International Journal of Foundations of Computer Science
Vol. 30, No. 2 (2019) 333–353

DOI: 10.1142/S0129054119500084

Computing Version Spaces in the Qualitative Approach to Multicriteria Decision Aid

Miguel Couceiro*
Université de Lorraine, CNRS, Inria
LORIA, F-54000 Nancy, France
miguel.couceiro@inria.fr

Miklós Maróti[†], Tamás Waldhauser[‡] and László Zádori[§]
Bolyai Institute, University of Szeged
Aradi vértanúk tere 1, H-6720 Szeged, Hungary
[†]*mmaroti@math.u-szeged.hu*
[‡]*twaldha@math.u-szeged.hu*
[§]*zadori@math.u-szeged.hu*

Received 27 January 2017
Accepted 20 May 2018
Communicated by Oscar Ibarra

We consider a lattice-based model in multiattribute decision making, where preferences are represented by global utility functions that evaluate alternatives in a lattice structure (which can account for situations of indifference as well as of incomparability). Essentially, this evaluation is obtained by first encoding each of the attributes (nominal, qualitative, numeric, etc.) of each alternative into a distributive lattice, and then aggregating such values by lattice functions. We formulate version spaces within this model (global preferences consistent with empirical data) as solutions of an interpolation problem and present their complete descriptions accordingly. Moreover, we consider the computational complexity of this interpolation problem, and show that up to 3 attributes it is solvable in polynomial time, whereas it is NP complete over more than 3 attributes. Our results are then illustrated with a concrete example.

Keywords: Multiple criteria analysis; complexity theory; preference modelling; Sugeno integral; decision under uncertainty.

1. Motivation

We consider a problem rooted in supervised learning and stated as an interpolation problem for functions $f : \mathbf{X} \to L$, where $\mathbf{X}$ is a set of objects (or alternatives) and L is a set of labels: *Given a finite $S \subseteq \mathbf{X} \times L$, decide whether there exists an $f : \mathbf{X} \to L$ interpolating S, i.e., such that $f(\mathbf{a}) = b$ for every $(\mathbf{a}, b) \in S$.* Our motivation is

*Corresponding author.

found in the field of decision making, more specifically, in the qualtitative approach to preference modeling and learning (prediction and elicitation).

As the starting point, we take the decomposable model to represent preferences over a set $\mathbf{X} = X_1 \times \cdots \times X_n$ of alternatives (e.g., houses to buy) described by n attributes $x_i \in X_i$ (e.g., price, size, location, color). In this setting, preference relations $\preceq$ are represented by "overall utility functions" $U : \mathbf{X} \to L$ valued in a scale L (also called the "evaluation space") using the following rule:

$$\mathbf{x} \preceq \mathbf{y} \quad \text{if and only if} \quad U(\mathbf{x}) \leq U(\mathbf{y}).$$

This representation of preference relations is usually refined by taking into account "local preferences" $\preceq_i$ on each X_i, modeled by mappings $\varphi_i : X_i \to L$ called "local utility functions", which are then merged through an aggregation function $A : L^n \to L$ into an overall utility function U:

$$U(\mathbf{x}) = A(\varphi_1(x_1), \ldots, \varphi_n(x_n)). \tag{1}$$

Loosely speaking, A merges the local preferences in order to obtain a global preference on the set of alternatives. In the qualitative setting, the aggregation function of choice is the *Sugeno integral*[25, 26] that can be regarded as an *idempotent lattice polynomial function* [6, 19], and the resulting global utility function (1) is then called a *pseudo-polynomial function* [10] or a *Sugeno utility function* [9] in the case when A is a Sugeno integral and the local utility functions are order-preserving. This observation brings the concept of Sugeno integral to domains more general than scales (linearly ordered sets) such as distributive lattices and Boolean algebras. Apart from the theoretic interest, such a generalization is both natural and useful as it allows incomparability amongst alternatives, a situation that is most common in real-life situations. Preferences modelled by (1) were axiomatized by different approaches in [1, 4, 17].

The interest of considering the interpolation problem in this model-based setting becomes apparent when dealing with supervised learning of preference relations in the qualitative setting, and which leads naturally to the following extension of the interpolation problem: *Given a finite $S \subseteq \mathbf{X} \times L$, find all pseudo-polynomial functions $U : \mathbf{X} \to L$ that interpolate S.* In other words, given a data set S consisting of pairs $(\mathbf{a}, b)$ of alternatives together with their evaluations, we would like to determine all models (1) that are consistent with S; in the terminology of machine learning (see, e.g., [3, 20]) the set of all such models is called the *version space.*

A complete solution of the interpolation problem thus provides an explicit description of version spaces in the multicriteria setting. Solutions to particular instances have been presented in the literature. In particular, the problem of covering a set of data by a set of Sugeno integrals was considered in the linearly ordered case [22, 23] where conditions that guarantee the existence of a Sugeno integral interpolating a set of data were provided. Essentially, the set of interpolating Sugeno integrals (if they exist) was characterized as being upper and lower bounded by particular Sugeno integrals (easy to build from data). These results were then generalized in two different directions. In [21] an approach by "splines" was proposed, which enables elicitation of families of generalized Sugeno integrals from pieces of data where local and global evaluations may be imprecisely known, whereas in [5, 11] lattice theoretic approaches were proposed not only to determine existence but also to provide explicit descriptions of all possible lattice polynomials interpolating a given data set S.

In the current paper we solve the above mentioned pseudo-polynomial interpolation problem and thus describe version spaces for models (1). An important special case is the case of *quasi-polynomial functions* [7, 8], where $X_1 = \cdots = X_n = X$ is an arbitrary set (not necessarily ordered) and $\varphi_1 = \cdots = \varphi_n = \varphi : X \to L$. Such a framework is pertaining to *decision under uncertainty* and it is used to model situations where we need to take into account different states of a given world.

The paper is organized as follows. In Sec. 2, we recall basic notions and terminology in lattice theory, and present results and constructions pertaining to interpolation by lattice polynomial functions. Extensions of the interpolation problem by pseudo- and quasi-polynomial functions are then proposed and solved in Sec. 3. For the sake of simplicity we present the solution in the setting of decision under uncertainty (interpolation by quasi-polynomials), but our method can be applied also in the multicriteria setting (interpolation by pseudo-polynomials). These results are then illustrated in Sec. 4 by a concrete example. In Sec. 5, we prove that for $n \geq 4$ it is an NP-complete problem to decide if the interpolation problem has a solution, while for $n \leq 3$ it can be decided in polynomial time. We conclude the paper in Sec. 6, where we indicate ongoing work and suggest other directions of future research.

Before proceeding, we would like to stress the fact that, despite being motivated by a problem rooted in preference learning (see [13] for general background and a thorough treatment of the topic), our setting differs from the standard setting in machine learning. This is mainly due to the fact that we aim to describe utility-based preference models that are consistent with existing data (version spaces) rather than aiming to learn utility-based models by optimization (minimizing loss measures and coefficients) such as in, e.g., the probabilistic approach of [2] or the approach based on the Choquet integral of [27], and that naturally accounts for errors and inconsistencies in the learning data. Another difference is that, in the latter, data is supposed to be given in the form of feature vectors (thus assuming that local utilities over attributes are known a priori), an assumption that removes the additional difficulty that we face, namely, that of describing local utility functions that enable models based on the Sugeno integral that are consistent with existing data. It is also worth noting that we do not assume any structure on attributes and that we allow incomparabilities in the evaluation space L, which thus subsumes preferences that are not necessarily rankings.

2. Preliminaries

Throughout this paper let L be a distributive lattice. Recall that a *polynomial function over* L is a mapping $p : L^n \to L$ that can be expressed as a combination of the lattice operations $\wedge$ and $\vee$, projections and constants. In the case when L is bounded, i.e., with a least and a greatest element, polynomial functions $p : L^n \to L$ can be represented in *disjunctive normal form* (DNF for short) by

$$p(\mathbf{y}) = \bigvee_{I \subseteq [n]} \left(c_I \wedge \bigwedge_{i \in I} y_i \right), \quad \text{where } \mathbf{y} = (y_1, \ldots, y_n) \in L^n. \tag{2}$$

Here, and throughout the paper, we denote the set $\{1, 2, \ldots, n\}$ by $[n]$. One can assume without loss of generality that the coefficients $c_I \in L$ are monotone in the sense that $c_I \leq c_J$ whenever $I \subseteq J$. Under this monotonicity assumption the coefficients of the DNF of the polynomial function p are uniquely determined.

As mentioned in Sec. 1, a natural model for supervised preference learning is the following interpolation problem, where a multivariable partial function on a lattice is to be interpolated by lattice polynomial functions.

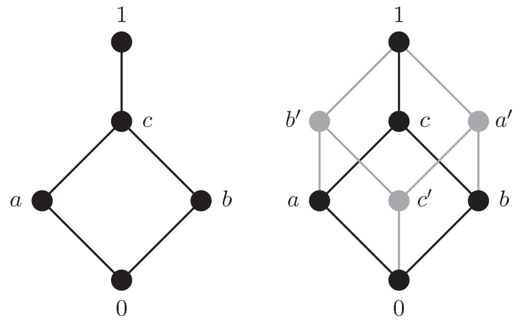
Polynomial Interpolation Problem. Let L be a distributive lattice. Given an arbitrary finite set $D \subseteq L^n$ and $g : D \to L$, find all polynomial functions $p : L^n \to L$ such that $p|_D = g$.

