Graphically Structured Value-Function Compilation

Ronen I. Brafman BRAFMAN@CS.BGU.AC.IL
Department of Computer Science
Ben-Gurion University
Beer Sheva, Israel 84105

Carmel Domshlak DCARMEL@IE.TECHNION.AC.IL
Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
Haifa, Israel 32000

Abstract

Classical work on eliciting and representing preferences over multi-attribute alternatives has attempted to recognize conditions under which value functions take on particularly simple and compact form, making their elicitation much easier. In this paper we consider preferences over discrete domains, and show that for a certain class of simple and intuitive qualitative preference statements, one can always generate compact value functions consistent with these statements. These value functions maintain the independence structure implicit in the original statements. For discrete domains, these representation theorems are much more general than previous results. However, we also show that it is not always possible to maintain this compact structure if we add explicit ordering constraints among the available outcomes.

1. Introduction

The spectrum of practical problems that require reasoning about preferences is extremely wide. In this paper we consider the problem of eliciting and reasoning about a user's *ordinal* preferences. We are motivated in part by the needs of large-scale, consumer product catalogs, an area that has received growing attention in the fields of the database systems and AI (e.g., see [1, 13, 5, 6, 9, 26, 31, 29].)

Online catalogs of products and information grow continuously, and with them grows the number of lay users accessing these catalogs. While keyword search provides users with some means to access these catalogs, user needs in such shopping contexts are typically more complex than in web search. In particular, users have personal preferences regarding price, quality, features, etc., and these preferences can be rather complex. Therefore, it is natural to expect that systems supporting this search process will aim to allow users to state their actual preferences, and that reasoning about such preferences can improve the understanding of user needs.

Unfortunately, it appears that achieving *both* user-friendly, robust preference elicitation and efficient reasoning about the elicited information is not easy. The conflict between these two desiderata is reflected by the conflicting forms in which a user might be asked to provide her preferences. On the one hand, if the user provides us with a *numerical value function* over the space of the products (henceforth referred to as *items*), ordering the catalog with respect to this function is easy. However, eliciting a quantitative description of preferences from the users is generally a long, involved and time-consuming process that is often unintuitive to users. Alternatively, we can consider allowing users to express their preferences using natural-language like *qualitative statements*, providing us with pieces of preference information like (i) "For a family car, I prefer white color to all other colors", or (ii) "This car is better for me than that car", or (iii) "This mini-van would be better in blue", or (iv) "I like ecologically friendly cars". This form of preference elicitation is considered to be more natural to users [18], and thus dealing with this form of preference information has received significant attention in the multi-disciplinary preference literature (to name just a few works, see [19, 32, 30, 8, 13].) Unfortunately, those preference expressions that can be reasoned about efficiently (at least for ordering a given set of items) are required to be "syntactically homogeneous", that is contain only statements in a certain specific form [14, 8]. For instance, to the best of our knowledge, there is no known general class of preference expressions containing statements of both forms (i) and (ii) as above [1] that can be reasoned with efficiently.

Striving to enjoy the pros of both a qualitative input and a quantitative representation of user preferences, in this paper we consider *compiling* qualitative preference expressions into value functions consistent with the information carried by these expressions. The main contributions of this paper are as follows:

1. We provide a new *representation theory* for generalized additive value functions [15, 2], and specify conditions under which there exists a particular factored value function consistent with (what is known about) the user's preference relation. Our representation theorems show that preference orders induced over the item space by certain

1. Later we define these forms of statements in a formal manner.

sets of qualitative statements of preference and importance can always be consistently captured by a compact generalized additive value function. In particular, our results extend the classical representation theorems for additive value functions over discrete variables [21]. As the conditions we require are much weaker than those required for an additive representation, we are able to capture a significantly wider spectrum of sets of natural preference statements, namely those representable by the TCP-net model [10, 12].

2. We show how our representation theory can be utilized in a *computationally efficient* methodology for eliciting and reasoning about ordinal preferences of the users. In this methodology, the user provides a set of qualitative preference statements, and these statements are used to efficiently generate a compact value function whose structure is based on the qualitative information supplied by the user. The key part is that the existence of such a compact value function, its consistency with the preference statements of the user, and efficiency of its generation are guaranteed by our representation theory.

3. In many applications, it is desirable to allow the users expressing not only structured preference information, but also direct rankings between pairs of concrete items (e.g.,see [26, 5, 20].) We consider the computational consequences of supporting both general statements of preference and such pair-wise item rankings. On the positive side, we show that such an extension can be straightforwardly supported in our methodology while preserving its soundness and efficiency. On the negative side, however, we formally show that completeness of structured value-function compilation is extremely sensitive to adding such item-level rankings. Specifically, we show that completeness of value-function compilation cannot be guaranteed even if the amount of such pair-wise item rankings is minimal, and that this impossibility result holds for most languages of generalizing preference statements.

The rest of the paper is organized as follows. In Section 2 we provide some essential background on qualitative preference statements targeted in this work, the TCP-model for modeling sets of such statements, and value functions. Section 3 is devoted to the value-function compilation of three progressively more complicated classes of TCP-nets. For clarity of presentation, the longer proofs are given in Appendix A. In Section 4 we consider extending structured preference information with pairwise comparisons between completely specified alternatives, provide an impossibility theorem on value-function compilation of such mixed sets of statements, and generalize this result to a general impossibility theorem. We summarize and list some open problems in Section 5.

2. Background and Notation

Let Ω be a space of alternative items, where each item is described by an assignment to a certain finite set of attributes (= variables) $\mathcal{X} = \{X_1, \ldots, X_n\}$ with domains $dom(X_i)$, respectively. Without loss of generality, the item space Ω is then considered to be $\Omega = \times dom(X_i)$. For instance, if Ω is the universe of descriptions of commercial flights for a certain time period, then the attributes might correspond to the departure time, airline, etc. Typically, the set of available items is a subset of Ω, and is described by some database. However, although the user is familiar with the item's attributes, she does not know, a-priori, what items appear in this database. In this work, we assume that all $dom(X_i)$ are finite, and thus Ω is finite as well.

To fix the basic notation, in what follows we use regular uppercase letters $X, Y, \ldots$ for variables, regular lowercase letters $x, y, \ldots$ for values of individual variables, calligraphic uppercase letters $\mathcal{X}, \mathcal{Y}, \ldots$ for sets of variables, and bold lowercase letters $\mathbf{x}, \mathbf{y}, \ldots$ for assignments to sets of variables.

In the rest of this section we define a set of qualitative preference statements, a graphical structure that is used to analyze them, and the type of value functions into which these statements will be compiled.

2.1 Qualitative Preference Statements

Our primary aim as a system is to build a model of the ordinal preferences of a given user over Ω. Our basic assumption is that there exists a relatively compact and sufficiently accurate representation of the user's preferences in terms of $\mathcal{X}$. In turn, to actually build an effective user model we obviously need information about the user's preference relation over different assignments to $\mathcal{X}$. Focusing on qualitative preference information, we begin with considering two types of natural preference statements, together with a set of preferential independence assumptions that are implicit in the user's specification.

(1) *(Conditional) preference over attribute values.*
For example, *"I prefer British Airways to Air France"* is a statement of unconditional value preference over the domain of the `airline` attribute. In turn, *"I prefer British Airways to Air France in morning flights"* is a statement of conditional value preference over the domain of the `airline` attribute, conditioned by the value of the `departure-time` attribute.

(2) *(Conditional) relative importance between pairs of attributes.*
For example, *"Departure time is more important to me than airline"* is a statement of unconditional relative importance between the attributes `departure-time` and `airline`, while *"Departure time is more important to me than airline if I'm flying business class"* is a conditional relative importance statement; the value of the `class` attribute affects the relative importance between the attributes `departure-time` and `airline`.

Each such statements is taken as defining a partial order over the items, using the *ceteris paribus* semantics [19]. That is, if the user states that he prefers Delta to United, we take it to mean that given two flights with identical attribute values, except for the airline, we prefer that in which the airline is Delta to that in which the airline is United. If the statement is conditioned on the fact that the flight is overnight, the same interpretation holds, except that it applies only to comparison between overnight flights. Similarly, for importance relations, if I state that departure time is more important than airline, then given two flights that are identical, except for their departure time and airline, I prefer the one that provides a more favorable departure time. A set of preference statements corresponds to the union of these partial orders.[2]

2.2 TCP-Nets

The language discussed above provides a relatively rich set of qualitative statements about outcomes with discrete-valued attributes. We wish to compile this language into compact value functions. In [12] it was shown that a set of preference statements from the above families can be organized within a graphical structure called TCP-net, an extension of the CP-nets model [8]. This graphical structure plays an important role in analyzing and compiling this language. Here we introduce TCP-nets in depth sufficient for our purposes only, and refer the interested reader to [12] for a detailed and systematic formalization of this model.

TCP-nets are annotated graphs with three types of edges. The nodes in TCP-nets correspond to the problem variables $\mathcal{X}$ (or to a subset of $\mathcal{X}$ if some variables are not addressed by the user statements.) The first type of (directed) edge captures direct preferential dependencies between the variables, that is, such an edge from X to Y implies that the user has different preferences over Y values given different values of X. The second (directed) edge type captures relative importance relations. The existence of such an edge from X to Y implies that it is more important to satisfy preferences with respect to X rather than preferences with respect to Y (denoted as $X \rhd Y$). The third (undirected) edge type captures conditional importance relations: Such an edge between nodes X and Y exists if different conditions (i.e., certain assignments to some set of variables $\mathcal{S}$) lead to $X \rhd Y$, $Y \rhd X$, or even to absence of relative importance between X and Y at all.

Each node X in a TCP-net is annotated with a *conditional preference table* (CPT). This table associates a preference ordering over $dom(X)$ with every possible value assignment to the parents of X (denoted $\mathcal{U}_X$). In addition, in TCP-nets, each undirected edge is annotated with a *conditional importance table* (CIT). The CIT associated with such an edge $\gamma = (X, Y)$ describes the relative importance of X and Y given the value of the conditioning variables $\mathcal{S}_\gamma$.

Definition 1 [12] A TCP-net N is a tuple $\langle \mathcal{X}, \mathsf{cp}, \mathsf{i}, \mathsf{ci}, \mathsf{cpt}, \mathsf{cit} \rangle$ where:

(1) $\mathcal{X}$ is a set of nodes, corresponding to the problem variables $\{X_1, \ldots, X_n\}$.

(2) cp is a set of directed cp-*arcs* $\{\alpha_1, \ldots, \alpha_k\}$ (where cp stands for *conditional preference*). A cp-arc $\langle \overrightarrow{X_i, X_j} \rangle$ is in N iff the preferences over the values of X_j depend on the actual value of X_i. For each $X \in \mathcal{X}$, let $\mathcal{U}_X = \{X' | \langle \overrightarrow{X', X} \rangle \in \mathsf{cp}\}$.

(3) i is a set of directed i-*arcs* $\{\beta_1, \ldots, \beta_l\}$ (where i stands for *importance*). An i-arc $(\overrightarrow{X_i, X_j})$ is in N iff $X_i \rhd X_j$.

(4) ci is a set of undirected ci-*arcs* $\{\gamma_1, \ldots, \gamma_m\}$ (where ci stands for *conditional importance*). A ci-arc $\gamma = (X_i, X_j)$ is in N iff there are certain conditions under which one variable

2. This begs the question of the consistency of such a set, which we touch upon later. See [12] for more details.

is more important than the other, but the relative importance between X_i and X_j is fully determined by the value of some *selector set* $\mathcal{S}_\gamma \subseteq \mathcal{X} \setminus \{X_i, X_j\}$.

(5) cpt associates a CPT with every node $X \in \mathcal{X}$, where $CPT(X)$ is a mapping from $dom(\mathcal{U}_X)$ (i.e., assignments to X's parent nodes) to strict partial orders over $dom(X)$.

(6) cit associates with every ci-arc $\gamma = (X_i, X_j)$ a (possibly partial) mapping $CIT(\gamma)$ from $dom\,(\mathcal{S}_\gamma)$ to orders over the set $\{X_i, X_j\}$.[3]

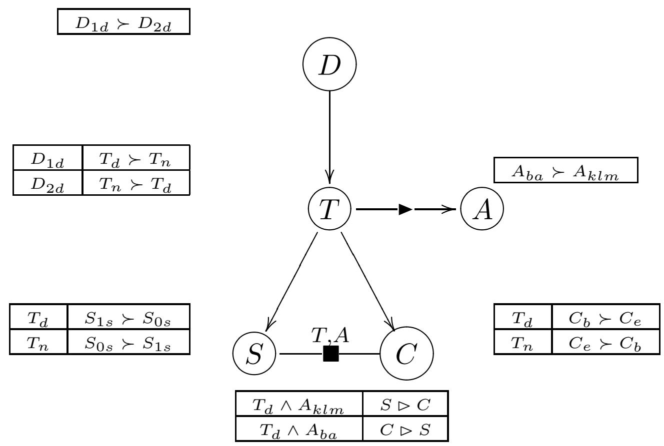


Figure 1: A TCP-net from the flight selection domain.

For those familiar with the original CP-net model, we note that a CP-net is simply a TCP-net in which the sets i and ci (and therefore cit) are empty. Figure 1 illustrates an example TCP-net for selecting a business flight from Israel to a conference in USA, borrowed from [12]. This network consists of five variables, standing for various parameters of the flight[4]:

- D – Departure day. Our busy, married user prefers flights leaving one day (D_{1d}) before the conference to flights leaving two days (D_{2d}) before the conference.
- A – Airline. Our user prefers British Airways (A_{ba}) to KLM (A_{klm}).
- T – Departure time. When leaving two days before the conference, our user prefers the later night flight (T_n) to the earlier day flight (T_d). When leaving just one day before the conference, these preferences are reversed.
- S – Stop-overs. On day flights, our user would like to have a smoking break, so he prefers an indirect flight (S_{1s}) to a direct flight (S_{0s}). On night flights, he sleeps well, and so prefers the shorter, direct flight.

3. That is, the relative importance relation between X_i and X_j may be specified only for certain values of the selector set.
4. Variables in this example are binary, but the semantics of TCP-nets is defined with respect to arbitrary finite domains.

