Scribe Notes
9/10/2009

Presenter: Shant
Scribe: Robert

Paper: An Efficient Consistency Algorithm for the Temporal Constraint Satisfaction Problem

Below, we summarize the presentation of the paper and the class discussion.

To solve the TCSP, Dechter proposes to model it as a meta-CSP:

- n vertices (time points), e edges (distance constraints), k intervals per edge.
- Edges in the TCSP are the variables in the meta-CSP
- Labels of the edge in the TCSP are the domains in the meta-CSP
- Global constraint in meta-CSP: edge-interval pairs must form a consistent STP. The size of the constraint is $O(k^e)$

What kind of constraint propagation can we do for the TCSP?

- Arc Consistency: Since there is a single *n*-ary constraint, GAC is NP-Hard
- NPC1/NPC2 : Cause fragmentation
- ULT (not mentioned in the talk) is an approximation.
- The paper proposes another approximation, which does not cause fragmentation but does some domain filtering in the TCSP.

New algorithm (ΔAC)

- Works on *existing* triangles, ΔAC is polynomial
  - o Given a triangle, an interval *a* on an edge is kept if we can find two intervals *b* and *c* on the other edges in the triangle whose composition (*bc*) intersects with the interval *a*.
  - o We say that *a* is supported by *b* and that *c* and *b* and *c* support *a*. If no two such intervals *b* and *c* exist, the interval *a* is removed.
    - ▪ Example: ([3,6] • [-3,-1]) ∩ [6,9] = [0,5] ∩ [6,9] = {}, so you would eliminate [6,9].
  - o This operation yields a domain filtering of the meta-CSP, thus reducing the size of the search tree for solving the TCSP.
  - o Conceptually, the algorithm "imposes" a new constraint on each existing triangle in the graph of the TCSP.
  - o Shant argued that we need to be careful about the orientation of the edges when checking whether a given interval is supported or not. When the direction of an edge does not allow to do the composition operation with another interval, the edge's direction can be inverted by inverting the polarity of the bounds of the interval considered: Example: [1,3] => [-3,-1].
- Check consistency of triangle by finding supports for all the values (intervals) for the variables (meta variables).
- Either ΔAC removes an interval or it leaves it: it does not cause fragmentation.
- ΔAC uses data structures that are support tables: Supports and SupportedBy like AC-4. Like AC2001, it canonically orders the values (intervals) in the domain and records where the last support was at. It only moves the "pointer" of the support forward, not backwards since we already checked the previous values.

- Shant explained that only triangles existing in the TCSP graph need to be considered because the intersection of any constraint with the universal constraint is always not null.
- uses a queue of (edge,interval)-pairs, and looping through the queue, checks if an edge-interval pair has a support in each triangle in which the edge appears.
    - Initializes the queue to all variable-value pairs
    - If a variable-value pair is not in a triangle, it is removed from the queue. (Alternatively, we could avoid putting in the queue pairs pertaining to edges that appear in any triangle.)
    - Else we check all the triangles of which the edge is a part of. When we remove a value, we put in the queue the vvps that are supported by the removed value.

Shant stated that when an interval *a* in a triangle is supported by two intervals *b* and *c*, then he claims that it is the case that *b* is supported by *a* and *c* and *c* is supported by *a* and *b*. He further says that if this claims is correct, then the performance of ΔAC can be improved by reducing the unnecessary search of supports for *b* or *c* once *b* and *c* have been found to support *a*. Berthe, who was first skeptical, thought that this may well be the case but it still needs to be formally established. She explained that one possible tool to be used for the proof is the Helly's theorem, which is important for proving the tractability of convex constraints.

- **Helly Theorm:** n objects that are convex, $\mathbb{R}^d$, n>d. If the intersection of every d+1 objects is not Ø, the intersection of all n objects is not Ø.
- In our case, d=1. Because of this, if we find a support for value in two other variables, we do not need to find a support for those two values. (Shant's improvement may be proven sound using this tool.)
- Any volunteer to write down the proof? Test the effect of the improvement on ΔAC?
- Alert: it may take more work than it appears to establish the improvement. Even if *c* is supported by *a* and *b* in the example above. It may be the case that *c* is also supported by *a* "previous" value appearing before *a* in the domain of the same variable than *a*. Ignoring that previous value and choosing *a* as support for *c* may yield an incomplete algorithm given that ΔAC uses the "last support" pointer of AC2001.

Shant presented the advantages of:

- More powerful in dense TCSPs (where there are more triangles, and thus, more filtering can be done).
- Sound and cheap $O(n|E|k^3)$.

Shant then showed two types of experiments to demonstrate the effectiveness of ΔAC and suggested to use ΔAC for look-ahead, during search, instead of only as a preprocessing step:

1. Comparing, over increasing densities:
    a. the size of the original TCSP (product of domain size),
    b. the size of the filtered TCSP (product of the filtered domain size)
    c. the number of solutions is the TCSP.

   This experiment shows that the ΔAC effectively reduces the size of the TCSP, increasingly reducing its size to almost the minimal network (very close to the number of solutions) as the density increases.
2. Comparing the effect on CPU, nodes visited, and constraint checks of:
    a. Solving the TCSP w/ three different search techniques
    b. Solving the TCSP w/ those techniques after preprocessing by ΔAC.

He showed up to orders of magnitude improvements in the performance of search, especially as the density increases.

Wesley asked how much more filtering would GAC do (If you were able to) beyond ΔAC?

- GAC: Every value can be extended to the remaining variables in the scope of the constraint. GAC on the TCSP, if it were possible, would make the domains minimal, and the TCSP decomposable, allowing us to build a solution in a backtrack-free manner. Life would be good. However, enforcing GAC would in fact correspond to solving the TCSP, which is NP-hard.
- ΔAC: Every variable-value pairs can be extended to 2 other variables (the three variables in a triangle). ΔAC filters the domain, but cannot guarantee the minimal domain. Search remains necessary.