Unlike in the case of interpolation by real polynomial functions, solutions do not necessarily exist, and it is a nontrivial problem to determine the necessary and sufficient conditions for the existence of an interpolating lattice polynomial function. Goodstein's theorem [15] provides a solution in the special case when the domain of g is the hypercube $D = \{0, 1\}^n$, where 0 and 1 are the least and greatest elements of the bounded distributive lattice L: a function $g : \{0, 1\}^n \to L$ can be interpolated by a polynomial function $p : L^n \to L$ if and only if g is monotone, and in this case p is unique. This result was generalized in [11] by allowing L to be an arbitrary (possibly unbounded) distributive lattice and by considering functions $g : D \to L$, where $D = \{a_1, b_1\} \times \cdots \times \{a_n, b_n\}$ with $a_i, b_i \in L$ and $a_i < b_i$, for each $i \in [n]$.

To describe the general solution of the Polynomial Interpolation Problem, which was given in [5], we need to recall that by the Birkhoff-Priestley representation theorem [12] we can embed any distributive lattice L into a Boolean algebra B, which can be assumed to be a subalgebra of the power set $\mathcal{P}(\Omega)$ of a set Ω. For the sake of canonicity, we assume that L generates B, so that B is uniquely determined up to isomorphism. The complement of an element $a \in B$ is denoted by a'. (See Fig. 1 for an example.)

Given a function $g : D \to L$, we define the following two elements in B for each $I \subseteq [n]$:

$$c_I^- := \bigvee_{\mathbf{a} \in D} \left(g(\mathbf{a}) \wedge \bigwedge_{i \notin I} a_i' \right) \quad \text{and} \quad c_I^+ := \bigwedge_{\mathbf{a} \in D} \left(g(\mathbf{a}) \vee \bigvee_{i \in I} a_i' \right).$$



(a) L_5 (b) $B(L_5)$

Fig. 1. A distributive lattice and its Boolean algebra.

Observe that $I \subseteq J$ implies $c_I^- \leq c_J^-$ and $c_I^+ \leq c_J^+$. Let p^- and p^+ be the polynomial functions over B given by these two systems of coefficients:

$$p^-(\mathbf{y}) := \bigvee_{I \subseteq [n]} \left(c_I^- \wedge \bigwedge_{i \in I} y_i \right) \quad \text{and} \quad p^+(\mathbf{y}) := \bigvee_{I \subseteq [n]} \left(c_I^+ \wedge \bigwedge_{i \in I} y_i \right).$$

As it turns out [5], p^- and p^+ are the least and greatest polynomial functions over B whose restriction to D coincides with g (whenever such a polynomial function exists). This yields the following explicit description of all possible interpolating polynomial functions over the Boolean algebra B.

Theorem 1 ([5]). *Let L be a distributive lattice, and let B be the Boolean algebra generated by L. Let $g : D \to L$ be a function defined on a finite set $D \subseteq L^n$, and let $p : B^n \to B$ be a polynomial function over B given by* (2).

Then the following conditions are equivalent:

(i) *p interpolates g, i.e., $p|_D = g$;*
(ii) *$c_I^- \leq c_I \leq c_I^+$ for all $I \subseteq [n]$;*
(iii) *$p^- \leq p \leq p^+$.*

From Theorem 1 it follows that a necessary and sufficient condition for the existence of a polynomial function $p : B^n \to B$ such that $p|_D = g$ is $c_I^- \leq c_I^+$, for every $I \subseteq [n]$. Moreover, if for every $I \subseteq [n]$, there is $c_I \in L$ such that $c_I^- \leq c_I \leq c_I^+$, then and only then there is a polynomial function $p : L^n \to L$ such that $p|_D = g$. For the special type of interpolation problem considered in [11], the condition for the existence of a solution was given by simple lattice inequalities, without referring to the Boolean algebra generated by the lattice. In the case when L is a finite chain such a condition was given in [23], where, rather than polynomial functions, the interpolating functions where assumed to be Sugeno integrals, i.e., idempotent polynomial functions (see [18, 19]). One can also obtain the solution of the Polynomial Interpolation Problem over L in this case from Theorem 1 by describing explicitly the Boolean algebra generated by a finite chain. This yields the following result, which basically reformulates Theorem 3 in [23] in the language of lattice theory [5].

Theorem 2 ([23]). *Let L be a finite chain, and let $g : D \to L$ be a function defined on a subset $D \subseteq L^n$. Then there is a polynomial function $p : L^n \to L$ such that $p|_D = g$ if and only if*

$$\forall \mathbf{a}, \mathbf{b} \in D : \; g(\mathbf{a}) < g(\mathbf{b}) \Rightarrow \exists i \in [n] : a_i \leq g(\mathbf{a}) < g(\mathbf{b}) \leq b_i. \tag{3}$$

In contrast to the above mentioned special cases, in general it is not possible to avoid the use of the Boolean algebra generated by L, as it is illustrated by the following example.

Example 3 ([5]). Let L_5 be the five-element lattice shown in Fig. 1(a), and let $B(L_5)$ be the Boolean algebra generated by L_5 (see Fig. 1(b)). Let $D = \{\mathbf{a}, \mathbf{b}\}$, where $\mathbf{a} = (1, c)$, $\mathbf{b} = (c, a)$ and consider $g : D \to L_5$ defined by

$$g(\mathbf{a}) = 1 \quad \text{and} \quad g(\mathbf{b}) = a. \tag{4}$$

As coefficients c_I^- and c_I^+ we obtain

$$c_\emptyset^- = 0, \quad c_{\{1\}}^- = c', \quad c_{\{2\}}^- = 0, \quad c_{\{1,2\}}^- = 1,$$

$$c_\emptyset^+ = a, \quad c_{\{1\}}^+ = b', \quad c_{\{2\}}^+ = 1, \quad c_{\{1,2\}}^+ = 1.$$

We see that $c_I^- \leq c_I^+$ holds for each $I \subseteq [2]$, hence this interpolation problem has a solution over $B(L_5)$ (in fact, it has 32 solutions), by Theorem 1. On the other hand, no element of L_5 lies between $c_{\{1\}}^-$ and $c_{\{1\}}^+$, hence there is no solution over L_5.

3. Generalized Lattice Interpolation

As mentioned in the introduction, the motivation for considering the interpolation problem is rooted in the qualitative approach to preference modeling, where preference relations $\preceq$ over a set $X_1 \times \cdots \times X_n$ of alternatives described by n attributes are represented by overall utility functions $U : X_1 \times \cdots \times X_n \to L$ valued in an ordered set L, by the rule:

$$\mathbf{x} \preceq \mathbf{y} \quad \text{if and only if} \quad U(\mathbf{x}) \leq U(\mathbf{y}).$$

Preferences on the attributes X_i are in turn modeled by local utility functions $\varphi_i : X_i \to L$, which are then aggregated through a lattice polynomial $p : L^n \to L$ thus giving rise to refined models

$$U(\mathbf{x}) = p(\varphi_1(x_1), \ldots, \varphi_n(x_n)), \tag{5}$$

which we referred to as *pseudo-polynomial functions.*

The interest of considering the interpolation problem in this setting becomes apparent when dealing with preference relations that are partially defined. This situation of incomplete information pertains to preference learning, where the set of interpolating pseudo-polynomial functions constitutes its version space. This motivates the following extension of the interpolation problem (stated as

Problem 5.1 in [11]):

Pseudo-polynomial Interpolation Problem. Let $X_1, \ldots, X_n$ be finite sets and L a finite distributive lattice. Given $C \subseteq X_1 \times \cdots \times X_n$ and a partial function $f : C \to L$, find all pseudo-polynomial functions $U : X_1 \times \cdots \times X_n \to L$ such that $U|_C = f$.

As mentioned in Sec. 1, uncertainty can be modeled by special kinds of pseudo-polynomials, where $X_1 = \cdots = X_n = X$ and $\varphi_1 = \cdots = \varphi_n = \varphi$. The resulting global utilty functions $U : X^n \to L$ are so-called *quasi-polynomial functions*:

$$U(\mathbf{x}) = p(\varphi(x_1), \ldots, \varphi(x_n)). \tag{6}$$

The corresponding interpolation problem can be formulated as follows:

Quasi-polynomial Interpolation Problem. Let X be a finite set and L a finite distributive lattice. Given $C \subseteq X^n$ and a partial function $f : C \to L$, find all quasi-polynomial functions $U : X^n \to L$ such that $U|_C = f$.

We present the solution of the Pseudo-polynomial Interpolation Problem in two steps. First, in Sec. 3.1 we show how to find the appropriate polynomials p provided that the local utility functions $\varphi_1, \ldots, \varphi_n$ are given. Then, in Sec. 3.2 we give an algorithm to construct all possible local utility functions that could appear in an interpolation. To simplify the formalism, in Sec. 3.2 we consider the special case of quasi-polynomials, but our method can be easily adapted to the more general problem of pseudo-polynomial interpolation, see Remark 9.

