Breaking of Partial Symmetries in the Photo and Alignment Problem

Sebastian Will and Rolf Backofen

Friedrich-Schiller-Universitaet Jena
Institute of Computer Science, Chair for Bioinformatics
Ernst-Abbe-Platz 1-4, D-07743 Jena, Germany,
{will,backofen}@inf.uni-jena.de

Abstract. Symmetry breaking by adding symmetric constraints during search usually assumes that symmetric constraints are simple. We identify symmetries with complex symmetric constraints, where the symmetries nevertheless can be handled by a similar method. For this aim, we introduce partial symmetries. We identify those symmetries in two problems. The photo problem is a well known example problem, while the alignment problem is a real world problem from bioinformatics.

1 Introduction

An important symmetry breaking method works by dynamically adding symmetry breaking constraints during search (e.g. see [2,4]). This method requires the notion of a symmetric constraint $S(c)$ for a constraint c and a symmetry S. $S(c)$ is satisfied by all symmetric assignments of c. Recent applications of this symmetry breaking method dealt with only simple symmetric constraints, where the symmetric constraint depends only on c and not on the rest of the assignment. A typical example are permutations of variables, where for a permutation π and constraint of the form $X_i = k$ we have $X_{\pi(i)} = k$ as a symmetric constraint.

Of course, there are many problems having symmetries, where such a simple definition of a symmetric constraint is not possible. Unfortunately, complex symmetric constraints undermine the efficiency of the considered breaking mechanism. We identify a class of *partial* symmetries, where one can apply a slightly modified breaking method efficiently.

The idea of partial symmetries is that we can describe a subset of assignments by constraints, where the symmetric constraint $S(c)$ again only depends on c. Previously, symmetries were defined as total, bijective functions on the assignments. In contrast, partial symmetries are defined as partial functions. On their domain, partial symmetries behave like usual symmetries. We discuss such symmetries and identify them in two different problems.

1.1 Preliminaries

Our notation follows the one used in [2,3]. In particular, recall that for a constraint c, $\|c\|$ denotes the set of assignments that satisfy c. Given a constraint problem C_{Pr}, we define similar to [2,3] the set of solution variables to be the variables whose valuation determines the valuation of all other variables by constraint propagation. A *permutation* π is defined as usual, a *reversion* ρ_{ij} *of* $\{1, \ldots, n\}$, is the permutation that just reverses the range $\{i, \ldots, j\}$. latex

2 The Photo Problem and Its Symmetries

2.1 Problem Description

In the photo problem, we align n persons in a row on a photo, such that maximally many preferences of the persons are satisfied. The persons prefer certain other persons as their immediate neighbors on the photo.

The *photo problem with n persons and preferences P* is a pair (n, P) and described formally as follows. The set of *preferences* is a subset $P \subseteq \{\{i, j\} \mid 1 \leq i, j \leq n\}$. The set of solution variables is denoted by $\{X_1, \ldots, X_n\}$. A solution α is a permutation $(\alpha_1, \ldots, \alpha_n)$ of the numbers $(1, \ldots, n)$, such that $\alpha(X_i) = \alpha_i$ for $1 \leq i \leq n$. We call a solution of the photo problem a *photo*. The *set of satisfied preferences* of a photo α is

$$\mathrm{satisfied}_{(n,P)}(\alpha) = \{\{\alpha_i, \alpha_{i+1}\} \in P \mid 1 \leq i \leq n\}.$$

An optimal solution of the problem is a photo α where the number of satisfied constraints $\mathrm{satisfaction}(\alpha) = |\mathrm{satisfied}_{(n,P)}(\alpha)|$ is maximal.

2.2 The Symmetries of the Photo Problem

For this problem, the notion of symmetry is not as obvious as for problems like N-queens (geometric symmetries) or graph coloring (permutation symmetries). Note that two symmetric photos α and β have to satisfy $\mathrm{satisfaction}(\alpha) = \mathrm{satisfaction}(\beta)$. As a definition of symmetry this leads to symmetries, which are far too complex to compute.

