An Adaptive Model Restarts Heuristic

Nina Narodytska and Toby Walsh

NICTA and UNSW, Sydney, Australia
{Nina.Narodytska,Toby.Walsh}@nicta.com.au

Abstract. We propose an adaptive heuristic for model restarts that aligns symmetry breaking with the dynamic branching heuristic. Experiments show that this method performs very well compared to other symmetry breaking methods.

1 Introduction

Symmetry is an important but often problematic feature of constraint satisfaction problems. One way to deal with symmetry is to add constraints to eliminate symmetric solutions [1–7]. Posting static symmetry breaking constraints has both good and bad features. On the positive side, static constraints are easy to post, and a few simple constraints can eliminate most symmetry in a problem. On the negative side, static symmetry breaking constraints pick out particular solutions in each symmetry class, and this may conflict with the branching heuristic. An alternative is a dynamic approach that modifies the search method to ignore symmetric states [8–11]. Whilst this reduces the conflict with the branching heuristic, we may get less propagation. In particular there is no pruning of symmetric values deeper in the search tree. An effective method to tackle this tension is model restarts [12]. This restarts search frequently with new and different symmetry breaking constraints. The hope is that we will find symmetry breaking constraints that do not clash with the branching heuristic. The original model restarts method proposed a random choice of symmetry breaking constraints. We show here that we can improve performance with an adaptive heuristic that aligns symmetry breaking with the dynamic branching heuristic.

2 An Adaptive Model Restarts Heuristic

Our adaptive heuristic collects information about branching decisions in earlier restarts in order to build a heuristic friendly ordering of variables within the static symmetry breaking constraints. This ordering is based on variable scores. We describe three different techniques to obtain these scores. The first two reuse statistics collected by the branching heuristics. If we use the domain over weighted degree variable heuristic, then we can use the DOWD ratio to compute variable scores. $Score(X_k) = D(X_k)/(w \times deg(X_k))$ [13], where $D(X_k)$ is the domain size, $deg(X_k)$ is the number of constraints involving the variable, and w the sum of the counters associated with these constraints. We order variables in increasing order of their scores. We call this the DOWD-based heuristic. Similarly, we can use statistics associated with impact based branching heuristics to build variable scores [14]. $Score(X_k) = \sum_{v \in D(x)}(1 - impact(x, v))$ where

C. Gomes and M. Sellmann (Eds.): CPAIOR 2013, LNCS 7874, pp. 369–377, 2013.

$impact(x, v)$ is the impact of a branching decision measured by the reduction of the search space induced when the decision was posted. We now order variables in decreasing order of their scores. We call this the IMPACT-based heuristic. Our third approach is based on the branching levels of variables. This offers some robustness to the choice of branching heuristic. For example, it could be used with some other branching heuristic than DOWD or IMPACT. If a variable is instantiated at level i then it gets a Borda type score of $n - i$, where n is the number of variables, and 0 otherwise. We order variables in decreasing order of the average Borda score over the last restart. We call this the ADAPT heuristic.

We use these three scoring heuristics within model restarts [12]. Model restarts was proposed to use a random variable ordering within symmetry breaking constraints in each restart. Frequent restarting ensures we eventually select a good representative symmetric solution that is aligned with the dynamic branching heuristic. Instead of using randomization, our adaptive heuristics build a variable ordering for symmetry breaking in each restart that is aligned with the branching heuristic. This variable ordering is a permutation of the original variables, and hence itself can be seen as a variable symmetry. As noted in [15], applying a symmetry to a (sound/complete) set of symmetry breaking constraints generates a new (sound/complete) set of symmetry breaking constraints. Thus, we can safely use this permutation to reorder the variables in the symmetry breaking constraints.

3 Experimental Results

We carried out experiments with 3 sets of commonly used benchmarks. We used Choco 2.1.2 on an Intel Core 8 CPU, 2.7 Ghz, 4Gb RAM with 1000 sec timeout. We branch with DOWD or IMPACT heuristics [13, 14].[1]

The first set of benchmarks, DIMACS graph colouring problems was used in earlier studies of symmetry breaking for interchangeable values [4, 16]. Such problems are particulary suitable to a dynamic symmetry breaking labeling rule that avoids symmetric solutions (DYN) [10]. We compared four symmetry breaking methods, including DYN, the static symmetry breaking precedence constraint (PREC) [4, 16], model restarts and one modification of model restarts. We use the suffix ADAPT, DOWD and IMPACT to denote that variables are reordered in the symmetry breaking PRECEDENCE constraint based on the corresponding scores.