3.1. *Interpolation with known local utility functions*

Assume that the local utility functions $\varphi_i : X_i \to L$ are given; our goal is to find all polynomial functions p over L such that the pseudo-polynomial function U given by (5) interpolates f. Let us consider an arbitrary polynomial function p over B in its disjunctive normal form (2). The corresponding pseudo-polynomial function $U = p(\varphi_1, \ldots, \varphi_n)$ interpolates f if and only if $p(\varphi_1(a_1), \ldots, \varphi_n(a_n)) = f(a_1, \ldots, a_n)$ for all $\mathbf{a} \in C$, i.e., if p interpolates the function $g : D \to L$ defined on the set

$$D = \{(\varphi_1(a_1), \ldots, \varphi_n(a_n)) : \mathbf{a} \in C\}$$

by

$$g(\varphi_1(a_1), \ldots, \varphi_n(a_n)) = f(a_1, \ldots, a_n).$$

Using the construction of Sec. 2 for this interpolation problem, we can define coefficients $c^-_{I,\varphi_1,\ldots,\varphi_n}$ and $c^+_{I,\varphi_1,\ldots,\varphi_n}$ for every $I \subseteq [n]$ as follows:

$$c^-_{I,\varphi_1,\ldots,\varphi_n} := \bigvee_{\mathbf{a}\in C} \left(f(\mathbf{a}) \wedge \bigwedge_{i \notin I} \varphi_i(a_i)' \right) \quad \text{and} \quad c^+_{I,\varphi_1,\ldots,\varphi_n} := \bigwedge_{\mathbf{a}\in C} \left(f(\mathbf{a}) \vee \bigvee_{i \in I} \varphi_i(a_i)' \right).$$

Denoting the corresponding polynomial functions by $p^-_{\varphi_1,\dots,\varphi_n}$ and $p^+_{\varphi_1,\dots,\varphi_n}$, Theorem 1 yields the following solution for the Pseudo-polynomial Interpolation Problem with known local utility functions.

Theorem 4. *Let $X_1,\dots,X_n$ be finite sets, let L be a finite distributive lattice, and let $f : C \to L$ be a function defined on a set $C \subseteq X_1 \times \cdots \times X_n$. For any maps $\varphi_i : X_i \to L (i \in [n])$ and any polynomial function $p : B^n \to B$ over B given by* (2), *the following conditions are equivalent*:

(i) $U = p(\varphi_1,\dots,\varphi_n)$ *interpolates* f, *i.e.*, $U|_C = f$;
(ii) $c^-_{I,\varphi_1,\dots,\varphi_n} \leq c_I \leq c^+_{I,\varphi_1,\dots,\varphi_n}$ *for all* $I \subseteq [n]$;
(iii) $p^-_{\varphi_1,\dots,\varphi_n} \leq p \leq p^+_{\varphi_1,\dots,\varphi_n}$.

Remark 5. Note that if there exist tuples $\mathbf{a},\mathbf{b} \in C$ such that $f(\mathbf{a}) \neq f(\mathbf{b})$ but $left(\varphi_1(a_1),\dots,\varphi_n(a_n)) = (\varphi_1(b_1),\dots,\varphi_n(b_n))$, then it is clearly impossible to find an interpolating pseudo-polynomial function (or any kind of function at all). We invite the reader to verify that this situation cannot occur if condition (ii) of Theorem 4 is satisfied.

3.2. *Interpolation with unknown local utility functions*

Now let us consider interpolation by quasi-polynomial functions

$$U(\mathbf{x}) = p(\varphi(x_1),\dots,\varphi(x_n)),$$

where the local utility function $\varphi : X \to L$ is not known. Our aim is to find all possible maps φ for which an interpolating polynomial exists. Specializing the results of the previous subsection to the case $\varphi_1 = \cdots = \varphi_n = \varphi$, we see that the necessary and sufficient condition for the existence of a solution over B is that the following inequalities hold:

$$\bigvee_{\mathbf{a}\in C}\left(f(\mathbf{a}) \wedge \bigwedge_{i\notin I}\varphi(a_i)'\right) \leq \bigwedge_{\mathbf{b}\in C}\left(f(\mathbf{b}) \vee \bigvee_{i\in I}\varphi(b_i)'\right) \quad \text{for all } I \subseteq [n]. \tag{7}$$

This gives a system of inequalities for the unknown values $\varphi(a)$ $(a \in X)$. To find all solutions of this system of inequalities, we make use of the fact that B can be embedded into the power set of a set Ω. We will encode a map $\varphi : X \to B$ by a system of sets $S_\omega \subseteq X (\omega \in \Omega)$, where $S_\omega = \{a \in X : \omega \in \varphi(a)\}$. (Note that from these sets we can uniquely recover the function φ as $\varphi(a) = \{\omega \in \Omega : a \in S_\omega\}$.) For each $\omega \in \Omega$, we define a hypergraph $\mathcal{H}_\omega = (X; \mathcal{E}_\omega)$ with vertex set X and edge set

$$\mathcal{E}_\omega = \{E_\omega(\mathbf{a},\mathbf{b}) : \mathbf{a},\mathbf{b} \in C \text{ such that } \omega \in f(\mathbf{a})\backslash f(\mathbf{b})\},$$

where $E_\omega(\mathbf{a},\mathbf{b}) = \{a_i : b_i \notin S_\omega\} \subseteq X$. (Here $f(\mathbf{a})\backslash f(\mathbf{b}) = f(\mathbf{a}) \wedge f(\mathbf{b})'$ is the difference of the sets $f(\mathbf{a}), f(\mathbf{b}) \subseteq \Omega$.) In the next theorem we will prove that (7) holds if and only if S_ω is a *transversal* (also called a *hitting set* or a *vertex cover*) of the hypergraph $\mathcal{H}_\omega$, i.e., S_ω intersects every edge of $\mathcal{H}_\omega$ for every $\omega \in \Omega$.

Theorem 6. *Let X be a finite set, let L be a finite distributive lattice, and let $f : C \to L$ be a function defined on a set $C \subseteq X^n$. For any map $\varphi : X \to L$, the following conditions are equivalent*:

(i) *there exists a polynomial function $p : B^n \to B$ such that the quasi-polynomial function $U = p(\varphi, \ldots, \varphi)$ interpolates f, i.e., $U|_C = f$;*
(ii) *for all $\mathbf{a}, \mathbf{b} \in C$, $\omega \in f(\mathbf{a}) \backslash f(\mathbf{b})$ and $I \subseteq [n]$, we have $\{b_i : i \in I\} \subseteq S_\omega \Rightarrow \{a_i : i \notin I\} \cap S_\omega \neq \emptyset$;*
(iii) *S_ω is a transversal of $\mathcal{H}_\omega$ for each $\omega \in \Omega$.*

Proof. First we prove the equivalence of (i) and (ii). By Theorem 4, condition (i) is equivalent to (7). The inequality in (7) holds if and only if each joinand on the left hand side is less than or equal to each meetand on the right hand side:

$$\forall \mathbf{a}, \mathbf{b} \in C \ \forall I \subseteq [n] : \quad f(\mathbf{a}) \wedge \bigwedge_{i \notin I} \varphi(a_i)' \leq f(\mathbf{b}) \vee \bigvee_{i \in I} \varphi(b_i)'. \tag{8}$$

Recall that we have embedded B into the power set of Ω, hence $\varphi(a_i)$ and $\varphi(b_i)$ in (8) are thought of as subsets of Ω as well as $f(\mathbf{a})$ and $f(\mathbf{b})$. If $\omega \notin f(\mathbf{a})$, then ω does not belong to the left hand side of (8), while if $\omega \in f(\mathbf{b})$, then the right hand side contains ω. Thus, it suffices to consider elements $\omega \in f(\mathbf{a}) \backslash f(\mathbf{b})$, and therefore (8) is equivalent to

$$\forall \mathbf{a}, \mathbf{b} \in C \ \forall \omega \in f(\mathbf{a}) \backslash f(\mathbf{b}) \ \forall I \subseteq [n] : \quad \omega \in \bigwedge_{i \notin I} \varphi(a_i)' \Rightarrow \omega \in \bigvee_{i \in I} \varphi(b_i)'.$$

Using De Morgan's laws and contraposition, we see that this is equivalent to

$$\forall \mathbf{a}, \mathbf{b} \in C \ \forall \omega \in f(\mathbf{a}) \backslash f(\mathbf{b}) \ \forall I \subseteq [n] : \quad \forall i \in I : \omega \in \varphi(b_i) \Rightarrow \exists i \notin I : \omega \in \varphi(a_i).$$

Now (ii) is just a reformulation of the condition above using the definition of the sets S_ω.

Next we prove that (ii) implies (iii). Let $E_\omega(\mathbf{a}, \mathbf{b}) = \{a_i : b_i \notin S_\omega\}$ be any edge of $\mathcal{H}_\omega$, and let us put $I = \{i : b_i \in S_\omega\}$. Applying (ii), we obtain an index $i \notin I$ such that $a_i \in S_\omega$. Since $i \notin I$, we have $b_i \notin S_\omega$, and this means that $a_i \in E_\omega(\mathbf{a}, \mathbf{b})$. This proves that $a_i \in E_\omega(\mathbf{a}, \mathbf{b}) \cap S_\omega$ for every edge $E_\omega(\mathbf{a}, \mathbf{b})$ of the hypergraph $\mathcal{H}_\omega$, hence S_ω is a transversal of $\mathcal{H}_\omega$.

