Crossword Puzzles as a Constraint Problem

Anbulagan and Adi Botea

NICTA* and Australian National University, Canberra, Australia
{anbulagan,adi.botea}@nicta.com.au

Abstract. We present new results in crossword composition, showing that our program significantly outperforms previous successful techniques in the literature. We emphasize phase transition phenomena, and identify classes of hard problems. Phase transition is shown to occur when varying problem parameters, such as the dictionary size and the number of blocked cells on a grid, of large-size realistic problems.

1 Introduction

In this paper we propose new ideas in solving crossword puzzles presented in a hybrid model with two viewpoints, one containing cell variables and the other containing word slot variables. We discuss an architecture where search and nogood learning exploit the strenghts of each viewpoint. Our program solves more crossword problems than previous successful techniques in the literature. We present the program performance on a collection of realistic puzzles, which has been used in previous studies [1,2,3].

We also analyze the behaviour of the crossword domain in detail. We emphasize phase transition phenomena and identify classes of hard problems. We discuss how the structure of a problem is affected by varying parameters such as the size of a dictionary and the number of blocked cells on a grid. Such structural changes exhibit phase transition phenomena. Unlike previous CSP contributions on phase transition (e.g., [4,5]), which always consider randomly generated problems, we experiment with large-size realistic problems.

The earliest contribution to crossword grid composition reported in the literature belongs to Mazlack [6]. In that work, a grid is filled with a letter-by-letter approach. Ginsberg *et al.* [7] focus on an approach that adds an entire word at a time. The list of matching words for each slot is updated dynamically based on the slot positions already filled with letters. Meehan and Gray [8] compare a letter-by-letter approach against a word-by-word encoding and conclude that the latter is able to scale up to harder puzzles.

Cheesman *et al.* [9] showed that the hard instances in NP-hard problems often exhibit a phase transition phenomenon. The classical phase transition in SAT [10] is observed when the ratio between the number of variables and the number of clauses varies.

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

P.J. Stuckey (Ed.): CP 2008, LNCS 5202, pp. 550–554, 2008.

2 Encoding Crosswords into CSP

Formally, a crossword puzzle consists of a grid size, a fixed configuration of blocked cells, and a dictionary. The problem is to fill the grid with words from the dictionary. No word can be placed more than once on the grid. As in [1,2], we adopt a hybrid encoding where both cells and word slots are used as CSP variables. Consider a slot s and its i-th cell c. A binary *intersection* constraint enforces that the letter assigned to c is the same as the i-th letter of the word assigned to s. Each pair of same-length slots defines a *repetition* constraint, which forbids to place the same word into two distinct slots. The hybrid encoding can be obtained from a basic model with only cell variables by applying the hidden variable transformation. It can also be seen as two combined viewpoints, one with *high-level* (or *dual*) slot variables and one with *low-level* cell variables.

3 Combus: A Crossword Composer

The solving engine highlighted in this section exploits the hybrid problem encoding by instantiating only dual variables in search and by using only low-level variables as part of nogood records. An instantiation to a dual variable in search is a macro of low-level instantiations. Macro-actions can reduce the depth of a search at the cost of increasing the branching factor per node (the utility problem). When the non-binary constraints that generated the dual variables are reasonably tight, the utility problem does not appear to be an issue. Since each dual variable contains several low-level variables, the search tree depth is reduced considerably. The branching factor can be kept low due to constraint propagation and to preferring variables with small domains to be instantiated.

In building nogoods we exploit that, in crosswords and other real-life, structured problems, a partial assignment to the dual variables can partition the uninstantiated variables into *clusters* that do not interact via common constraints. In crosswords, clustering is possible if we ignore the repetition constraints, which can connect slots on any two grid areas. A cluster is initialized to a seed slot and extended iteratively up to a fix point by adding new uninstantiated dual variables (i.e., empty or partially filled slots) that intersect the cluster.

To extract a nogood from a deadlocked node n, a *deadlock cluster* is built around the variable selected for instantiation, whose possible assignments have been invalidated either through further search or statically (via arc-consistency propagation). If n is not a leaf node and its subtree has explored instantiating variables in other clusters too, an additional condition is needed to ensure that the deadlock is contained within the current cluster, being independent of the rest of the problem. Specifically, we require that no parts of n's subtree have been pruned because of deadlocks that involve variables from other clusters. Then the instantiated cells in the deadlock cluster are a superset of a nogood.

