Supersymmetric Modeling for Local Search

Steven Prestwich

Cork Constraint Computation Centre
University College, Cork, Ireland
s.prestwich@cs.ucc.ie

Abstract. A previous paper showed that adding binary symmetry breaking constraints to a model can degrade local search performance. This paper shows that even unary constraints are not necessarily beneficial because they can transform solutions into local minima. Furthermore, performance can be improved by the reverse of symmetry breaking: using a *supersymmetric* model with extra symmetry. Some improved results for the highly symmetrical Social Golfer problem are presented.

1 Introduction

Considerable research has been devoted to symmetry breaking techniques for reducing search space size. Probably the most popular approach is to add constraints to the problem formulation, so that each equivalence class of solutions to the original problem corresponds to a single solution in the new problem. This avoids the need to modify search algorithms, which are often rather complex, and is the only option available to a researcher using SAT solvers or other algorithms downloaded from the Internet. (Dynamic approaches to symmetry breaking such as [3] are not considered in this paper.)

Symmetry breaking is usually applied when complete backtrack search is to be performed and its benefits are often considerable. However, it is well known that it does not necessarily improve search for a single solution. A previous paper [4] showed that adding binary symmetry breaking constraints to SAT models can have a negative effect on local search for clique, set cover and block design problems. This was not merely overhead caused by extra constraint processing; the number of search steps needed to find a solution was also increased. Several researchers have demonstrated unexpected effects when combining two or more techniques: backtracking has been shown to interact badly with the normally beneficial techniques of domain pruning [6], arc consistency preprocessing [7] and removal of inconsistent or redundant domain values or subproblems [2]. The effects of combining techniques can be surprising and unpredictable, so it should not be a complete surprise that local search and symmetry breaking do not always combine well. However, in our experiments local search performance was *almost always* degraded by symmetry breaking, suggesting that the effects are not isolated anomalies.

This paper investigates the effect of *unary* symmetry breaking constraints on local search performance. These might be expected to benefit any search algorithm but Section 2 shows that, in principle, they can have a negative effect on a variety of algorithms. Section 3 shows that this can occur on real problems, using a hybrid local search algorithm and an ILP model of the Social Golfer problem. Section 4 investigates the following question: if reducing symmetry in a model is bad for local search, might *increasing* symmetry help?

2 Symmetry breaking and local minima

First we show that adding unary symmetry breaking constraints can have a negative effect on local, backtrack and hybrid search. Consider the SAT problem

$$\overline{a} \vee b \quad \overline{a} \vee c \quad a \vee \overline{b} \quad a \vee \overline{c}$$

There are two solutions: [a=T, b=T, c=T] and [a=F, b=F, c=F]. Suppose a problem modeler realises that every solution to a problem has a symmetrical solution in which all truth values are negated, as in this example. Then a simple way to break symmetry is to fix the value of any variable by adding a new clause such as

$$a$$

Denote the first model by M and the model with symmetry breaking by M_s.

Now suppose we apply a local search algorithm such as GSAT [9] to the problem. GSAT starts by making a random truth assignment to all variables, then repeatedly flipping truth assignments to try to reduce the number of violated clauses. In model M the state [a=F, b=F, c=F] is a solution, but in M_s the added clause a is violated; moreover, any flip leads to a state in which *two* clauses are violated. In other words this state has been transformed from a solution to a *local minimum*. (The unary clause does not preclude this as a random first state, nor does it necessarily prevent a randomized local search algorithm from reaching this state.) In contrast M has no local minima: any non-solution state contains either two T or two F assignments, so a single flip leads to a solution (respectively TTT or FFF). GSAT will actually escape this local minimum because it makes a "best" flip even when that flip increases the number of violations, but examples can be constructed with deeper local minima to defeat this heuristic. The point is that a new local minimum has been created, and local minima generally degrade local search performance by requiring more noise.

Unit propagation applied to the added clause gives a reduced problem

$$b \quad c$$

which contains no local minima; will unary constraints always benefit search algorithms with unit propagation? Consider a Davis-Logemann-Loveland algorithm [1] (backtracking with unit propagation) applied to another SAT problem

$$a \vee b \vee c \vee d \quad a \vee \overline{b} \vee c \vee d \quad \overline{a} \vee b \vee c \vee d \quad \overline{a} \vee \overline{b} \vee c \vee d \quad \overline{c} \vee \overline{d}$$

There are eight solutions:

	1	2	3	4	5	6	7	8
a:	T	T	T	T	F	F	F	F
b:	T	T	F	F	T	T	F	F
c:	T	F	T	F	T	F	T	F
d:	F	T	F	T	F	T	F	T

Suppose a problem modeler realises that each solution has a symmetric version in which the values of c and d are exchanged. To exclude solutions 1, 3, 5 and 7 a unary constraint

$$\bar{c}$$

may be added. Applying unit propagation and removing redundant constraints gives a reduced problem

$$a \vee b \vee d \quad a \vee \bar{b} \vee d \quad \bar{a} \vee b \vee d \quad \bar{a} \vee \bar{b} \vee d$$

However, even though the unary constraint has been propagated, the backtracker can still move part of the way towards one of the excluded solutions. Assign d=F (as in excluded solutions 1, 3, 5 and 7) and apply unit propagation:

$$a \vee b \quad a \vee \bar{b} \quad \bar{a} \vee b \quad \bar{a} \vee \bar{b}$$

No empty clause has been derived so the algorithm will not backtrack at this point. However, this state cannot be extended to a solution because all combinations of a and b assignments lead to an empty clause. Only after a further variable assignment will backtracking occur. Adding the unary symmetry breaking constraint $\bar{c}$ has increased the number of backtracks needed to find a solution.

This phenomenon is well known for backtrackers and was a motivation for approaches such as Symmetry Breaking During Search [3]. However, the example also applies to a hybrid local search algorithm: Saturn [5], which performs local search in a space of partial assignments, uses unit propagation to prune the search space, and attempts to minimize the number of unassigned variables. For Saturn adding the unary symmetry breaking constraint transforms the state [c=F, d=F] from a partial solution to a local minimum.

To summarize: adding a unary symmetry breaking constraint may increase the number of search steps needed to find a solution, whether we use local search, backtracking or a hybrid such as Saturn. But does this effect occur in practice or are these contrived examples mere curiosities? The next section provides evidence that it does occur.

3 Experiments on the Social Golfer problem

As a symmetrical problem we choose the Social Golfer problem which is studied extensively in [10]. In a golf club there are 32 social golfers, each of whom play golf once a week and always in groups of 4. The problem is to find a schedule of play for these golfers, to last as many weeks as possible, such that no golfer plays in the same group as any other golfer on more than one occasion. This can be generalized to P golf players playing in G groups of size S for W weeks, where $G \times S = P$. An example is shown in Figure 1 for instance 3-3-4 (G-S-W). The related Kirkman's Schoolgirls problem corresponds to instance 5-3-7. These problems have a huge number of symmetries because players, groups, weeks and variable names may all be permuted.

week	group 1	group 2	group 3
1	1,2,3	4,5,6	7,8,9
2	1,4,7	2,5,8	3,6,9
3	1,6,8	2,4,9	3,5,7
4	1,5,9	2,6,7	3,4,8

Fig. 1. A schedule for Social Golfer instance 3-3-4

3.1 A model

A pure 0/1 integer linear program (ILP) model is as follows. Define a 0/1 variable v_{pgw} which is 1 if and only if player p plays in group g in week w. The constraints are as follows. Each group contains exactly S players:

$$\sum_{p=1}^{P} v_{pgw} = S$$

where $1 \leq g \leq G$ and $1 \leq w \leq W$. Each player plays in exactly one group per week:

$$\sum_{g=1}^{G} v_{pgw} = 1$$

where $1 \leq p \leq P$ and $1 \leq w \leq W$. Finally the "sociability constraints": no two players can play in the same group as each other more than once. Define auxiliary variables $s_{pp'w}$ ($1 \leq p < p' \leq P$, $1 \leq w \leq W$) to denote that in week w players p and p' play in the same group:

$$v_{pgw} + v_{p'gw} \leq 1 + s_{pp'w}$$

where $1 \leq p < p' \leq P$, $1 \leq g \leq G$ and $1 \leq w \leq W$. Now the sociability constraints can be expressed on the s:

$$\sum_{w=1}^{W} s_{pp'w} \leq 1$$

where $1 \leq p < p' \leq P$ and $1 \leq w < w' \leq W$. Symmetry breaking can be applied by adding two sets of constraints to the model. We can fix the groups in the first week:

$$v_{ij1} = 1$$

where $1 \leq i \leq P$ and $j = (i-1)/S+1$ rounded down to the nearest integer. We can also fix player 1 to be in group 1 in every subsequent week:

$$v_{11w} = 1$$

where $2 \leq w \leq W$. This is Walser's model as described in CSPLib. [1]

3.2 Results

If adding unary symmetry breaking constraints can create local minima, this should require additional noise in a local search algorithm. We test this using the Saturn local search algorithm, which has been extended from SAT to pure 0/1 ILP models. Saturn has an integer noise parameter B and we examine the effect of varying B on its performance. Performance is measured as the number of search steps (non-systematic backtracks, denoted by "bt") required to find a solution, taking medians over 1000 runs. The use of search steps filters out overheads caused by processing the extra symmetry breaking constraints, and the use of medians rather than means reduces distortion by outliers. The results for two instances, which are representative of more extensive experiments, are shown in Figures 2 and 3.