Finally, let us show that (iii) implies (ii). Let $\mathbf{a}, \mathbf{b} \in C$, $\omega \in f(\mathbf{a}) \backslash f(\mathbf{b})$ and $I \subseteq [n]$, and assume that $\{b_i : i \in I\} \subseteq S_\omega$. Since S_ω is a transversal of $\mathcal{H}_\omega$, there exists a vertex $v \in E_\omega(\mathbf{a}, \mathbf{b}) \cap S_\omega$. From $v \in E_\omega(\mathbf{a}, \mathbf{b})$ it follows that $v = a_i$ for some $i \in [n]$ such that $b_i \notin S_\omega$. Since $\{b_i : i \in I\} \subseteq S_\omega$, we must have $i \notin I$, and this means that $v \in \{a_i : i \notin I\} \cap S_\omega$, as claimed in (ii). □

Theorem 6 yields Algorithm 1 for finding all local utility functions φ that provide a quasi-polynomial function interpolating f on C. We start with $S_\omega = \emptyset$, and at every iteration we extend S_ω if necessary to obtain a transversal of $\mathcal{H}_\varepsilon$. It suffices

Algorithm 1 Constructing all sets S_ω that satisfy the conditions of Theorem 6

1: **procedure** EXTEND(S_ω) ▷ extends S_ω in all possible ways
2: **for all** transversals T of $\tilde{\mathcal{H}}_\omega$ **do**
3: $S_\omega \leftarrow S_\omega \cup T$
4: compute $\tilde{\mathcal{E}}_\omega$ by (9)
5: **if** $\tilde{\mathcal{E}}_\omega = \emptyset$ **then** ▷ S_ω is a transversal of $\tilde{\mathcal{H}}_\omega$
6: **output** S_ω
7: **else if** $\emptyset \in \tilde{\mathcal{E}}_\omega$ **then** ▷ $\tilde{\mathcal{H}}_\omega$ has an empty edge
8: **output fail**
9: **else** ▷ S_ω might extend to a transversal
10: EXTEND(S_ω)
11: **end if**
12: **end for**
13: **end procedure**

14: $S_\omega \leftarrow \emptyset$
15: compute $\tilde{\mathcal{E}}_\omega$ by (9)
16: EXTEND(S_ω)

to consider only those edges of $\mathcal{H}_\omega$ that are disjoint from S_ω. Let $\tilde{\mathcal{E}}_\omega$ be the set of these edges, and let $\tilde{\mathcal{H}}_\omega = (X; \tilde{\mathcal{E}}_\omega)$ be the corresponding hypergraph:

$$\tilde{\mathcal{E}}_\omega = \{E_\omega(\mathbf{a}, \mathbf{b}) : \mathbf{a}, \mathbf{b} \in C \text{ such that } \omega \in f(\mathbf{a}) \setminus f(\mathbf{b}) \text{ and } E_\omega(\mathbf{a}, \mathbf{b}) \cap S_\omega = \emptyset\}. \quad (9)$$

We add a transversal of $\tilde{\mathcal{H}}_\omega$ to S_ω and then we recompute $\tilde{\mathcal{E}}_\omega$ by (9). We iterate these steps until S_ω either becomes a transversal of $\mathcal{H}_\omega$ (i.e., $\tilde{\mathcal{E}}_\omega = \emptyset$) or one of the edges becomes empty (i.e., $\emptyset \in \tilde{\mathcal{E}}_\omega$). In the latter case $\mathcal{H}_\omega$ has no transversal at all, and this cannot be resolved by continuing the iteration, since the edges $E_\omega(\mathbf{a}, \mathbf{b})$ can only get smaller as S_ω is being extended. We will eventually reach one of the terminating conditions $\tilde{\mathcal{E}}_\omega = \emptyset$ or $\emptyset \in \tilde{\mathcal{E}}_\omega$. Indeed, if the algorithm does not terminate by finding a vertex cover of $\mathcal{H}_\omega$, then sooner or later S_ω will contain all the elements $b_1, \ldots, b_n$ (as X is finite), which implies that $E_\omega(\mathbf{a}, \mathbf{b}) = \emptyset$ for all $\mathbf{a}, \mathbf{b}$.

Remark 7. In order to make sure that we find all solutions, we must try *every* transversal of $\tilde{\mathcal{H}}_\omega$ in line 2 of the algorithm in every iteration, and proceed with these extensions recursively, in a depth-first search manner. If we would like to find just one solution (if there is one at all), then it is sufficient to add a minimal transversal of $\mathcal{H}$, but still we must try *every* minimal transversal in every iteration, leading to an exponential running time. The example of Sec. 4 below shows that this cannot be avoided, since it is possible that certain transversals lead to a contradiction, while other transversals give a solution. Also, in Sec. 5 we prove that even deciding the existence of an interpolating quasi-polynomial function is an NP-complete problem, hence an effective algorithm cannot be expected unless P = NP.

To determine the whole version space, i.e., the set of all interpolating quasi-polynomial functions, one needs to compute all possible systems of sets $S_\omega (\omega \in \Omega)$, and then one can define the corresponding local utility functions $\varphi : X \to B$ by $\varphi(a) = \{\omega \in \Omega : a \in S_\omega\}$. After computing all such maps φ, one can select those for which $\varphi(a) \in L$ holds for all $a \in X$. Then using the construction of Sec. 3.1 one can determine the corresponding polynomial functions p for each φ. Recall that the coefficients $c^-_{I,\varphi}, c^+_{I,\varphi}$ belong to B, but we need only the elements of L that lie between $c^-_{I,\varphi}$ and $c^+_{I,\varphi}$.

Example 8. Note that the current setting is strictly more general than that of the previous section. To illustrate this, let $X = \{0, a, 1\} = L$ and $C = \{(0,1), (1,0), (a,a), (1,1)\}$. (The ordering on L is $0 < a < 1$, i.e., L is a three-element chain. Then B can be chosen as $\{0, a, a', 1\}$ with $0 < a, a' < 1$.) Consider $f : C \to L$ given by

$$\begin{aligned} f(a,a) &= 0, \\ f(0,1) &= f(1,0) = a, \\ f(1,1) &= 1. \end{aligned}$$

Using Theorem 1, we can verify that there is no polynomial function that would interpolate f on C (even if considered over the Boolean lattice B extending L). However, taking $\varphi : X \to L$ given by $\varphi(0) = \varphi(a) = 0$ and $\varphi(1) = 1$, we get

$$\begin{aligned} c^-_{\emptyset,\varphi} &= c^+_{\emptyset,\varphi} = 0, \\ c^-_{\{1\},\varphi} &= c^+_{\{1\},\varphi} = c^-_{\{2\},\varphi} = c^+_{\{2\},\varphi} = a, \\ c^-_{\{1,2\},\varphi} &= c^+_{\{1,2\},\varphi} = 1. \end{aligned}$$

Hence, $p = p^-_\varphi = p^+_\varphi = (a \wedge x_1) \vee (a \wedge x_2) \vee (1 \wedge x_1 \wedge x_2)$, and it is not difficult to verify that $U = p \circ \varphi$ indeed interpolates f.

Remark 9. Let $U : X_1 \times \cdots \times X_n \to L$ be a pseudo-polynomial function of the form (5). Assume (without loss of generality) that the sets $X_1, \ldots, X_n$ are pairwise disjoint, and let $X = X_1 \cup \cdots \cup X_n$ and $\varphi = \varphi_1 \cup \cdots \cup \varphi_n$. Consider the quasi-polynomial function $\widetilde{U} : X^n \to L$ defined by $\widetilde{U}(\mathbf{x}) = p(\varphi(x_1), \ldots, \varphi(x_n))$. Observe that $X_1 \times \cdots \times X_n \subseteq X^n$ and the restriction of $\widetilde{U}$ to $X_1 \times \cdots \times X_n$ coincides with U. Thus, every pseudo-polynomial function can be viewed as a restriction of a quasi-polynomial function. Conversely, if $p(\varphi(x_1), \ldots, \varphi(x_n))$ is a quasi-polynomial function over X, then its restriction to $X_1 \times \cdots \times X_n$ is a pseudo-polynomial function corresponding to the local utility functions $\varphi_i = \varphi|_{X_i} (i = 1, \ldots, n)$. This observation allows us to use Algorithm 1 almost verbatim to solve the Pseudo-polynomial Interpolation Problem.