Nogoods are stored in a database and used for pruning in the future. As nogood learning ignores repetition constraints, nogoods might be built that are actually part of a correct solution, giving up the method completeness. As repetition constraints are handled in the main search, all found solutions are correct.

4 Empirical Results on Composer Performance

We examine the performance of COMBUS on a suite of problems introduced in [1] and subsequently used in other studies [2,3]. The suite contains ten grids of each of the following sizes: 5x5, 15x15, 19x19, 21x21 and 23x23. There are two dictionaries: the smaller one, called "words", contains 45,000 words and "UK" contains 220,000 words. Each combination of a grid and a dictionary creates a problem instance. Thus, we obtain a set of 100 instances.

Table 1. Time (T) in seconds and expanded nodes (N) of the 80 instances. The 5x5 problems are too easy and their details are not presented. A dash means that no solution was found in the given time limit.

Inst	15x15 grids				19x19 grids				21x21 grids				23x23 grids			
	words		UK		words		UK		words		UK		words		UK	
ance	T	N	T	N	T	N	T	N	T	N	T	N	T	N	T	N
01	86	83	287	71	56	118	320	123	113	471	851	128	1	0	209	157
02	11	75	216	74	23	96	429	109	134	143	1133	141	81	178	697	172
03	41	71	239	73	34	315	245	108	76	139	624	130	455	12121	1185	160
04	18	304	290	66	75	127	738	116	740	15367	525	139	180	1700	715	162
05	25	65	354	70	13	112	121	115	56	238	288	132	105	178	707	170
06	84	1678	521	64	96	126	313	121	99	140	560	137	–	–	462	227
07	146	118	548	65	34	118	296	126	128	152	571	136	86	241	572	162
08	41	76	196	78	62	122	396	120	68	145	551	142	320	7842	766	156
09	25	77	210	75	23	121	342	121	64	141	479	138	689	18109	366	160
10	114	506	462	65	14	119	120	121	–	–	857	119	–	–	680	135

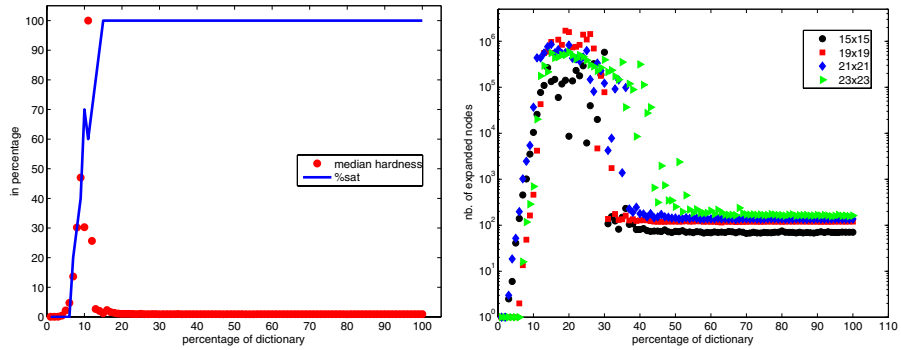
In Table 1, we present the results obtained by switching on the nogood recording procedure. COMBUS solves 97 instances in less than 20 minutes per instance on a 2.4GHz Intel Duo Core. The results show that problems corresponding to the "UK" dictionary require few node expansions. The number of expanded nodes is close to the depth of the search tree, indicating that, often, solutions are found with no backtracking. The "words" dictionary generates harder problems with respect to the number of nodes. All three unsolved problems are in this category. In general, our results are significantly better than the results presented in [1,2,3].

5 Empirical Results on Crossword Phase Transition

The experiments were run using the COMBUS engine. The incomplete method of nogood learning is switched off, to ensure that an instance reported as UNSAT has no solution indeed. In this study, we vary the dictionary size and the percentage of blocked cells.

5.1 Changing Dictionary Size

Here, we study the problem hardness and the phase transition by varying the dictionary size. The full dictionary has 220,000 words [1]. Subsets of it are



a. On 9x9 grids. b. On larger grids.

Fig. 1. Phase transition and problem hardness when percentage of dictionary varies

obtained by adding 2,200 words (1%) at a time. We created a set of 10 9x9 grids, having 12 blocked cells each, to be able to solve them all in a reasonable time and compute the percentage of SAT instances. Figure 1a shows the problem hardness (the median expanded nodes) and the percentage of SAT instances on 9x9 grids. The hard instances occur in the phase transition region, where the dictionary ranges from about 10,000 to 35,000 words.