- C – Class. On a night flight, our user prefers the cheaper economy (C_e) seats because he sleeps anyways, but wants to enjoy business class (C_b) service on day flights.

CP-arcs and CPTs in Figure 1 capture these preference statements and the underlying preferential dependencies. Our user's relative importance relations are as follows: There is an i-arc from T to A: getting a more suitable flying time is more important than getting the preferred airline. There is a ci-arc between S and C, where the relative importance of S and C depends on the values of T and A (see the corresponding CIT): (i) On KLM day flights, an intermediate stop in Amsterdam is more important than flying business class. (Our user likes the casino in Amsterdam's airport.) (ii) On British Airways day flights, business class is more important than a stop-over. (Smoking areas in Heathrow are depressing.)

The semantics of a TCP-net N is defined in terms of preference rankings consistent with the constraints imposed by cpt and cit of N, that is, strict partial orders consistent with the partial order induced by N. The local constraints are interpreted *ceteris paribus*. For example, the fact that in the CPT for departure time (T) we have that $T_m \succ T_n$ given $D = D_{1d}$ implies that, given two flights departing one day before the conference that differ *only* in their departure time, the user prefers the one leaving in the morning to the one leaving at night. (Preference between alternative values x_1, x_2 of a variable $X \in \mathcal{X}$ given an assignment $\mathbf{u}$ to the parents $\mathcal{U}_X$ is denoted as $N \models x_1 \succ_{\mathbf{u}} x_2$, or simply as $N \models x_1 \succ x_2$ if the assignment to parent variables $\mathbf{u}$ is clear from the context.) The fact that T is more important than A implies that given two flights that are identical, except for the value of T and A, the user prefers the one in which T is assigned a better value regardless of the value of A. Similar semantics is given to conditional importance relation, taking into account the requirement for the conditioning variables (the selector set). A TCP-net N is consistent iff there is some strict partial order $\succ$ consistent with it. For all $o, o' \in \Omega$, $o \succ o'$ is implied by a TCP-net N (denoted as $N \models o \succ o'$) iff it holds in *all* strict partial orders consistent with N, and this preferential entailment with respect to a consistent TCP-net is transitive. (For the formal semantics in detail, see [12].)

The structure of the TCP-net was shown to be useful for (a) recognizing the (in)consistency of user preference statements, and (b) performing efficient inference. In Section 3 we will show that it can also be exploited in identifying compact value functions consistent with the user's preferences. We reemphasize that this graphical structure is used for analysis purpose (although it can be used for describing preferences, if so desired). Users are not expected to specify the explicit graphical model nor need they be aware of its existence. They simply need to verbalize statements of the two kinds discussed earlier. The system can easily construct the corresponding TCP-net automatically. Nor are the users required to provide statements that specify the CPTs completely (e.g., as in Bayes nets). This property is especially important in practice as users should not be required to express every nuance of their preferences.

2.3 Value functions and GA-decomposition

We wish to map a set of preference statements into a numeric value function. A *value function* $v : \Omega \rightarrow \mathbb{R}$ is a real-valued function defined over the space of all possible assignments on $\mathcal{X}$, that is, over our item space. Value function v is consistent with a (possibly partial) preference ordering $\succeq$ of the user iff $v(o) > v(o')$ for all $o \succ o'$, and $v(o) \geq v(o')$ for all

$o \succeq o'$. As the size of Ω is exponential in the number of problem variables, only compactly representable value functions can be practically useful.

In this work we focus on one such family of compactly representable value functions, namely *generalized additive* (GA) value functions. The notion of GA value functions closely corresponds to the notion of *generalized additive independence* for cardinal utility functions [15, 2], but addresses only the structural assumptions of the latter. A value function v over the variables $\mathcal{X}$ is GA if there exists a cover of $\mathcal{X}$ by some of its subsets $\mathcal{X}_1, \ldots, \mathcal{X}_k$ such that $v(\mathcal{X}) = \sum_{i=1}^{k} v_i(\mathcal{X}_i)$. In what follows, we refer to these variable subsets $\mathcal{X}_1, \ldots, \mathcal{X}_k$ as the *factors* of v. Notice that *any* value function can be seen as GA (for $k = 1$), but working with a GA value function is practically feasible only if its factors are sufficiently compact and the number of essential factors k is small, e.g., both k and the size of each $\mathcal{X}_i$ are bounded by a constant. In general, this brings us to the specify the notion of GA-decomposition of a preference ordering.

Definition 2 Given a (possibly partial) preference ordering $\succeq$ over $\Omega = dom(\mathcal{X})$, and a cover $\mathcal{X}_1, \ldots, \mathcal{X}_k$ of $\mathcal{X}$, we say that $\succeq$ *is GA-decomposable over* $\mathcal{X}_1, \ldots, \mathcal{X}_k$ if there exists a real-valued function

$$v(\mathcal{X}) = \sum_{i=1}^{k} \phi_i(\mathcal{X}_i), \tag{1}$$

consistent with $\succeq$. In particular, we say that *a TCP-net N over $\mathcal{X}$ is GA-decomposable over* $\mathcal{X}_1, \ldots, \mathcal{X}_k$ if there exists a real-valued function v as in Eq. 1 such that, for all $\emptyset, o' \in \Omega$, if $N \models o \succ o'$, then $v(o) > v(o')$.

3. From Qualitative Preferences to GA Value Functions

Value functions provide a mathematically general and efficient way of representing and reasoning with preference information. Given a value function, we can quickly sort a given database of items or determine the top-k of its items. However, obtaining a value function directly from the user is significantly more involved than obtaining a set of simple preference statements. Therefore, we propose to

1. use a TCP-net to initially organize the qualitative preference statements obtained from the user,

2. compile this information to a value function that *maintains the qualitative structure and independence assumptions implicit in this TCP-net*,

3. use the obtained value function as the model of user's ordinal preference,

4. as new information comes from the user, refine this value function, while still maintaining independence assumptions implied by the original TCP-net, if possible.

Note that the basic idea of such value-function compilation framework is not new, and it was considered in the literature before with respect to some other forms of qualitative preference information [3, 17, 27]. The precise relation to previous works is established later in Section 5.

The main theoretical question we face is: *Given a TCP-net N as input, can we efficiently generate a value function v that is consistent with N?* Of course, it is quite trivial to see that, for any consistent TCP-net, there exists at least one value function consistent with it. Indeed, this is true for any partial order. However, what we would really like to know is whether we can find a *structured* value function that, in some sense, is *as compact as the original TCP-net*. Specifically, we would like to know whether there exists a GA value function defined over small factors "implied" by the structure of the network.

To answer this question, we consider three progressively more complicated classes of TCP-nets, and show how the factors of GA value functions representing these TCP-nets relate to their graphical structure. We prove this relation by showing that for *every* TCP-net in each of these network classes there exists a GA value function over a particular set of factors. Likewise, our representation results provide us with a concrete computational mechanism for generating such GA value functions, tractable for a wide class of TCP-nets.

3.1 GA-decomposition of CP-nets

First, we consider CP-nets [8], that is, TCP-nets with only cp-arcs, and start with some notation. Given a CP-net $N = \langle \mathcal{X}, \mathsf{cp}, \emptyset, \emptyset, \mathsf{cpt}, \emptyset \rangle$ over variables $\mathcal{X}$, let $\mathcal{U}_X$ and $\mathcal{Y}_X$ be the sets of parents and (immediate) children of X in N, respectively. Let $\mathcal{F}_X = \{X\} \cup \mathcal{U}_X$ denote the *CP-family* of X, and $\mathcal{EF}_X = \mathcal{F}_X \cup \bigcup_{Y \in \mathcal{Y}_X} \mathcal{F}_Y$ denote the *extended CP-family* of X. That is, the set $\mathcal{EF}_X$ contains X, its parents, its children, and the parents of its children. This set is also known as the Markov Blanket of X, and it is illustrated in Figure 2. Finally, given two assignments $\mathbf{z}, \mathbf{z}'$ to a pair of subsets $\mathcal{Z}, \mathcal{Z}' \subseteq \mathcal{X}$, respectively, we say that $\mathbf{z}$ and $\mathbf{z}'$ are *compatible*, denoted by $\mathbf{z} \bowtie \mathbf{z}'$, if $\mathbf{z}$ and $\mathbf{z}'$ provide the same value to all the shared variables $\mathcal{Z} \cap \mathcal{Z}'$.

The following CP-condition, originally introduced in [7] for defining real-valued value functions representable as UCP-nets, plays a central role in our discussion. To understand this condition better, one has to understand that our goal now is to show that for a CP-net, we can define a consistent value function that is a sum of smaller functions, each of which depends on a single family within the network. That is, *we would like a GA value function whose factors are the families of the CP-net.*

Consider the family $\mathcal{F}_X$ of some variable X. Let ϕ^X be the component of a GA value function as above that corresponds to this family. Consider some assignment $\mathbf{u}$ to the parents of X. $\phi^X(\cdot, \mathbf{u})$ is then a function of X. We would expect it to provide higher values to assignment to X that the user prefers given $\mathbf{u}$. However, this condition is not sufficient to ensure that the value function be consistent with the stated preferences. To see this, suppose that $N \models x_1 \succ_{\mathbf{u}} x_2$. It is possible that $\phi^X(x_1, \mathbf{u}) > \phi^X(x_2, \mathbf{u})$, yet for some assignment $\mathbf{w}$ to the rest of the variables, $v(x_1, \mathbf{u}, \mathbf{w}) < v(x_2, \mathbf{u}, \mathbf{w})$. This can be the case because in the context of $\mathbf{w}$, the value of some child of X is much higher given x_1 than given x_2. Thus, while x_2 does not seem to contribute too much directly, it makes the contribution of some other assignment much higher. For an illustration, consider a CP-net N over two variable X and Y, with (parent-less) X being the parent of Y. Let $N \models x_1 \succ x_2$ for some $x_1, x_2 \in dom(X)$. From the semantics of TCP-nets, we then have $N \models x_1 y \succ x_2 y$ for any $y \in dom(Y)$. Now, let $v(X, Y) = \phi^X(X) + \phi^Y(Y, X)$ be a GA value function with the structure as required such that $\phi^X(x_1) = 1$, $\phi^X(x_2) = 0$, $\phi^Y(y, x_1) = 0$, and $\phi^Y(y, x_2) = 2$.

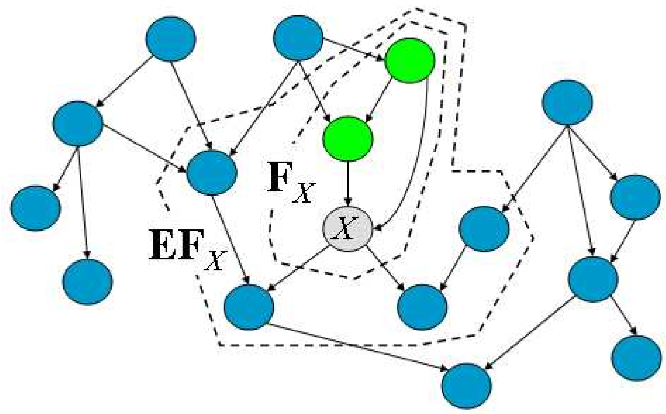


Figure 2: CP-family and extended CP-family of X.

While the condition $\phi^X(x_1) > \phi^X(x_2)$ is satisfied, we have $\phi^Y(y, x_1) < \phi^Y(y, x_2)$, violating the consistency of v with N.

The CP-condition as specified in Definition 3 below rules out this possibility, making the sum of direct and indirect contributions of a less favored value smaller than that of a more favored value. Note that the direct and indirect contributions of an assignment to X depend on (and only on) the extended family of X: its parents determine its direct contribution, and its children's parents determine its indirect contribution.

Definition 3 Given a CP-net N, and a set of non-negative real-valued functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ over $\mathcal{F}_{X_1}, \cdots, \mathcal{F}_{X_n}$, respectively, we say that Φ satisfies the *CP-conditions* of N if and only if for each variable $X \in \mathcal{X}$, each $\mathbf{u} \in \mathit{dom}(\mathcal{U}_X)$, and each $x_1, x_2 \in \mathit{dom}(X)$, if $N \models x_1 \succ_{\mathbf{u}} x_2$, then for each $\mathbf{v} \in \mathit{dom}(\mathcal{EF}_X \setminus \{X\})$ compatible with $\mathbf{u}$ we have:

$$\phi^X(x_1, \mathbf{u}) + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_1) > \phi^X(x_2, \mathbf{u}) + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_2) \tag{2}$$

where $\mathbf{v}_i$ is the value provided by $\mathbf{v}$ to $(\mathcal{F}_{Y_i} \setminus \{X\})$.

Developed by Boutilier *et al.* [7], Lemma 1 below exploits the CP-conditions of N to provide a necessary and sufficient condition for a GA value function with factors $\mathcal{F}_{X_1}, \cdots, \mathcal{F}_{X_n}$ to be consistent with N.

Lemma 1 ([7]) *Given a CP-net N, and a function*

$$v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}_{X_i}), \tag{3}$$

we have v consistent with N iff $\{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy the CP-conditions of N.

In fact, Lemma 1 provides us with even stronger knowledge on GA-decomposability of N. First, reasonably assuming that no preferential dependency between X and $\mathcal{U}_X$ is redundant, one could not expect a more compact GA-decomposition of N. Second, CP-conditions actually provide us with a concrete procedure for generating such a value function v:

(1) Given a CP-net N, construct a system of linear inequalities L, the variables of which stand for the entries of the factors $\phi^{X_1}, \ldots, \phi^{X_n}$ and inequalities correspond to all the required instances of Eq. 2. Let $\mathscr{H}_L$ be the polytope defined by L.

(2) If L is satisfiable (that is, $\mathscr{H}_L$ is not empty), pick *any* solution for L. The latter selection can be done, for instance, by solving a linear program defined by L and an arbitrary linear objective function bounded on $\mathscr{H}_L$ [4], or by sampling a point from $\mathscr{H}_L$).

