

A Tool to Graphically Edit CP-nets

Aidan Shafran

Sam Saarinen

University of Kentucky
Lexington, KY 40506
samuel.saarinen@uky.edu

Judy Goldsmith

University of Kentucky
Lexington, KY 40506
goldsmi@cs.uky.edu

Abstract

Conditional preference networks (CP-nets) are a mathematical formalism for compactly representing preferences over combinatorial domains. The software package presented allows editing of CP-nets through a graphical interface, loads and saves to an XML-based file format, and detects properties of the currently loaded CP-net.

Background Theory

Qualitative preferences over outcomes in a combinatorial domain (where many variables jointly describe the outcome) are useful in automated decision making and modeling human preferences in real world domains. Conditional Preference Networks (CP-nets), also known as Ceteris Paribus Networks, are a compact graph-based mathematical formalism for representing such preferences (Boutilier et al. 2004).

A CP-net is a directed graph where nodes represent domain variables, and where an edge from v to w indicates that preferences over the domain of w depend (in part) on the value of the v variable. In addition, each node in the graph (each preference variable) is annotated with a conditional preference table (CPT), which expresses, for each combination of possible values for the parent nodes, the preference ordering over values of the child node. In Figure 1, each row of the CPT for the “Model” variable corresponds to one possible combination of values for parent nodes. Here, a_1 and a_2 represent the values of “binary” and “multi-valued” respectively for the “DomainSize” variable. The row a_2b_3 : $CP - net \succ TotalOrder$ says that when the “DomainSize” variable is “multi-valued” and the “Features” variable has value 3, the value “CP-net” is preferred to the value “Total Order” for the “Model” variable.

Each CP-net implicitly defines a graph on outcomes, which is exponentially larger than the CP-net. Directed edges in this “preference graph” correspond to changing the value of one variable from a less-preferred to a more-preferred value, given the fixed values of its parent’s values. Given a CP-net N and two outcomes o_1 and o_2 , we say that $o_1 \prec o_2$ (o_1 “is dominated by” o_2) if there is a directed path from o_1 to o_2 in the preference graph. Since each edge in the

path corresponds to an improving flip of a variable’s value, we call the path an improving flipping sequence.

The problem of finding such flipping sequences (if they exist) can be cast as a planning problem and in the case that only CP-nets whose graphs are acyclic are considered, the problem of finding flipping sequences is in NP (Goldsmith et al. 2008). The nature of the relationship between CP-net structure and expected flipping sequence length is an area of open research.

The Presented Technology

The CP-net visualizer presented is useful for researchers eliciting human preferences, building CP-nets for specific experiments, visualizing generated CP-nets, and for the general public learning more about preference modeling. It has an interface consisting of three vertical panels. On the left is a control panel for managing the entire CP-net, and it currently has support for import and export in a file format extending XML, export as an SVG (for including in papers or sharing with others), and editing the structure of the CP-net graph by adding, removing, or connecting nodes. There are also options to enforce the acyclicity of the constructed graph, and to restrict the indegree of nodes in the graph (effectively limiting the length of the CPTs).

In the center, the CP-net is displayed, and nodes can be repositioned manually, or allowed to position themselves automatically. If a node is selected, it appears highlighted, and edges are highlighted if the CPT of the child node is not dependent on the value of the parent.

The rightmost panel allows data for a selected node to be edited, including its name, domain values, parents, and CPT. The initial conditional preference table is generated automatically, and each row can be edited conveniently by clicking on the values that should be changed.

This tool is designed to be easy to extend with functionality to animate a provided flipping sequence between two outcomes, and eventually client-side plugins to generate CP-nets and flipping sequences between those CP-nets. It is also possible to add support for more general expressions of incompleteness of data and indifference across domain values in a single row of the CPT.

