Scribe Notes for CSE921 January 23, 2013

Daniel Dobos

February 6, 2013

Contents

1 The Simple Temporal Problem (STP)

We discussed algorithms to determine the consistency of an STP (see Table 1). Those algorithms basically enforce Path Consistency. Given that the constraints are convex, then they also they compute the minimal network in most cases.

Table 1: Algorithms for computing consistency (and minimality and decomposability) of the STP. For the complete table, see lecture slides. * can compute minimality, but must be applied back and forth.

Algorithm	**Consistency**	**Minimality**
Floyd-Warshall	*yes*	*yes*
(Incremental) Bellman-Ford	*yes*	*no*
DPC	*yes*	*no**
ΔSTP	*yes*	*yes*
PPC	*yes*	*yes*

When solving the STP, we are more concerned with the relations between variables than with the variables' domains. Because once we have ensured decomposability, we can generate any solution for any value of any variable in a backtrack-free manner.

2 Directional Path Consistency (DPC)

The algorithm for Directional Path Consistency (Algorithm 1) can be applied to the STP:

Algorithm 1: DPC (from Dechter page 355).

```
Data: A STP, S, and a variable ordering $d = (x_1, x_2, ..., x_n)$
Result: The directionally path-consistent network along d
for $r := n$ downto 1 by $-1$ do
    for all $i, j < k$ such that $(i, k), (j, k) \in E$ do
        $T_{ij} \leftarrow T_{ij} \oplus T_{ik} \otimes T_{kj}$
        $E \leftarrow E \cup (i, j)$
        if $T_{ij} = \emptyset$ then
            exit(network is inconsistent)
```

- The variables must follow some ordering, given as input, considered to be the variables' instantiation order. The ordering is implicitly/indirectly moralized.
- Vertices are traversed in reverse order (bottom up). At depth k, we revise the constraint, when it exists, between every two vertices $i, j < k$ given the constraints between (i, k) and (j, k).
- DPC determines consistency of the STP when we sweep the ordering bottom up (one direction). Once DPC gets to the top, and there are no empty intersections, then the network is consistent.
- If a constraint doesn't exist, then it is the universal constraint $(-\infty, \infty)$ and must be added if its between the parents of some variable. Remember that any interval intersected with the universal constraint is just the interval itself.
- On second pass DPC is run from the top down to filtering the rest, and ensuring minimality and decomposability in the case of the STP. Planken et al. provide the proof [4].[1] Two passes with DPC guarantees

[1]Response to question from Tony.

minimality only for temporal problems.[2]

3 Relating Consistency to Structure

Reminders from CSCE421/821:

- Given an ordering of the variables of a graph, the width of a *vertex in the ordering* is the number of its parents.
- The *width of the ordering* is the maximum widths of the vertices in the ordering.
- The *width of the graph*, w, is the minimum width of all its orderings. The minimal width can be found in quadratic time (see CSE421/821).
- The *induced width of an ordering* is the width of the moralized graph.
- The *induced width of the graph*, w^*, is the minimum induced width of all its orderings. Computing the induced width of a graph is NP-hard.

When the network is triangulated, $w = w^*$. (Guess why?)

Now, an important result by Freuder [3] links the level of consistency of a CSP to its width as a sufficient condition for a backtrack-free search: A CSP can be solved in a backtrack-free manner if its level of consistency is equal to $w + 1$.[3] As an example, consider a tree-structured CSP and apply DAC on it from the leaves up, any path in the resulting tree from the root to a leaf yields a solution in a backtrack-free manner (by simply applying backchecking).

However, the same does not necessarily hold for a path from the leaves to the root, because any value for a domain lower on the tree may not have a support higher in the tree. To fix this issue and guarantee minimality, we apply DAC again, this time from the root to the leaves. The above illustrates we have to sweep in both direction and is the basis for tree-structured methods for solving CSPs (Chapter 9 of [2]):

- Consider a tree embedding of the constraint network, where the largest cluster has k variables, then the maximum number of parents any node can have is k.
- Induced width determines level of consistency required for minimality: $W^* + 1 =$ *consistency level*

[2]Response to question from Robert.

[3]In practice, k-consistency is not useful because enforcing it may change the structure of a graph, thus increasing its (induced) width.

- The result about tree decomposition is very important and exploited in many areas in Computer Science (Bayesian Networks, Databases, etc.). More generally, we talk about reasoning in *graphical models.* Within graphical models the concept of width is also important, as width bounds many things (e.g., the cost of reasoning, of answering queries).

4 ΔSTP

ΔSTP is another method of solving the STP [5]:

- Like DPC but does not generate all edges as DPC does, just those edges required to triangulate the graph.
- In ΔSTP each triangle is like a ternary constraint. When a change is made we just update the neighboring triangles.
- Effort is bound by size of largest connected component, instead of by triangles[4].
- Also, any update remains local within the biconnected component.[5]

Definition 1 Articulation point/node and bi-connected component *An articulation point is vertex that, when removed from the graph, the graph becomes disconnected into two or more biconnected components.*

5 Incremental Bellman-Ford

Incremental Bellman-Ford is another algorithm for determining the consistency of an STP. It was proposed for temporal in space mission applications [1]. It is extension of the Bellman-Ford (single source shortest path) algorithm in order to handle incrementality.

References

[1] Amadeo Cesta and Angelo Oddi. Gaining Efficiency and Flexibility in the Simple Temporal Problem. In *International Symposium on Temporal Representation and Reasoning (TIME 96)*, 1996. Available from instructor.

[2] Rina Dechter. *Constraint Processing.* Morgan Kaufmann, 2003.

[4] Also holds for any tree decomposition.

[5] Result does not hold in general.

[3] Eugene C. Freuder. A Sufficient Condition for Backtrack-Free Search. *JACM*, 29 (1):24–32, 1982.

[4] Léon Planken, Mathijs de Weerdt, and Roman van der Krogt. P3C: A New Algorithm for the Simple Temporal Problem. In *Processings of the 18th International Conference on Automated Planning & Scheduling (ICAPS 2008)*, pages 256–263, 2008.

[5] Lin Xu and Berthe Y. Choueiry. A New Efficient Algorithm for Solving the Simple Temporal Problem. In Mark Reynolds and Abdul Sattar, editors, *10th International Symposium on Temporal Representation and Reasoning and Fourth International Conference on Temporal Logic (TIME-ICTL 03)*, pages 212–222. IEEE Computer Society Press, 2003.