Step (2) is correct because *any solution of L constitutes a value function v of form (3), consistent with N.* The complexity of L is only locally exponential: the number of variables and inequalities in L is $O(nd^{\lambda})$ and $O(nd^{2\mu})$, respectively, where $d = \max_{X \in \mathcal{X}} \{|dom(X)|\}$, $\lambda = \max_{X \in \mathcal{X}} \{|\mathcal{F}_X|\}$, and $\mu = \max_{X \in \mathcal{X}} \{|\mathcal{EF}_X|\}$. Finally, since linear programming is in P, we obtain the following corollary of practical interest.

Corollary 2 *If a CP-net N is GA-decomposable over its CP-families, and we have $\max_{X \in \mathcal{X}} \{|\mathcal{EF}_X|\} = k$ for some constant k, then a value function $v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}_{X_i})$ consistent with N can be constructed in time polynomial in the size of N.*

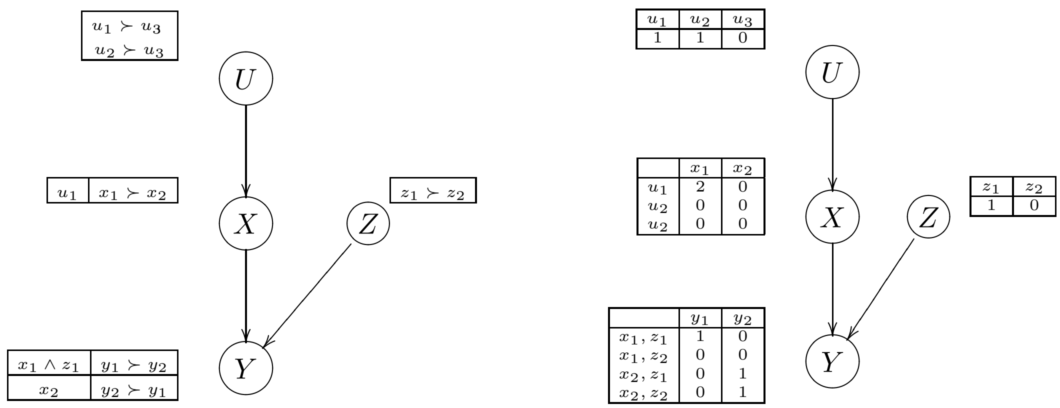
Corollary 2 presents a wide class of efficiently GA-decomposable CP-nets. However, notice that nothing so far prevents $\mathscr{H}_L$ from being empty, since Lemma 1 provides no guarantees for the actual GA-decomposability. It is possible that, for some CP-nets, value functions of form (3) simply do not exist. As we would like to assume that user's statements provide us with sufficient information about value independence, such incompleteness would clearly be problematic. Fortunately, Theorem 3 below shows that polytopes $\mathscr{H}_L$ for *acyclic CP-nets* are always non-empty.

Theorem 3 *Every acyclic CP-net is GA-decomposable over its CP-families.*

To relate Theorem 3 to the classical results in multi-attribute decision theory, consider a CP-net without any edges. According to Theorem 3, such a CP-net induces an additive value function, that is, for $1 \leq i \leq n$, the factor $\mathcal{X}_i$ consists of exactly one variable X_i. Indeed, variables in such a CP-net are mutually preferentially independent, a necessary and sufficient condition for additive decomposition (see Theorem 3.6 in [21]). Thus, a representation theorem for standard additive value functions over discrete variables[5] is a special case of our Theorem 3. As far as we know, results on conditional structures and generalized additive decomposability exist for cardinal *utility* functions [2], but require complex conditions which do not seem to relate in any simple manner to the above result.

Example 1 To illustrate the above procedure for value-function generation, consider the CP-net N depicted in Figure 3(a). This CP-net is defined over three binary-valued variables

5. Here we note that the classical results on additive decomposition cover continuous variables as well, whereas we deal with discrete variables only.



(a) (b)

Figure 3: (a) The CP-net for Example 1; (b) The value function generated for that CP-net–each table associated with a variable captures the sub-function over the corresponding factor.

X, Y, and Z, and one ternary variable U. The linear system L encoding the CP-conditions of N is as follows.

First, for the parent-less variable U, we have $N \models u_1 \succ u_3$ and $N \models u_2 \succ u_3$, $\mathcal{Y}_U = \{X\}$, and $\mathcal{F}_X \setminus \{U\} = \{X\}$. Given that, the CP-conditions "for U" are

$$
\begin{aligned}
\phi^U(u_1) + \phi^X(x_1, u_1) &> \phi^U(u_3) + \phi^X(x_1, u_3) \\
\phi^U(u_1) + \phi^X(x_2, u_1) &> \phi^U(u_3) + \phi^X(x_2, u_3) \\
\phi^U(u_2) + \phi^X(x_1, u_2) &> \phi^U(u_3) + \phi^X(x_1, u_2) \\
\phi^U(u_2) + \phi^X(x_2, u_2) &> \phi^U(u_3) + \phi^X(x_2, u_2)
\end{aligned} \tag{4}
$$

Similarly, for the parent-less variable Z, we have $N \models z_1 \succ z_2$, children $\mathcal{Y}_Z = \{Y\}$, but not the set $\mathcal{F}_Y \setminus \{Z\} = \{Y, X\}$ is not a singleton. The CP-conditions "for Z" are thus

$$
\begin{aligned}
\phi^Z(z_1) + \phi^Y(y_1, x_1, z_1) &> \phi^Z(z_2) + \phi^Y(y_1, x_1, z_2) \\
\phi^Z(z_1) + \phi^Y(y_1, x_2, z_1) &> \phi^Z(z_2) + \phi^Y(y_1, x_2, z_2) \\
\phi^Z(z_1) + \phi^Y(y_2, x_1, z_1) &> \phi^Z(z_2) + \phi^Y(y_2, x_1, z_2) \\
\phi^Z(z_1) + \phi^Y(y_2, x_2, z_1) &> \phi^Z(z_2) + \phi^Y(y_2, x_2, z_2)
\end{aligned} \tag{5}
$$

Next, consider the variable X with $\mathcal{U}_X = \{U\}$. Considering $dom(U)$ and the CPT of X, we have only $N \models x_1 \succ_{u_1} x_2$. Thus, given $\mathcal{Y}_X = \{Y\}$, and $\mathcal{F}_Y \setminus \{X\} = \{Y, Z\}$, we have

$$
\begin{aligned}
\phi^X(x_1, u_1) + \phi^Y(y_1, x_1, z_1) &> \phi^X(x_2, u_1) + \phi^Y(y_1, x_2, z_1) \\
\phi^X(x_1, u_1) + \phi^Y(y_1, x_1, z_2) &> \phi^X(x_2, u_1) + \phi^Y(y_1, x_2, z_2) \\
\phi^X(x_1, u_1) + \phi^Y(y_2, x_1, z_1) &> \phi^X(x_2, u_1) + \phi^Y(y_2, x_2, z_1) \\
\phi^X(x_1, u_1) + \phi^Y(y_2, x_1, z_2) &> \phi^X(x_2, u_1) + \phi^Y(y_2, x_2, z_2)
\end{aligned} \tag{6}
$$

Finally, from the CPT of the child-less variable Y with $\mathcal{U}_Y = \{X, Z\}$, we have $N \models y_1 \succ_{x_1 z_1} y_2$, $N \models y_2 \succ_{x_2 z_1} y_1$, and $N \models y_2 \succ_{x_2 z_1} y_1$. Hence, the CP-conditions "for Y" are

$$\begin{aligned} \phi^Y(y_1, x_1, z_1) &> \phi^Y(y_2, x_1, z_1) \\ \phi^Y(y_2, x_2, z_1) &> \phi^Y(y_1, x_2, z_1) \\ \phi^Y(y_2, x_2, z_2) &> \phi^Y(y_1, x_2, z_2) \end{aligned} \tag{7}$$

Together, Eqs. 4-7 provide us with 15 linear constraints L over 19 variables, and these constraints constitute the CP-conditions of N. At the second step of the value-function generation procedure we should then pick an arbitrary solution to L, and this can be done in numerous ways. For instance, solving (in polynomial time [4]) the quadratic program aiming at resolving L with margin $\epsilon > 0$ while (i) keeping the vector of variables in the positive quadrant, and (ii) minimizing the ℓ^2-norm of that vector, provides us with the solution (= value function) depicted in Figure 3(b). (For readability, and without loss of generality, we have subtracted ϵ from the values of all the variables of L.)

3.2 GA-decomposition of TCP-nets with no ci-arcs

Now, let us consider a wider class of TCP-nets, namely *TCP-nets with no* ci*-arcs*. Here we show that, assuming the TCP-net remains acyclic, the GA-decomposability of this class of networks $N = \langle \mathcal{X}, \mathsf{cp}, \mathsf{i}, \emptyset, \mathsf{cpt}, \emptyset \rangle$ is not affected by the relative importance statements. We do need, however, to change the constraints used to generate this value function. We begin by formalizing a new set of conditions essential for analysis of GA-decomposability of this class of networks.

Definition 4 Given a TCP-net N with no ci-arcs, and a set of non-negative real-valued functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ over $\mathcal{F}_{X_1}, \cdots, \mathcal{F}_{X_n}$, respectively, we say that Φ satisfies the *I-conditions* of N if and only if for each i-arc $(\overrightarrow{X, X'}) \in N$, each $\mathbf{u} \in dom(\mathcal{U}_X)$, and each $x_1, x_2 \in dom(X)$, if $N \models x_1 \succ_{\mathbf{u}} x_2$ then, for each $x'_1, x'_2 \in dom(X')$, each $\mathbf{u}' \in dom(\mathcal{U}_{X'})$ compatible with $\mathbf{u}$, and each $\mathbf{v} \in dom(\mathcal{EF}_X \setminus \{X, X'\})$, $\mathbf{v}' \in dom(\mathcal{EF}_{X'} \setminus \{X, X'\})$ compatible with $\mathbf{u}$ and $\mathbf{u}'$, we have:

$$\begin{aligned} \phi^X(x_1, \mathbf{u}) + \phi^{X'}(x'_1, \mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_1, x'_1) + \sum_{i=1}^{|\mathcal{Y}_{X'} \setminus \mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_1, x_1) \quad > \\ \phi^X(x_2, \mathbf{u}) + \phi^{X'}(x'_2, \mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_2, x'_2) + \sum_{i=1}^{|\mathcal{Y}_{X'} \setminus \mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_2, x_2) \end{aligned} \tag{8}$$

where $\mathbf{v}_i$ and $\mathbf{v}'_i$ are the values provided by $\mathbf{v}$ and $\mathbf{v}'$ to $(\mathcal{F}_{Y_i} \setminus \{X, X'\})$ and $(\mathcal{F}_{Y'_i} \setminus \{X, X'\})$, respectively. Note that x'_1 and x'_2 (similarly, x_1 and x_2) might be redundant parameters for some ϕ^{Y_i} (respectively, $\phi^{Y'_i}$).

This condition may look complicated, but the intuition behind it is simple. The idea is to provide constraints on the value function that ensure that X is more important than X'. Recall that if X is more important than X' then given two assignments that are identical on variables other than X and X', we prefer the one that has a better X value, regardless of its value on X'. This requirement translates into the above I-condition. Previously, the CP-condition required that if x_1 is preferred to x_2 given $\mathbf{u}$ then the sum of contributions of the factors that correspond to the extended family of X be higher for x_1 than for x_2 given any fixed context. This context consisted of the extended family of X. Now, we require that this condition will hold even if we change the value of X'. Thus, we must take into account the direct and indirect effects of this change in the value of X'. These depend only on the extended family of X'. Thus, at each side of the inequality in Eq. 8, we see a sum that corresponds to the extended families of X and X'.

It turns out that the I and CP-conditions together constitute for TCP-nets with no ci-arcs *exactly* what the CP-conditions alone constitute for CP-nets.

Lemma 4 *Given a TCP-net N with no ci-arcs, and a function $v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}_{X_i})$, we have that v is consistent with N if and only if $\{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy both the CP and I-conditions of N.*

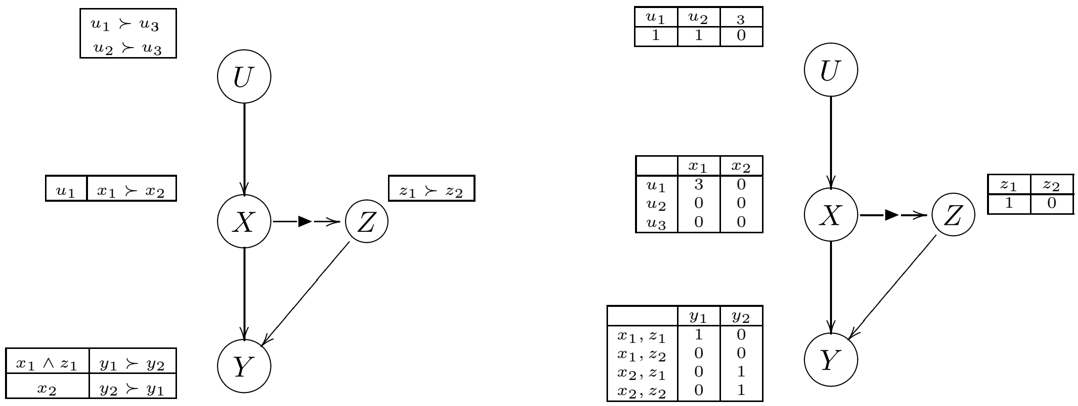
Notice that, similarly to the case of CP-nets, Lemma 4 provides TCP-nets with no ci-arcs with a procedure for generating consistent GA value functions of the form as in Eq. 3. However, an immediate concern should be its usefulness: At first sight, such decomposition does not seem to be very likely, as the functional form in Eq. 3 is based only on preference dependencies, completely ignoring the importance relations induced by the i-arcs. Theorem 5 shows that these concerns are not entirely justified, and that value decomposition of form (3) is complete for *acyclic TCP-nets with no ci-arcs.* (Since all the arcs in such networks are directed, the corresponding notion of acyclicity is straightforward.)