Model restarts constructs a random permutation of variables in the scope of the symmetry breaking PRECEDENCE constraint (MR). Our adapted method works in the following way. The search starts on the model without symmetry breaking constraints. Until the first restart, we collect statistic about the search tree. If we use the ADAPT heuristic, we store the information about variables that the solver branched on as described in Section 2. If we use DOWD or IMPACT heuristics then the solver accumulates statistics in weights and impact factors. On the first restart, we order variables based on their scores obtained from the heuristic. The scores are described in Section 2. Then we post the PRECEDENCE constraint and align variables in the scope of the constraint

[1] We would like to thank Charles Prud'homme for his help in implementing the model restarts technique.

Table 1. Graph coloring. The average **t**ime(sec) over ten runs with a random seed to initialize the branching heuristic. **s** is the number of runs that finished within the timeout. #vals is the number of values in a problem.

problem	#val	Domain over weighted degree branching								Impact based branching							
		DYN		PREC		MR + DOWD ADAPT				MR_{sh} + DOWD ADAPT				DYN		PREC		MR + IMPACT ADAPT				MR_{sh} + IMPACT ADAPT			
		s	time	s	time	s	time	s	time	s	time	s	time	s	time	s	time	s	time	s	time	s	time	s	time
SAT (Satisfiable instances)																									
queen8 8	9	**10**	18.22	**10**	4.39	**10**	16.59	**10**	6.80	**10**	5.47	**10**	8.25	**10**	5.65	**10**	**3.50**	**10**	3.70	**10**	3.55	**10**	20.91	9	105.28	**10**	5.72	**10**	5.94	9	75.80	**10**	4.26
DSJC125.1	5	**10**	**0.22**	**10**	1.92	**10**	1.27	**10**	0.84	**10**	0.96	**10**	0.58	**10**	0.40	**10**	0.42	**10**	0.80	**10**	3.86	**10**	1.44	**10**	94.40	**10**	1.60	**10**	0.82	**10**	6.87	**10**	0.82
school1	14	**10**	1.41	**10**	4.53	-	-	**10**	9.34	**10**	8.76	-	-	**10**	1.75	**10**	1.79	**10**	0.26	**10**	7.30	9	**0.14**	9	**0.14**	**10**	0.41	9	0.19	9	0.16	**10**	0.30
school1	15	**10**	0.85	**10**	3.33	1	156.20	**10**	3.17	**10**	6.09	-	-	**10**	1.46	**10**	1.54	**10**	0.26	**10**	9.41	9	0.19	9	**0.18**	**10**	0.48	9	0.22	9	0.22	**10**	0.38
DSJC125.5	19	**10**	**13.30**	-	-	-	-	-	-	-	-	2	99.79	6	372.39	6	128.78	1	128.41	-	-	6	550.26	-	-	7	380.84	7	285.50	1	353.33	9	264.50
DSJC125.5	20	**10**	**0.25**	**10**	205.94	**10**	59.58	**10**	58.85	9	65.31	**10**	1.49	**10**	0.84	**10**	0.86	8	0.44	**10**	269.34	**10**	9.73	9	67.45	**10**	3.56	**10**	3.24	**10**	38.65	**10**	1.89
