Supersymmetric Modeling for Local Search

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

introduction

popular SB approach: add constraints to the problem formulation

- avoids the need to modify search algorithms (often complex)
- only option available to a researcher using (eg) SAT solvers

of course there’s also SBDS etc...

SB is usually combined with backtrack search, though it's well known that it may not improve search for a single solution (hence SBDS)

but does it help local search? I added binary SB constraints to models for *cliques*, *covers*, *BIBDs* and transformed *k-SAT* problems in

S. D. Prestwich.
Negative Effects of Modeling Techniques on Search Performance. *Annals of Operations Research* (to appear).

S. D. Prestwich.
First-Solution Search with Symmetry Breaking and Implied Constraints. *CP'01 Workshop on Modelling and Problem Formulation.*

result: *SB almost always increased the number of local moves*

other bad combinations of techniques have been reported, eg backtracking can interact badly with

- domain pruning [Prosser]
- arc consistency preprocessing [Sabin & Freuder]
- removal of inconsistent or redundant domain values or subproblems [Freuder, Hubbe & Sabin]

some people think this effect is another anomaly, others that it's completely unsurprising!

the results are pretty consistent and therefore (I believe) *not* anomalies — and they surprise at least some researchers

this paper investigates further:

- why does SB harm LS? (previous explanation: reduced number of solutions)

- are *unary* SB constraints harmless? (at first sight they should be)

- does it make sense to *add* symmetry to models for LS? (opposite strategy to SB)

unary SB constraints

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 2 solutions: [a=T,b=T,c=T] and [a=F,b=F,c=F]

suppose a problem modeler realises that every solution has a symmetrical solution in which all truth values are negated

then a simple way to break symmetry is to fix the value of any variable by adding a unary constraint, eg

$$a$$

denote the 1st model by M and the 2nd by M_s

what if we apply GSAT, which makes a random truth assignment to all variables then flips to remove violations?

in M [a=F,b=F,c=F] is a solution; but in M_s clause a is violated, and any flip leads to *two* violations

so [a=F,b=F,c=F] has been transformed from a *solution* in M to a *local minimum* in M_s

local minima degrade local search performance by requiring more noise

I propose this as a general explanation: if it applies to unary constraints then it should apply even more to binary, ternary etc

but what if we apply unit propagation to the unary constraints?

applying UP to this example gives

$$b \quad c$$

which contains no local minima; will unary SB constraints always benefit search algorithms with UP?

consider DLL 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 8 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 add a unary constraint

$$\overline{c}$$

applying UP and removing redundant constraints gives

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

but the backtracker can still move smoe way towards an excluded solution:

assign d=F and apply UP

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

no empty clauses, yet d=F prevents this from being extended to a solution excluded by $\overline{c}$

so the unary constraint:

- may slow down backtrack search for a solution
- transforms a solution into a local minimum for hybrid LS such as Saturn

social golfer experiments

so adding unary SB constraints may create local minima for LS, requiring more noise and perhaps more search steps to find a solution

but does this occur in practice?

take Walser's ILP model for the Social Golfer problem, with and without SB — very symmetrical (see CSPLib problem 10)

aim to detect the effect by measuring optimum noise levels and search effort

apply Saturn LS hybrid, which has an integer noise parameter B; take medians over 1000 runs per data point

the model

main 0/1 variables $v_{pgw} = 1$ iff player p plays in group g in week w

each group has S players

$$\sum_{p} v_{pgw} = S$$

each player plays in one group per week

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

auxiliary variables $s_{pp'w} = 1$ iff in week w players p and p' play in the same group