Theorem 5 *Every acyclic TCP-net with no ci-arcs is GA-decomposable over its CP-families.*

Theorem 5 shows that additional unconditional relative importance relation do not affect GA-decomposability of the network (assuming it remains acyclic). Lemma 4 shows that any such GA value function corresponds to a solution of a linear system L, as in the case of CP-nets. Still locally exponential, the complexity of L, however, is affected by i-arcs, since now L consists of both instances of Eq. 2 and Eq. 8. As a result, the number of variables in L is still $O(nd^{\lambda})$, but the number of equations grows to $O((n+l)d^{2\mu})$, where l is the number of i-arcs in N. Notice that the order of description complexity of L remains the same as for CP-nets, thus Corollary 2 can be re-stated for TCP-nets with no ci-arcs, all else being equal.

Example 2 Consider the TCP-net N in Figure 4(a) that extends the CP-net from Example 1 by an i-arc $(\overrightarrow{X,Z})$. The CP-conditions of N are identical to these in Example 1, that is, given by the linear constraints in Eqs. 4-7. The I-conditions of N are (only) due to the i-arc $(\overrightarrow{X,Z})$, and these correspond (only) to the preference of $X = x_1$ to $X = x_2$ given $U = u_1$. Thus, the I-conditions of N are

$$\begin{aligned}
\phi^X(x_1,u_1) + \phi^Z(z_1) + \phi^Y(y_1,x_1,z_1) &> \phi^X(x_2,u_1) + \phi^Z(z_2) + \phi^Y(y_1,x_2,z_2)\\
\phi^X(x_1,u_1) + \phi^Z(z_1) + \phi^Y(y_2,x_1,z_1) &> \phi^X(x_2,u_1) + \phi^Z(z_2) + \phi^Y(y_2,x_2,z_2)\\
\phi^X(x_1,u_1) + \phi^Z(z_2) + \phi^Y(y_1,x_1,z_2) &> \phi^X(x_2,u_1) + \phi^Z(z_1) + \phi^Y(y_1,x_2,z_1)\\
\phi^X(x_1,u_1) + \phi^Z(z_2) + \phi^Y(y_2,x_1,z_2) &> \phi^X(x_2,u_1) + \phi^Z(z_1) + \phi^Y(y_2,x_2,z_1)
\end{aligned} \tag{9}$$



(a) (b)

Figure 4: (a) TCP-net with no ci-arcs for Example 2; (b) The value function generated for that TCP-net.

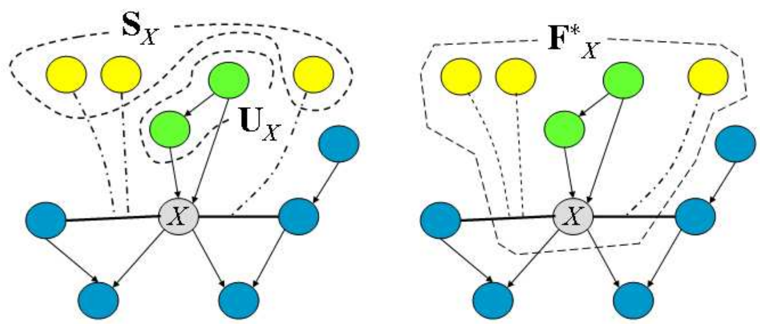


Figure 5: TCP-family of X.

Solving the linear system L obtained from the union of Eqs. 4-7 and Eq. 9 using the same quadratic programming approach as in Example 1 we generate the value function depicted in Figure 4(b). Note that the only difference between the value functions in Figures 3(b) and 4(b) is that the former has $\phi^X(x_1, u_1) = 2$ while the latter has $\phi^X(x_1, u_1) = 3$—the relative importance of preference over X is getting captured.

3.3 GA-decomposition of TCP-nets

We now consider TCP-nets capturing all our types of preference statements, thus consisting of both directed (cp and i) and undirected (ci) arcs. Recall that the selector set of a ci-arc is the set of variables that determine which end of this edge is more important. Let $\mathcal{S}_X$ be the union of selector sets of ci-arcs involving X, that is,

$$\mathcal{S}_X = \bigcup_{\gamma=(X,X')} \mathcal{S}_\gamma.$$

Reversely, let $\mathcal{W}_X$ be the set of end-point variables of ci-arcs where X acts as a selector, that is,

$$\mathcal{W}_X = \left\{ X' \mid X \in \mathcal{S}_{X'} \right\}.$$

Let $\mathcal{Y}^\star_X = \mathcal{Y}_X \cup \mathcal{W}_X$ be the set of all X's "direct dependents". Let $\mathcal{I}_{X|\mathbf{s}}$ be the set of all variables X' that are directly more important than X given $\mathbf{s} \in dom(\mathcal{S}_X)$. That is, for each $X' \in \mathcal{I}_{X|\mathbf{s}}$, either we have an i-arc $(\overrightarrow{X', X}) \in N$, or we have a ci-arc $(X', X) \in N$ and CIT of this arc stipulates that, given $\mathbf{s}$, we have that X' is more important than X. Finally, let $\mathcal{F}^\star_X = \mathcal{F}_X \cup \mathcal{S}_X$ denote the *TCP-family* of X (see Figure 5, where the dashed arcs *schematically* connect between $\mathcal{S}_X$ and X), and $\mathcal{EF}^\star_X = \mathcal{F}^\star_X \bigcup_{Y \in \mathcal{Y}^\star_X} \mathcal{F}^\star_Y$ denote the *extended TCP-family* of X.

TCP-nets with ci-arcs are significantly richer than these without, and a GA-decomposition as in Eq. 3 is not expressive enough to cover this type of networks. However, here we show that there exists a sufficiently expressive (yet often compact) extended counterpart of Eq. 3, namely:

$$v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}^\star_{X_i}). \tag{10}$$

Definition 5 Consider a TCP-net N, and a set of non-negative, real-valued functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ over $\mathcal{F}^\star_{X_1}, \cdots, \mathcal{F}^\star_{X_n}$, respectively. We say that Φ satisfies the *CI-conditions* of N if and only if for each ci-arc $(X, X') \in N$ (and, similarly, each i-arc $(\overrightarrow{X, X'}) \in N$), each $x_1, x_2 \in dom(X)$, and each $\mathbf{u} \in dom(\mathcal{U}_X)$ and $\mathbf{s} \in dom(\mathcal{S}_X)$, if $X \in \mathcal{I}_{X'|\mathbf{s}}$ and $N \models x_1 \succ_{\mathbf{u}} x_2$, then, for each $x'_1, x'_2 \in dom(X')$, and each set of (all pairwise compatible) $\mathbf{u}' \in dom(\mathcal{U}_{X'})$, $\mathbf{s}' \in dom(\mathcal{S}_{X'})$, $\mathbf{v} \in dom(\mathcal{EF}^\star_X \setminus \{X, X'\})$, $\mathbf{v}' \in dom(\mathcal{EF}^\star_{X'} \setminus \{X, X'\})$, we have:

$$\begin{aligned} &\phi^X(x_1, \mathbf{u}, \mathbf{s}) + \phi^{X'}(x'_1, \mathbf{u}', \mathbf{s}') + \sum_{i=1}^{|\mathcal{Y}^\star_X|} \phi^{Y_i}(\mathbf{v}_i, x_1, x'_1) + \sum_{i=1}^{|\mathcal{Y}^\star_{X'} \setminus \mathcal{Y}^\star_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_1, x_1) \quad > \\ &\quad \phi^X(x_2, \mathbf{u}, \mathbf{s}) + \phi^{X'}(x'_2, \mathbf{u}', \mathbf{s}') + \sum_{i=1}^{|\mathcal{Y}^\star_X|} \phi^{Y_i}(\mathbf{v}_i, x_2, x'_2) + \sum_{i=1}^{|\mathcal{Y}^\star_{X'} \setminus \mathcal{Y}^\star_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_2, x_2) \end{aligned} \tag{11}$$

where $\mathbf{v}_i$ and $\mathbf{v}'_i$ are the value provided by $\mathbf{v}$ and $\mathbf{v}'$ to $(\mathcal{F}^\star_{Y_i} \setminus \{X, X'\})$ and $(\mathcal{F}^\star_{Y'_i} \setminus \{X, X'\})$, respectively. As in Definition 4, x'_1 and x'_2 (similarly, x_1 and x_2) might be redundant in some ϕ^{Y_i} (respectively, $\phi^{Y'_i}$).

The form of the CI-condition is identical to that of the I-condition. The difference is that it is more constrained, requiring a particular assignment to the appropriate selector sets.

For GA-decomposition as in Eq. 10, the CP-conditions that work for CP-nets and TCP-nets with no ci-arcs will not work anymore. To understand this, recall that the original CP-conditions ensured that the total contribution of a more preferred value for X will be larger than the total contribution of a less preferred value. This total contribution included the effect of an assignment to X on its children. Now, the total contribution of X depends on additional elements. First, X might participate in a ci-arc, in which case its value depends on the selector set for this edge. Moreover, X might influence the relative importance of other variables in whose selector sets it belongs. Here we provide a modified set of CP-conditions compatible with value decomposition as in Eq. 10. In fact, the modification brought by Definition 6 is schematically simple: To fit the functional form 10, the CP-conditions as in Definition 3 should be simply reformulated from CP- to TCP-families.

Definition 6 Consider a TCP-net N, and a set of non-negative real-valued functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ over $\mathcal{F}^\star_{X_1}, \cdots, \mathcal{F}^\star_{X_n}$, respectively. We say that Φ satisfies the *CP-conditions* of N if and only if for each $X \in \mathcal{X}$, each $\mathbf{u} \in dom(\mathcal{U}_X)$ and $\mathbf{s} \in dom(\mathcal{S}_X)$, and each $x_1, x_2 \in dom(X)$, if $N \models x_1 \succ_{\mathbf{u}} x_2$ then, for each $\mathbf{v} \in dom(\mathcal{EF}^\star_X \setminus \{X\})$ compatible with $\mathbf{u}$ and $\mathbf{s}$ we have:

$$\phi^X(x_1, \mathbf{u}, \mathbf{s}) + \sum_{i=1}^{|\mathcal{Y}^\star_X|} \phi^{Y_i}(\mathbf{v}_i, x_1) > \phi^X(x_2, \mathbf{u}, \mathbf{s}) + \sum_{i=1}^{|\mathcal{Y}^\star_X|} \phi^{Y_i}(\mathbf{v}_i, x_2) \tag{12}$$

where $\mathbf{v}_i$ is the value provided by $\mathbf{v}$ to $\mathcal{F}^\star_{Y_i} \setminus \{X\}$.

Lemma 6 below shows that the (modified) CP-conditions and CI-conditions are necessary and sufficient for GA-decomposability of general TCP-nets along the functional form as in Eq. 10.

Lemma 6 *Given a TCP-net N, and a function $v(\mathcal{X}) = \sum_{i=1}^n \phi^{X_i}(\mathcal{F}^\star_{X_i})$, we have v consistent with N if and only if $\{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy CP- and CI-conditions of N.*

Again, Lemma 6 provides us with a mechanism for generating GA value functions consistent with TCP-nets, similar to the procedures provided by Lemmas 1 and 4. Of course, the complexity of the corresponding linear system L is not as before: The number of variables and inequalities in L is now $O(nd^{\lambda'})$ and $O(nd^{2\mu'})$, where $\lambda' = \max_{X \in \mathcal{X}} \{|\mathcal{F}^\star_X|\}$, and $\mu' = \max_{X \in \mathcal{X}} \{|\mathcal{EF}^\star_X|\}$. Clearly, adding ci-arcs reduce the general compactness of GA-decomposition, but factoring on TCP-families instead of CP-families seems unavoidable.

Corollary 7 *If a TCP-net N is GA-decomposable over its TCP-families, and we have $\max_{X \in \mathcal{X}} \{|\mathcal{EF}^\star_X|\} = k$ for some constant k, then a corresponding value function $v(\mathcal{X}) = \sum_{i=1}^n \phi^{X_i}(\mathcal{F}^\star_{X_i})$ can be constructed in time polynomial in the size of N.*

Considering the completeness of GA-decomposition for general TCP-nets, it is unlikely that every consistent TCP-net is GA-decomposable along the functional form as in Eq. 10. Yet, in Theorem 8 below we show that such decomposability is complete for *acyclic TCP-nets*. Since TCP-nets may contain both directed and annotated undirected arcs, the corresponding notion of acyclicity is non-standard.

Definition 7 ([12]) The *dependency graph* $N^\star$ of TCP-net N contains all the nodes and edges of N. Additionally, for every ci-arc (X_i, X_j) in N and every $X_k \in \mathcal{S}_{(X_i,X_j)}$, $N^\star$ contains a pair of directed edges (X_k, X_i) and (X_k, X_j), if these edges are not already in N.

Figure 6 illustrates the notion of the dependency graph on the "Flight to USA" TCP-net example from [12].

$T_m \wedge A_{klm}$	$S \rhd C$
$T_m \wedge A_{ba}$	$C \rhd S$
$T_n \wedge A_{ba}$	$S \rhd C$

(a) (b)

Figure 6: "Flight to USA" TCP-net (a), and its dependency graph (b) (from [12].)

Definition 8 A TCP-net N is *acyclic* if each cycle in the undirected graph induced by the dependency graph $N^\star$, when projected back to $N^\star$, contains a pair of directed arcs in different directions[6].

For instance, the TCP-net N in Figure 6(a) is acyclic—considering the cycle T—C—S in the undirected graph induced by dependency graph $N^\star$, we see that the directed arcs from T to S and from T to C in $N^\star$ are oriented in two different directions with respect to that cycle (and the same property holds for all the cycles in the undirected graph induced by $N^\star$.)

Based on the notion of acyclic TCP-nets, Theorem 8 finalizes our representation theory.

Theorem 8 *Every acyclic TCP-net is GA-decomposable over its TCP-families.*

Example 3 Consider the TCP-net N in Figure 7(a) that extends the CP-net from Example 1 by a ci-arc $\gamma = (X, Z)$ with $\mathcal{S}_\gamma = \{U\}$ and $CIT(\gamma)$ as in Figure 7(a).

First, note that the TCP-families of U, X, and Y are identical to the CP-families of these variables. While it is easy to see for U and Y, for X this is the case because the only selector U of X already belongs to the parents $\mathcal{U}_X$ of X. From that, it is not hard to verify that the extended TCP-families of U, X, and Y are identical to their extended TCP-families, and thus the CP-conditions of N for U, X, and Y are given by Eq. 4, Eq. 6, and Eq. 7, respectively.