Figure 1b presents the median expanded nodes for larger grids. The data contain ten grids of each of the following sizes: 15x15, 19x19, 21x21 and 23x23 [1]. The percentage of blocked cells is around 15 to 20%. Here, the time limit is set to 5 hours per problem. The results clearly show easy-hard-easy transitions for each grid collection. The hard region size increases with the grid size. The 15x15 data set shows a hard region from around 22,000 to 66,000 words, whereas hard 23x23 problems occur from about 22,000 to 110,000 words. Problems with a small dictionary size have no solution. The search cost to prove this is low. As we progress along the horizontal axis, problems become solvable. Finding a solution is hard at the beginning of the SAT range. The search effort decreases as larger and larger dictionaries are used.

5.2 Changing Number of Blocked Cells

In this experiment we use the 23x23 grids and the 220,000 words dictionary. To vary the number of blocked cells, we started from a configuration with 192 blocked cells (36% of all cells) placed symmetrically on the grid. We gradually removed pairs of symmetrical cells until an entirely blank grid was obtained.

Table 2 presents data showing the occurence of phase transition phenomena when solving the crossword puzzles on 23x23 grids. In the table, "Time" represents the mean runtime in seconds and "Total" represents the number of instances in the given solution region. There are three distinct experiments, one with the full dictionary (D=100%), one with half of it (D=50%), and one with

Table 2. The phase transition of crossword puzzles on 23x23 grids

Solution Region	D=100%			D=50%		
	blocked cells	Total	Time	blocked cells	Total	Time
UNSAT	0-36	19	60	0-48	25	1835
HARD	38-66	15	>86400	50-72,76	13	>86400
SAT	68-192	60	734	74, 78-192	56	228

Solution Region	D=30%		
	blocked cells	Total	Time
UNSAT	0-48	25	13
HARD	50-80,84,104,110,114,116	21	>86400
SAT	82,86-100,106,108,112,118-192	48	516

D=30%. The data show that there are more hard problems for the D=30% case. As the dictionary size gets smaller, the number of UNSAT instances increases.

The problem set is partitioned into three regions. Problems with few blocked cells have no solutions and are easy to solve. We call this the UNSAT region. Instances with a large number of blocked cells have solutions that are easy to compute (SAT region). Since we work with large problems, the hard instances between the previous two regions cannot be solved in 24 hours (HARD region).

References

1. Beacham, A., Chen, X., Sillito, J., van Beek, P.: Constraint Programming Lessons Learned from Crossword Puzzles. In: Stroulia, E., Matwin, S. (eds.) Canadian AI 2001. LNCS (LNAI), vol. 2056, pp. 78–87. Springer, Heidelberg (2001)
2. Samaras, N., Stergiou, K.: Binary Encodings of Non-binary Constraint Satisfaction Problems: Algorithms and Experimental Results. JAIR, 641–684 (2005)
3. Katsirelos, G., Bacchus, F.: Generalized NoGoods in CSPs. In: Proc. of 20th AAAI, pp. 390–396 (2005)
4. Smith, B.: Phase Transition and the Mushy Region in Constraint Satisfaction Problems. In: Proc. of 11th ECAI, pp. 100–104 (1994)
5. Gent, I.P., MacIntyre, E., Prosser, P., Walsh, T.: Scaling Effects in the CSP Phase Transition. In: Proc. of 1st CP, pp. 70–87 (1995)
6. Mazlack, L.J.: Computer Construction of Crossword Puzzles Using Precedence Relationships. Artificial Intelligence, 1–19 (1976)
7. Ginsberg, M.L., Frank, M., Halpin, M.P., Torrance, M.C.: Search Lessons Learned from Crossword Puzzles. In: Proc. of 8th AAAI, pp. 210–215 (1990)
8. Meehan, G., Gray, P.: Constructing Crossword Grids: Use of Heuristics vs Constraints. In: Proc. of R & D in Expert Systems, pp. 159–174 (1997)
9. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the Really Hard Problems Are. In: Proc. of 12th IJCAI, pp. 163–169 (1991)
10. Gent, I.P., Walsh, T.: The SAT Phase Transition. In: Proc. of 11th ECAI, pp. 105–109 (1994)