4. An Example

We illustrate the construction of the version space outlined in the previous section on a simple example. Although this is only a "toy" example, it might be helpful to give a concrete interpretation. Assume that we are evaluating patients using four different medical tests, and we would like to decide the course of action to follow based on the results of these tests. Our setup is the following:

- $L = \{0, a, 1\}$, where 0 means "no action to take", a means "no treatment is necessary, but the patient should be monitored" and 1 means "start treatment". We take the natural ordering $0 < a < 1$ on L.
- $X = \{\mathtt{P}, \mathtt{S}, \mathtt{I}, \mathtt{N}\}$, where $\mathtt{P}$ means "the test is positive", $\mathtt{S}$ means "only slight abnormality is detected", $\mathtt{I}$ means "the test is inconclusive" and $\mathtt{N}$ means "the test is negative". We do not need an order structure on X, however, it seems natural to consider the following partial order: $\mathtt{N} < \mathtt{S}, \mathtt{I} < \mathtt{P}$.
- $C = \{(\mathtt{S}, \mathtt{P}, \mathtt{N}, \mathtt{S}),\ (\mathtt{P}, \mathtt{N}, \mathtt{S}, \mathtt{S}),\ (\mathtt{N}, \mathtt{P}, \mathtt{I}, \mathtt{I}),\ (\mathtt{S}, \mathtt{I}, \mathtt{P}, \mathtt{N}),\ (\mathtt{I}, \mathtt{I}, \mathtt{P}, \mathtt{S})\}$, and the function $f: C \to L$ is given by

$$\begin{aligned} f(\mathtt{S}, \mathtt{P}, \mathtt{N}, \mathtt{S}) &= 0, \quad & f(\mathtt{S}, \mathtt{I}, \mathtt{P}, \mathtt{N}) &= 1, \\ f(\mathtt{P}, \mathtt{N}, \mathtt{S}, \mathtt{S}) &= a, \quad & f(\mathtt{I}, \mathtt{I}, \mathtt{P}, \mathtt{S}) &= 1, \\ f(\mathtt{N}, \mathtt{P}, \mathtt{I}, \mathtt{I}) &= a. \end{aligned}$$

The lattice L can be embedded into the power set of a two-element set $\Omega = \{\omega_1, \omega_2\}$, hence we have $B = \mathcal{P}(\Omega)$, and we regard the elements of L as subsets of Ω:

$$0 = \emptyset, \quad a = \{\omega_1\}, \quad 1 = \{\omega_1, \omega_2\}.$$

Note that $B = \{0, a, a', 1\}$, where $a' = \{\omega_2\}$. One can interpret ω_1 as "monitor patient" and ω_2 as "start treatment". Then a' would mean "start treatment without monitoring the patient", which is naturally excluded from the set of possible options.

Let us compute (some of) the sets S_{ω_1} that satisfy the conditions of Theorem 6. Starting with $S_{\omega_1} = \emptyset$, we obtain $\tilde{\mathcal{E}}_{\omega_1} = \big\{\{\mathtt{P}, \mathtt{S}, \mathtt{I}, \mathtt{N}\}, \{\mathtt{P}, \mathtt{I}, \mathtt{N}\}, \{\mathtt{P}, \mathtt{S}, \mathtt{N}\}, \{\mathtt{P}, \mathtt{S}, \mathtt{I}\}\big\}$ by (9). The hypergraph $\tilde{\mathcal{H}}_{\omega_1} = \big(X, \tilde{\mathcal{E}}_{\omega_1}\big)$ has 4 minimal transversals, namely $\{\mathtt{P}\}, \{\mathtt{S}, \mathtt{I}\}, \{\mathtt{S}, \mathtt{N}\}, \{\mathtt{I}, \mathtt{N}\}$. Any subset of X containing one of these sets is a transversal; there are altogether 12 transversals, and we should examine each one of them in order to find all solutions. This is rather tedious, hence we give the details only for the minimal transversals.

Setting $S_{\omega_1} = \{\mathtt{P}\}$, we obtain $\tilde{\mathcal{E}}_{\omega_1} = \big\{\{\mathtt{I}, \mathtt{N}\}\big\}$, hence we must add either $\mathtt{I}$ or $\mathtt{N}$ to S_{ω_1}. In the former case we get $\tilde{\mathcal{E}}_{\omega_1} = \emptyset$, which yields the solution $S_{\omega_1} = \{\mathtt{P}, \mathtt{I}\}$. In the latter case we have $S_{\omega_1} = \{\mathtt{P}, \mathtt{N}\}$ and $\tilde{\mathcal{E}}_{\omega_1} = \big\{\{\mathtt{S}, \mathtt{I}\}\big\}$, hence one of $\mathtt{S}$ and $\mathtt{I}$ must be added to S_{ω_1}. The case $S_{\omega_1} = \{\mathtt{P}, \mathtt{S}, \mathtt{N}\}$ gives $\tilde{\mathcal{E}}_{\omega_1} = \{\emptyset\}$, and the corresponding hypergraph has no transversals. The case $S_{\omega_1} = \{\mathtt{P}, \mathtt{I}, \mathtt{N}\}$ gives $\tilde{\mathcal{E}}_{\omega_1} = \emptyset$, and this means that there are no edges that need to be covered, i.e., $S_{\omega_1} = \{\mathtt{P}, \mathtt{I}, \mathtt{N}\}$ is a transversal of $\mathcal{H}_{\omega}$. The rest of the computation is shown on Fig. 2. Note that if we

- $S_{\omega_1} = \emptyset,\ \tilde{\mathcal{E}}_{\omega_1} = \{\{\mathtt{P},\mathtt{I},\mathtt{N}\},\{\mathtt{P},\mathtt{S},\mathtt{N}\},\{\mathtt{P},\mathtt{S},\mathtt{I}\},\{\mathtt{P},\mathtt{S},\mathtt{I},\mathtt{N}\}\}$
  - $S_{\omega_1} = \{\mathtt{P}\},\ \tilde{\mathcal{E}}_{\omega_1} = \{\{\mathtt{I},\mathtt{N}\}\}$
    - $S_{\omega_1} = \{\mathtt{P},\mathtt{I}\},\ \tilde{\mathcal{E}}_{\omega_1} = \emptyset \longrightarrow$ **output** $\{\mathtt{P},\mathtt{I}\}$
    - $S_{\omega_1} = \{\mathtt{P},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \{\{\mathtt{S},\mathtt{I}\}\}$
      - $S_{\omega_1} = \{\mathtt{P},\mathtt{S},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \{\emptyset\} \longrightarrow$ **output fail**
      - $S_{\omega_1} = \{\mathtt{P},\mathtt{I},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \emptyset \longrightarrow$ **output** $\{\mathtt{P},\mathtt{I},\mathtt{N}\}$
  - $S_{\omega_1} = \{\mathtt{S},\mathtt{I}\},\ \tilde{\mathcal{E}}_{\omega_1} = \emptyset \longrightarrow$ **output** $\{\mathtt{S},\mathtt{I}\}$
  - $S_{\omega_1} = \{\mathtt{S},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \{\{\mathtt{P}\},\{\mathtt{I}\}\}$
    - $S_{\omega_1} = \{\mathtt{P},\mathtt{S},\mathtt{I},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \{\emptyset\} \longrightarrow$ **output fail**
  - $S_{\omega_1} = \{\mathtt{I},\mathtt{N}\},\ \tilde{\mathcal{E}}_{\omega_1} = \emptyset \longrightarrow$ **output** $\{\mathtt{I},\mathtt{N}\}$

Fig. 2. Computing S_{ω_1}.

had started with $S_{\omega_1} = \{\mathtt{S},\mathtt{N}\}$ instead of $S_{\omega_1} = \{\mathtt{P}\}$ at the beginning, then we would have gotten no solutions. This illustrates that one must search the whole tree of possibilities in order to guarantee that a solution will be found if there is one.

Figure 3 shows the computations for S_{ω_2}, again only working with minimal transversals. Taking into account non-minimal transversals as well, one obtains all possible sets S_{ω_1} and S_{ω_2}:

$$S_{\omega_1} : \{\mathtt{P},\mathtt{I}\},\ \{\mathtt{S},\mathtt{I}\},\ \{\mathtt{I},\mathtt{N}\},\ \{\mathtt{P},\mathtt{S},\mathtt{I}\},\ \{\mathtt{P},\mathtt{I},\mathtt{N}\};$$
$$S_{\omega_2} : \{\mathtt{P}\},\ \{\mathtt{I}\},\ \{\mathtt{S},\mathtt{I}\},\ \{\mathtt{I},\mathtt{N}\},\ \{\mathtt{P},\mathtt{S},\mathtt{I}\},\ \{\mathtt{S},\mathtt{I},\mathtt{N}\}.$$

- $S_{\omega_2} = \emptyset,\ \tilde{\mathcal{E}}_{\omega_2} = \{\{\mathtt{P},\mathtt{S},\mathtt{I}\},\{\mathtt{P},\mathtt{S},\mathtt{I},\mathtt{N}\}\}$
  - $S_{\omega_2} = \{\mathtt{P}\},\ \tilde{\mathcal{E}}_{\omega_2} = \emptyset \longrightarrow$ **output** $\{\mathtt{P}\}$
  - $S_{\omega_2} = \{\mathtt{S}\},\ \tilde{\mathcal{E}}_{\omega_2} = \{\{\mathtt{I}\},\{\mathtt{P},\mathtt{I}\}\}$
    - $S_{\omega_2} = \{\mathtt{S},\mathtt{I}\},\ \tilde{\mathcal{E}}_{\omega_2} = \emptyset \longrightarrow$ **output** $\{\mathtt{S},\mathtt{I}\}$
  - $S_{\omega_2} = \{\mathtt{I}\},\ \tilde{\mathcal{E}}_{\omega_2} = \emptyset \longrightarrow$ **output** $\{\mathtt{I}\}$

Fig. 3. Computing S_{ω_2}.