The situation with the variable Z is different because for Z we have $\mathcal{F}_Z = \{Z\}$, $\mathcal{S}_Z = \{U\}$, and thus $\mathcal{F}^\star_Z = \{Z, U\}$. Hence, in contrast to Example 1, the function ϕ^Z should

6. In particular, acyclic CP-nets are a sub-class of what is called *conditionally acyclic TCP-nets*, which today is probably the widest known sub-class of consistent TCP-nets defined in terms of structural properties of the networks [12].

now be defined from $dom(Z) \times dom(U)$, and the CP-conditions for Z should be formulated according to Definition 6. These CP-conditions are given by Eq. 13 below.

$$
\begin{aligned}
&\forall u \in \{u_1, u_2, u_3\} : \\
&\qquad \phi^Z(z_1, u) + \phi^Y(y_1, x_1, z_1) > \phi^Z(z_2) + \phi^Y(y_1, x_1, z_2) \\
&\qquad \phi^Z(z_1, u) + \phi^Y(y_1, x_2, z_1) > \phi^Z(z_2) + \phi^Y(y_1, x_2, z_2) \\
&\qquad \phi^Z(z_1, u) + \phi^Y(y_2, x_1, z_1) > \phi^Z(z_2) + \phi^Y(y_2, x_1, z_2) \\
&\qquad \phi^Z(z_1, u) + \phi^Y(y_2, x_2, z_1) > \phi^Z(z_2) + \phi^Y(y_2, x_2, z_2)
\end{aligned}
\tag{13}
$$

Next, the CI-conditions of N are given by Eqs. 14 and 15 below. For the domain of the selector set $\mathcal{S}_\gamma = \{U\}$ we have

(1) $\mathcal{I}_{X|u_1} = \emptyset$, $\mathcal{I}_{Z|u_1} = \{X\}$ (X being more important than Z given u_1),

(2) $\mathcal{I}_{X|u_2} = \emptyset$, $\mathcal{I}_{Z|u_2} = \emptyset$ (given u_2, the ci-arc γ simply vanishes), and

(3) $\mathcal{I}_{X|u_3} = \{Z\}$, $\mathcal{I}_{Z|u_3} = \emptyset$ (Z being more important than X given u_3).

The CI-constraints corresponding to the case (1) are

$$
\begin{aligned}
\phi^X(x_1, u_1) + \phi^Z(z_1, u_1) + \phi^Y(y_1, x_1, z_1) &> \phi^X(x_2, u_1) + \phi^Z(z_2, u_1) + \phi^Y(y_1, x_2, z_2) \\
\phi^X(x_1, u_1) + \phi^Z(z_1, u_1) + \phi^Y(y_2, x_1, z_1) &> \phi^X(x_2, u_1) + \phi^Z(z_2, u_1) + \phi^Y(y_2, x_2, z_2) \\
\phi^X(x_1, u_1) + \phi^Z(z_2, u_1) + \phi^Y(y_1, x_1, z_2) &> \phi^X(x_2, u_1) + \phi^Z(z_1, u_1) + \phi^Y(y_1, x_2, z_1) \\
\phi^X(x_1, u_1) + \phi^Z(z_2, u_1) + \phi^Y(y_2, x_1, z_2) &> \phi^X(x_2, u_1) + \phi^Z(z_1, u_1) + \phi^Y(y_2, x_2, z_1)
\end{aligned}
\tag{14}
$$

Note that Eq. 14 is similar to Eq. 9 with $\phi^Z(z_1)$ and $\phi^Z(z_2)$ being respectively replaced with $\phi^Z(z_1, u_1)$ and $\phi^Z(z_2, u_1)$, all else being identical. This similarity is not incidental as, given u_1, the ci-arc (X, Z) is equivalent to the i-arc $(\overrightarrow{X, Z})$ from Example 2.

The CI-constraints corresponding to the case (3) are

$$
\begin{aligned}
\phi^Z(z_1, u_3) + \phi^X(x_1, u_3) + \phi^Y(y_1, x_1, z_1) &> \phi^Z(z_2, u_3) + \phi^X(x_2, u_3) + \phi^Y(y_1, x_2, z_2) \\
\phi^Z(z_1, u_3) + \phi^X(x_1, u_3) + \phi^Y(y_2, x_1, z_1) &> \phi^Z(z_2, u_3) + \phi^X(x_2, u_3) + \phi^Y(y_2, x_2, z_2) \\
\phi^Z(z_1, u_3) + \phi^X(x_2, u_3) + \phi^Y(y_1, x_2, z_1) &> \phi^Z(z_2, u_3) + \phi^X(x_1, u_3) + \phi^Y(y_1, x_1, z_2) \\
\phi^Z(z_1, u_3) + \phi^X(x_2, u_3) + \phi^Y(y_2, x_2, z_1) &> \phi^Z(z_2, u_3) + \phi^X(x_1, u_3) + \phi^Y(y_2, x_1, z_2)
\end{aligned}
\tag{15}
$$

Eq. 15 is structurally identical to Eq. 15, with the roles of X of Z being reversed.

Solving the linear system L obtained from the union of Eq. 4, Eqs. 6-7, and Eqs. 13-15 sing the same quadratic programming approach as in Examples 1-2 we generate the value function depicted in Figure 7(b).

4. Refinement by Item-level Rankings

The goal of most preference elicitation systems is to help the user recognize the most preferred item among the set of candidate items. The *ceteris paribus* semantics we used to model natural preference statements is intuitive, but also weak. This implies that, typically,

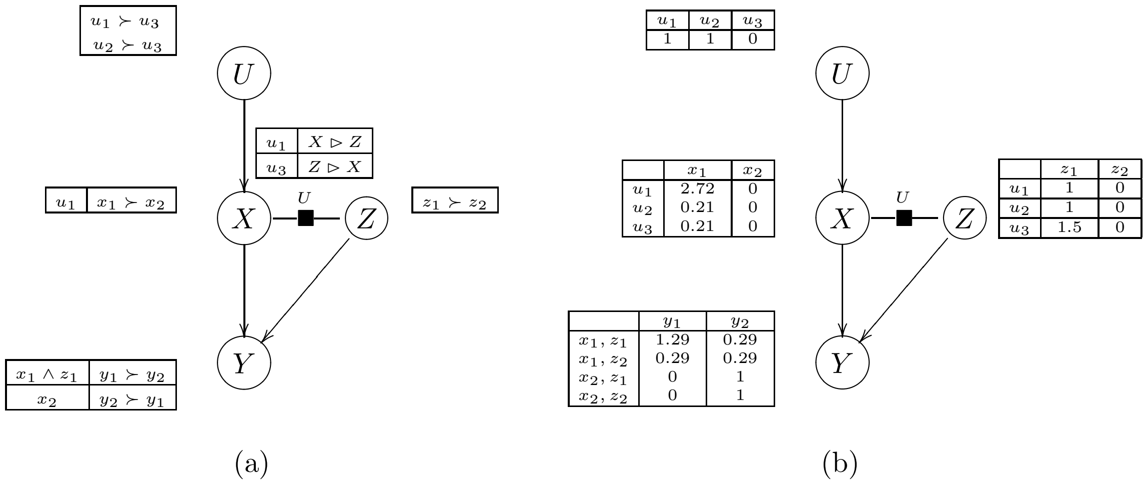


Figure 7: (a) Acyclic TCP-net with a ci-arc for Example 2; (b) The value function generated for that TCP-net.

generalizing preference statements will only specify a partial order over the items. Phrased differently, the set of value functions consistent with a set of statements will correspond to many different orderings. Various systems attempt to further refine the user's preference information at this point. Here, we concentrate on the case the user is asked to rank a small set of items. For instance, in [11], the user is presented with the 10 top items according to one value function, and asked to point to the one most preferred amongst them. This immediately yields 9 pairwise orderings between items. The question that we ask in this section is whether we can integrate such explicit item rankings with the generalizing preference statements, while maintaining the compact structure derived from these generalizing preference statements.

Our results are somewhat surprising. Obviously, if the user's ranking contradict the generalizing statements (e.g., she ranks a red car above an identical blue car, despite stating that blue is her preferred color for cars), no value function can model both. But what happens when the user's rankings are consistent with her generalizing preference statements? That is, there exists an ordering that satisfies both the user's generalizing preference statements, as well as her explicit relative ranking of some items. Clearly, such information can be modeled by some value function. What is perhaps surprising, is that it is not always possible to model this information using the type of compact value functions that can model the generalizing information alone. The proof of this result is unconstructive, and what we find puzzling about it, is that it is not obvious to see how such rankings violate the structural assumptions of the underlying TCP-net.

For reasons that will be clear shortly, let us begin by considering preference expressions from the *simplest* class for which we provide a polynomial time GA-decomposition, that is, preference expressions representable as acyclic CP-nets with extended CP-families of the variables all being of size bounded by a constant. According to Corollary 2, given such an expression N we can efficiently generate a value function v as in Eq. 3 consistent with N.

In addition, Theorem 3 shows that this value-function compilation is not only efficient and sound, but also complete, that is, our ability to generate v is guaranteed.

Now, suppose that the user provides the system with a set of m item-level rankings $R = \{o^i_{i_1} \succ o^i_{i_2}\}_{i=1}^m$, $o_{i_j} \in \Omega$. The task of the system is now to incorporate R in the process of generating a value-function.

Definition 9 Given a set of finite-domain variables $\mathcal{X}$ and $k, m \in \mathbb{N}^+$, $\Phi^{(k)}$ is the language of CP-nets with $\max_{X \in \mathcal{X}} \{|\mathcal{EF}_X|\} \leq k$, and $\Phi^{(k,m)}$ is the language of expressions $\langle N, R \rangle$ consisting of a CP-net $N \in \Phi^{(k)}$ and a set of item-level rankings $R = \{o^i_{i_1} \succ o^i_{i_2}\}_{i=1}^{m'}$, $m' \leq m$, consistent with N. (In particular, we have $\Phi^{(k,0)} = \Phi^{(k)}$.) We use $\Phi^{(k)}_a$ and $\Phi^{(k,m)}_a$ to denote the restrictions of $\Phi^{(k)}$ and $\Phi^{(k,m)}$ to acyclic CP-nets.

For a CP-net N, let L_N be the system of linear constraints corresponding to the CP-conditions of N (Definition 3.) Considering the GA-decomposiiton of $\langle N, R \rangle \in \Phi^{(k,m)}$ over the CP-families of $\mathcal{X}$ (Eq. 3), as we have done so far, observe that each pairwise ranking $o_1 \succ o_2$ in R can be encoded as a linear constraint

$$\sum_{X \in \mathcal{X}} \phi^X(x_1, \mathbf{u}_1) > \sum_{X \in \mathcal{X}} \phi^X(x_2, \mathbf{u}_2), \tag{16}$$

where $x_1, \mathbf{u}_1$ (respectively $x_2, \mathbf{u}_2$) are the values provided by o_1 (respectively o_2) to the variable X and its parents $\mathcal{U}_X$, respectively. Let L_R denote the set of all m such constraints corresponding to R. Let $\mathscr{H}_{L_{N,R}}$ be the polytope defined by the $L_N \cup L_R$. It is easy to verify that any point in $\mathscr{H}_{L_{N,R}}$ provides us with a value function as in Eq. 3 consistent with both N and R. Likewise, for any fixed $k \in \mathbb{N}^+$ and any $N \in \Phi^{(k)}$, we can construct such a value function efficiently. Hence, it appears that the value-function compilation scheme as in Section 3.1 preserves both its soundness and efficiency when extended from $\Phi^{(k)}$ to $\Phi^{(k,m)}$.

Corollary 9 *Given a preference expression $\langle N, R \rangle \in \Phi^{(k,m)}$, if $\langle N, R \rangle$ is GA-decomposable over the CP-families of N, then a value function $v(\mathcal{X}) = \sum_{i=1}^n \phi^{X_i}(\mathcal{F}_{X_i})$ consistent with N and R can be constructed in time polynomial in k and m.*

Having the positive result provided by Corollary 9, what is left to be studied is the completeness of GA-decomposition of $\langle N, R \rangle \in \Phi^{(k,m)}$ over the CP-families of N, and in particular, the completeness of this procedure for acyclic CP-nets. For $\langle N, \emptyset \rangle \in \Phi^{(k)}_a$, completeness is guaranteed by Theorem 3. At first view, there seems to be no reason why it should not hold for $\langle N, R \rangle \in \Phi^{(k,m)}_a$, too. After all, (i) R *is* consistent with N, and thus a value function consistent with both N and R does exists, and (ii) the assumptions that the *ceteris paribus* interpretation of preference statements underlying CP-nets makes about the information these statements communicate are arguably minimal [18]. However, Theorem 10 shows that, in general, extending the preference specification language from $\Phi^{(k)}$ to $\Phi^{(k,m)}$ does not preserve the completeness of GA-decomposition as in Eq. 3, unless we have $P = NP$. Even more surprisingly, it shows that the completeness is not guaranteed even if R contains only a single ranking between a pair of items.

Theorem 10 *Unless* $\mathbf{P} = \mathbf{NP}$*, for* $k \geq 22$*,* $m \geq 1$*, there exists a preference expression* $\langle N, R\rangle \in \Phi_a^{(k,m)}$ *such that* $\mathscr{H}_{L_{N,R}}$ *is empty.*

Proof: The proof combines the results provided by Lemma 1 and Corollary 2, and the results on complexity of dominance testing in CP-nets established in [8]. Given a CP-net N and a pair of complete assignments o, o' on the variables of N, the problem of testing dominance of o over o' is this of deciding $N \models o \succ o'$. By a reduction from the classical 3-SAT problem, Theorem 15 in [8] shows that dominance testing in acyclic CP-nets is NP-hard. For the detailed reduction, see [8]. Two points about the reduction are important for us here:

1. The 3-SAT problem remains NP-hard for the subclass of 3-SAT formulae in which no propositional variable appears (in both its positive and negated forms) in more than three clauses (e.g., see [16], p. 259).

