Group-graphs associated with Row and Column Symmetries of Matrix Models: some observations

Zeynep Kiziltan[1] and Michela Milano[2]

[1] Department of Information Science, Uppsala University, Uppsala, Sweden
Zeynep.Kiziltan@dis.uu.se
[2] DEIS, University of Bologna, Bologna, Italy
mmilano@deis.unibo.it

Abstract. The effect of symmetry-breaking constraints is often evaluated empirically. In order to understand which symmetric configurations are removed by a set of constraints, we have to understand the underlying structure of the symmetry group in concern. A class of symmetry that frequently occurs in constraint programming is the row and column symmetries of a matrix model. In this paper, we study these symmetries from a structural viewpoint, and show how the graph of the associated group can be built. The graph can help us to understand the effect of certain symmetry-breaking constraints posted on a matrix model, though some questions remain open. This paper is a preliminary study on the relations between group properties and the corresponding graph and symmetry-breaking constraints.

1 Introduction

Symmetries are ubiquitous in many Constraint Satisfaction Problems (CSPs). Symmetry in a CSP can involve the variables, the values, or both, and map each search state (e.g., a partial assignment, a solution and a failure) to an equivalent one. An exhaustive search method spends time in visiting equivalent states if symmetries are not eliminated [5].

Symmetries of a CSP form a group. Thus, we can exploit results coming from the *group theory* to understand the structure of each particular symmetry. In addition, each group has a corresponding graph which again can help for this purpose.

A class of symmetries that frequently occur in constraint programming is the row and column symmetries of a matrix model [2]. A matrix model is a constraint program that contains one or more matrices of decision variables. Many CSPs can be easily represented by matrix models [3] in which the matrices may have symmetry between their rows and/or columns. Such symmetries are referred to as *row and column symmetries*. Two matrices are symmetric if one can be obtained from the other by row and/or column permutations.

In this paper, we analyse the group describing the row and column symmetries, in order to understand their underlying structure. The elements of the group are all the symmetric matrices of a given matrix. To obtain all permutations, only two generators for rows, and two for columns should be considered: the flip of the first two rows (resp. columns), and a shift that leads the first row (resp. column) to the last position. We have observed that the resulting group-graph has a very interesting structure.

Our ultimate aim is to provide some considerations for studying, from a structural point of view, which configurations are removed when we add different sets of symmetry-breaking constraints to a matrix model so as to remove row and column symmetries. Thus, this work can be seen as a starting point for a deeper insight on this topic. In general, approaches proposing symmetry-removal algorithms or constraints experimentally evaluate the effectiveness of the method. Here, we propose a more formal perspective which could possibly be used to compare different approaches from a structural point of view, and could help to devise new algorithms and symmetry-breaking constraints.

This paper is a preliminary step towards this more general and ambitious aim, and we think it deserves more investigation. We here restrict our focus on small square matrices (3×3), but our observations could be generalized for bigger matrices.

2 Groups and their graphs

Recently, group theory has been used to describe the symmetries of a CSP, and to reduce the effort to remove them (e.g., [4]). A group is a tuple $G = (S, Op)$ where S is a set and Op is a closed binary operation over S. A group has the following properties:

- the operation Op is associative, i.e., for any $x, y, z \in S,\ (x\ Op\ y)\ Op\ z = x\ Op\ (y\ Op\ z)$.
- there is a neutral element I, i.e., for any $x \in S,\ x\ Op\ I = I\ Op\ x = x$.
- each element x in S has an inverse x^{-1}, i.e., for any $x \in S,\ x\ Op\ x^{-1} = x^{-1}\ Op\ x = I$.

As an example, we consider the permutation of n elements. We have a group $G = (S, Op)$ (called *permutation group*), where S contains $n!$ elements, each corresponding to a permutation of n elements, i.e., a bijective mapping from S to itself. The operation Op is the function composition, if we consider functions as the elements of S. It can easily be shown that the function composition is closed and associative, the neutral element is the identity permutation, and every permutation has an inverse. In a permutation group, we can consider a minimal set of permutations whose compositions gives all possible permutations. Permutations belonging to this minimal set are called *generators*. Each generator has a *period*, i.e., the number of applications of the generator to a permutation so as to obtain itself.