There are 30 possibilties for the systems of sets $S_\omega (\omega \in \Omega)$, hence there are 30 maps $\varphi : X \to B$ for which an interpolating polynomial exists over B. However, if there is an element $u \in S_{\omega_2} \backslash S_{\omega_1}$, then $\varphi(u) = a' \notin L$. Therefore, it sufficies to consider the cases where $S_{\omega_2} \subseteq S_{\omega_1}$, giving 13 local utility functions $\varphi : X \to L$.

If we consider the partial ordering $\mathtt{N} < \mathtt{S}, \mathtt{I} < \mathtt{P}$ on X and we look only for order-preserving maps φ, then we have only 3 possibilities. We give the corresponding polynomial functions p_φ^- and p_φ^+ only for these cases (to facilitate readability we omit the symbol $\wedge$ and use juxtaposition instead):

- $S_{\omega_1} = \{\mathtt{P}, \mathtt{S}, \mathtt{I}\}$, $S_{\omega_2} = \{\mathtt{P}, \mathtt{S}, \mathtt{I}\}$: In this case we have

$$\varphi(\mathtt{P}) = 1, \quad \varphi(\mathtt{S}) = 1, \quad \varphi(\mathtt{I}) = 1, \quad \varphi(\mathtt{N}) = 0;$$

$$p_\varphi^- = y_1 y_2 y_3 \vee a y_1 y_3 y_4 \vee a y_2 y_3 y_4, \quad p_\varphi^+ = a y_3 \vee y_1 y_2 y_3.$$

- $S_{\omega_1} = \{\mathtt{P}, \mathtt{I}\}$, $S_{\omega_2} = \{\mathtt{P}\}$: In this case we have

$$\varphi(\mathtt{P}) = 1, \quad \varphi(\mathtt{S}) = 0, \quad \varphi(\mathtt{I}) = a, \quad \varphi(\mathtt{N}) = 0;$$

$$p_\varphi^- = a y_1 \vee a' y_3 \vee y_1 y_3 \vee y_2 y_3, \quad p_\varphi^+ = a y_1 \vee y_3 \vee y_4 \vee y_1 y_2.$$

  Here p_φ^- involves a' as a coefficient, hence it is not a polynomial over L. The least polynomial p over L satisfying $p_\varphi^- \leq p$ is obtained by replacing a' by 1:

$$p = a y_1 \vee 1 y_3 \vee y_1 y_3 \vee y_2 y_3 = a y_1 \vee y_3.$$

  Probably this is the simplest polynomial over L that lies between p_φ^- and p_φ^+; the corresponding quasi-polynomial $U(\mathbf{x}) = a\varphi(x_1) \vee \varphi(x_3)$ depends only on x_1 and x_3, which shows that the first and the third tests are sufficient in order to choose the action to take.
- $S_{\omega_1} = \{\mathtt{P}, \mathtt{S}, \mathtt{I}\}$, $S_{\omega_2} = \{\mathtt{P}\}$: In this case we have

$$\varphi(\mathtt{P}) = 1, \quad \varphi(\mathtt{S}) = a, \quad \varphi(\mathtt{I}) = a, \quad \varphi(\mathtt{N}) = 0;$$

$$p_\varphi^- = a' y_3 \vee y_1 y_2 y_3 \vee y_1 y_3 y_4 \vee y_2 y_3 y_4, \quad p_\varphi^+ = y_3 \vee a' y_4 \vee a' y_1 y_2.$$

  Again a' appears in the polynomials; we need to replce it by 1 in p_φ^- and by 0 in p_φ^+ to all find polynomials p over L such that $p_\varphi^- \leq p \leq p_\varphi^+$. After simplification, we get the polynomial y_3 in both cases. This means that for this local utility function the interpolating quasi-polynomial is unique: $U(\mathbf{x}) = \varphi(x_3)$; revealing the fact that the third test alone can determine the recommended action to take.

5. Complexity of Quasi-Polynomial Interpolation

In Sec. 3, we gave an algorithm that constructs all quasi-polynomial functions interpolating a given partial function $f : C \to L$ $(C \subseteq X^n)$. We noticed that even if one looks for only one interpolating quasi-polynomial, the algorithm still involves finding minimal transversals in hypergraphs, which is an NP-complete problem [14]. In this section we prove that this difficulty is not avoidable, as already for $n = 4$, it is an NP-complete problem to decide whether an interpolating quasi-polynomial exists. However, as we shall see, for $n \leq 3$ this problem can be solved in polynomial time. For background on complexity theory we refer the reader to [14].

We introduce and study the following two decision problems. We call the first one the n-ary *existential pseudo-polynomial interpolation problem* for a finite distributive lattice L with $|L| \geq 2$.

Problem Pseudo(n, L)**:** Given finite sets $X_1, \ldots, X_n$, a subset $C \subseteq X_1 \times \cdots \times X_n$ and a partial function $f : C \to L$, decide whether there exist an n-ary lattice polynomial operation p on L and maps

$$\varphi_1 : X_1 \to L, \ldots, \varphi_n : X_n \to L$$

such that

$$p(\varphi_1(x_1), \ldots, \varphi_n(x_n)) = f(x_1, \ldots, x_n)$$

for all $(x_1, \ldots, x_n) \in C$.

The corresponding n-ary *existential quasi-polynomial interpolation problem* for L can be formulated as follows.

Problem Quasi(n, L)**:** Given a finite set X, a subset $C \subseteq X^n$ and a partial function $f : C \to L$, decide whether there exist an n-ary lattice polynomial operation p on L and a map $\varphi : X \to L$ such that

$$p(\varphi(x_1), \ldots, \varphi(x_n)) = f(x_1, \ldots, x_n)$$

for all $(x_1, \ldots, x_n) \in C$.

We note that both problems are in the complexity class NP. Indeed, for problem **Pseudo**(n, L),

$$p, \varphi_1, \ldots, \varphi_n$$

is a linear size certificate that witnesses all of the equalities

$$f(x_1, \ldots, x_n) = p(\varphi_1(x_1), \ldots, \varphi_n(x_n)), \quad (x_1, \ldots, x_n) \in C,$$

in polynomial time. One obtains similarly that **Quasi**(n, L) is in NP. We also note that by Remark 9, it follows immediately that **Pseudo**(n, L) has a polynomial time reduction to **Quasi**(n, L).

Our aim is to determine the complexity of **Quasi**(n, L) for each n. First let us observe that it is sufficient to consider the case where L is the two-element lattice. Indeed, if L is any finite distributive lattice, then, as before, we embed L into a power set $\mathcal{P}(\Omega)$ of a finite set Ω, and consider the elements $\omega \in \Omega$ separately, as we did in Sec. 3.2. In this way we can translate **Quasi**(n, L) to $|\Omega|$ many problems with two-element lattices $\mathcal{P}(\{\omega\})$. By Theorem 6, a "global" solution exists if and only if each one of these "local" problems has a solution. Therefore, in the sequel we will always assume that $L = \{0, 1\}$.

We will examine the complexity of our interpolation problem with the help of certain constraint satisfaction problems that are related to upsets in the Boolean lattice $L^n = \{0,1\}^n$. We say that a subset $\alpha \subseteq L^n$ is an *upset* (order filter) if for all $\mathbf{a}_1, \mathbf{a}_2 \in L^n$, $\mathbf{a}_1 \in \alpha$ and $\mathbf{a}_1 \leq \mathbf{a}_2$ (in the componentwise ordering) imply $\mathbf{a}_2 \in \alpha$. We will denote the complement of α by β, i.e., $\beta = L^n \backslash \alpha$. Observe that β is a *downset* (order ideal): $\mathbf{b}_1 \in \beta$ and $\mathbf{b}_1 \geq \mathbf{b}_2$ imply $\mathbf{b}_2 \in \beta$ for all $\mathbf{b}_1, \mathbf{b}_2 \in L^n$. For every upset $\alpha \subseteq L^n$ we define a problem $\mathbf{P}(\alpha)$ as follows.

Problem $\mathbf{P}(\alpha)$. Given a finite set V of variables and sets of n-tuples $A, B \subseteq V^n$, find an assignment $\psi : V \to L$ such that $\psi(\mathbf{a}) \in \alpha$ for all $\mathbf{a} \in A$ and $\psi(\mathbf{b}) \in \beta = L^n \backslash \alpha$ for all $\mathbf{b} \in B$.

Note that $\mathbf{P}(\alpha)$ is a Boolean constraint satisfaction problem, hence, by Schaefer's dichotomy theorem for Boolean CSP, it is either in P or it is NP-complete [24].