2. The reduction of satisfiability of this class of 3-SAT formulae to the problem of dominance testing in acyclic CP-nets is effectively a reduction to the problem of dominance testing in CP-nets $\Phi_a^{(22)}$. Thus, dominance testing in CP-nets $\Phi_a^{(22)}$ is NP-hard as well.

Now, assume to the contrary that the polytope $\mathscr{H}_{L_{N,R}}$ is empty for no $\langle N, R\rangle \in \Phi_a^{(22,1)}$, where $k \geq 22$ and $m \geq 1$, and consider the following procedure for dominance testing in CP-nets $\Phi_a^{(22)}$.

(1) Given a CP-net $N \in \Phi_a^{(22)}$ and a pair of complete assignments o, o' on the variables of N, construct two instances $\langle N, R_1\rangle, \langle N, R_2\rangle \in \Phi_a^{(22,1)}$ with $R_1 = \{o \succ o'\}$ and $R_1 = \{o' \succ o\}$, respectively.

(2) Create two linear programs LP_1 and LP_2 with constraints $L_N \cup L_{R_1}$ and $L_N \cup L_{R_2}$, and arbitrary linear objective function bounded on $\mathscr{H}_{L_{N,R_1}}$ and $\mathscr{H}_{L_{N,R_2}}$, respectively.

(3) Solve LP_1 and LP_2. If LP_1 is feasible and LP_2 is infeasible, return *true*. Otherwise, return *false*.

First, Corollary 9 implies that the time complexity of this procedure is polynomial. Second, our assumption of non-emptiness of $\mathscr{H}_{L_{N,R}}$ for *all* $\langle N, R\rangle \in \Phi_a^{(22,1)}$ implies that the procedure returns *true* if and only if $N \models o \succ o'$. However, unless we have $P = NP$, this contradicts our previous result that dominance testing in CP-nets $\Phi_a^{(22)}$ is NP-hard. Given the correctness of Corollary 9, we arrive into contradiction with our assumption regarding universal non-emptiness of $\mathscr{H}_{L_{N,R}}$ for $\Phi_a^{(22,1)}$, and thus prove our theorem. ∎

We end this section by generalizing the impossibility result of Theorem 10 to a wide range of other languages for preference specification, including some yet to be proposed.

Theorem 11 (Generalized Impossibility) *Given a language* Φ *of preference expressions over a space of alternatives* Ω*, let* Φ' *be an extension of* Φ *to include consistent item level rankings, i.e.,*

$$\Phi' = \{N \cup R \mid N \in \Phi \;\wedge\; R = \{o_1 \succ o_2\}, o_1, o_2 \in \Omega \;\wedge\; N \models o_1 \succ o_2\}\,.$$

Let $\mathscr{V}$ be the space of real-value functions over Ω, and let $\Gamma_{\Phi'} : \Phi' \mapsto \mathscr{V} \cup \{\otimes\}$ be a sound value-function compilation scheme that either maps preference expressions $N' \in \Phi'$ to value functions consistent with them (if $\Gamma_{\Phi'}(N') \in \mathscr{V}$), or fails (if $\Gamma_{\Phi'}(N') = \otimes$.)

Given that, at least one of the following three properties holds.

(1) Dominance testing in Φ is in **P**.

(2) $\Gamma_{\Phi'}$ is not in **P**.

(3) $\Gamma_{\Phi'}$ is incomplete, that is, there exists $N' \in \Phi'$ such that $\Gamma_{\Phi'}(N') = \otimes$.

It is not hard to verify that the basic idea underlying the proof of Theorem 10 extends in a straightforward manner to a proof for Theorem 11. Note that the generality of the claim in Theorem 11 is due to the fact that it poses no syntactical conditions on the preference language Φ, or on the way the functions in $\mathscr{V}$ are physically specified. And since value-function compilation is of a wide theoretical and practical interest, while dominance testing has been shown to be **NP**- or **coNP**-hard for most non-trivial existing qualitative preference specification languages [8, 25], the relevance of Theorem 11 is apparent.

In fact, below we show that Theorem 11 already provides an answer to at least one open problem from the literature. Considering a preference specification language (referred to here as Φ_{cp}) that *strictly extends* the language of TCP-nets, McGeachie and Doyle [27] suggest a sound and complete value-function compilation scheme for this language. While the compilation schemes suggested in [27] was shown by the authors to be worst-case computationally intractable, McGeachie and Doyle left the necessity of such intractability as an open question ([27], p. 174). Now, given that

(i) dominance testing in Φ_{cp} is known to be **PSPACE**-complete [25], and

(ii) $\Phi'_{cp} = \Phi_{cp}$ because pairwise comparisons between completely specified elements of Ω are simply part of the language Φ_{cp},

Theorem 11 implies that no sound and complete compilation scheme for Φ_{cp} can be computationally efficient, unless, of course, **PSPACE** collapses to **P**.

5. Summary, related work and open problems

In this work we have studied representational and computational issues of compiling a set of qualitative statements of ordinal preference into a value-function consistent with these statements. Specifically, we considered partial orders induced by certain sets of qualitative statements of conditional preference and conditional relative importance, namely the sets of statements representable by the TCP-net model [10, 12]. We presented a new representation theory for factored value functions that allow for more useful preferential independence structures than those appearing in classical textbook results in this area [21]. In particular, these representation results show that preference orders induced by a wide class of TCP-nets can be consistently captured by compact generalized additive value functions. Moreover, we show that for many such sets of statements, the corresponding generalized additive value function can be efficiently generated. Next we considered a practically important problem of value-function compilation of a mixed set of generalizing and item-level preference statements. Adding item-level preference statements does not affect neither the complexity nor the soundness of compiling a set of generalizing preference statements. However, we showed that the completeness of such compilation is affected, and provide a general impossibility theorem on this matter.

The idea of representing partial preference information as a constraint over a space of candidate value functions lies in the very foundations of measurement and multi-attribute decision theory [24]. Given that, numerous works in the area of multi-attribute decision making consider the computational and algorithmic issues of generating value functions consistent with given preference information (e.g., see [33, 23, 22, 28].) Targeting the complexity issues, these works a priori restrict the space of candidate value functions to be (in increasing order of generality) linear, quasi-concave, or monotonic. However, it is easy to show that, by definition, such functions cannot capture many intuitive preference statements, such as, for instance, statements of conditional preference and/or importance. On the other hand, some recent work in the field of artificial intelligence considers value-function compilation of these (as well as some other) statements of preference [17, 27]. However, to the best of our knowledge, our work is the first to provide a non-trivial preference specification language for which value functions can be generated efficiently in a sound and complete manner.

Finally, our work raises numerous open theoretical questions, such as:

1. When (if at all) GA-decomposition is complete for cyclic TCP-nets, or even just cyclic CP-nets?

2. What is the most compact form of GA-decomposition that is complete for all consistent TCP-nets?

3. Can we characterize the representation theorem purely in terms of conditional independence, without using the graph structure explicitly, or, alternatively, what are the core properties of the graph that allow for a compact GA decomposition?

4. What are the concrete limits of our impossibility theorem for various preference specification languages? For instance, for what values of m and $k < 22$ can we obtain completeness of GA-decomposition of $\Phi^{(k,m)}$ over the CP-families of the variables?

We believe that addressing these questions will provide a better understanding of the practical expressiveness and limitations of reasoning about ordinal preference information.

References

[1] R. Agrawal and E. L. Wimmers. A framework for expressing and combining preferences. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 297–306, 2000.

[2] F. Bacchus and A. Grove. Graphical models for preference and utility. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pages 3–10, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[3] F. Bacchus and A. Grove. Utility independence in qualitative decision theory. In *Proceedings of the Fifth Conference on Knowledge Representation (KR–96)*, pages 542–552, Cambridge, 1996. Morgan-Kauffman.

[4] D. Bertsekas, A. Nedic, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[5] J. Blythe. Visual exploration and incremental utility elicitation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 526–532, 2002.

[6] C. Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 239–246, 2002.

[7] C. Boutilier, F. Bacchus, and R. I. Brafman. UCP-networks: A directed graphical representation of conditional utilities. In *Proceedings of Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64, 2001.

[8] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning about conditional *ceteris paribus* preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[9] C. Boutilier, R. Brafman, H. Hoos, and D. Poole. Reasoning with conditional *ceteris paribus* preference statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 71–80. Morgan Kaufmann Publishers, 1999.

[10] R. Brafman and C. Domshlak. Introducing variable importance tradeoffs into CP-nets. In *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 69–76, Edmonton, Canada, August 2002.

[11] R. Brafman, C. Domshlak, and T. Kogan. Compact value-function representations for qualitative preferences. In *Proceedings of the Twentieth Annual Conference on Uncertainty in Artificial Intelligence*, pages 51–58, Banff, Canada, 2004.

[12] R. Brafman, C. Domshlak, and S. E. Shimony. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research*, 25:389–424, 2006.

[13] J. Chomicki. Preference formulas in relational queries. *ACM Transactions on Database Systems*, 28(4):427–466, 2003.

[14] C. Domshlak and R. Brafman. CP-nets - reasoning and consistency testing. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning*, pages 121–132, Toulouse, France, April 2002.

[15] P. C. Fishburn. *The Foundations of Expected Utility*. Reidel, Dordrecht, 1982.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New-York, 1978.

[17] V. Ha and P. Haddawy. A hybrid approach to reasoning with partially elicited preference models. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden, July 1999. Morgan Kaufmann.

[18] S. O. Hansson. Preference logic. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 319–394. Kluwer, 2 edition, 2001.

[19] S. O. Hansson. *The Structure of Values and Norms*. Cambridge University Press, 2001.

[20] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 154–161, 2002.

[21] R. L. Keeney and H. Raiffa. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, 1976.

[22] M. Koksalan and P. Sagala. Interactive approaches for discrete alternative multiple criteria decision making with monotone utility functions. *Management Science*, 41(7):1158–1171, 1995.

[23] P. Korhonen, J. Wallenius, and S. Zionts. Solving the discrete multiple criteria problem using convex cones. *Management Science*, 30(11):1336–1345, 1984.

[24] D. H. Krantz, R. D. Luce, P. Suppes, and A. Tversky. *Foundations of Measurement*. New York: Academic, 1971.

[25] J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1-3):37–71, 2004.

[26] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In *Proceedings of the Sixth International Conference on User Modeling*, pages 67–78, 1997.

[27] M. McGeachie and J. Doyle. Utility functions for ceteris paribus preferences. *Computational Intelligence*, 20(2):158–217, 2004. (Special Issue on Preferences in AI).

[28] S. Prasad, M. Karwan, and S. Zionts. Use of convex cones in interactive multiple objective decision making. *Management Science*, 43(5):723–734, 1997.

[29] P. Pu and B. Faltings. Decision tradeoff using example critiquing and constraint programming. *Constraints: An International Journal*, 9(4):289–310, 2004.

[30] S. W. Tan and J. Pearl. Qualitative decision theory. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 928–933, Seattle, 1994. AAAI Press.

[31] M. Torrens, B. Faltings, and P. Pu. SmartClients: Constraint satisfaction as a paradigm for scaleable intelligent information systems. *Constraints*, 7:49–69, 2002.

[32] M. Wellman and J. Doyle. Preferential semantics for goals. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 698–703, July 1991.

[33] S. Zionts and J. Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–633, 1976.

Appendix A. Proofs

Theorem 3 Every acyclic CP-net is GA-decomposable over its CP-families.

Proof: To prove this claim we constructively show that, for every acyclic CP-net N, there exists a function $v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}_{X_i})$ such that $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy the CP-conditions of N.

Given an acyclic CP-net N, for each variable $X \in N$, each value $x \in dom(X)$, and each assignment $\mathbf{u} \in dom(\mathcal{U}_X)$ we define a parameter $p(x, \mathbf{u})$ to quantify the local preference for x given $\mathbf{u}$:

$$p(x, \mathbf{u}) = |\{x' \mid x' \in dom(X) \text{ and } N \models x \succ_{\mathbf{u}} x'\}|$$

For instance, if x is one of the least preferred value of X given $\mathbf{u}$, then we have $p(x, \mathbf{u}) = 0$, and for each $x \in dom(X)$ we have $p(x, \mathbf{u}) \leq |dom(X)| - 1$. (Recall that $\succ_{\mathbf{u}}$ is a *partial* order over $dom(X)$.)

Next, we define weight coefficients $w^{X_1}, \ldots, w^{X_n}$. They are defined recursively, in a top-down manner, as follows: If X is a root node of N (i.e., $\mathcal{U}_X = \emptyset$), then $w^X = K$ for some arbitrary constant $K > 0$. Now, consider a variable X that is already assigned its weight w^X. We will distribute the weight of X is evenly between its children. This is done by defining the following coefficient for each variable $Y \in \mathcal{Y}_X$.

$$\alpha_{X \to Y} = \frac{w^X}{|\mathcal{Y}_X| \cdot |dom(Y)|} \tag{17}$$

Since N is assumed to be acyclic, prior to processing a non-root node Y, the parameters $\alpha_{X \to Y}$ are known for each $X \in \mathcal{U}_Y$, and we assign:

$$w^Y = \min_{X \in \mathcal{U}_Y} \{\alpha_{X \to Y}\} \tag{18}$$

Having the parameters $p(x, \mathbf{u})$ and w^X as above, we define the set of functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ in terms of these parameters as follows. For each variable X, each $x \in dom(X)$, and each $\mathbf{u} \in dom(\mathcal{U}_X)$, we have:

$$\phi^X(x, \mathbf{u}) = w^X \cdot p(x, \mathbf{u}) \tag{19}$$

To prove the correctness of the construction, it is sufficient to show that $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy the CP-conditions of N. Consider a variable X, a pair of values $x_1, x_2 \in dom(X)$, and an assignment $\mathbf{u} \in dom(\mathcal{U}_X)$, such that $N \models x_1 \succ_{\mathbf{u}} x_2$. For each $\mathbf{v} \in dom(\mathcal{EF}_X - \{X\})$

compatible with $\mathbf{u}$ we have:

$$\begin{aligned}
\sum_{i=1}^{|\mathcal{Y}_X|} \left[\phi^{Y_i}(\mathbf{v}_i, x_2) - \phi^{Y_i}(\mathbf{v}_i, x_1)\right] &= \sum_{i=1}^{|\mathcal{Y}_X|} w^{Y_i} \left(p(\mathbf{v}_i, x_2) - p(\mathbf{v}_i, x_1)\right) \\
&\leq \sum_{i=1}^{|\mathcal{Y}_X|} \alpha_{X \to Y_i} \left(p(\mathbf{v}_i, x_2) - p(\mathbf{v}_i, x_1)\right) \\
&< \sum_{i=1}^{|\mathcal{Y}_X|} \alpha_{X \to Y_i} |dom(Y_i)| \\
&= w^X \;\; < \;\; w^X \left(p(x_1, \mathbf{u}) - p(x_2, \mathbf{u})\right) \\
&= \phi^X(x_1, \mathbf{u}) - \phi^X(x_2, \mathbf{u})
\end{aligned}$$

∎

Lemma 4 Given a TCP-net N with no ci-arcs, and a function $v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}_{X_i})$, we have that v is consistent with N if and only if $\{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfy both the CP and I-conditions of N.