For the permutation group, we have two generators. The first is identified as f and corresponds to the flip of the elements in positions 1 and 2. The second is

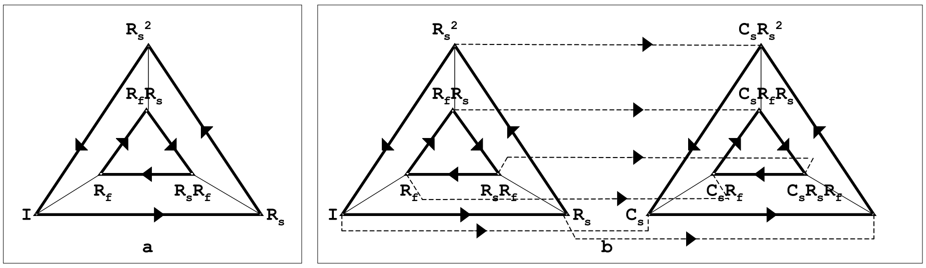

Fig. 1. a. Group-graph of row symmetries. **b.** Group-graph of row symmetries plus a column movement.

identified as s and corresponds to a shift, carrying the first element to the last position.

Every group has a corresponding graph, where a vertex corresponds to a configuration (i.e., an element of the group), and an arc represents the application of a generator to a configuration. Two configurations A and B are characterised by a distance: the number of generator applications transforming A to B.

As a further example, the size of the permutation group of 3 elements is 6 (=3!), and thus the graph associated with this group has 6 vertices. Each vertex has 2 outgoing and 2 incoming arcs. Each arc corresponds to a generator. Each generator has a period, i.e., the number of applications of the generator to a permutation so as to obtain itself. The period of f is 2, and the period of s is 3.

3 The group describing row and column symmetries

In this section, we use group theory to understand the structure of the row and column symmetries of a matrix model. This is a preliminary study: in fact, we restrict ourselves to small matrices (3×3) but we believe this study can be extended for larger matrices.

In a 3×3 matrix, we have 9 variables $I = [X_1, \ldots, X_9]$ ordered from the top left corner to the bottom right one. Let us now consider only the row symmetry. We have 3 rows $[X_1, X_2, X_3]$, $[X_4, X_5, X_6]$, and $[X_7, X_8, X_9]$, subject to permutation. The set of all possible row permutations forms a group $GR = (S_r, \circ)$. This group has the same structure (permutation) of the one described in Section 2: S_r is the set of configurations symmetric to the identity matrix I, and the operation $\circ$ is the function composition.

The row symmetry group has two generators: the flip of the first two rows R_f, and the row shift R_s that leads the first row to the last position. The period of R_f is 2, while the period of R_s is 3. Each configuration in S_r can be identified by various composition of the 2 generators applied to the identity matrix I. Figure 1.a reports the corresponding graph which has 6 (=3!) vertices. Each vertex has 2 outgoing and 2 incoming arcs. Each arc has a direction. Note that in the figure, a bi-directional arc (corresponding to R_f and its inverse) is replaced by

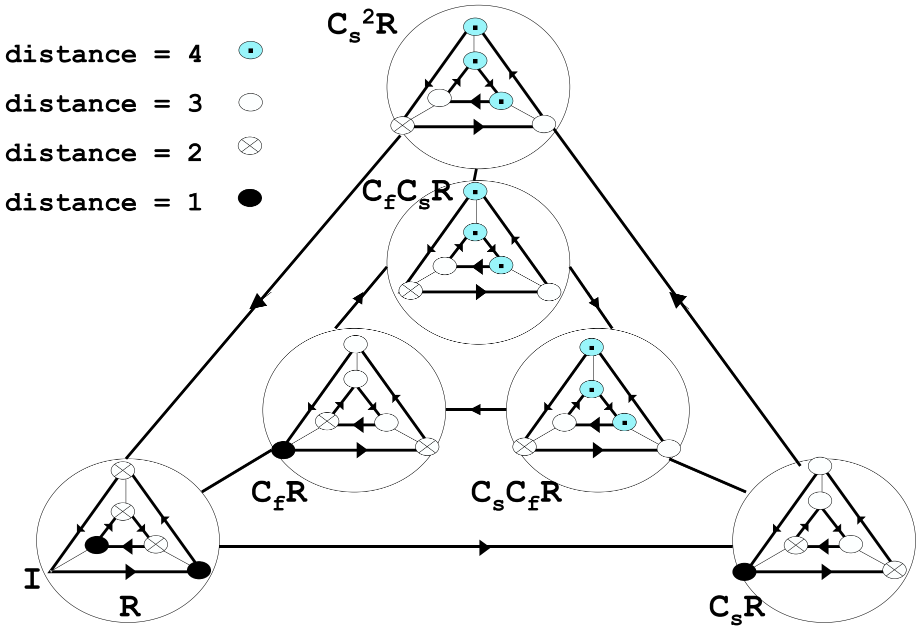

Fig. 2. Group-graph of the row and column symmetries of a 3×3 matrix.