Lemma 10. *Let $L = \{0,1\}$ be the two-element lattice, let X be a finite set and consider a function $f : C \to L$, where $C \subseteq X^n$. There exists a quasi-polynomial function interpolating f if and only if $\mathbf{P}(\alpha)$ has a solution for some upset $\alpha \subseteq L^n$ with $V = X$ and*

$$A = \{\mathbf{a} \in C : f(\mathbf{a}) = 1\}, \quad B = \{\mathbf{b} \in C : f(\mathbf{b}) = 0\}.$$

Proof. Let us apply Theorem 6 to the lattice $L = \{0,1\}$. In this case we have $\Omega = \{\omega\}$, and the elements 0 and 1 of L are represented by the sets $\emptyset$ and $\{\omega\}$. A map $\varphi : X \to L$ can be given by a single set $S_\omega = \{a \in X : \varphi(a) = 1\}$, and the hyperedges of the hypergraph $\mathcal{H}_\omega$ are of the form $E_\omega(\mathbf{a}, \mathbf{b}) = \{a_i : b_i \notin S_\omega\} = \{a_i : \varphi(b_i) = 0\}$ for $\mathbf{a}, \mathbf{b} \in C$ with $\omega \in f(\mathbf{a}) \backslash f(\mathbf{b})$. Since both $f(\mathbf{a})$ and $f(\mathbf{b})$ are either $\emptyset$ or $\{\omega\}$, the condition $\omega \in f(\mathbf{a}) \backslash f(\mathbf{b})$ is satisfied if and only if $f(\mathbf{a}) = \{\omega\} = 1$ and $f(\mathbf{b}) = \emptyset = 0$. Thus we have

$$\mathcal{E}_\omega = \{\{a_i : \varphi(b_i) = 0\} : \mathbf{a}, \mathbf{b} \in C \text{ such that } f(\mathbf{a}) = 1 \text{ and } f(\mathbf{b}) = 0\}.$$

If $v \in E_\omega(\mathbf{a}, \mathbf{b}) \cap S_\omega$, then $v = a_i$ with $\varphi(b_i) = 0$ (since $v \in E_\omega(\mathbf{a}, \mathbf{b})$) and $\varphi(a_i) = 1$ (since $v \in S_\omega$). Therefore, the intersection $E_\omega(\mathbf{a}, \mathbf{b}) \cap S_\omega$ is nonempty iff there is an $i \in [n]$ such that $\varphi(a_i) = 1$ and $\varphi(b_i) = 0$. Note that the latter condition means that $\varphi(\mathbf{a}) \nleq \varphi(\mathbf{b})$ in the componentwise ordering of n-tuples over $L = \{0,1\}$. (We use the shorthand notation $\varphi(\mathbf{a}) = (\varphi(a_1), \ldots, \varphi(a_n))$.) We conclude that S_ω is a transversal of $\mathcal{H}_\omega$ if and only if

$$\forall \mathbf{a}, \mathbf{b} \in C : \ (f(\mathbf{a}) = 1 \text{ and } f(\mathbf{b}) = 0) \Rightarrow \varphi(\mathbf{a}) \nleq \varphi(\mathbf{b}), \tag{10}$$

and by Theorem 6, this is equivalent to the existence of an interpolating quasi-polynomial function with the local utility function φ.

(Note that the implication in (10) can be reformulated as $\varphi(\mathbf{a}) \leq \varphi(\mathbf{b}) \Rightarrow f(\mathbf{a}) \leq f(\mathbf{b})$. This gives an alternative way of proving that (10) is equivalent to the existence of a polynomial p such that $f(\mathbf{c}) = p(\varphi(\mathbf{c}))$ for all $\mathbf{c} \in C$, since lattice polynomial functions coincide with nondecreasing functions over the two-element lattice.)

Assume that φ satisfies (10), and let α be the least upset containing $\varphi(\mathbf{a})$ for all $\mathbf{a} \in A$:

$$\alpha := \{\mathbf{y} \in L^n : \mathbf{y} \geq \varphi(\mathbf{a}) \text{ for some } \mathbf{a} \in A\}.$$

Obviously, we have $\varphi(\mathbf{a}) \in \alpha$ for all $\mathbf{a} \in A$, and (10) implies that $\varphi(\mathbf{b}) \notin \alpha$ for all $\mathbf{b} \in B$. Thus, φ is a solution of the problem $\mathbf{P}(\alpha)$ with X being the set of variables.

Conversely, if $\alpha \subseteq L^n$ is an arbitrary upset and φ is a solution of $\mathbf{P}(\alpha)$, then it is immediate that φ satisfies (10). □

According to Lemma 10, we can split $\mathbf{Quasi}(n, L)$ into finitely many subproblems $\mathbf{P}(\alpha)$ with α running through the set of upsets of L^n. If each of these subproblems can be solved in polynomial time, then the whole problem is in P. As the next theorem shows, this is the case for $n \leq 3$.

Theorem 11. *If $n \leq 3$, then $\mathbf{Quasi}(n, L)$, and hence $\mathbf{Pseudo}(n, L)$, belongs to the complexity class* P*.*

Proof. Clearly, it suffices to prove the theorem for $n = 3$. By Lemma 10, we only need to show that $\mathbf{P}(\alpha)$ is in P for every upset $\alpha \subseteq L^3$. Up to permutations of variables, we have the 8 cases listed below. For each upset α we give a polymorphism h of the constraint language $\{\alpha, \beta\}$ that shows that $\mathbf{P}(\alpha)$ belongs to P by Schaefer's dichotomy theorem. (For better readability we write elements of L^3 as words.)

$$\begin{array}{ll}
\alpha = \{111\} & h = x \wedge y \\
\alpha = \{101, 111\} & h = x \wedge y \\
\alpha = \{101, 110, 111\} & h = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \\
\alpha = \{100, 101, 110, 111\} & h = x \wedge y \\
\alpha = \{011, 101, 110, 111\} & h = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \\
\alpha = \{011, 100, 101, 110, 111\} & h = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \\
\alpha = \{010, 011, 100, 101, 110, 111\} & h = x \vee y \\
\alpha = \{001, 010, 011, 100, 101, 110, 111\} & h = x \vee y
\end{array}$$

□

For $n \geq 4$ one can find upsets $\alpha \subseteq L^n$ such that $\mathbf{P}(\alpha)$ is NP-complete. This does not yield immediately NP-completeness of the interpolation problem, since there might be "easy" solutions corresponding to some other upsets. Nevertheless, in the next theorem we prove that $\mathbf{Quasi}(n, L)$ is indeed NP-complete for $n \geq 4$.

Theorem 12. *If $n \geq 4$, then problem $\mathbf{Quasi}(n, L)$ is* NP*-complete.*

Proof. Clearly, it suffices to prove the theorem for $n = 4$. Let $\alpha \subseteq \{0, 1\}^4$ be the upset consisting of tuples of Hamming weight at least 3, that is, $\alpha := \{0111, 1011, 1101, 1110, 1111\}$. In this case the constraint language $\{\alpha, \beta\}$ admits only projections as polymorphisms, thus $\mathbf{P}(\alpha)$ is NP-complete, by Schaefer's dichotomy theorem.

For every instance of $\mathbf{P}(\alpha)$ we construct an instance of the quasi-polynomial interpolation problem with $L = \{0, 1\}$ and $n = 4$ such that the solutions ψ of the former are in a one-to-one correspondence with the local utility functions φ that solve the latter. So assume that V and $A, B \subseteq V^4$ are given, as in $\mathbf{P}(\alpha)$. Let $X = V \dot{\cup} \{0, 1\}$, $C = A \cup B \dot{\cup} \{0, 1\}^4$ (where $\dot{\cup}$ denotes disjoint union) and $f : C \to L^4$ such that

$$\forall \mathbf{a} \in A \cup \alpha : f(\mathbf{a}) = 1 \quad \text{and} \quad \forall \mathbf{b} \in B \cup \beta : f(\mathbf{b}) = 0.$$

(Note that 0 and 1 belong to both X and L, hence they play the role of "variables" as well as the role of "values".) We claim that a map $\varphi : X \to L$ satisfies (10), which, as we have seen in Lemma 10, is equivalent to the existence of an interpolating quasi-polynomial function, if and only if $\varphi(0) = 0$ and $\varphi(1) = 1$, and the restriction $\psi := \varphi|_V$ of φ to V is a solution of $\mathbf{P}(\alpha)$.

First suppose that φ satisfies (10). This immediately implies that $\varphi(\mathbf{a}) \nleq \varphi(\mathbf{b})$ for all $\mathbf{a} \in \alpha$ and $\mathbf{b} \in \beta$, and it easy to see that this holds if and only if $\varphi(0) = 0$ and $\varphi(1) = 1$. Now applying (10) with $\mathbf{a} \in A$, $\mathbf{b} \in \beta$, we get $\varphi(\mathbf{a}) \nleq \varphi(\mathbf{b}) = \mathbf{b}$; in particular, $\varphi(\mathbf{a}) \neq \mathbf{b}$. Since this holds for all $\mathbf{b} \in \beta$, we have that $\varphi(\mathbf{a}) \notin \beta$, i.e., $\varphi(\mathbf{a}) \in \alpha$. A similar argument shows that $\varphi(\mathbf{b}) \in \beta$ for all $\mathbf{b} \in B$, and this proves that $\psi = \varphi|_V$ is indeed a solution to $\mathbf{P}(\alpha)$.

Next assume that ψ is a solution of $\mathbf{P}(\alpha)$, and let $\varphi : X \to L$ coincide with ψ on V, and let $\varphi(0) = 0$, $\varphi(1) = 1$. Then we have $\varphi(\mathbf{a}) \in \alpha$ for all $\mathbf{a} \in A \cup \alpha$ (if $\mathbf{a} \in A$ then by the constraints of $\mathbf{P}(\alpha)$, if $\mathbf{a} \in \alpha$ then by the fact that $\varphi(\mathbf{a}) = \mathbf{a}$), and similarly, $\varphi(\mathbf{b}) \in \beta$ for all $\mathbf{b} \in B \cup \beta$. Therefore, if $f(\mathbf{a}) = 1$ and $f(\mathbf{b}) = 0$, then $\varphi(\mathbf{a}) \in \alpha$ and $\varphi(\mathbf{b}) \in \beta$, and this implies that $\varphi(\mathbf{a}) \nleq \varphi(\mathbf{b})$, hence (10) holds.

