72	268	4.70
24	54	1	—	∞	144	32.93	144	15.03
32	85	1	—	∞	240	726.33	240	765.16

Table 2: computing $F_4(v)$ (time in seconds; timeout 4hrs)

v	$\mathrm{f}_5(\mathrm{v})$	no sym break	sb_ℓ	sb_ℓ^+	sb_ℓ^*
4	3	0.00	0.00	0.01	0.02
5	4	0.01	0.01	0.01	0.01
6	6	0.01	0.01	0.01	0.01
7	7	0.14	0.03	0.03	0.02
8	9	2.01	0.06	0.04	0.04
9	10	76.07	0.13	0.08	0.07
10	12	2224.65	0.30	0.02	0.15
11	14	∞	1.29	0.61	0.41
12	16	∞	4.97	3.49	1.30
13	18	∞	21.18	11.19	7.93
14	21	∞	85.73	43.78	20.29
15	22	∞	801.03	418.05	203.08
16	24	∞	∞	6076.86	1613.60
17	26	∞	∞	∞	13903.10

Table 3: computing $f_5(v)$ (time in seconds; timeout 4hrs)

Proposition 1 ***(a)*** *Every extremal graph G with 24 nodes contains a star $S_{5,3}$ in which the 5 children of the root have degrees $5, 5, 5, 4, 4$ in G.* ***(b)*** *Every extremal graph G with 32 nodes contains a star $S_{6,4}$ in which the 6 children of the root have degrees $6, 5, 5, 5, 5, 5$ in G.*

4.3 Experiment 3: computing $f_5(v)$

For our final experiment we consider the extremal graphs which contain no cycles of length 5 or less. To this end we extend the basic constraint model of Figure 5 with an additional constraint that states that every sequence of five vertices does not form a cycle, and we consider the optimization problem which computes values of $f_5(v)$. Table 3 shows our results. To better illustrate the impact of improved lexicographic symmetry breaking we consider also a predicate sb_ℓ^+ which is like sb_ℓ^* but only compares consecutive rows of the matrix. It is clear that the much larger sb_ℓ^* pays off, reducing computation time considerably for the larger instances.

5 Conclusion

Symmetry breaking for graph representations using sb_ℓ is considered also in [Miller and Prosser, 2012] where it is shown to have a considerable impact. However, no formal justification is provided. We address in general terms the application of symmetry breaking to improve the search for an undirected graph satisfying a given property. We formally justify the use of sb_ℓ. We also introduce and formally justify the use of sb_ℓ^* which is a more powerful symmetry breaking predicate. We demonstrate the impact of symmetry breaking to extremal graph theory where we also close several open instances.

References

[Abajo and Diánez, 2010] E. Abajo and Ana Diánez. Exact values of ex(ν;$\{C_3, C_4, ..., C_n\}$). *Discrete Applied Mathematics*, 158(17):1869–1878, 2010.

[Abajo *et al.*, 2010] E. Abajo, Camino Balbuena, and Ana Diánez. New families of graphs without short cycles and large size. *Discrete Applied Mathematics*, 158(11):1127–1135, 2010.

[Arvind and Torán, 2005] Vikraman Arvind and Jacobo Torán. Isomorphism testing: Perspective and open problems. *Bulletin of the EATCS*, 86:66–84, 2005.

[Bollobás, 1978] Béla Bollobás. *Extremal Graph Theory*. Courier Dover Publications, 1978.

[Crawford *et al.*, 1996] James M. Crawford, Matthew L. Ginsberg, Eugene M. Luks, and Amitabha Roy. Symmetry-breaking predicates for search problems. In Luigia Carlucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *KR*, pages 148–159. Morgan Kaufmann, 1996.

[Faulon, 1998] Jean-Loup Faulon. Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. *Journal of Chemical Information and Computer Sciences*, 38(3):432–444, 1998.

[Flener *et al.*, 2002] Pierre Flener, Alan M. Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, Justin Pearson, and Toby Walsh. Breaking row and column symmetries in matrix models. In Pascal Van Hentenryck, editor, *CP*, volume 2470 of *Lecture Notes in Computer Science*, pages 462–476. Springer, 2002.

[Frisch *et al.*, 2003] Alan M. Frisch, Christopher Jefferson, and Ian Miguel. Constraints for breaking more row and column symmetries. In Francesca Rossi, editor, *CP*, volume 2833 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2003.

[Gardiner, 2011] Eleanor J. Gardiner. Graph applications in chemoinformatics and structural bioinformatics. In Sherif Sakr and Eric Pardede, editors, *Graph Data Management*, pages 386–420. IGI Global, 2011.

[Garnick and Nieuwejaar, 1992] David K. Garnick and A. Nieuwejaar. Non-isomorphic extremal graphs without three-cycles or four-cycles. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 12:33–56, 1992.

[Garnick *et al.*, 1993] David K. Garnick, Y. H. Harris Kwong, and Felix Lazebnik. Extremal graphs without three-cycles or four-cycles. *Journal of Graph Theory*, 17(5):633–645, 1993.

[Gent *et al.*, 2002] I.P. Gent, P. Prosser, and B.M. Smith. A 0/1 encoding of the gaclex constraint for pairs of vectors. In *ECAI 2002 workshop W9: Modelling and Solving Problems with Constraints*, 2002.

[Laburthe and Jussien, 2004] François Laburthe and Narendra Jussien. *The Choco Solver*, 2004. `http://choco.mines-nantes.fr/`.

[McKay, 1990] B. McKay. *nauty* user's guide (version 1.5). Technical Report TR-CS-90-02, Australian National University, Computer Science Department, 1990.

[Metodi and Codish, 2012] Amit Metodi and Michael Codish. Compiling finite domain constraints to SAT with BEE. *TPLP*, 12(4-5):465–483, 2012.

[Metodi *et al.*, 2011] Amit Metodi, Michael Codish, Vitaly Lagoon, and Peter J. Stuckey. Boolean equi-propagation for optimized SAT encoding. In Jimmy Ho-Man Lee, editor, *CP*, volume 6876 of *Lecture Notes in Computer Science*, pages 621–636. Springer, 2011.

[Miller and Prosser, 2012] Alice Miller and Patrick Prosser. Diamond-free degree sequences. *Acta Univ. Sapientiae, Informatica*, 4(2):189–200, 2012.

[OEIS, 2010] The on-line encyclopedia of integer sequences. published electronically at `http://oeis.org`, 2010.

[Read, 1978] Ronald C. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Annals of Discrete Mathematics*, 2:107120, 1978.

[Shlyakhter, 2007] Ilya Shlyakhter. Generating effective symmetry-breaking predicates for search problems. *Discrete Applied Mathematics*, 155(12):1539–1548, 2007.

[Soos, 2010] Mate Soos. CryptoMiniSAT, v2.5.1. `http://www.msoos.org/cryptominisat2`, 2010.

[Wang *et al.*, 2001] Ping Wang, Gerhard W. Dueck, and S. MacMillan. Using simulated annealing to construct extremal graphs. *Discrete Mathematics*, 235(1-3):125–135, 2001.

[Washio and Motoda, 2003] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *SIGKDD Explorations*, 5(1):59–68, 2003.