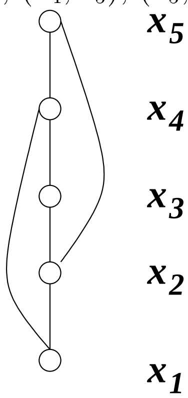
Rina Dechter, Fall-01

ICS 275A, Assignment 5

1. Read chapters 6 throughly and provide a review while answering the following questions.
   - Is the chapter clear? readable?
   - Is the chapter technically difficult?
   - Enough examples and illustrations?
   - Any typos? technical inaccuracies?
   - Are the algorithm's code clear?
   - What do you want to see more of? less of?
   - Please provide an overall evaluation

2. Let G be a constraint graph of a problem having 5 variables x_1, x_2, x_3, x_4, x_5 with arcs (x_1, x_2), (x_2, x_3), (x_3, x_4), (x_4, x_1), (x_4, x_5), (x_5, x_2).



   For a problem having the constraint graph G and using ordering $d_1 = (x_1, x_2, x_3, x_4, x_5)$, confirm, reject, and justify your answer either with an argument or a counter example:

   (a) Graph-based backjumping will always behave exactly as backtracking.

   (b) Gaschnig's backjumping will always behave like backtracking.

   (c) Conflict-directed backjumping and Gaschnig's backjumping are identical on G.

   (d) Graph-based learning over G and ordering d_1 will never record constraints of size greater than two.

   (e) If a leaf dead-end occurs at x_5, what is the induced-ancestor set $I_5(\{x_5\}$? what is the induced ancestors $I_3(x_3, x_4)$? of $I_3(x_3, x_4, x_5)$.

   (f) Propose a better ordering for graph-based jumping. Justify your answer.

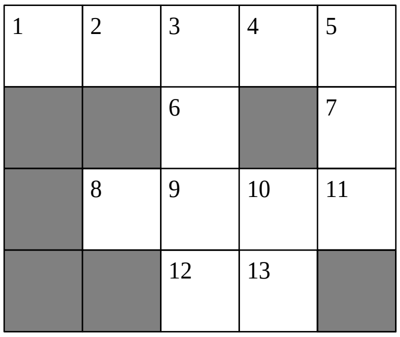


Figure 1: A crossword puzzle

(g) If a leaf dead-end occurs at x_5 and another internal dead-end at x_4, what is the conflict-set recorded by graph-based learning?

(h) How would your answer on all previous questions be different if constraint (x_2, x_3) is omitted.

3. Prove that in a "leaf deadend" jumping back to the parent in the graph is safe.

4. Let m_d be the depth of a DFS tree of an induced graph along some ordering d_1. Let d be its DFS ordering. a) prove that graph-based backjumping always jumps to its parent in the DFS tree. (b) (extra credit) Prove that backjumping along d is bounded exponentially by m_d.

5. (extra credit) Describe algorithms backjump and graph-based learning for CNF propositional theories. You can write two separate algorithms or incorporate both ideas into one algorithm.

6. Consider the crossword puzzle in Figure 1 as described in Chapter 2 formulated using the dual-graph representation. Using the ordering $(x_1, x_2, x_5, x_6, x_4, x_3)$, show a trace, whose length is limited to a 1-page description, or provide the required answer, for each of the following algorithms:

   (a) graph-based backjumping.

   (b) graph-based backjumping-learning.

   (c) jumpback-learning.

   (d) Bound the complexity of graph-based backjumping on the crossword puzzle along the ordering above.

   (e) Bound the complexity of graph-based backjumping-learning using that ordering.

   (f) Can you suggest a better ordering for graph-based backjumping for this problem?

7. Suppose we try to solve the problem in Figure 2 along the ordering $d = (x_1, x_2, x_3, x_4, x_5)$.

   (a) Show the trace and nogoods recorded by graph-based learning.

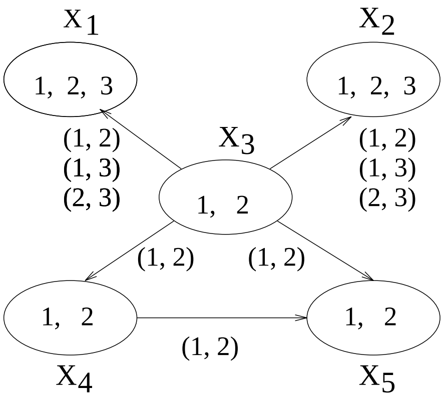


Figure 2: A small CSP. The constraints are: $x_3 < x_1, x_3 < x_2,\ x_3 < x_5,\ x_3 < x_4,\ x_4 < x_5$. The allowed pairs are shown on each arc.

(b) Discuss the performance of deep learning on that problem

(c) Show how jumpback learning will behave upon a dead-end at x_5.

8. (extra credit) Prove the correctness of FC-CBJ

9. Prove that when using the same variable ordering, Gaschnig's backjumping always explores every node explored by forward-checking.