Summary: Robert Woodward’s presentation on
Properties of Tree Convex Constraints

Peter Schlette

November 17, 2009

1 Story

• This paper builds on the concept of tree convexity.
• (Tree convexity ∧ path consistency) → global consistency.
• If the CSP is not path consistent, we need to enforce PC.
• Path consistency algorithms utilize the ∩ and ◦ operators
  – While the ∩ operator preserves tree the tree convexity property, the ◦ operator may damage it, so how can we ‘recuperate’ global consistency?
• The authors propose the consecutiveness property, which is closed under ◦, but not ∩.
• Then they propose the “Locally Chain Convex” (LCC) property, which is closed under ∩, but not ◦.
• When graphs are LCC and also Strictly Union Closed (SUC), they’re closed under both ∩ and ◦, so we can safely enforce path consistency!
  – Thus (LCC ∧ SUC) → (PC ∧ → global consistency).

2 Introduction

What is tree convexity?
• Defined on a binary constraint network.
• The constraint $C_{xy}$ is tree convex iff the domain of the variable $Y$ can be arranged into a tree such that the *image* (defined below) of each of the values of variable $X$ are all subtrees of this tree.

• A linear time algorithm to find a tree that satisfies this property exists [Conitzer+ Al104].

• Tree convexity can also be defined for a CSP: if every edge in the graph is tree convex, then we say that the CSP itself is also tree convex.

3 Definitions

• $C_{xy}$ and $C_{yx}$ are considered as distinct constraints, but one is the inverse of the other.

• *Image*: With respect to a constraint $C_{xy}$, the image of $u \in X$ is the set of all values $v \in Y$ that support it. Can also be defined for multiple values of $X$ (or the whole domain) by unioning the individual images. This definition is obvious when you think of the constraint as a relation between two sets.

• Reviewed graph concepts:
  - A tree is a connected graph with no cycles.
  - A chain has at most one child per node.
  - A forest is a set of trees. The paper assumes that the root of each tree is given.

• If a CSP is tree convex and path consistent, then it must also be globally consistent.
  - Robert covered the inductive proof of this claim on the slides.

From the paper:

• **Tree Convex (TC)** - Sets $E_1, \ldots, E_k$ are tree convex with respect to a forest $T$ on $\bigcup_{i \in 1..l}$ if every $E_i$ is a subtree of $T$.
  A constraint $C_{xy}$ is tree convex with respect to a forest $T$ on $D_y$ if the images of all values of $D_x$ are tree convex with respect to $T$.

  A constraint network is tree convex if there exists a forest on the domain of each variable such that every constraint $C_{xy}$ of the network is tree convex with respect to the forest on $D_y$.

• **Consecutive** - A tree convex constraint $C_{xy}$ with respect to a forest $T_y$ on $D_y$ is consecutive with respect to a forest $T_x$ on $D_x$ if and only if for every two neighboring values $a, b$ on $T_x$, $I_y(a) \cup I_y(b)$ is a subtree of $T_y$. 

• **Locally Chain Convex (LCC)** - A constraint $C_{xy}$ is locally chain convex with respect to a forest on $D_y$ if and only if the image of every value in $D_x$ is a subchain of the forest.

A constraint network is locally chain convex iff there exists a forest on each domain such that every constraint $C_{xy}$ is locally chain convex with respect to the forest on $D_y$.

• A constraint $C_{xy}$ is locally chain convex and strictly union closed (LCC & SUC) with respect to forest $T_x$ on $D_x$ and $T_y$ on $D_y$ iff the image of any subchain on $T_x$ is a subchain of $T_y$.

A constraint network is LCC & SUC iff there exists a forest on each domain such that every constraint $C_{xy}$ of the network is LCC & SUC with respect to the forests on $D_x$ and $D_y$.

### 4 Properties

• Tree Convexity (TC)
  - The intersection of two tree convex constraints is still TC, due to the intersection of two subtrees still being a subtree of $T$.
  - However, this is not the case for composition; it will destroy TC in the general case.

• Consecutiveness
  - Consecutive constraints are closed under composition!
  - As before with TC, this can be generalized to CSPs by defining a consecutive CSP to be a CSP whose edges are all consecutive.

### 5 Tractable Networks

• Locally Chain Convex (LCC)
  - If a constraint is LCC, it will remain LCC after the removal of a value from the domain
  - However, composition can still destroy LCC.

• We’ll want both LCC (for closure over intersection) and consecutiveness (for closure over composition) combined. How to do this?
• LCC & SUC (Locally Chain Convex and Strictly Union Closed).
  – This set of properties means that the network can be rendered globally consistent in polynomial time.
  – We can prove that LCC & SUC CSPs are closed under both intersection and compositions; Robert reviewed the steps during the presentation.

6 Wrap-Up

• Robert demonstrated an application of tree convexity, a specific instance of the scene labeling problem.

• However, the existence of efficient algorithms for finding local chain convexity and strict union closedness is still open, so present application is limited.

• The approach is compared to tractable languages by Jeavons et al. and to another convexity property by Kumar.
  – The authors argue a fundamental distinction between the tractability of constraint languages and that of constraint problems.
  – The authors argue that the convexity property introduced by Kumar guarantee tractability assuming the use of randomized algorithm, and not deterministic algorithms as in this paper.