Proof: By definition, a value function v is consistent with N iff, for each pair of complete assignments $\mathbf{x}_1, \mathbf{x}_2 \in dom(\mathcal{X})$ such that $N \models \mathbf{x}_1 \succ \mathbf{x}_2$, we have $v(\mathbf{x}_1) > v(\mathbf{x}_2)$. By the semantics of TCP-nets, for TCP-nets with no ci-arcs, we can reduce this test to just two cases of $\mathbf{x}_1, \mathbf{x}_2$ (the rest of the cases will be implied by the transitivity of the preference relation induced by N):

(1) $\mathbf{x}_1, \mathbf{x}_2$ differ only in the value of a single variable X, and $N \models x_1 \succ_{\mathbf{u}} x_2$, and

(2) $\mathbf{x}, \mathbf{x}'$ differ only in the values of a pair of variables X, X' with i-arc $(\overrightarrow{X, X'}) \in N$, and $N \models x_1 \succ_{\mathbf{u}} x_2$.

where x_1 and x_2 are the values provided to X by $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, and $\mathbf{u}$ is the assignment provided by $\mathbf{x}_1$ (and $\mathbf{x}_2$) to $\mathcal{U}_X$.

Considering case (1), recall the notation used in Definition 3. From GA-decomposition of v it follows that $v(\mathbf{x}_1) > v(\mathbf{x}_2)$ iff

$$\phi^X(x_1, \mathbf{u}) + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_1) > \phi^X(x_2, \mathbf{u}) + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_2)$$

since all other sub-value functions of v take on the same values on $\mathbf{x}_1$ and $\mathbf{x}_2$. However, since $N \models x_1 \succ_{\mathbf{u}} x_2$, this is exactly the formulation of the CP-condition of N with respect to X.

Considering case (2), recall the notation used in Definition 4. Let x_1' and x_2' be the values provided to X' by $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. From GA-decomposition of v it follows that $v(\mathbf{x}_1) > v(\mathbf{x}_2)$ iff

$$\begin{aligned} \phi^X(x_1,\mathbf{u}) + \phi^{X'}(x'_1,\mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_1, x'_1) + \sum_{i=1}^{|\mathcal{Y}_{X'}-\mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_1, x_1) > \\ \phi^X(x_2,\mathbf{u}) + \phi^{X'}(x'_2,\mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_2, x'_2) + \sum_{i=1}^{|\mathcal{Y}_{X'}-\mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_2, x_2) \end{aligned} \tag{20}$$

since all other sub-value functions of v take on the same values on $\mathbf{x}_1$ and $\mathbf{x}_2$. Again, since $N \models x_1 \succ_{\mathbf{u}} x_2$, this is exactly the formulation of the I-condition of N with respect to X.

Finally, suppose that Φ does not satisfy the I-conditions of N. It means that there exists an i-arc $\overrightarrow{(X, X')} \in N$, an assignment $\mathbf{u} \in dom(\mathcal{U}_X)$, and a pair of values $x_1, x_2 \in dom(X)$, such that $N \models x_1 \succ_{\mathbf{u}} x_2$, yet there exist a pair of values $x'_1, x'_2 \in dom(X')$, and a set of (all compatible) $\mathbf{u}' \in dom(\mathcal{U}_{X'})$, $\mathbf{v} \in dom(\mathcal{EF}_X - \{X, X'\})$, $\mathbf{v}' \in dom(\mathcal{EF}_{X'} - \{X, X'\})$ such that:

$$\begin{aligned} \phi^X(x_1,\mathbf{u}) + \phi^{X'}(x'_1,\mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_1, x'_1) + \sum_{i=1}^{|\mathcal{Y}_{X'}-\mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_1, x_1) \leq \\ \phi^X(x_2,\mathbf{u}) + \phi^{X'}(x'_2,\mathbf{u}') + \sum_{i=1}^{|\mathcal{Y}_X|} \phi^{Y_i}(\mathbf{v}_i, x_2, x'_2) + \sum_{i=1}^{|\mathcal{Y}_{X'}-\mathcal{Y}_X|} \phi^{Y'_i}(\mathbf{v}'_i, x'_2, x_2) \end{aligned} \tag{21}$$

Let $\mathcal{A} = \mathcal{X} - (\mathcal{EF}_X \cup \mathcal{EF}_{X'})$. Due to the GA-decomposition of v, for all $\mathbf{a} \in dom(\mathcal{A})$, Eq. 21 implies $v(\mathbf{x}_1) < v(\mathbf{x}_2)$, where $\mathbf{x}_1 = x_1 x'_1 \mathbf{u}\mathbf{u}'\mathbf{v}\mathbf{v}'\mathbf{a}$, and $\mathbf{x}_1 = x_2 x'_2 \mathbf{u}\mathbf{u}'\mathbf{v}\mathbf{v}'\mathbf{a}$. However, this implies that $N \not\models \mathbf{x}_1 \succ \mathbf{x}_2$, which contradicts the semantics of i-arc $\overrightarrow{(X, X')}$. Hence, we accomplished the proof that $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfying the CP-conditions and I-conditions of N is a necessary and sufficient condition for v as in Lemma 4 to be consistent with N. ∎

Theorem 5 Every acyclic TCP-net with no ci-arcs is GA-decomposable over its CP-families.

Proof: First, extending our notation, for each variable $X \in \mathcal{X}$, let $\mathcal{I}_X$ denote the set of all variables $X' \in \mathcal{X}$ that are directly and unconditionally more important that X, i.e., for each $X' \in \mathcal{I}_X$, there is an i-arc $\overrightarrow{(X', X)} \in N$.

The proof of Theorem 5 is based on the construction similar to this in the proof of Theorem 3, but with Eq. 18 replaced by:

$$w^Y = \min\left\{ \min_{X \in \mathcal{U}_Y} \{\alpha_{X \to Y}\}, \min_{X \in \mathcal{I}_Y} \left\{ \frac{w^X}{\delta^2} \right\} \right\} \tag{22}$$

where

$$\delta = 1 + \max_{X \in \mathcal{X}} |dom(X)|$$

First, observe that replacing Eq. 18 by Eq. 22 has no impact on the satisfying the CP-conditions, since $w(Y)$ in Eq. 22 is at least as small (comparatively to all w^X, $X \in \mathcal{U}_Y$) as this in Eq. 18. Therefore, we only should prove that $\Phi = \{\phi^{X_1}, \dots, \phi^{X_n}\}$ constructed this way will satisfy the I-conditions of N. Consider an i-arc $\overrightarrow{(X, X')}$, rewrite Eq. 8 as:

$$\begin{aligned}\phi^X(x_1, \mathbf{u}) - \phi^X(x_2, \mathbf{u}) - \sum_{i=1}^{|\mathcal{Y}_X|} \left[\phi^{Y_i}(\mathbf{v}_i, x_2, x_2') - \phi^{Y_i}(\mathbf{v}_i, x_1, x_1')\right] > \\ \phi^{X'}(x_2', \mathbf{u}') - \phi^{X'}(x_1', \mathbf{u}') - \sum_{i=1}^{|\mathcal{Y}_{X'} - \mathcal{Y}_X|} \left[\phi^{Y_i'}(\mathbf{v}_i', x_1', x_1) - \phi^{Y_i'}(\mathbf{v}_i', x_2', x_2)\right]\end{aligned} \tag{23}$$

and denote the left and right sides of Eq. 23 by $(\star)$ and $(\star\star)$, respectively.

$$\begin{aligned}(\star) &= w^X\left(p(x_1, \mathbf{u}) - p(x_2, \mathbf{u})\right) - \sum_{i=1}^{|\mathcal{Y}_X|} w^{Y_i}\left(p(\mathbf{v}_i, x_2, x_2') - p(\mathbf{v}_i, x_1, x_2')\right) \geq \\ &\geq w^X - \sum_{i=1}^{|\mathcal{Y}_X|} w^{Y_i}\left(|dom(Y_i)| - 1\right) \geq \\ &\geq w^X - \sum_{i=1}^{|\mathcal{Y}_X|} \alpha_{X \to Y_i}\left(|dom(Y_i)| - 1\right) = \\ &= w^X - \sum_{i=1}^{|\mathcal{Y}_X|} \left(\frac{w^X}{|\mathcal{Y}_X|} - \frac{w^X}{|\mathcal{Y}_X| \cdot |dom(Y_i)|}\right) = \\ &= \frac{w^X}{|\mathcal{Y}_X|} \sum_{i=1}^{|\mathcal{Y}_X|} \frac{1}{|dom(Y_i)|} > \\ &> \frac{w^X}{\delta}\end{aligned} \tag{24}$$

$$\begin{aligned}(\star\star) &= w^{X'}\left(p(x_2', \mathbf{u}') - p(x_1', \mathbf{u}')\right) - \sum_{i=1}^{|\mathcal{Y}_{X'}|} w^{Y_i'}\left(p(\mathbf{v}_i', x_1', x_1) - p(\mathbf{v}_i', x_2', x_1)\right) \leq \\ &\leq w^{X'}\left(|dom(X')| - 1\right) + \sum_{i=1}^{|\mathcal{Y}_{X'}|} w^{Y_i'}\left(|dom(Y_i')| - 1\right) \leq \\ &\leq w^{X'}\left(|dom(X')| - 1\right) + \sum_{i=1}^{|\mathcal{Y}_{X'}|} \alpha_{X' \to Y_i'}\left(|dom(Y_i')| - 1\right) = \\ &= w^{X'}\left(|dom(X')| - 1\right) + \sum_{i=1}^{|\mathcal{Y}_{X'}|} \left(\frac{w^{X'}}{|\mathcal{Y}_{X'}|} - \frac{w^{X'}}{|\mathcal{Y}_{X'}| \cdot |dom(Y_i')|}\right) = \\ &= w^{X'}\left(|dom(X')| - \frac{1}{|\mathcal{Y}_{X'}|} \sum_{i=1}^{|\mathcal{Y}_{X'}|} \frac{1}{|dom(Y_i')|}\right) < \\ &< w^{X'}|dom(X')|\end{aligned} \tag{25}$$

From Eqs. 24 and 25 it follows that $\Phi = \{\phi^{X_1}, \dots, \phi^{X_n}\}$ will satisfy the I-conditions of N if:

$$\frac{w^X}{\delta} \geq w^{X'}|dom(X')|,$$

which is ensured by the new definition of w^X in Eq. 22. ∎

Lemma 6 Given a TCP-net N, and a function $v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}^\star_{X_i})$, we have v consistent with N if and only if $\{\phi^{X_1}, \dots, \phi^{X_n}\}$ satisfy CP- and CI-conditions of N.

Proof: By definition, a value function v is consistent with N iff, for each pair of complete assignments $\mathbf{x}_1, \mathbf{x}_2 \in dom(\mathcal{X})$ such that $N \models \mathbf{x}_1 \succ \mathbf{x}_2$, we have $v(\mathbf{x}_1) > v(\mathbf{x}_2)$. By the semantics of TCP-nets, we can reduce this test to just two cases of $\mathbf{x}_1, \mathbf{x}_2$ (the rest of the cases will be implied by the transitivity of the preference relation induced by N):

(1) $\mathbf{x}_1, \mathbf{x}_2$ differ only in the value of a single variable X, and $N \models x_1 \succ_{\mathbf{u}} x_2$, and

(2) $\mathbf{x}, \mathbf{x}'$ differ only in the values of a pair of variables X, X', such that $N \models x_1 \succ_{\mathbf{u}} x_2$, and we have either i-arc $(\overrightarrow{X, X'}) \in N$, or ci-arc $(X, X') \in N$ and X being more important than X' given $\mathbf{s}_\gamma$,

where x_1 and x_2 are the values provided to X by $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, while $\mathbf{u}$ and $\mathbf{s}_\gamma$ are the assignments provided by $\mathbf{x}_1$ (and $\mathbf{x}_2$) to $\mathcal{U}_X$ and $\mathcal{S}_{(X,X')}$, respectively. Analysis of these two cases with respect to Eq. 12 and 11 is similar to the analysis of the corresponding two cases in the proof of Lemma 4 regarding Eqs. 2 and 8, respectively. ∎

Theorem 8 Every acyclic TCP-net is GA-decomposable over its TCP-families.