an undirected arc. The configurations are labelled[1] in the figure as I, R_s, $R_s \circ R_s$ (hereinafter referred to as R_s^2), R_f, $R_s \circ R_f$, $R_f \circ R_s$. Note that the configuration $R_s \circ R_f$ can also be obtained by $R_f \circ R_s^2$.

We now consider the column symmetry. We have 3 columns $[X_1, X_4, X_7]$, $[X_2, X_5, X_8]$, and $[X_3, X_6, X_9]$, subject to permutation. Similar to the row symmetry case, the set of all possible column permutations forms a group with two generators: the column flip C_f and the column shift C_s. The size of the group is 6 (=3!) and its elements are I, C_s, C_s^2, C_f, $C_s \circ C_f$, $C_f \circ C_s$. Clearly, each of these permutations can be applied to each vertex of the group-graph in Figure 1.a. For instance, if we apply the permutation C_s to each vertex of the graph in Figure 1.a, then we obtain another graph, depicted in Figure 1.b. In this graph, whilst the leftmost triangle represents all the row permutations of the identity matrix I, the rightmost triangle represents all the row permutations of the matrix C_s obtained by applying a shift on the columns of the matrix I.

Since we have in total 6 column permutations, we obtain the group-graph of the row and column symmetries of a 3×3 matrix by applying all the 6 column permutations to each vertex of the graph in Figure 1.a. The resulting graph has 6 vertices (called *meta-vertices*), each of which is a graph with 6 vertices representing all the row permutations of a column permutation of the identity

[1] Vertices are labelled by the sequence of the generators applied to the identity matrix I.

matrix I, as depicted in Figure 2. The leftmost vertex of every meta-vertex is obtained from I by permuting its columns. In the rest of this paper, we will refer to every column permutation of the identity matrix I as the identity of the corresponding meta-vertex.

Each vertex in the graph has a distance with respect to the starting point, i.e., the minimum number of generators applied to reach the vertex from the identity matrix I. In Figure 2, each meta-vertex is labelled with the generator sequence to reach the vertex from the initial meta-vertex R, and each vertex is labelled with its distance.

4 Equivalence classes

In an equivalence class of matrices, any two matrices are symmetric, i.e., any matrix can be obtained from the other via row and/or column permutations. In Figure 2 we see that every 3×3 matrix has 36 symmetric configurations. Does this mean that the size of every equivalence class is 36? This is the case if the values in the matrix are all different. However, this is not always the case if there are repeating values in the matrix, because in a such case two symmetric matrices are not necessarily distinct. For instance, given $I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$, the matrix R_f is identical to I.

Although not every equivalence class has necessarily the same size, it is possible to know how big the equivalence classes can be. For instance, the columns of the matrix $I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ are all the same. Hence, any column permutation leaves the matrix unchanged. That is, in the group-graph associated with row and column symmetries, all the meta-vertices fall into the (leftmost) meta-vertex R that represents the row permutations of I. Moreover, two rows of I are the same, which means that a matrix within a meta-vertex falls into the one obtained by swapping those rows. This can be visualised in Figure 3. The size of the equivalence class containing the matrix I is thus $36/(2!3!)$, which is 3.