This proves that $\mathbf{P}(\alpha)$ reduces to $\mathbf{Quasi}(n, L)$ in polynomial time, showing that the latter problem is also NP-complete. □

Summarizing Theorems 11 and 12, we obtain the following dichotomy result.

Corollary 13. *If $n \leq 3$ then the problem of deciding the existence of an interpolating quasi-polynomial function is in* P, *whereas for $n \geq 4$ it is* NP*-complete.*

Remark 14. In order to determine the complexity of $\mathbf{Pseudo}(n, L)$, one could study the following analogue of problem $\mathbf{P}(\alpha)$ (here $\alpha \subseteq \{0, 1\}^n$ is an upset, as before):

Given a finite set V of variables partitioned into n parts $V = V_1 \dot{\cup} \cdots \dot{\cup} V_n$ and sets of n-tuples $A, B \subseteq V_1 \times \cdots \times V_n$, find an assignment $\psi : V \to L$ such that $\psi(\mathbf{a}) \in \alpha$ for all $\mathbf{a} \in A$ and $\psi(\mathbf{b}) \in \beta = L^n \backslash \alpha$ for all $\mathbf{b} \in B$.

The difference compared to $\mathbf{P}(\alpha)$ is that the set V is partitioned into n disjoint sets, and in the constraints $\mathbf{a} \in A$ and $\mathbf{b} \in B$ the i-th coordinate must come from V_i. Thus the above problem is a subproblem of $\mathbf{P}(\alpha)$, where we have some restriction on the structure of the sets $A, B \subseteq V^n$. Further research is needed to decide whether this restriction makes the problem easier. Thus, the question if $\mathbf{Pseudo}(n, L)$ is NP-complete remains open for $n \geq 4$ and $|L| \geq 2$.

6. Concluding Remarks and Future Work

In this paper, we considered the problem of interpolating empirical data given as couples consisting of a tuple specified by several attributes, together with its evaluation in a distributive lattice. The interpolating objects are lattice-valued functions, called quasi- and pseudo-polynomial functions, that can be factorized into a composition of a lattice polynomial function with possibly different local utility functions that evaluate each attribute in a distributive lattice. We presented necessary and sufficient conditions for the existence of quasi- and pseudo-polynomial functions interpolating a given finite set of examples. In doing so, we actually presented explicit descriptions of such solutions when they exist. Looking into complexity issues in computing them, we established a dichotomy result stating that, up to 3 attributes, the existence of an interpolationg quasi-polynomial function can be decided in polynomial time, whereas this problem for sets of examples over more than 3 attributes becomes NP-complete. The analogous complexity question for pseudo-polynomial functions remains open.

Now our framework was motivated by problems typically arising in the qualitative approach to multicriteria decision making. The basic aggregation functions considered, namely, lattice polynomial functions (that include Sugeno integrals), have neat representations, e.g., by disjunctive normal forms, and played a key role in the constructions provided. Other noteworthy aggregation functions in decision making, such as Lovász extensions (that include Choquet integrals), also share similar representation features. The natural step is to make use of them when considering analogous interpolation problems for these aggregation models.

Furthermore, simplified notions of Sugeno and Choquet integrals (parametrized versions arising from the notions of k-maxitivity and k-additivity; see [16] for a general reference) have been proposed in the literature and could provide alternatives to avoid intractable complexity classes when dealing with interpolation problems.

These constitute few topics of our current interest, and that will be tackled in forthcoming research work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their suggestions that helped improving the manuscript. Research supported by the Hungarian National Foundation for Scientific Research (grants no. K104251 and K115518), by the János Bolyai Research Scholarship and by the Ministry of Human Capacities 20391-3/2018/FEKUSTRAT.

References

[1] D. Bouyssou, T. Marchant and M. Pirlot, A conjoint measurement approach to the discrete Sugeno integral, in *The Mathematics of Preference, Choice and Order*. Essays in Honor of Peter C. Fishburn, Brams, S., Gehrlein, W. V., Roberts, F. S. (eds.) (Springer, 2009), pp. 85–109.

[2] K. Cao-Van, B. De Baets and S. Lievens, A probabilistic framework for the design of instance-based supervised ranking algorithms in an ordinal setting, *Annals Operations Research* **163** (2008) 115–142.
[3] A. Cornuéjols and L. Miclet, *Apprentissage Artificiel — Concepts et Algorithmes* (Eyrolles, 2010).
[4] M. Couceiro, D. Dubois, H. Prade and T. Waldhauser, Decision-making with Sugeno integrals. Bridging the gap between multicriteria evaluation and decision under uncertainty, *Order* **33**(3) (2016) 517–535.
[5] M. Couceiro, D. Dubois, H. Prade, A. Rico and T. Waldhauser, General interpolation by polynomial functions of distributive lattices. Information Processing and Management of Uncertainty in Knowledge-Based Systems, *Communications in Computer and Information Science*, Vol. 299 (Springer-Verlag, 2012), pp. 347–355.
[6] M. Couceiro and J.-L. Marichal, Characterizations of discrete Sugeno integrals as polynomial functions over distributive lattices, *Fuzzy Sets and Systems* **161**(5) (2010) 694–707.
[7] M. Couceiro and J.-L. Marichal, Axiomatizations of quasi-polynomial functions on bounded chains, *Aequationes Mathematicae* **78**(1) (2009) 195–213.
[8] M. Couceiro and J.-L. Marichal, Quasi-polynomial functions over bounded distributive lattices, *Aequationes Mathematicae* **80** (2010) 319–334.
[9] M. Couceiro and T. Waldhauser, Axiomatizations and factorizations of Sugeno utility functions, *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems* **19**(4) (2011) 635–658.
[10] M. Couceiro and T. Waldhauser, Pseudo-polynomial functions over finite distributive lattices, *Fuzzy Sets and Systems* **239** (2014) 21–34.
[11] M. Couceiro and T. Waldhauser, Interpolation by polynomial functions of distributive lattices: A generalization of a theorem of R. L. Goodstein, *Algebra Universalis* **69**(3) (2013) 287–299.
[12] B. A. Davey and H. Priestley, *Introduction to Lattices and Order* (Cambridge University Press, New York, 2002).
[13] J. Fürnkranz, E. Hüllermeier (eds.), *Preference Learning* (Springer, Berlin, 2011).
[14] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences (W. H. Freeman and Co., San Francisco, CA, 1979).
[15] R. L. Goodstein, The solution of equations in a lattice, *Proc. Roy. Soc. Edinburgh* Section A **67** (1965/1967) 231–242.
[16] M. Grabisch, J.-L. Marichal, R. Mesiar and E. Pap, *Aggregation Functions*, Encyclopedia of Mathematics and Its Applications, Vol. 127 (Cambridge University Press, Cambridge, 2009).
[17] S. Greco, B. Matarazzo and R. Słowiński, Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules, *European Journal of Operational Research* **158** (2004) 271–292.
[18] J.-L. Marichal, On Sugeno integral as an aggregation function, *Fuzzy Sets and Systems* **114** (2000) 347–365.
[19] J.-L. Marichal, Weighted lattice polynomials, *Discrete Mathematics* **309**(4) (2009) 814–820.
[20] T. Mitchell, *Machine Learning* (McGraw-Hill, 1997).
[21] H. Prade, A. Rico and M. Serrurier, Elicitation of Sugeno Integrals: A version space learning perspective, *Proc. 18th Inter. Symp. on Methodologies for Intelligent Systems* (*ISMIS'09*), eds. J. Rauch, Z. W. Ras, P. Berka and T. Elomaa, Prague, LNCS, Vol. 5722 (Springer, 2009), pp. 392–401.

[22] H. Prade, A. Rico, M. Serrurier and E. Raufaste, Eliciting Sugeno integrals: Methodology and a case study, in *Proc. European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'09* (2009), pp. 712–723.

[23] A. Rico, M. Grabisch, Ch. Labreuche and A. Chateauneuf, Preference modeling on totally ordered sets by the Sugeno integral, *Discrete Applied Math.* **147**(1) (2005) 113–124.

[24] T. J. Schaefer, The complexity of satisfiability problems, *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing* (San Diego, Calif., 1978), pp. 216–226 (ACM, New York, 1978).

[25] M. Sugeno, *Theory of Fuzzy Integrals and Its Applications*, PhD thesis (Tokyo Institute of Technology, Tokyo, 1974).

[26] M. Sugeno, Fuzzy measures and fuzzy integrals — A survey, in *Fuzzy Automata and Decision Processes*, eds. M. M. Gupta, G. N. Saridis and B. R. Gaines (North-Holland, New York, 1977), pp. 89–102.

[27] A. F. Tehrani, W. Cheng and E. Hüllermeier, Preference learning using the choquet integral: The case of multipartite ranking, *IEEE Transactions on Fuzzy Systems* **20**(6) (2012) 1102–1113.