Thus, define two photos α and β as symmetric iff

$$\mathrm{satisfied}_{(n,P)}(\alpha) = \mathrm{satisfied}_{(n,P)}(\beta).$$

Obviously, two symmetric photos still have the same satisfaction, which makes this definition reasonable. The symmetry functions are permutations under the condition that the permutation preserves the satisfied preferences. Instead of dealing with all permutations, it is reasonable to handle only reversions that keep the set of satisfied preferences constant, this is an example of breaking a subset of symmetry as e.g. described in [6]. We discuss both possibilities.

We define permutations and reversions on photos α of (n, P) as follows. Let π be a permutation. Then, $\pi^{\mathrm{var}} : \|C_{\mathrm{Pr}}\| \to \|C_{\mathrm{Pr}}\|$ denotes a *variable permutation*, such that $\forall \alpha \in \|C_{\mathrm{Pr}}\| \forall 1 \leq i \leq n : \alpha(X_i) = (\pi^{\mathrm{var}}(\alpha))(X_{\pi(i)})$. Variable reversions ρ_{ij}^{var} may be defined analogously.

a) 1-2 3 4-5-6 7-8 9
b) 3 2-1 9 8-7 4-5-6
c) 1-2 6-5-4 3 7-8 9

Fig. 1. Permutation and reversion symmetries of the photo problem. Each line represents a photo for a problem instance with 9 persons, i.e. it shows the order of the persons 1,...,9. Preferred neighbors are shown by dashes between them. The three photos are symmetric, in the sense that they satisfy the same preferences. The symmetry that maps *a*) to *b*) is a permutation symmetry and a) is mapped to *c*) by a reversion symmetry.

The set of *permutation symmetries* $\mathcal{S}_\pi$ *of a photo problem* (n, P) is defined as the set of functions s_π, where

$$s_\pi(\alpha) = \begin{cases} \pi^{\text{var}}(\alpha) & \text{if satisfied}_{(n,P)}(\alpha) = \text{satisfied}_{(n,P)}(\pi^{\text{var}}(\alpha)) \\ \text{undef.} & \text{otherwise} \end{cases}$$

where $\{\pi_1, \ldots, \pi_m\}$ is the set of all permutations of the solution variables. For (n, P), we define a *reversion symmetry between* i *and* j for $1 \le i < j \le n$ by

$$s_{ij}(\alpha) = \begin{cases} \rho_{ij}^{\text{var}}(\alpha) & \text{if satisfied}_{(n,P)}(\alpha) = \text{satisfied}_{(n,P)}(\rho_{ij}(\alpha)) \\ \text{undef.} & \text{otherwise.} \end{cases}$$

The *set of reversion symmetries* $\mathcal{S}_\rho$ *of* (n, P) is defined by $\{s_{ij} \mid 1 \le i < j \le n\} \cup \{\text{id}\}$, where id is the identity.

Breaking only reversions raises the question whether the breaking of the all reversion symmetries is capable of breaking all permutation symmetries. While in general, the set of reversions is a set of generators for the group of permutations, we are considering reversions that are only *partially* defined. Hence, when combining reversions $s_{\rho 1}, \ldots, s_{\rho m} \in \mathcal{S}_\rho$ such that $\forall \alpha \in \text{dom}(s_\pi) : s_{\rho 1} \circ \cdots \circ s_{\rho m}(\alpha) = s_\pi(\alpha)$, it might happen that the partial domain of $s_{\rho i}$ does not fit with the image of $s_{\rho(i+1)}$. Such an incompatibility would forbid to use the composition $s_{\rho i} \circ s_{\rho(i+1)}$.

Hence, we do not show that the (partial) reversions symmetries generate the permutation symmetries, i.e. $\exists s_{\rho 1}, \ldots, s_{\rho m} \in \mathcal{S}_\rho \forall \alpha \in \text{dom}(s_\pi) : s_{\rho 1} \circ \cdots \circ s_{\rho m}(\alpha) = s_\pi(\alpha)$. Instead, we show that for each valuation $\alpha \in \text{dom}(s_\pi)$, we find a specific sequence of reversions preserving the preferences satisfied in α, i.e. $\forall \alpha \in \text{dom}(s_\pi) \exists s_{\rho 1}, \ldots, s_{\rho m} \in \mathcal{S}_\rho : s_{\rho 1} \circ \cdots \circ s_{\rho m}(\alpha) = s_\pi(\alpha)$.