DSJC250.5	35	9	65.09	-	-	-	-	-	-	-	-	9	132.61	9	10.12	9	20.60	7	48.81	-	-	1	**0.22**	1	0.24	1	0.28	8	197.88	3	415.35	**10**	295.45
DSJC250.5	36	**10**	0.62	-	-	5	0.20	6	2.26	7	4.69	**10**	0.66	**10**	0.92	**10**	0.60	**10**	3.20	1	499.66	6	0.19	6	**0.18**	8	8.26	**10**	6.69	**10**	32.15	**10**	1.87
DSJC250.5	37	**10**	0.29	**10**	18.65	9	0.20	**10**	0.53	**10**	0.40	**10**	0.25	**10**	0.29	**10**	0.29	**10**	0.28	6	172.89	**10**	1.65	9	**0.18**	**10**	2.25	**10**	0.24	**10**	0.24	**10**	0.26
queen9 9	10	**10**	194.14	**10**	23.13	9	103.86	**10**	46.18	**10**	31.30	**10**	148.92	**10**	88.69	**10**	**11.60**	9	22.77	**10**	65.37	**10**	66.40	8	205.80	**10**	18.88	**10**	49.78	8	126.22	**10**	20.39
le450 15b	15	**10**	93.00	**10**	**1.99**	5	213.62	**10**	21.52	**10**	15.97	**10**	8.73	**10**	2.24	**10**	3.86	6	132.62	-	-	9	60.84	**10**	175.59	**10**	60.84	**10**	153.89	**10**	72.78	**10**	40.84
school1 nsh	14	**10**	0.32	9	777.52	-	-	**10**	1.76	**10**	3.51	-	-	**10**	0.87	**10**	1.07	**10**	0.32	-	-	3	**0.12**	3	0.13	**10**	1.40	3	0.14	3	0.13	**10**	0.71
school1 nsh	15	**10**	0.21	**10**	60.39	**10**	**0.16**	**10**	0.18	**10**	0.18	**10**	**0.16**	**10**	0.18	**10**	0.20	**10**	0.73	4	261.06	3	**0.16**	3	**0.16**	**10**	1.35	3	**0.16**	3	**0.16**	**10**	0.57
school1 nsh	16	**10**	0.22	**10**	95.76	**10**	0.17	**10**	0.18	**10**	0.18	**10**	0.18	**10**	0.21	**10**	0.19	9	1.10	-	-	4	**0.15**	5	5.01	**10**	2.07	4	**0.15**	5	67.12	**10**	0.76
queen10 10	11	-	-	2	190.35	-	-	1	**120.96**	-	-	1	979.08	-	-	**5**	516.73	2	496.60	3	500.89	-	-	-	-	-	-	3	517.91	1	414.93	1	836.35
queen10 10	12	**10**	8.81	**10**	1.29	**10**	1.26	**10**	1.17	**10**	0.86	**10**	**0.34**	**10**	0.36	**10**	0.40	**10**	0.41	**10**	1.15	**10**	1.27	**10**	1.07	**10**	1.00	**10**	0.53	**10**	1.66	**10**	0.47
TOTALS																																	
solved problems/total		15 /16		13 /16		11 /16		14 /16		13 /16		13 /16		15 /16		**16** /16		**16** /16		11 /16		15 /16		14 /16		15 /16		**16** /16		**16** /16		**16** /16	
UNSAT (Unatisfiable instances)																																	
myciel5	5	**10**	**0.95**	**10**	2.98	**10**	23.96	**10**	14.93	**10**	11.27	**10**	13.66	**10**	6.84	**10**	6.53	**10**	1.52	**10**	2.39	**10**	15.89	**10**	16.45	**10**	12.05	**10**	9.30	**10**	12.55	**10**	6.73
R50 5gb	9	**10**	**0.28**	**10**	2.92	**10**	8.80	**10**	4.61	**10**	2.18	**10**	5.16	**10**	1.97	**10**	1.80	**10**	0.68	**10**	3.45	**10**	12.42	**10**	84.77	**10**	2.90	**10**	10.14	**10**	59.13	**10**	2.09
queen8 8	8	**10**	**5.16**	**10**	10.64	7	713.59	**10**	169.65	**10**	201.60	**10**	443.13	**10**	202.86	**10**	140.87	**10**	18.25	**10**	18.83	-	-	8	352.89	**10**	214.68	2	729.62	8	235.29	**10**	153.14