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

no two players can play in the same group as each other more than once

$$\sum_{w} s_{pp'w} \leq 1$$

SB fix the groups in the first week

$$v_{ij1} = 1$$

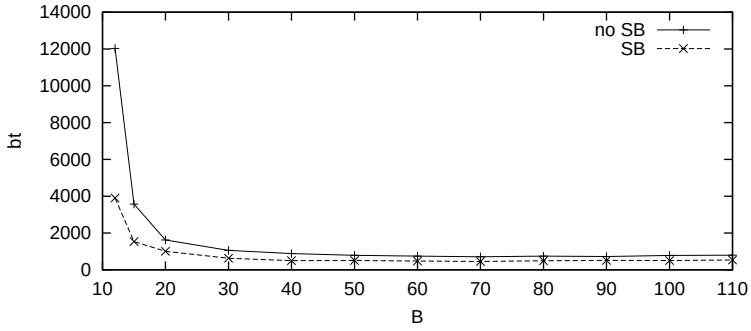
($j = (i-1)/S + 1$ rounded down) and fix player 1 in group 1 after that

$$v_{11w} = 1$$

($w > 1$)

results

instance 5-4-3

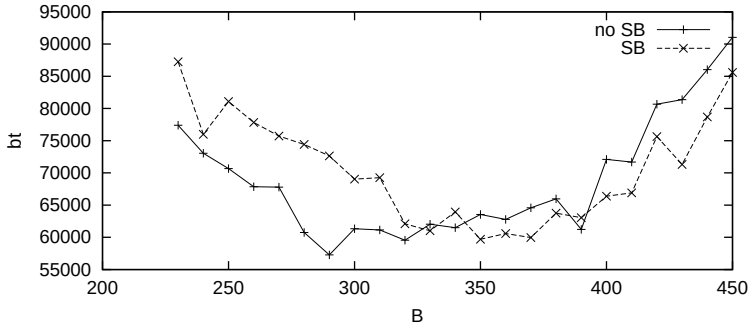


for such easy instances the added constraints consistently improve performance

because the number of search variables has been effectively reduced via UP on the unary constraints?

other easy instances give similar results

instance 6-4-5



for harder instances the results are different

the optimum noise level has increased: evidence for extra local minima

but optimum search effort is similar in both cases: positive effect of fewer search variables vs negative effect of extra local minima?

other hard instances give similar results

recommendation: *apply LS to symmetric models*

there may be other problems on which the negative effect is greater

also, extra SB constraints increase runtime overheads

LS without SB can be very effective: Saturn 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, 10-10-3 (and Kirkman's Schoolgirls in a few seconds)

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

supersymmetry

if SB harms LS, can LS be improved by *adding* symmetry?

I'll call models with added symmetry *supersymmetric*, and propose supersymmetry as a new modeling technique

an example: Golomb rulers, ie an ordered sequence of integers $0 = x_1 < x_2 < \ldots < x_m = \ell$ such that the $m(m-1)/2$ differences $x_j - x_i$ are distinct

finding a ruler with given m and ℓ is a CSP

binary/ternary model [Gent & Smith]

main integer variables $x_1 \ldots x_m$, auxiliary variables d_{ij}

ordering constraints: $x_i < x_{i+1}$

ternary constraints: $d_{ij} = x_j - x_i$

binary constraints: $d_{ij} \neq d_{i'j'}$

unary constraints: $x_1 = 0$ and $x_m = \ell$

SB constraint: $d_{12} < d_{m-1,m}$

supersymmetric model

ordering constraints: relaxed to $x_i \neq x_j$ (supersymmetry: each ruler has many permutations)

ternary constraints: changed to $d_{ij} = |x_i - x_j|$

binary constraints: unchanged

unary constraints: unchanged

SB constraint: removed

now a solution is not a Golomb ruler, but we can derive one by sorting the x_i (polynomial time)

results

compare Saturn and Walksat on several instances via direct SAT encoding [Walsh]: mean results over 50 runs for best found noise parameters (Natural/Supersymmetr models)

			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

on these problems Walksat is faster than Saturn, but both are consistently faster on the supersymmetric models, in search steps and time

conclusion

further evidence that (static) SB harms LS

likely explanation: SB constraints (even unary ones) do not prevent movement towards excluded solutions, which become local minima

new modeling technique for LS: maximize symmetry in models (but SB may help GAs because offspring of symmetrically equivalent solutions are likely to be “lethals”)

bonus: no need for complex and expensive SB constraints, so modeling for LS can be easier than for backtrack search

future work

supersymmetry seems potentially useful and I hope to find other examples in future work — perhaps some symmetry expertise could be diverted to increasing instead of removing symmetry?

I tried a supersymmetric model for the Social Golfer, allowing extra members of each group that can be dropped to get a true solution, but this involved new variables and gave worse results