Theorem 1. *Fix an instance of the photo problem. Let* $\mathcal{S}_\pi$ *be the set of permutation symmetries and* $\mathcal{S}_\rho$ *be the set of reversion symmetries for this problem instance. Then, for any* $s_\pi \in \mathcal{S}_\pi$ *and for any photo* α *of the problem instance, where* s_π *is defined, there exist* $s_{\rho 1}, \ldots, s_{\rho m} \in \mathcal{S}_\rho$*, such that* $s_{\rho 1} \circ \cdots \circ s_{\rho m}(\alpha) = s_\pi(\alpha)$.

The main idea of our proof of Theorem 1 is to cluster persons who are connected by preferences. Then, permutations and reversion on these clusters do not break preferences any more. Thus, any permutation on these clusters can be expressed by reversions.

3 Partial Symmetries

The symmetries of the previous section are intuitively formulated as partial functions. Albeit those symmetries can be represented by total functions[1], this is counter-intuitive and does not help for an efficient implementation due to the complexity of those functions. This motivates introducing the notion of partial symmetry.

Definition 1 (Partial Symmetry). *A* partial symmetry *is a partial function* $s : \|C_{Pr}\| \to \|C_{Pr}\|$, *such that the* domain restriction *of* s, *which is* $s_{dom} : dom(s) \to dom(s), \alpha \mapsto s(\alpha)$, *is a bijective function.*

Note that by this definition a partial symmetry is not a symmetry as defined in [2,3], since it is not total and bijective. Nevertheless, partial symmetries can be handled by the same symmetry exclusion algorithms as presented there with a small modification.

Similar to a symmetry, a partial symmetry s leads to a *symmetry function on constraints* s_{con} with the property that for every $\alpha \in \mathrm{dom}(s)$

$$\alpha \models c \;\Leftrightarrow\; s(\alpha) \models \mathrm{s}_{\mathrm{con}}(c).$$

A partial symmetry s may be intuitively interpreted as a symmetry on a subset $\mathrm{dom}(s)$ of all solutions $\|C_{\mathrm{Pr}}\|$. In general, there is a *condition* $c_s \in \mathcal{C}$ *for* s to decide if a solution is in $\mathrm{dom}(s)$. That is, for the condition c_s $\mathrm{dom}(s) = \|C_{\mathrm{Pr}} \wedge c_s\|$ is satisfied. Thus, the condition c_s tests, whether a partial symmetry can be applied to a solution.

Proposition 1. *Let* $s\colon \|C_{Pr}\| \to \|Cpr\|$ *be a partial symmetry and* c_s *be the condition for* s*. Then,* $\alpha \models c_s$ *iff* $s(\alpha) \models c_s$*.*

Note that Proposition 1 implies for symmetry s and condition c_s for s holds $s(c_s) \models\!\!\!\dashv c_s$ and $c_{s^{-1}} \models\!\!\!\dashv c_s$, where $c_{s^{-1}}$ is the condition for s^{-1}.

The concept of a condition c_s for each symmetry s allows to extend the breaking mechanism for usual symmetries to *partial symmetry breaking search*. Recall that for symmetry breaking we add constraints of the form $\mathrm{s}_{\mathrm{con}}(C_{\mathrm{p}}) \to \neg\, \mathrm{s}_{\mathrm{con}}(c)$. Here, $\mathrm{s}_{\mathrm{con}}(C_{\mathrm{p}})$ actually tests, whether the insertion of $\neg\, \mathrm{s}_{\mathrm{con}}(c)$ exactly breaks the symmetry s. Similarly for partial symmetries, $\neg\, \mathrm{s}_{\mathrm{con}}(c)$ has to take effect only if the symmetry s is defined on all possible solutions. That is, we have to add the test for c_s to the antecedent of the implication and thus, get constraints of the form $s_c \wedge \mathrm{s}_{\mathrm{con}}(C_{\mathrm{p}}) \to \neg\, \mathrm{s}_{\mathrm{con}}(c)$. [2]

Finally, we give a short comparison of partial symmetries to our treatment of non-partial symmetries in [2,3]. There we defined the terms $\mathcal{S}$*-reduced* and C_{Pr}*-complete w.r.t.* $\mathcal{S}$. These terms are extended to the case of partial symmetry sets

[1] The corresponding non-partial symmetry extends the partial function to a total one by mapping values outside of the domain to themselves.