mulsol.i.1	47	-	-	-	-	-	-	1	11.48	3	41.62	-	-	2	24.42	**10**	56.00	5	75.09	-	-	-	-	-	-	3	0.60	-	-	-	-	3	**0.38**
mulsol.i.1	48	-	-	-	-	-	-	1	3.65	-	-	-	-	1	98.82	**10**	48.21	5	108.72	-	-	-	-	-	-	3	0.59	-	-	-	-	3	**0.50**
school1	13	**10**	1.11	**10**	2.75	-	-	**10**	15.02	**10**	7.27	-	-	**10**	0.91	**10**	1.85	**10**	**0.15**	**10**	2.58	-	-	-	-	**10**	0.67	-	-	-	-	**10**	0.35
fpsol2.i.2	29	**10**	**1.14**	-	-	-	-	**10**	7.13	-	-	-	-	**10**	1.32	**10**	80.80	8	1.32	-	-	-	-	-	-	5	9.83	-	-	-	-	6	89.98
myciel6g	6	**10**	**0.70**	**10**	1.36	**10**	41.13	**10**	29.35	**10**	9.22	**10**	30.73	**10**	6.83	**10**	6.49	**10**	7.51	**10**	2.01	**10**	63.49	**10**	52.29	**10**	19.82	**10**	36.22	**10**	27.88	**10**	14.86
DSJC125.5	12	**10**	**4.90**	-	-	-	-	-	-	7	67.18	-	-	**10**	90.04	**10**	65.71	3	477.69	-	-	-	-	-	-	1	225.70	-	-	-	-	3	480.15
DSJC250.5	14	**10**	**131.62**	-	-	-	-	-	-	-	-	-	-	3	820.79	1	749.73	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
le450 15b	13	**10**	150.46	-	-	-	-	**10**	0.52	**10**	0.63	-	-	**10**	**0.22**	**10**	**0.22**	9	70.15	-	-	-	-	-	-	**10**	6.65	-	-	-	-	**10**	1.40
school1 nsh	12	**10**	0.14	**10**	3.64	-	-	**10**	1.80	**10**	0.85	2	815.90	**10**	0.67	**10**	0.50	**10**	**0.11**	**10**	2.12	-	-	-	-	**10**	0.69	1	563.31	-	-	**10**	0.35
school1 nsh	13	**10**	**0.14**	**10**	133.17	-	-	**10**	1.41	**10**	0.92	-	-	**10**	0.63	**10**	0.53	**10**	0.18	2	514.70	-	-	-	-	**10**	2.46	-	-	-	-	**10**	0.88
4-FullIns 4	7	**10**	121.38	-	-	**10**	407.39	**10**	3.61	**10**	1.97	**10**	46.99	**10**	1.01	**10**	0.71	6	213.38	9	241.43	8	257.65	4	791.36	**10**	1.61	**10**	171.45	**10**	206.65	**10**	**0.51**
2-FullIns 4	5	**10**	0.95	**10**	5.12	**10**	3.33	**10**	1.09	**10**	0.57	**10**	1.32	**10**	0.36	**10**	**0.22**	**10**	5.90	**10**	2.20	**10**	3.91	**10**	7.00	**10**	0.82	**10**	1.60	**10**	3.15	**10**	0.41
4-Insertions 3	3	2	896.16	**10**	599.48	-	-	-	-	-	-	-	-	-	-	-	-	**10**	**56.72**	**10**	92.25	8	689.13	7	567.20	8	564.31	9	566.94	9	464.88	9	533.42
huck	10	**10**	2.09	**10**	13.84	**10**	46.62	**10**	0.19	**10**	0.34	**10**	14.96	**10**	**0.11**	**10**	0.14	**10**	63.85	**10**	19.25	**10**	36.45	4	741.95	**10**	0.44	**10**	16.49	**10**	541.38	**10**	0.21
homer	12	**10**	158.25	-	-	-	-	**10**	2.73	**10**	2.51	-	-	**10**	**0.48**	**10**	0.58	3	752.27	-	-	-	-	-	-	**10**	4.02	-	-	-	-	**10**	0.92
TOTALS																																	
solved problems/total		16 /18		10 /18		7 /18		15 /18		14 /18		8 /18		**17** /18		**17** /18		**17** /18		11 /18		7 /18		8 /18		**17** /18		9 /18		8 /18		**17** /18	