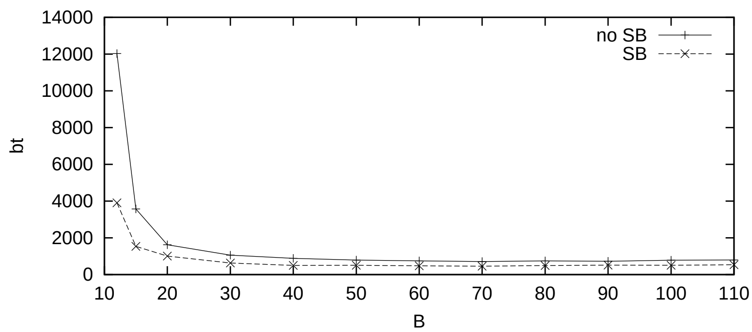

Fig. 2. Results for instance 5-4-3

For easy instances such as 5-4-3 (Figure 2) the added constraints consistently improve performance. We believe that this is simply because the number of search variables has been reduced via unit propagation on the unary constraints. For harder instances such as 6-4-5 (Figure 3) the results are more complex. The optimum noise level has increased, which is evidence for extra local minima. However, the search effort at the optimum noise levels is similar in both cases. This may be the positive effect of fewer search variables balancing the negative effect of extra local minima. These results are not a strong argument against

[1] http://www.csplib.org, problem 10

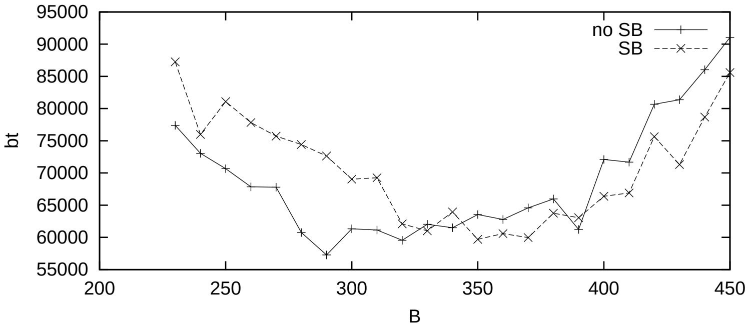

Fig. 3. Results for instance 6-4-5

using symmetry breaking with local search, but there may be other problems on which the negative effect is greater. Moreover, extra constraints increase runtime overheads, so avoiding them seems preferable.

Local search on symmetric models can be very effective. Saturn was applied to the model without symmetry breaking, and at the time of writing it has found the longest schedules for several large instances: 9-5-6, 9-6-5, 9-8-3, 9-9-3, 10-5-7, 10-7-5, 10-8-4, 10-9-3 and 10-10-3, each taking a few seconds or minutes. The optimal 7 week solution for Kirkman's Schoolgirls problem was found in a few seconds, a 7 week solution for the instance with 8 groups of 4 golfers was found in approximately 1 minute, and an 8 week solution was found after approximately 6 hours. On the other hand, Saturn cannot reproduce some solutions found by other algorithms, for example 8-4-9. Current results for the Social Golfer are summarized on a web page. [2]

4 The opposite of symmetry breaking

These results suggest a novel approach to modeling for local search, mentioned in [4] but not tested: *adding* symmetry. We might call this reversal of symmetry breaking *supersymmetric modeling*.

As an example, consider the usual definition of a Golomb Ruler: an ordered sequence of integers $0 = x_1 < x_2 < \ldots < x_m = \ell$ such that the $m(m-1)/2$ differences $x_j - x_i$ $(j > i)$ are distinct, where ℓ is the permitted ruler length. The constraint $|x_2 - x_1| < |x_m - x_{m-1}|$ is usually added to break a reflective symmetry. Finding a ruler with given m and ℓ is a CSP. Several models are given in [11] and we use their *binary/ternary* model. Define variables $x_1 \ldots x_m$ each with domain $\{0, \ldots, \ell\}$, and auxiliary variables d_{ij} where $1 \leq i < j \leq m$. Impose ordering constraints $x_i < x_{i+1}$ where $1 \leq i \leq m-1$. Impose ternary constraints $d_{ij} = x_j - x_i$ and binary constraints $d_{ij} \neq d_{i'j'}$ where $i < i'$, or $i = i'$ and $j < j'$. Also impose unary constraints $x_1 = 0$ and $x_m = \ell$, and a symmetry

[2] http://www.icparc.ic.ac.uk/~wh/golf/

breaking constraint $d_{12} < d_{m-1,m}$. This CSP can be SAT-encoded using the direct encoding.