The CP-net visualizer is written in JavaScript and packaged as a webpage that can be accessed on a website <http://cs.uky.edu/~goldsmi/visualizer/> or saved

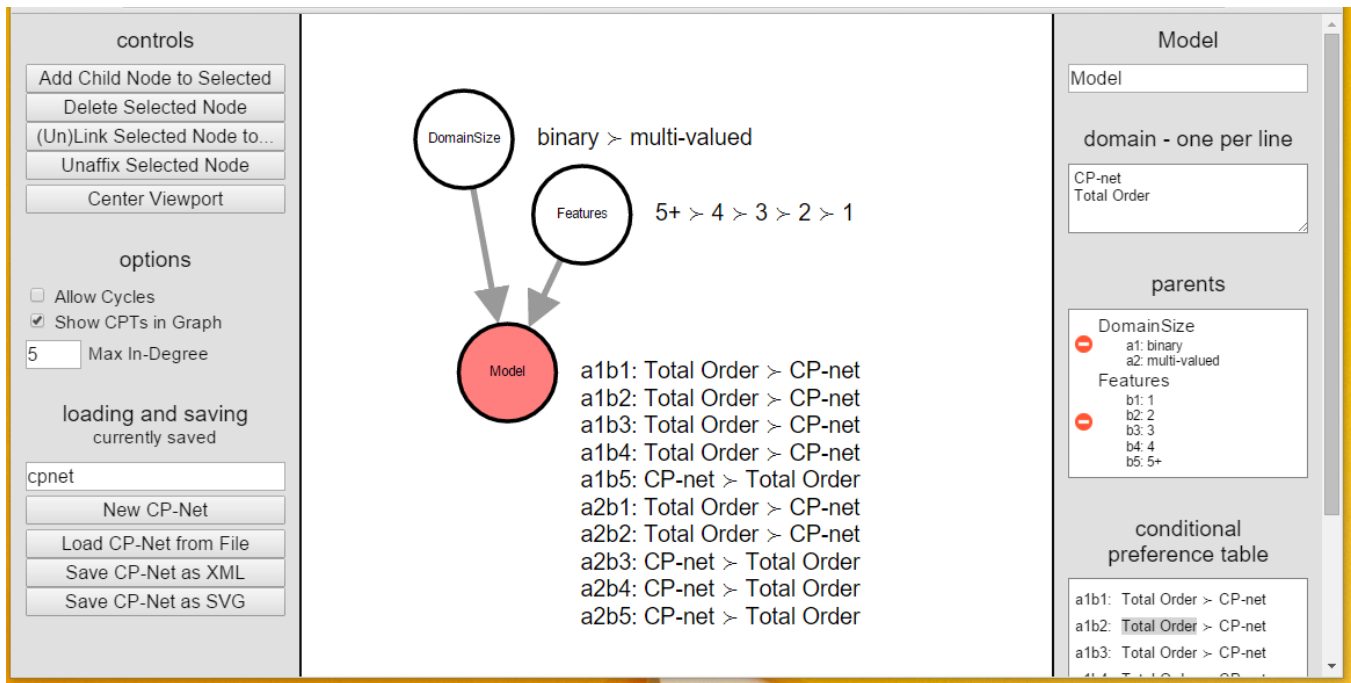


Figure 1: Here, the “Model” node is selected.

locally. It can be run in most web browsers on most platforms. It uses multiple third-party open-source JavaScript libraries, including D3 and WebCoLa. D3 provides Scalable Vector Graphic (SVG) rendering capabilities, and WebCoLa provides the constraint-based physics engine used to position the nodes in the graph. The visualizer is provided open-source and free of charge under the MIT License, as are most of the third-party libraries it uses.

Purpose of the Demonstration

We will demonstrate the current functionalities of the CP-net visualizer, including its use for eliciting human preferences and visualizing generated CP-nets. This tool is designed to be easy to extend. For instance, Allen et al. (Allen et al. 2016) have code for generating CP-nets uniformly at random, given the number of nodes and, optionally, bounds on in-degree. We will be enabling client-side plugins for that code, and for code to find *and animate* flipping sequences. Support can be added for learning CP-nets, and for handling incomplete CPTs and indifference across domain values in a single row of a CPT.

The demonstration is designed to make this tool known and available to researchers around the world who are working with CP-nets, both for research purposes and for outreach. An easy-to-understand tool like the one presented will make it easier to communicate with the public about work in understanding decision-making, and will make the mathematical models used by researchers more accessible to the general public.

References

- Allen, T. E.; Goldsmith, J.; Justice, H. E.; Mattei, N.; and Raines, K. 2016. Generating CP-nets uniformly at random. In *Proc. AAAI*.
- Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135–191.
- Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2008. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research* 33:403–432.