Table 2. Graph coloring. The branching heuristic is impact based branching. The average #**b**ack**t**racks over ten runs with a random seed to initialize the branching heuristic. #vals is the number of values in a problem.

problem	#val	Domain over weighted degree branching								Impact based branching							
		DYN	PREC	MR + DOMD		MR + ADAPT		MR_{sh} + DOMD	MR_{sh} + ADAPT	DYN	PREC	MR + IMPACT		MR + ADAPT	MR_{sh} + IMPACT		MR_{sh} + ADAPT
		bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt	bt
SAT (Satisfiable instances)																	
queen8 8	9	909599	109529	426741	159069	115148	380833	229315	122541	99789	**59922**	387503	1.72e+06	82926	130842	1.99e+06	78942
DSJC125.1	5	**504**	7087	2551	994	928	2847	937	826	9902	40019	5977	1.08e+06	3518	6512	142994	3896
school1	14	8037	5667	-	6397	6669	-	2673	5699	231	7325	**4**	**4**	93	**4**	**4**	93
school1	15	3299	3472	166212	1192	3904	-	1987	3452	39	10283	**10**	**10**	125	**10**	**10**	127
DSJC125.5	19	**230634**	-	-	-	-	2.07e+06	7.61e+06	2.42e+06	1.45e+06	-	4.32e+06	-	2.88e+06	3.31e+06	3.51e+06	3.06e+06
DSJC125.5	20	**826**	2.29e+06	1e+06	1.05e+06	1.05e+06	12855	2591	2256	2324	1.89e+06	53539	390780	14711	27984	421366	11432
DSJC250.5	35	572804	-	-	-	-	1.05e+06	47883	138271	237564	-	**87**	**87**	**87**	914841	2.03e+06	1.42e+06
DSJC250.5	36	2693	-	**1**	1125	3846	229	493	145	15455	1.34e+06	17	17	7277	25888	159019	4204
DSJC250.5	37	8	85164	**5**	86	44	15	23	15	86	763464	925	9	1379	24	44	20
queen9 9	10	8.87e+06	575929	2.45e+06	1.15e+06	763242	6.75e+06	3.75e+06	426910	589342	1.21e+06	1.02e+06	2.61e+06	**278081**	1.18e+06	2.57e+06	450170
le450 15b	15	1.79e+06	**11684**	1.26e+06	81512	45835	111520	13011	32670	1.24e+06	-	297147	714565	317851	1.43e+06	679485	322694
school1 nsh	14	134	2.17e+06	-	376	1423	-	483	1274	367	-	**1**	**1**	440	**1**	**1**	440
school1 nsh	15	**0**	111659	**0**	**0**	**0**	**0**	**0**	**0**	2088	478604	**0**	**0**	437	**0**	**0**	305
school1 nsh	16	**0**	170079	**0**	**0**	**0**	**0**	**0**	**0**	4404	-	**0**	3342	999	**0**	255628	739
queen10 10	11	-	4.11e+06	-	**2.66e+06**	-	3.8e+07	-	1.89e+07	1.05e+07	7.62e+06	-	-	-	1.04e+07	7.71e+06	1.66e+07
queen10 10	12	353154	10618	7400	3688	2120	**1168**	1427	1432	3007	6614	4640	3149	2481	2139	17143	2178
TOTALS																	
solved problems/total		15 /16	13 /16	11 /16	14 /16	13 /16	13 /16	15 /16	**16** /16	**16** /16	11 /16	15 /16	14 /16	15 /16	**16** /16	**16** /16	**16** /16
UNSAT (Unatisfiable instances)																	
myciel5	5	**27100**	47759	1.1e+06	727665	518428	1.03e+06	450168	423985	52560	30503	540891	459642	327131	500700	633030	271010
R50 5gb	9	**653**	26311	150714	47840	10339	128823	12867	10848	7068	40257	174693	1.24e+06	16270	207598	1.37e+06	13847
queen8 8	8	**222329**	337913	1.86e+07	4.87e+06	5.93e+06	1.93e+07	8.64e+06	6.06e+06	521777	425590	-	6.97e+06	4.15e+06	1.73e+07	6.76e+06	4.01e+06
mulsol.i.1	47	-	-	-	8022	83800	-	145618	494667	986138	-	-	-	**101**	-	-	**101**
mulsol.i.1	48	-	-	-	1634	-	-	1.06e+06	371767	1.39e+06	-	-	-	**101**	-	-	**101**
school1	13	5935	1645	-	14005	6040	-	1038	5910	211	903	-	-	**125**	-	-	**125**
fpsol2.i.2	29	6241	-	-	1915	-	-	**1249**	530532	13968	-	-	-	5036	-	-	563975
myciel6g	6	**7757**	8552	771308	516916	140871	806771	148121	142850	161052	21137	806651	520652	232815	738818	557867	267651
DSJC125.5	12	**68793**	-	-	-	520442	-	1.29e+06	933692	5.93e+06	-	-	-	1.26e+06	-	-	4.23e+06
DSJC250.5	14	**760284**	-	-	-	-	-	4.75e+06	4.64e+06	-	-	-	-	-	-	-	-
le450 15b	13	3.64e+06	-	-	**108**	113	-	**108**	113	606155	-	-	-	8237	-	-	8237
school1 nsh	12	**13**	3458	-	521	136	8.31e+06	321	133	43	2011	-	-	115	3.87e+06	-	115
school1 nsh	13	**28**	330336	-	279	156	-	236	156	150	1.53e+06	-	-	1293	-	-	1293
4-FullIns 4	7	2.96e+06	-	881513	2679	827	854747	2506	1019	1.83e+06	582695	300621	632518	**424**	1.37e+06	1.5e+06	**424**
2-FullIns 4	5	11263	31267	9325	657	179	10679	554	**173**	118198	9011	15523	29506	1618	15523	31130	1833
4-Insertions 3	3	1.23e+08	6.22e+07	-	-	-	-	-	-	**4.75e+06**	5.97e+06	4.35e+07	3.49e+07	3.56e+07	5e+07	4.42e+07	4.43e+07
huck	10	361738	334880	1.04e+06	**214**	322	1.04e+06	**214**	334	5.37e+06	339937	1.02e+06	3.95e+07	395	974031	4.41e+07	395
homer	12	1.15e+07	-	-	1133	1279	-	**813**	1450	8.01e+06	-	-	-	2422	-	-	3181
TOTALS																	
solved problems/total		16 /18	10 /18	7 /18	15 /18	14 /18	8 /18	**17** /18	**17** /18	**17** /18	11 /18	7 /18	8 /18	**17** /18	9 /18	8 /18	**17** /18