[2] Note that an implementation can use reified constraints and boolean variables to compute $s_c \wedge \mathrm{s}_{\mathrm{con}}(C_{\mathrm{p}})$ incrementally. Compared to a naive implementation, this reduces the computational work significantly.

S straightforwardly. Then, we proof, that the partial symmetry breaking search tree is reduced and complete. The detailed treatment and proof is omitted due to space restrictions.

4 The Alignment Problem

A further problem with partial symmetries is the alignment problem. The alignment problem is a very important problem in bioinformatics, where one searches for an alignment of two strings, optimizing a certain score. The strings represent biological macromolecules as DNAs or proteins. The scoring scheme evaluates the aligned columns.

In the simplest case, alignment is identical to computing the edit distance of strings. This problem is usually solved by dynamic programming (DP) as e.g. by Needleman and Wunsch in [7].

However, dynamic programming approaches suffer from their inflexibility. If the problem is slightly modified, one has to develop a new DP algorithm (if one exists at all). For example, it is an unsolved problem to align two sequences incorporating biological knowledge that tells us which sub-sequences/domains should be aligned.

Further there are many biologically motivated extensions to sequence alignment, e.g. protein threading (e.g., [1]) or the contact map problem [5] that are NP-complete and not solveable by DP at all.

To investigate such problems, where the usual DP approach fails, a constraint-based formulation of sequence alignment is desirable. We believe that such a formulation will allow many biologically interesting extension. An extension to sequence structure alignment as in the contact map problem is discussed.

4.1 A Constraint Model for Alignment

The alignment problem is given by sequences $a = a_1 \dots a_n$ and $b = b_1 \dots b_m$. It is modelled as a bipartite graph, where the sequence positions are vertices and the edges connect aligned positions. We consider the score that sums over weights $w(a_i, b_j)$ for aligned edges (i, j) and adds a *(linear) gap penalty*, i.e. the number of all unaligned positions times a factor g.

In our constraint model, we introduce variables X_i for every position i in sequence a. Their domain contains the positions in sequence b and a "gap value",

```
1 2 3 4 5 6 7   8
A C G T G G A   A
  | |     | |
  C G     G A T   T T
  1 2     3 4 5   6 7
```

Fig. 2. The figure shows a graph representing an alignment of the two sequences `ACGTGGAA` and `CGGATTT`. For this alignment, the variables X_i $(1 \leq i \leq 8)$ are assigned to values $(0, 1, 2, 0, 0, 3, 4, 0)$ and the variables Y_i have the values $(0, 1, 2, 2, 2, 3, 4, 4)$.

here 0. The assignment $X_i \doteq j$ means positions i and j are aligned. $X_i \doteq 0$ means a_i is aligned to a gap. Further variables have to be introduced to compute the score of the alignment, which is optimized by branch-and-bound. Concerning the constraints, we need to avoid crossing of alignment edges, i.e. the values of variables $X_i \neq 0$ have to be ascending. To avoid the use of quadratically many non-standard ordering constraints, we introduce a second representation of the alignment. There, we use one variable Y_i for each position in a. The domain of Y_i is the set of positions in sequence b. If $Y_{i-1} < Y_i$ then i is aligned to the value of Y_i, else if $Y_{i-1} = Y_i$, i is aligned to a gap. On the variables Y_i, it suffices to impose linearly many standard $\leq$-constraints. For clarifying the model, an example is given in Fig. 2.

4.2 Symmetries

We discuss the symmetry that occurs if the sequences contain repeats as in `CTAAAGT`, where the `A` is repeated 3 times. For example, if we align this sequence to `CAGTT` the following 6 alignments are intuitively symmetric to each other, since our score cannot distinguish them.

```
CTAAAGT-  CTAAAGT-  CTAAAGT-  CTAAAG-T  CTAAAG-T  CTAAAG-T
C-A--GTT  C--A-GTT  C---AGTT  C-A--GTT  C--A-GTT  C---AGTT
```

These symmetries are permutations of values and variables, however as in the case of the photo problem the permutations yield symmetric assignments only on a part of all assignments, where for other assignments the same permutation yields non-symmetric assignments. Again these symmetries are intuitively expressed as partial symmetries in the sense of our previous definition.