If a 3×3 matrix I has only the values 0 and 1 then there are 9 possible scenarios of the group-graph, giving rise to 9 possible size for the equivalence class C_I containing the matrix I:

1. Rows are all the same, and columns are all the same: in this case, all the meta-vertices fall into the meta-vertex R representing the row permutations of I. Also, all matrices within a meta-vertex fall into the identity matrix of the meta-vertex. Hence, $|C_I| = 36/(3!3!) = 1$.
2. Only 2 rows are the same, and columns are all the same: in this case, all the meta-vertices fall into R. Also, a matrix within a meta-vertex falls into the one obtained by swapping those rows. Hence, $|C_I| = 36/(2!3!) = 3$.
3. Rows are all the same, and only 2 columns are the same: this case is analogues to the previous case.
4. Only 2 rows are the same, and only 2 columns are the same: in this case, a meta-vertex falls into the one obtained by swapping those columns. Also, a

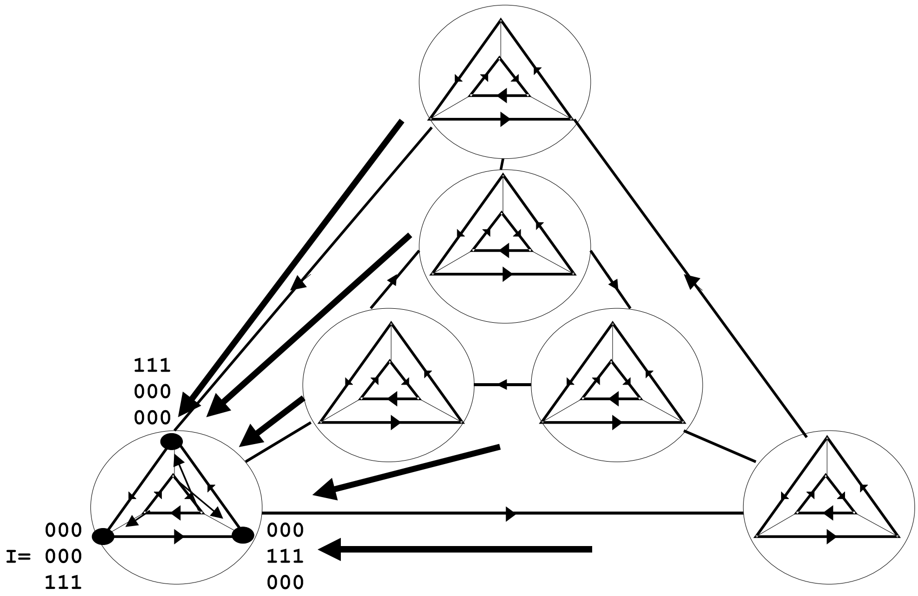

Fig. 3. Symmetric configurations of the matrix $I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$.

matrix within a meta-vertex falls into the one obtained by swapping those rows. Hence, $|C_I| = 36/(2!2!) = 9$.

5. Only 2 rows are the same: in this case, a matrix within a meta-vertex falls into the one obtained by swapping those rows. Hence, $|C_I| = 36/2! = 18$.
6. Only 2 columns are the same: in this case, a meta-vertex falls into the one obtained by swapping those columns. Hence, $|C_I| = 36/2! = 18$.
7. Every row permutation gives the same effect as a column permutation: in this case, all the meta-vertices fall into R. Hence, $|C_I| = 36/3! = 6$.
8. Swapping two columns gives the same effect as swapping two rows: in this case, a meta-vertex representing the swap of the columns falls into the one representing the corresponding row swaps. Hence, $|C_I|$ is 36/2, which is 18.
9. None of the above: in this case, clearly $|C_I| = 36$.

5 Symmetry-breaking constraints

A way to break all row and column symmetries is to add to the model a complete set of symmetry-breaking constraints, i.e., one constraint for each symmetry [1]. Consider a 3×3 matrix I, which we represent as $[X_1, \ldots, X_9]$. Now, we suppose to have an ordering relation $\leq$ among matrices. The complete set of symmetry-breaking constraints is composed by imposing $I \leq A$, where A is any matrix obtained by permuting the rows and/or columns of I. Since the number of symmetries is 36 (=3!3!), we need the same number of constraints. This complete set of constraints removes all symmetries but the identity.

As the matrix size enlarges, it becomes impractical to impose the complete set of symmetry-breaking constraints. In such a case, only a subset of these constraints could be used and thus not all symmetries are removed. In general, the effect of such symmetry-breaking constraints are evaluated experimentally, by for instance counting the number of symmetric solutions left unbroken. Here we provide some observations which could be generalized and used to evaluate the symmetry-breaking constraint methods.

One way of reducing much of row and column symmetries of a matrix model is to impose that the rows and the columns of the matrix are lexicographically ordered [2]. These constraints are a subset of the complete set of symmetry-breaking constraints, and prevent the row (resp. column) permutations. Hence, in an equivalence class, the symmetric configurations that are surely removed by these constraints reside on the leftmost meta-vertex R, as well as on the identity matrix of every other meta-vertex of the group-graph in Figure 2.