with the obtained variable ordering. Statistics for the ADAPT heuristic are reset to zero after first and later restarts. Statistics for DOWD and IMPACT heuristics have built-in mechanisms to gradually forget pervious decisions. Between the first and the second restarts we again collect search statistics. On the second and later restarts, we remove the PRECEDENCE constraint that we posted on the previous restart from the model and post a new PRECEDENCE constraint where the variables in its scope are aligned with the ordering obtained from the heuristic. We continue this procedure until we find a solution or timeout. Note that our adaptive approach can be applied to all problems where model restarts can be applied as we replace a random ordering of variables with one derived from heuristics.

MR + DOWD and MR + IMPACT use the DOWD and IMPACT heuristics, whilst MR + ADAPT uses our adaptive version of model restarts.

We also consider limiting the cost of symmetry breaking. In MR_{sh}, MR_{sh} + ADAPT, MR_{sh} + DOWD and MR_{sh} + IMPACT, we shorten the PRECEDENCE constraint to the first $2m$ variables, where m is the number of values. The intuition behind this idea is based on an empirical observation that an instantiation of a relatively small number of variables in the scope of the PRECEDENCE constraint entails the constraint in most benchmarks. The value $2m$ was chosen based on statistical analysis of the benchmarks. We use a geometric restart policy with the base of 100 backtracks and a growth coefficient of 1.1. This ensure that restarts are rapid as in [12]. Tables 1–2 give average times and the number of backtracks for the DOWD and IMPACT branching heuristics over 10 runs. In addition, Table 1 shows the number of runs where a problem was solved. We also computed geometric means for these instances to reduce impact of outliers. However, as this gives the same picture of results and we have limited space, we do not include these results here. We removed instances solved by all methods in under 3 seconds and separated results for satisfiable and unsatisfiable.

Effect of the adaptive heuristic. By comparing PREC, MR and MR_{sh} with their adaptive counterparts, we see that our adaptive heuristic ADAPT dramatically improves performance on the majority of instances. For example, the adaptive heuristic helps solve 9 additional benchmarks if we compare MR and MR + ADAPT. The adaptive heuristic is especially useful on unsatisfiable instances. Note that many of these additionally solved benchmarks are easy once we remove conflict between the branching heuristic and static symmetry breaking. We observed that DOWD-based adaptive ordering also performs well. Unfortunately, the IMPACT ordering does not perform well on these benchmarks.

Effect of shortening. By comparing MR + ADAPT and MR_{sh} + ADAPT, as well as other models with their shortened counterparts, we see that shortening achieves much better performance. However, it slightly increases the number of backtracks in some cases. Shortening does not increase significantly the number of solved instances, or change substantially the search tree. However, it improves the efficiency of search. Overall, MR_{sh} + ADAPT gives the best performance over all benchmarks among all symmetry breaking methods using the DOWD and IMPACT branching heuristic.

Our second and third case studies consider classes of problems on which model restarts has been shown to outperform other static and dynamic symmetry breaking methods [12]. We ran experiments with the "signature" based static symmetry breaking

constraints proposed for variable and value interchangeability in [11] and denoted here as GCC-based. We decomposed the GCC constraint into AMONG constraints so we can have access to the cardinality variables. Following [12] we only order partitions within the symmetry breaking constraints. We compute a score for all variables in each partition with respect to the used heuristic and sort the partitions according to these scores. Again, the main advantage of our approach is that instead of random ordering of partitions in model restarts we align them with branching heuristics.

We generated 20 problems of each size and averaged statistics over these problems. We report time to find an optimum solution and prove optimality. Note that all results are shown on instances that are solved by all techniques for at least 10 generated problems.

As in [4, 12], we tested on graph colouring and Concert Hall scheduling problems. In [12], the model restarts technique was shown to outperform other symmetry breaking methods on these benchmarks. Hence, we only compare our adaptive strategy with the simple static symmetry breaking constraints and the highly effective model restarts technique (GCC-based +MR). As previously, we biased the ordering of variables in the simple static symmetry breaking constraint to put large partitions first. Figure 1 (left part) shows the results for uniform and biased graph colouring problems with $q = 0.5$ using IMPACT branching heuristic. The results confirm that model restarts is better than static symmetry breaking. Our adaptive ordering of partitions significantly improves performance of model restarts. In particular, the ADAPT heuristic is more robust compared to the IMPACT heuristic.