We restrict our discussion to symmetries for repeats in sequence a. The case for sequence b can be handled similarly. Again, it is reasonable to handle only a subset of all permutations. We deal with translation symmetries $s_{i\leftrightarrow j}$, which swap the value of variables x_i and x_j. The translation $i \leftrightarrow j$ is a symmetry, if there is only one match of the positions in the range $\{i, \ldots, j\}$, which forms our condition for the symmetry $s_{i\leftrightarrow j}$. The condition is expressed easily by the constraint $Y_{i-1} + 1 \doteq Y_j$.

While we are aware that the use of two representations for the alignment seems circumstantial, this is justified by the following consideration. On the one hand the non-crossing of alignment edges and the symmetry conditions are not expressed easily on the variables X_i, on the other hand the symmetric constraint $s_{i\leftrightarrow j}(Y_i \doteq j)$ does not only depend on i and j, but on further variables (especially it is not $Y_j \doteq i$). In contrast, symmetric constraints of the constraints $X_i \doteq j$ are straightforward, namely $s_{i\leftrightarrow j}(X_i \doteq j) = X_j \doteq i$ as long as the condition for $s_{i\leftrightarrow j}$ holds.

4.3 Extension to Sequence Structure Alignment

Sequence structure alignment is the problem of aligning two macromolecules not only according to their sequence, but taking account of their structure.

While the sequence is given as a string, the structure can be represented as a set of arcs that connect positions in one sequence. The constraint model for sequence alignment can be extended by additional variables for the arcs in the first sequence, where their domains contains the arcs of the second sequence, analogously to the variables for sequence positions. The discussed symmetry still occurs in this problems in repeats that are not disrupted by incidenting structure.

5 Results

In [2,3], we show that a subset of symmetries can break symmetries from the generated group. In the photo problem, we observe this effect. The breaking of all reversion symmetries breaks many symmetries from the group of permutation symmetries. Some illustrating data is shown in Table 1. There, in column *problem*, the instance of the photo problem is specified in the form (n, P). *sat.* gives the satisfaction for the problem. The column n_{rev} gives the number of symmetry classes w.r.t. reversion symmetries, n_{perm} gives analogously the number of classes for permutation symmetries. Note that the breaking of reversion symmetries by our breaking mechanism reduces the number of solutions at least to n_{rev}. By Theorem 1, the breaking of reversion symmetries can reduce the number of solutions to n_{perm}. $\#_{nobr}$ (resp. $\#_{br}$) is the number of nodes in the search for all solutions without (resp. with) symmetry breaking of reversion symmetries. Column n_{nobr} (resp. n_{br}) gives the number of solutions found without (resp. with) breaking of reversion symmetries.

We are interested in how close the breaking of reversion symmetries comes to the breaking of all permutation symmetries. Therefore, we give the ratio $\frac{n_{nobr}-n_{br}}{n_{nobr}-n_{perm}}$ for each problem instance. Further, we compare the number of reversion symmetry classes to the number of remaining solutions by the ratio $\frac{n_{rev}}{n_{br}}$.

As of yet, we have not done extensive tests and evaluation on the alignment problem. The use of the current implementation is mainly to demonstrate the proposed symmetry breaking, further improvements are not investigated yet. Our implementation breaks the symmetries that are caused by repeats in the first sequence. We give a few examples in Table 2. There, $\#_{\mathrm{br}}$ (resp. $\#_{\mathrm{nobr}}$ denotes the number of search steps with (resp. without) symmetry breaking, which is shown for the search for the best alignment as well as for the search for all optimal alignments. n_{br} (resp. n_{nobr}) denotes the number of solutions with (resp. without) breaking of symmetries.