With the lexicographic ordering constraints, we observed that many other symmetric configurations in any equivalence class are also removed. One reason is that some matrices on the group-graph *collapse* into some others when there are repeating values in the matrix. If these matrices happen to fall into R and/or on the identity matrix of every other meta-vertex then lexicographic ordering constraints will remove these symmetric matrices. For instance, if a matrix is formed by only 0 and 1 then we know that there 9 possible scenarios of the group-graph as discussed in Section 4. In the 1st, 2nd, 3rd, and the 7th cases, all matrices fall into the ones that are removed by lexicographic ordering constraints. Hence, all symmetries are surely broken for such kind of equivalence classes. In all the other cases, matrices reside on the parts of the graph that are not reachable by the lexicographic ordering constraints. This hints that if there is any unbroken symmetric matrix, it must belong to one of the equivalence classes described by these cases.

As an example, consider the matrix $I = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, where two rows as well as two columns are identical. The equivalence class of this matrix is described by case 4. The configuration $C_s^2 R_s^2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ is obtained by applying two shifts on the columns and two shifts on the rows of I. This matrix resides on the topmost vertex of the graph and is not broken by the lexicographic ordering constraints.

In fact, for 0/1 values, 9 symmetric matrices remain to be unbroken, and they belong to the equivalence classes described by the 4th, 5th, 6th, and the 9th cases. It appears that all configurations at distance 1 and 2 from the identity are removed. Those 9 configurations left are all of distance 3 and 4.

Although we understand why some certain configurations are removed by this set of constraints, it is still an open question why all symmetries are broken for certain equivalence classes. For instance, for a 0/1 valued matrix I, all its symmetric configurations are removed if I belongs to an equivalence class described by case 8.

Note that if a matrix has few values then many symmetric configurations fall into the ones that are surely removed by the lexicographic ordering constraints. The more values a matrix can have the more it becomes difficult to remove the symmetric configurations using these symmetry-breaking constraints (i.e., lexicographically ordering the rows and columns).

6 Conclusions and Future Work

In this paper, we studied the structure of the group describing row and column symmetries of a matrix model, and showed how the associated graph can be constructed. We focused on a particular set of symmetry-breaking constraints that are posted on a matrix model so as to remove much of such symmetries. By studying the graph of the group describing these symmetries, we can see which symmetric configurations are removed by the constraints we considered. However, we still fully do not know how some symmetric configurations can be removed for some kind of equivalence classes.

Although this paper is a preliminary study on the relation between a group and its graph and symmetry-breaking constraints, we believe that it is the starting point of understanding the effect of certain symmetry-breaking constrains or algorithms from a formal perspective. We here considered only 3×3 matrices but our study could be generalised.

In the future, we will study this group further. An important question is whether there exist a characterisation that uniquely identifies a matrix in its equivalence class. The group-graph can help us to answer this question. If such a characterisation exists then all symmetric configurations of a matrix can be removed without having to consider each of them one by one. Also, such a graph can help to devise new symmetry-removal algorithms or constraints for matrix models. Least but not last, we plan to repeat this study for other symmetry-breaking constraints for matrix models. This will allow us to compare the relative strengths of the methods.

Acknowledgements

We would like to thank Andrea Roli, Marco Gavanelli, and Luca Benini for useful discussions, suggestions, and ideas.

References

1. J. Crawford, G. Luks, M. Ginsberg, and A. Roy. Symmetry breaking predicates for search problems. In *Proc. KR'96*, pp. 148–159. 1996.
2. P. Flener, A. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, J. Pearson, and T. Walsh. Breaking row and column symmetries in matrix models. In *Proc. of CP'2002*. Springer, 2002.

3. P. Flener, A. M. Frisch, B. Hnich, Z. Kiziltan, I. Miguel, and T. Walsh. Matrix modelling. Technical report APES-36-2001. Available from http://www.dcs.st-and.ac.uk/∼apes/reports/apes-36-2001.ps.gz.
4. I.P. Gent, W. Harvey, and T. Kelsey. Groups and constraints: symmetry breaking during search. In *Proc. of CP'2002*. Springer, 2002.
5. J.-F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. In *Proc. ISMIS'93*, pp. 350–361. Springer, 1993.