For the Concert Hall Problem, we generated problems as in [4]. As it is important to put large partitions first, we assumed that any partition with size greater than 4 is a large partition (the maximum partition size is 8 in this setup). The number of halls is 12 or 14. Figure 1 (right part) shows the results for 14 halls. As can be seen from the graphs, using an adaptive heuristic to order partitions improves model restarts significantly using both DOWD or IMPACT branching heuristics. Moreover, ADAPT shows the best performance across all instances.

4 Other Related Work

Crawford *et al.* proposed a general method to break symmetry statically using lex-leader constraints [17]. Most static symmetry breaking constraints (including the PRECEDENCE constraints used here) can be derived from such constraints. Efficient algorithms have been developed to propagate many static symmetry breaking constraints (e.g. [21–24]). Lex-leader constraints pick out the lexicographically smallest solution in each symmetry class. However, this may conflict with the branching heuristic. A number of dynamic methods have been proposed to deal with this conflict. For example, *SBDS* posts lex-leader constraints dynamically during search [8]. Another dynamic method for breaking symmetry is *SBDD* [9]. This checks if a node of the search tree is symmetric to some previously explored node. *GAPLex* is a hybrid method that combines together static and dynamic symmetry breaking [25]. However, it is limited to dynamically posting lex-leader constraints, and to searching with a fixed variable ordering (which can be a considerable burden). *Dynamic Lex* is another hybrid method that dynamically posts static symmetry breaking constraints during search which works with dynamic variable

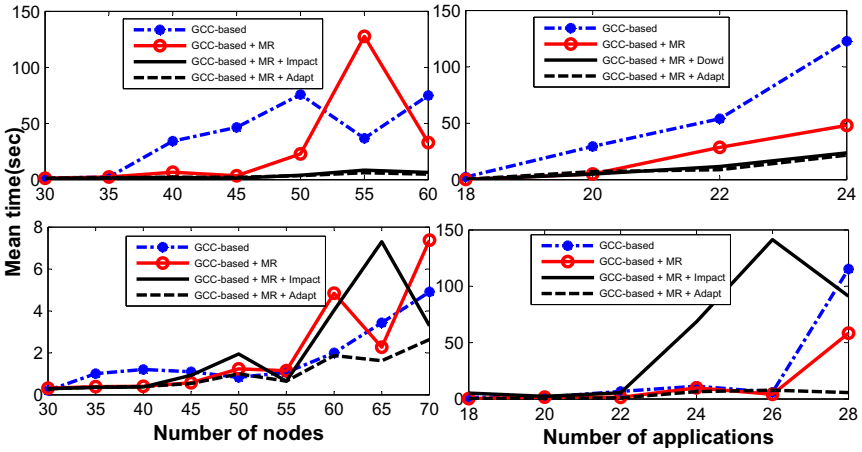

Fig. 1. Mean times for uniform (top left) and biased (bottom left) graph colouring benchmarks with q=0.5 using IMPACT based branching. Mean times for Concert Hall Problem benchmarks with 14 halls using DOWD (top right) and IMPACT based branching (bottom right).

ordering heuristics [26]. This method adds lex-leader constraints during search that are compatible with the current partial assignment. Hence the first solution found is not removed by symmetry breaking. However, unlike here, the method assumes that values are tried in a fixed order.

5 Conclusions

Static symmetry breaking constraints are often an easy and effective way to deal with symmetry in a constraint or optimisation problem. However, there can be a conflict between static symmetry breaking constraints and branching heuristics. To reduce this conflict, we propose a simple adaptive heuristic for model restarts. This orders variables within symmetry breaking constraints to align with the dynamic branching heuristic. Experimental results suggest that it is a very promising alternative between purely static and purely dynamic symmetry breaking methods. In particular, the results show that the proposed ADAPT heuristic works well across all benchmarks and two state-of-the-art branching heuristics. Our adaptive method thus appears to be more robust compared to the original model restarts algorithm.