To obtain a supersymmetric model we remove the symmetry breaking constraint and the ordering constraints on the x_i, merely constraining them to be distinct: $x_i \neq x_j$ where $1 \leq i < j \leq m$. The binary constraints $d_{ij} \neq d_{i'j'}$ are as above, but because the x_i are no longer ordered the ternary constraints are modified to $d_{ij} = |x_i - x_j|$. The symmetry breaking constraint $d_{12} < d_{m-1,m}$ is not used but the unary constraints $x_1 = 0$ and $x_m = \ell$ are (these decisions are somewhat arbitrary but gave best results). Though a solution to this problem is no longer necessarily a Golomb ruler, one can be derived in polynomial time by sorting the x_i into ascending order.

Figure 4 shows mean results over 50 runs for Golomb rulers with various lengths ℓ and numbers of marks m, and the model M either "natural" (N) or supersymmetric (S). Best results for Walksat [8] are reported using noise values from 5% to 95% in steps of 5%; and best results for Saturn with noise values {5, 10, 15, 20, 25, 30, 40, 50, 70, 90, 110, 130, 150} (less tuning effort was required at higher noise levels).

			Walksat		Saturn	
m	ℓ	M	flips	sec.	back.	sec.
4	6	S	139	0.002	126	0.002
4	6	N	492	0.005	1467	0.020
5	13	S	397	0.033	1751	0.35
5	13	N	1042	0.058	8460	1.71
5	11	S	564	0.023	1534	0.19
5	11	N	1509	0.050	8435	1.12
6	21	S	1897	0.35	12688	9.0
6	21	N	4579	0.75	68250	49.0
6	19	S	2390	0.30	14101	8.3
6	19	N	3007	0.33	111128	66.5
6	17	S	3736	0.31	36304	17.0
6	17	N	11233	0.82	166549	81.5

Fig. 4. Results on Golomb rulers

On these problems Walksat is much faster than Saturn. However, both consistently perform better on the supersymmetric models, both in terms of steps and execution times, showing that supersymmetry can improve local search performance. We have not yet found a successful way to apply this approach to the Social Golfer problem: a first attempt (details omitted for space reasons) introduced many new variables and gave inferior results. But we believe that the approach is potentially useful and hope to find other examples in future work.

Acknowledgment The Cork Constraint Computation Centre is supported by Science Foundation Ireland.

References

1. M. Davis, G. Logemann, D. Loveland. A Machine Program for Theorem Proving. *Communications of the ACM* vol. 5, 1962, pp. 394–397.
2. E. C. Freuder, P. D. Hubbe, D. Sabin. Inconsistency and Redundancy do not Imply Irrelevance. *Proceedings of the AAAI Fall Symposium on Relevance*, New Orleans, LA, USA, 1994.
3. I. P. Gent, B. Smith. Symmetry Breaking During Search in Constraint Programming. *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, 2000, pp. 599–603.
4. S. D. Prestwich. First-Solution Search with Symmetry Breaking and Implied Constraints. *CP'01 Workshop on Modeling and Formulation*, Cyprus, 2001.
5. S. D. Prestwich. Randomised Backtracking for Linear Pseudo-Boolean Constraint Problems. *Fourth International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems*, le Croisic, France, 2002, pp. 7–20.
6. P. Prosser. Domain Filtering Can Degrade Intelligent Backjumping Search. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Chambery, France, Morgan Kaufmann 1993, pp. 262–267.
7. D. Sabin, E. C. Freuder. Contradicting Conventional Wisdom in Constraint Satisfaction. *Proceedings of the Second International Workshop on Principles and Practice of Constraint Programming, Lecture Notes in Computer Science* vol. 874, Springer-Verlag 1994, pp. 125–129.
8. B. Selman, H. Kautz, B. Cohen. Noise Strategies for Improving Local Search. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, AAAI Press 1994, pp. 337–343.
9. B. Selman, H. Levesque, D. Mitchell. A New Method for Solving Hard Satisfiability Problems. *Proceedings of the Tenth National Conference on Artificial Intelligence*, MIT Press 1992, pp. 440–446.
10. B. Smith. Reducing Symmetry in a Combinatorial Design Problem. *Third International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Ashford, Kent, England, 2001, pp. 351–359.
11. B. Smith, K. Stergiou, T. Walsh. Modeling the Golomb Ruler Problem. Research Report 1999.12, University of Leeds, England, June 1999 (presented at the IJCAI'99 Workshop on Non-binary Constraints).