Proof: Given an acyclic TCP-net N, we define a set of functions $\Phi = \{\phi^{X_1}, \dots, \phi^{X_n}\}$ over $\left\{\mathcal{F}^\star_{X_1}, \cdots, \mathcal{F}^\star_{X_n}\right\}$, respectively, where for $1 \le i \le n$, each $x \in dom(X_i)$, $\mathbf{u} \in dom(\mathcal{U}_{X_i})$, $\mathbf{s} \in dom(\mathcal{S}_{X_i})$, we have:

$$\phi^{X_i}(x, \mathbf{u}, \mathbf{s}) = w^{X_i}(\mathbf{s}) \cdot p^{X_i}(x, \mathbf{u}) \tag{26}$$

The functions p^X are defined exactly as in the proof for Theorem 3. Let $\mathbf{a} \bowtie \mathbf{b}$ denote the fact that two assignments $\mathbf{a}$ and $\mathbf{b}$ on some sets of variables are compatible, i.e. agree on the assignment provided to their overlapping variables. The weight function w^X is defined as:

$$w^X(\mathbf{s}) = \min\left\{ \min_{X' \in \mathcal{U}_X \cup \mathcal{S}_X} \{\alpha_{X' \to X}(\mathbf{s})\}, \min_{X'' \in \mathcal{I}_{X|\mathbf{s}}} \{\beta_{X'' \to X}(\mathbf{s})\} \right\} \tag{27}$$

where, for each $X' \in \mathcal{U}_X \cup \mathcal{S}_X$,

$$\alpha_{X' \to X}(\mathbf{s}) = \min_{\substack{\mathbf{s}' \in dom(\mathcal{S}_{X'}) \\ \mathbf{s}' \bowtie \mathbf{s}}} \left\{ \frac{w^{X'}(\mathbf{s}')}{|\mathcal{Y}^\star_{X'}| \cdot |dom(X)|} \right\} \tag{28}$$

and, for each $X'' \in \mathcal{I}_{X|\mathbf{s}}$,

$$\beta_{X'' \to X}(\mathbf{s}) = \min_{\substack{\mathbf{s}'' \in dom(\mathcal{S}_{X''}) \\ \mathbf{s}'' \bowtie \mathbf{s}}} \left\{ \frac{w^{X''}(\mathbf{s}'')}{\delta^2} \right\} \tag{29}$$

We claim that the function

$$v(\mathcal{X}) = \sum_{i=1}^{n} \phi^{X_i}(\mathcal{F}^\star_{X_i})$$

with $\Phi = \{\phi^{X_1}, \dots, \phi^{X_n}\}$ constructed according to Eq. 26:

1. satisfies the CP- and CI-conditions of N, and
2. have well-defined weight functions $w^{X_1}, \cdots, w^{X_n}$.

CI-conditions: Let us rewrite Eq. 11 as:

$$\begin{aligned}\phi^X(x_1, \mathbf{u}, \mathbf{s}) - \phi^{X'}(x_2, \mathbf{u}, \mathbf{s}) - \sum_{i=1}^{|\mathcal{Y}^\star_X|} \left[\phi^{Y_i}(\mathbf{v}_i, x_2, x_2') - \phi^{Y_i}(\mathbf{v}_i, x_1, x_1')\right] > \\ \phi^{X'}(x_2', \mathbf{u}', \mathbf{s}') - \phi^{X'}(x_1', \mathbf{u}', \mathbf{s}') - \sum_{i=1}^{|\mathcal{Y}^\star_{X'} - \mathcal{Y}^\star_X|} \left[\phi^{Y_i'}(\mathbf{v}_i', x_1', x_1) - \phi^{Y_i'}(\mathbf{v}_i', x_2', x_2)\right]\end{aligned} \tag{30}$$

and denote the right and left sides of Eq. 30 by $(\star)$ and $(\star\star)$, respectively. For $(\star)$, we have:

$$\begin{aligned}(\star) = & \, w^X(\mathbf{s}) \left(p(x_1, \mathbf{u}) - p(x_2, \mathbf{u})\right) \\ & + \sum_{i=1}^{|\mathcal{Y}^\star_X|} \left[w^{Y_i}(\mathbf{s}_i, x_1) p(\mathbf{y}_i, \mathbf{u}_i, x_1, x_1') - w^{Y_i}(\mathbf{s}_i, x_2) p(\mathbf{y}_i, \mathbf{u}_i, x_2, x_2')\right]\end{aligned} \tag{31}$$

where $\mathbf{y}_i$ is the value provided by $\mathbf{v}_i$ to Y_i, while $\mathbf{s}_i$ and $\mathbf{u}_i$ are the projections of $\mathbf{v}_i$ on $\mathcal{S}_{Y_i} - \{X'\}$ and $\mathcal{U}_{Y_i} - \{X'\}$, respectively. (Note that x_1' and x_2' might be redundant in some $p(\mathbf{v}_i, \ldots)$ and w^{Y_i}.) Now, since $p(x_1, \mathbf{u}) > p(x_2, \mathbf{u})$, from Eq. 31 we have:

$$\begin{aligned}(\star) & \geq w^X(\mathbf{s}) - \sum_{i=1}^{|\mathcal{Y}^\star_X|} w^{Y_i}(\mathbf{s}_i, x_2)(|dom(Y_i)| - 1) \geq \\ & \geq w^X(\mathbf{s}) - \sum_{i=1}^{|\mathcal{Y}^\star_X|} \alpha_{X \to Y_i}(\mathbf{s}_i, x_2)(|dom(Y_i)| - 1) = \\ & = w^X(\mathbf{s}) - \sum_{i=1}^{|\mathcal{Y}^\star_X|} \min_{\substack{\bar{\mathbf{s}} \in dom(\mathcal{S}_X) \\ \bar{\mathbf{s}} \bowtie \mathbf{s}_i}} \left\{\frac{w^X(\bar{\mathbf{s}})}{|\mathcal{Y}^\star_X| \cdot |dom(Y_i)|}\right\} \cdot (|dom(Y_i)| - 1) \geq \\ & \geq w^X(\mathbf{s}) - \sum_{i=1}^{|\mathcal{Y}^\star_X|} \left(\frac{w^X(\mathbf{s})}{|\mathcal{Y}^\star_X|} - \frac{w^X(\mathbf{s})}{|\mathcal{Y}^\star_X| \cdot |dom(Y_i)|}\right) = \\ & = \frac{w^X(\mathbf{s})}{|\mathcal{Y}^\star_X|} \sum_{i=1}^{|\mathcal{Y}^\star_X|} \frac{1}{|dom(Y_i)|} > \\ & > \frac{w^X(\mathbf{s})}{\delta}\end{aligned} \tag{32}$$

Writing $(\star\star)$ similarly to $(\star)$ in Eq. 31, we have:

$$\begin{aligned}(\star\star) = & \, w^{X'}(\mathbf{s}') \left(p(x_2', \mathbf{u}') - p(x_1', \mathbf{u}')\right) \\ & + \sum_{i=1}^{|\mathcal{Y}^\star_{X'}|} \left[w^{Y_i'}(\mathbf{s}_i', x_2') p(\mathbf{u}_i', x_2', x_2) - w^{Y_i'}(\mathbf{y}_i', \mathbf{s}_i', x_1') p(\mathbf{y}_i', \mathbf{u}_i', x_1', x_1)\right]\end{aligned} \tag{33}$$

where $\mathbf{y}_i'$ is the value provided by $\mathbf{v}_i'$ to Y_i', while $\mathbf{s}_i'$ and $\mathbf{u}_i'$ are the projections of $\mathbf{v}_i'$ on $\mathcal{S}_{Y_i'} - \{X\}$ and $\mathcal{U}_{Y_i'} - \{X\}$, respectively. (Again, note that x_1 and x_2 might be redundant in some $p(\mathbf{v}'_i, \ldots)$ and $w^{Y'_i}$.) From Eq. 33 we have:

$$
\begin{aligned}
(\star\star) &\leq w^{X'}(\mathbf{s}')\left(|dom(X')|-1\right) + \sum_{i=1}^{|\mathcal{Y}^\star_{X'}|} w^{Y'_i}(\mathbf{s}'_i, x'_2)\left(|dom(Y'_i)|-1\right) \leq \\
&\leq w^{X'}(\mathbf{s}')\left(|dom(X')|-1\right) + \sum_{i=1}^{|\mathcal{Y}^\star_{X'}|} \alpha_{X' \rightarrow Y'_i}(\mathbf{s}'_i, x'_{max})\left(|dom(Y'_i)|-1\right) = \\
&= w^{X'}(\mathbf{s}')\left(|dom(X')|-1\right) + \sum_{i=1}^{|\mathcal{Y}^\star_{X'}|} \min_{\substack{\bar{\mathbf{s}} \in dom(\mathcal{S}_{X'}) \\ \bar{\mathbf{s}} \bowtie \mathbf{s}'_i}} \left\{ \frac{w^{X'}(\bar{\mathbf{s}})}{|\mathcal{Y}^\star_{X'}| \cdot |dom(Y'_i)|} \right\} \left(|dom(Y'_i)|-1\right) = \\
&= w^{X'}(\mathbf{s}')\left(|dom(X')|-1\right) + \min_{\substack{\bar{\mathbf{s}} \in dom(\mathcal{S}_{X'}) \\ \bar{\mathbf{s}} \bowtie \mathbf{s}'_i}} \left\{ w^{X'}(\bar{\mathbf{s}}) \right\} - \frac{\min_{\substack{\bar{\mathbf{s}} \in dom(\mathcal{S}_{X'}) \\ \bar{\mathbf{s}} \bowtie \mathbf{s}'_i}} \left\{ w^{X'}(\bar{\mathbf{s}}) \right\}}{|\mathcal{Y}^\star_{X'}|} \sum_{i=1}^{|\mathcal{Y}^\star_{X'}|} \frac{1}{|dom(Y'_i)|} \leq \\
&\leq w^{X'}(\mathbf{s}') \cdot |dom(X')| < \\
&< w^{X'}(\mathbf{s}') \cdot \delta
\end{aligned}
\tag{34}
$$

From Eq. 32 and 34 it follows that our set of functions $\Phi = \{\phi^{X_1}, \ldots, \phi^{X_n}\}$ satisfies the CI-conditions of N if we have:

$$
\frac{w^X(\mathbf{s})}{\delta} \geq w^{X'}(\mathbf{s}') \cdot \delta,
$$

and, since $X \in \mathcal{I}_{X'|\mathbf{s}'}$, this relation is ensured by the construction of $w^{X_1}, \cdots, w^{X_n}$ (Eqs. 27 and 29).

CP-conditions: Considering Eq. 12, let $\mathbf{y}_i$ is the value provided by $\mathbf{v}_i$ to Y_i, while $\mathbf{s}_i$ and $\mathbf{u}_i$ be the projections of $\mathbf{v}_i$ on $\mathcal{S}_{Y_i}$ and $\mathcal{U}_{Y_i}$, respectively. Satisfaction of Eq. 12 can be shown as follows:

$$
\begin{aligned}
&\sum_{i=1}^{|\mathcal{Y}_X \cup \mathcal{W}_X|} \left[\phi^{Y_i}(\mathbf{y}_i, x_2, \mathbf{u}_i, \mathbf{s}_i) - \phi^{Y_i}(\mathbf{y}_i, x_1, \mathbf{u}_i, \mathbf{s}_i)\right] \\
&\qquad = \sum_{i=1}^{|\mathcal{Y}_X \cup \mathcal{W}_X|} w^{Y_i}(\mathbf{s}_i, x_2) p(\mathbf{y}_i, x_2, \mathbf{u}_i) - w^{Y_i}(\mathbf{s}_i, x_1) p(\mathbf{y}_i, x_1, \mathbf{u}_i) \\
&\qquad \leq \sum_{i=1}^{|\mathcal{Y}_X \cup \mathcal{W}_X|} \alpha_{X \rightarrow Y_i}(\mathbf{s}_i, x_2) p(\mathbf{y}_i, x_2, \mathbf{u}_i, \mathbf{z}_i) \\
&\qquad < \sum_{i=1}^{|\mathcal{Y}_X \cup \mathcal{W}_X|} \alpha_{X \rightarrow Y_i}(\mathbf{s}_i, x_2) |dom(Y_i)| \\
&\qquad \leq \sum_{i=1}^{|\mathcal{Y}_X \cup \mathcal{W}_X|} \frac{w^X(\mathbf{s})}{|\mathcal{Y}_X \cup \mathcal{W}_X|} \\
&\qquad = w^X(\mathbf{s}) \\
&\qquad < w^X(\mathbf{s})\left(p(x_1, \mathbf{u}) - p(x_2, \mathbf{u})\right) \\
&\qquad = \phi^X(x_1, \mathbf{u}, \mathbf{s}) - \phi^X(x_2, \mathbf{u}, \mathbf{s})
\end{aligned}
\tag{35}
$$

Well-definedness: The last thing that remains to be shown is that the functions $w^{X_1}, \cdots, w^{X_n}$

as in Eq. 27 are well-defined. Given a variable $X \in \mathcal{X}$ and an assignment $\mathbf{s} \in dom(\mathcal{S}_X)$, from Eq. 27 it follows that (i) $w^X(\mathbf{s}) = f_{\mathbf{s}}(w^{X'}(\mathbf{s}'))$ for some $X' \in (\mathcal{U}_X \cup \mathcal{S}_X \cup \mathcal{I}_{X|\mathbf{s}})$ and some $\mathbf{s}' \in dom(\mathcal{S}_{X'})$ compatible with $\mathbf{s}$, and (ii) $w^X(\mathbf{s}) < f_{\mathbf{s}}(w^{X'}(\mathbf{s}'))$. Since for the root variables X of N the (only) function $w^X(\emptyset)$ is explicitly specified as $w^X(\emptyset) = K$, the set of functions $w^{X_1}, \cdots, w^{X_n}$ is not well-defined if and only if there exists a sequence of variables $X_{i_1}, \ldots, X_{i_k}$ and a set of assignments $\mathbf{s}_1, \ldots, \mathbf{s}_k$ on $\mathcal{S}_{X_{i_1}}, \ldots, \mathcal{S}_{X_{i_k}}$, respectively, such that:

$$\begin{aligned} w^{X_{i_1}}(\mathbf{s}_1) &= f_{\mathbf{s}_1}(w^{X_{i_2}}(\mathbf{s}_2)) \\ w^{X_{i_2}}(\mathbf{s}_2) &= f_{\mathbf{s}_2}(w^{X_{i_3}}(\mathbf{s}_3)) \\ &\cdots \\ w^{X_{i_{k-1}}}(\mathbf{s}_{k-1}) &= f_{\mathbf{s}_{k-1}}(w^{X_{i_k}}(\mathbf{s}_k)) \\ w^{X_{i_k}}(\mathbf{s}_k) &= f_{\mathbf{s}_k}(w^{X_{i_1}}(\mathbf{s}_1)) \end{aligned} \tag{36}$$

However, assuming that such a sequence exists will immediately violate the acyclicity of N. Hence, $w^{X_1}, \cdots, w^{X_n}$ as in Eq. 27 are well-defined. ∎