References

1. Puget, J.F.: On the satisfiability of symmetrical constrained satisfaction problems. In: Komorowski, J., Raś, Z.W. (eds.) ISMIS 1993. LNCS (LNAI), vol. 689, pp. 350–361. Springer, Heidelberg (1993)
2. Shlyakhter, I.: Generating effective symmetry-breaking predicates for search problems. In: Proc. of Workshop on Theory and Applications of Satisfiability Testing, SAT 2001 (2001)

3. Flener, P., Frisch, A.M., Hnich, B., Kiziltan, Z., Miguel, I., Pearson, J., Walsh, T.: Breaking row and column symmetries in matrix models. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 462–477. Springer, Heidelberg (2002)
4. Law, Y., Lee, J.: Symmetry Breaking Constraints for Value Symmetries in Constraint Satisfaction. Constraints 11(2-3), 221–267 (2006)
5. Walsh, T.: Symmetry breaking using value precedence. In: Proc. of ECAI 2006, pp. 168–172 (2006)
6. Walsh, T.: General Symmetry Breaking Constraints. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 650–664. Springer, Heidelberg (2006)
7. Walsh, T.: Breaking value symmetry. In: Proc. of the 23rd National Conf. on AI, pp. 1585–1588. AAAI (2008)
8. Gent, I., Smith, B.: Symmetry breaking in constraint programming. In: Proc. of ECAI 2000, pp. 599–603 (2000)
9. Fahle, T., Schamberger, S., Sellmann, M.: Symmetry breaking. In: Walsh, T. (ed.) CP 2001. LNCS, vol. 2239, pp. 93–107. Springer, Heidelberg (2001)
10. Hentenryck, P.V., Agren, M., Flener, P., Pearson, J.: Tractable symmetry breaking for CSPs with interchangeable values. In: Proc. of the 18th IJCAI (2003)
11. Flener, P., Pearson, J., Sellmann, M., Van Hentenryck, P.: Static and dynamic structural symmetry breaking. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 695–699. Springer, Heidelberg (2006)
12. Heller, D., Panda, A., Sellmann, M., Yip, J.: Model restarts for structural symmetry breaking. In: Stuckey, P.J. (ed.) CP 2008. LNCS, vol. 5202, pp. 539–544. Springer, Heidelberg (2008)
13. ChocoTeam: Documentation. CHOCO is a java library for constraint satisfaction problems (CSP) and constraint programming (CP), `http://choco.mines-nantes.fr/`
14. Refalo, P.: Impact-based search strategies for constraint programming. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 557–571. Springer, Heidelberg (2004)
15. Katsirelos, G., Walsh, T.: Symmetries of symmetry breaking constraints. In: Proc. of ECAI 2010 (2010)
16. Walsh, T.: Breaking value symmetry. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 880–887. Springer, Heidelberg (2007)
17. Crawford, J., Luks, G., Ginsberg, M., Roy, A.: Symmetry breaking predicates for search problems. In: Proc. of the 5th Int. Conf. on Knowledge Representation and Reasoning (KR 1996), pp. 148–159 (1996)
18. Luby, M., Sinclair, A., Zuckerman, D.: Optimal speedup of Las Vegas algorithms. Information Processing Letters 47, 173–180 (1993)
19. Boussemart, F., Hemery, F., Lecoutre, C., Sais, L.: Boosting systematic search by weighting constraints. In: Proc. of the 16th ECAI 2004, pp. 146–150 (2004)
20. Gomes, C., Selman, B., Crato, N.: Heavy-tailed distributions in combinatorial search. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 121–135. Springer, Heidelberg (1997)
21. Frisch, A.M., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Global constraints for lexicographic orderings. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 93–108. Springer, Heidelberg (2002)
22. Frisch, A., Hnich, B., Kiziltan, Z., Miguel, I., Walsh, T.: Propagation algorithms for lexicographic ordering constraints. Artificial Intelligence 170(10), 803–908 (2006)
23. Law, Y.C., Lee, J.H.M., Walsh, T., Yip, J.Y.K.: Breaking symmetry of interchangeable variables and values. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 423–437. Springer, Heidelberg (2007)

24. Katsirelos, G., Narodytska, N., Walsh, T.: Combining symmetry breaking and global constraints. In: Oddi, A., Fages, F., Rossi, F. (eds.) CSCLP 2008. LNCS, vol. 5655, pp. 84–98. Springer, Heidelberg (2009)
25. Jefferson, C., Kelsey, T., Linton, S., Petrie, K.: GAPLex: Generalised static symmetry breaking. In: Proc. of 6th Int. Workshop on Symmetry in Constraint Satisfaction Problems, SymCon 2006 (2006)
26. Puget, J.-F.: Symmetry breaking using stabilizers. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp. 585–599. Springer, Heidelberg (2003)