Dimitris Sideris
1115201300158

Sorry but I couldn’t type in Greek, was in a rush

1)Modelopoihsh
variables = (I, J) ,I = 0…. n j=0 …..n
domain = 1 ……. n
neighbours = elements of same column or row
constraint = if neighbouring elements then different value
and satisfy the “clique “ that the element belongs to
NxN variables
k cliques
dictionary cliques[i] = (type , wanted result , list of elements)
I = 0...k
dictionary elementToclique[element] = I (which clique it belongs to)
dictionary values[element] = val (what value element currently has)
0(n) xwrikh poluplokothta where n = NxN
constraint complexity = 0(n)

times after 100 samples in seconds with clock() time function

	3x3			4x4		
	least	max	aver	least	max	aver
BT search	0.0004	0.0023	0.0010	0.0026	0.0261	0.0051
BT+MRV	0.0003	0.0029	0.0013	0.0032	0.4221	0.0661
BT+ FC	0.0002	0.0029	0.0010	0.0017	0.0141	0.0031
BT+MRV +FC	0.0008	0.0037	0.0016	0.0026	0.0102	0.0041
BT+mac	0.0007	0.0062	0.0015	0.0039	0.0168	0.0064
minconflicts	0.0008	40.75	27.9	106.78	143.23	119.79

	3x3		4x4	
	#assignments	#constraints	#assignments	#constraints
BT search	9	71	18	235
BT+MRV	9	78.5	336.67	4364.5
BT+ FC	9	84	17	243
BT+MRV +FC	9	86	18.15	259.23
BT+mac	9	126	16	417

minconflicts|9 / 1000000_80 / 4640782| 1000000 _ 12000192 |

	5x5			6x6		
	least	max	aver	least	max	aver
BT search	0.0055	0.02199	0.0082	0.0109	0.3450	0.0191
BT+MRV	0.0070	10.52	0.4873	0.0109	0.3450	0.0191
-						
BT+ FC	0.0026	0.0261	0.0051	0.0098	0.0251	0.0137
BT+MRV +FC	0.0073	0.01547	0.0099	0.0134	0.1056	0.0193
BT+mac	0.0090	0.1659	0.0144	0.0181	0.3466	0.0269
minconflicts	270.67	290.79	280.79	~570		

	5x5		6x6	
	#assignments	#constraints	#assignments	#constraints
BT search	27	538	40	1014
BT+MRV	1428.89	33462.2		-
BT+ FC	26	495	40	919
BT+MRV +FC		25	503.4	41.5 _ 996.82
BT+mac	25	790	40	1449
minconflicts	1000000	23741769	1000000	42001080

	6x6(hard)			7x7		
	least	max	aver	least	max	aver
BT search	0.0117	1.8236	0.0329	0.0210	1.8327	0.0458
BT+MRV	-			-		
BT+ FC	0.0104	0.2035	0.0159	0.0162	0.2580	0.0234
BT+MRV +FC	0.0145	0.0457	0.0213	0.0235	2.0025	0.0515
BT+mac	0.0193	1.2683	0.0403	0.0312	1.277	0.0515
minconflicts	-			-		

	6x6(hard)		7x7	
averages	#assigns	#constraints	#assigns	#constraints
BT search	39 _	1008	56 _	2013
BT+MRV	-		-	
BT+ FC	39 _	923	51 _	1518
BT+MRV +FC	4.4 _	927.8	51.5 _	2938
BT+mac	39 _	1514	50 _	2633
minconflicts	-		-	

	7x7(hard)			9x9		
	least	max	aver	least	max	aver
BT search	0.0198	22.756	0.2502	0.0651	306.96	3.1456
BT+MRV	-			-		
BT+ FC	0.0168	3.839	0.0595	0.0375	1.6877	0.0633
BT+MRV +FC	0.0229	1.817	0.0485	0.0535	7.7428	0.1425
BT+mac	0.0296	26.944	0.3058	0.0668	21.601	0.2912
minconflicts	-			-		

	7x7(hard)		9x9	
	#assignments	#constraints	#assignments	#constraints
BT search	54	1787	126	6352
BT+MRV	-		-	
BT+ FC	54	1516	84	3289
BT+MRV +FC	49	1508.5	81	3270
BT+mac	54	2464	82	5583
minconflicts	-		-	

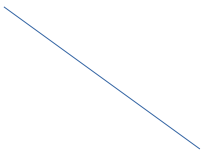
Conclusion: BT scales bad, makes sense since it has no way to “dodge” deadends
BT+MRV was bugged but is should have been than plain BT.
BT + FC scaled much better than plain BT, since it can avoid deadends and use less pointless constraints check
BT + MRV+ FC not better than BT + FC, which is surprising, proprably because of the “bug”
BT + mac also better than plain BT, but loses to BT+FC, because it wastes a lot of time in checking constraints

4) minconflicts was bad overall because it isn’t suited for the problem, since minconflicts thrives the more the solutions there are, but in this type of the problem there are cases that there can be even only one solution, which makes it very hard with minconflicts’ “randomness” to find it the vastness of possible states

3)1_
Variables = [A1,A2,A3,A4,A5]
Domains = [9:00, 10:00, 11:00]
Constraints = [A1 >A3, A3 <A4 , A3>A5, A2!=A4,A2!=A1, A4 !=10.00]
3.3 (den prolavaina analutika)
staring domains:
A1 [11:00]
A2[9:00, 10:00, 11:00]
A3[9:00, 10:00, 11:00]
A4 [9:00, 11:00]
A5 [9:00, 10:00, 11:00]

after AC-3
A1 [11:00]
A2[9:00, 10:00]
A3[10:00]
A4 [11:00]
A5 [9:00]

3.2)
A5 --------- A3----------A4



A2 -------A1