Question by Daniel Dobbs:

How do we know when a constraint is convex?

a. In a CSP, we can look at the bit-matrix representation of the binary constraints:

\[
\begin{bmatrix}
0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1
\end{bmatrix}
\rightarrow
\begin{bmatrix}
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0
\end{bmatrix}
\]

If we can swap rows in the bit-matrix representation such that all of the ones in each row are consecutive (consecutive-ones property), then we say that the constraints are row convex. If the constraints remain row convex after enforcing consistency (a posteriori condition), then a known result guarantees that PC => Minimality => Decomposable.

b. In a CSP where constraints or domains can be arranged as intervals with a total ordering and that those intervals are not fragmented by the operations of enforcing consistency (composition and intersection), then the constraints are convex.

Temporal Reasoning: \(\Delta STP\)

The algorithm:

- Is a refinement of the algorithm for PPC
- Works by updating all edges in a triangle at every propagation step
- Changes are propagated to adjacent triangles

Advantages of \(\Delta STP\)

- Cheaper than PPC and F-W (empirical proof)
- Guarantees the minimal network
- Automatically decomposes graph into its bi-connected components
  - Binds effort in the size of the largest component

---

1 See textbook page 231—237.
allows for parallelization
• sweep through back and forth (relation to tree decomposition made by N. Wilson in 2005).

Recent Advances in ∆STP ⁴
• exploit structure
  o order variables linearly
  o use a PEO (bottom-up ordering) or Max Cardinality ordering (top down ordering)
• apply directional path consistency (Ch 4. Dechter): determines consistency
• propagate down: provides minimal network

Temporal Reasoning: Solving the TCSP

Review of TCSP
• variables represent time points, each with a continuous domain
• each constraint is a disjunction of intervals
• solution is an assignment which does not violate any constraints
• deciding consistency of a TCSP instance is NP-complete

Solving the TCSP
• formulate TCSP as meta-CSP
• find all solutions to the meta-CSP
• use ULT (Dechter’s textbook), or ∆STP to solve the individual STPs efficiently (Xu Lin’s CP 2003)

Preprocessing the TCSP

- Arc consistency on the meta-CSP (with GAC) is NP-hard because it corresponds to solving the TCSP
- Use $\Delta AC$ to filter the domains of TCSP with ternary constraints
- $\Delta AC$ removes values that are not supported in the ternary constraint.

$\Delta STP$ is very effective at low density, $\Delta AC$ is very effective at high density. We can combine them to create a powerful tool.