References

1. Tatsuya Akutsu and Satoru Miyano. On the approximation of protein threading. In *Proc. of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*. ACM Press, 1997.
2. Rolf Backofen and Sebastian Will. Excluding symmetries in constraint-based search. In Joxan Jaffar, editor, *Proceedings of 5^{th} International Conference on Principle*

				without sym.br.		with sym.br.		ratio	
problem	sat.	n_{rev}	n_{perm}	$\#_{nobr}$	n_{nobr}	$\#_{br}$	n_{br}	$\frac{n_{nobr}-n_{br}}{n_{nobr}-n_{perm}}$	$\frac{n_{rev}}{n_{br}}$
(10,P_1)	7	84	8	1,710	192	999	14	96.7%	6.0
(11,P_2)	8	184	27	15,531	720	8,059	86	91.5%	2.1
(11,P_3)	8	120	15	2,734	480	1,896	52	92.0%	2.3
(12,P_4)	9	17	2	5,346	96	2,714	12	89.4%	1.4
(12,P_5)	8	53	3	1,427	240	974	20	92.3%	2.7
(13,P_6)	9	180	4	68,337	768	17,478	17	98.3%	10.6
(14,P_7)	8	582	1	6,401	2880	1,173	1	100.0%	582.0

$$
\begin{aligned}
P_1 =& \{\{1,3\},\{2,5\},\{2,7\},\{2,8\},\{3,4\},\{5,9\},\{5,10\},\{7,10\},\{8,9\}\} \\
P_2 =& \{\{1,7\},\{2,6\},\{3,5\},\{3,7\},\{3,8\},\{4,7\},\{5,7\},\{5,9\},\{6,10\},\{7,2\}, \\
& \{8,11\},\{9,11\}\} \\
P_3 =& \{\{1,7\},\{2,6\},\{3,5\},\{3,8\},\{4,7\},\{5,9\},\{6,10\},\{7,2\},\{8,11\},\{9,11\}\} \\
P_4 =& \{\{1,3\},\{1,8\},\{1,11\},\{2,3\},\{2,7\},\{3,5\},\{3,7\},\{4,9\},\{4,12\},\{5,9\},\{6,10\}\} \\
P_5 =& \{\{1,9\},\{2,5\},\{3,7\},\{4,3\},\{5,12\},\{10,2\},\{10,9\},\{11,4\},\{12,3\}\} \\
P_6 =& \{\{1,5\},\{1,6\},\{1,7\},\{3,5\},\{3,13\},\{4,7\},\{4,12\},\{5,8\},\{7,12\},\{8,13\}, \\
& \{9,10\},\{11,13\}\} \\
P_7 =& \{\{1,4\},\{1,7\},\{2,14\},\{2,5\},\{3,8\},\{3,9\},\{4,9\},\{5,13\}\}
\end{aligned}
$$

Table 1. Effect of symmetry breaking in the photo problem.

		search best	search all	
sequence a	sequence b	$\frac{\#_{\rm nobr}}{\#_{\rm br}}$	$\frac{\#_{\rm nobr}}{\#_{\rm br}}$	$\frac{n_{\rm nobr}}{n_{\rm br}}$
AAAACCCTTTCCCAAATTT	GAACTTAAT	4.34	5.32	324.0
CAAACCTCCAAATTT	GAACTTAATC	1.85	1.93	34.0
AAATTTTGGGGCCC	ATGCATGCATGC	2.44	2.54	9.0
TTAAAATTGGCCCCGG	TGCATGCATG	2.77	2.78	196.0

Table 2. Symmetry breaking in the alignment problem.

and Practice of Constraint Programming (CP'99), volume 1713 of *Lecture Notes in Computer Science*, pages 73–87, Berlin, 1999. Springer–Verlag.

3. Rolf Backofen and Sebastian Will. Excluding symmetries in constraint-based search. *Constraints*, 7(3):333–349, 2002.
4. Ian P. Gent and Barbara M. Smith. Symmetry Breaking in Constraint Programming. In Werner Horn, editor, *Proceedings ECAI 2000*, pages 599–603. IOS Press, 2000.
5. Giuseppe Lancia, Robert Carr, Brian Walenz, and Sorin Istrail. 101 optimal PDB structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. In *Proc. of the Fifth Annual International Conferences on Compututational Molecular Biology (RECOMB01)*. ACM Press, 2001.
6. Iain McDonald and Barbara M. Smith. Partial symmetry breaking. In Pascal van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002*, volume 2470 of *LNCS*, pages 431–445. Springer-Verlag, 2002.
7. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–53, 1970.