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 vėrtanak tere 1, H-6720 Szeged, Hungary
†mmaroti@math.u-szeged.hu
‡waldha@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 \( f : X \rightarrow L \), where \( X \) is a set of objects (or alternatives) and \( L \) is a set of labels: \textit{Given a finite } \( S \subseteq X \times L \), \textit{decide whether there exists an } \( f : X \rightarrow L \) \textit{interpolating } \( S \), \textit{i.e., such that } \( f(a) = b \) \textit{for every } \((a, b) \in S\). Our motivation is

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

As the starting point, we take the decomposable model to represent preferences over a set $X = X_1 \times \cdots \times X_n$ 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” $U : X \to L$ valued in a scale $L$ (also called the “evaluation space”) using the following rule:

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

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 $U$:

$$U(x) = A(\varphi_1(x_1), \ldots, \varphi_n(x_n)).$$

(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\cite{25, 26} that can be regarded as an idempotent lattice polynomial function\cite{6, 19}, and the resulting global utility function (1) is then called a pseudo-polynomial function\cite{10} or a Sugeno utility function\cite{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\cite{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 $S \subseteq X \times L$, find all pseudo-polynomial functions $U : X \to L$ that interpolate $S$. In other words, given a data set $S$ consisting of pairs $(a, b)$ 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.,\cite{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\cite{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 \( \land \) and \( \lor \), 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

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

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 \( \Omega \). 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 \( I \subseteq [n] \):

\[
c^+_I := \bigvee_{a \in D} \left( g(a) \land \bigwedge_{i \notin I} a_i' \right) \quad \text{and} \quad c^-_I := \bigwedge_{a \in D} \left( g(a) \lor \bigvee_{i \in I} a_i' \right).
\]
Observe that $I \subseteq J$ implies $c_I^- \leq c_J^- \leq c_J^+ \leq c_I^+$. Let $p^-$ and $p^+$ be the polynomial functions over $B$ given by these two systems of coefficients:

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

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 $g : D \rightarrow L$ be a function defined on a finite set $D \subseteq L^n$, and let $p : B^n \rightarrow B$ 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 $p : B^n \rightarrow B$ 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 $p : L^n \rightarrow L$ 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

![A distributive lattice and its Boolean algebra.](image_url)
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

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

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 $D = \{a, b\}$, where $a = (1, c)$, $b = (c, a)$ and consider $g : D \to L_5$ defined by

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

As coefficients $c_I^-$ and $c_I^+$ we obtain

$$c_0^- = 0, \quad c_{(1)}^- = c', \quad c_{(2)}^- = 0, \quad c_{(1,2)}^- = 1,$$

$$c_0^+ = a, \quad c_{(1)}^+ = b', \quad c_{(2)}^+ = 1, \quad c_{(1,2)}^+ = 1.$$

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:

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

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

$$U(x) = p(\varphi_1(x_1), \ldots, \varphi_n(x_n)), \quad (5)$$

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 $f : C \rightarrow L$, find all pseudo-polynomial functions $U : X_1 \times \cdots \times X_n \rightarrow L$ 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 utility functions $U : X^n \rightarrow L$ are so-called quasi-polynomial functions:

$$U(x) = p(\varphi(x_1), \ldots, \varphi(x_n)).$$

(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 $f : C \rightarrow L$, find all quasi-polynomial functions $U : X^n \rightarrow L$ 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 $\varphi_i : X_i \rightarrow L$ 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 $a \in C$, i.e., if $p$ interpolates the function $g : D \rightarrow L$ defined on the set

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

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:

$$c_{I, \varphi_1, \ldots, \varphi_n}^- := \bigvee_{a \in C} \left( f(a) \wedge \bigwedge_{i \notin I} \varphi_i(a_i) \right) \quad \text{and} \quad c_{I, \varphi_1, \ldots, \varphi_n}^+ := \bigwedge_{a \in C} \left( f(a) \vee \bigvee_{i \in I} \varphi_i(a_i) \right).$$
Denoting the corresponding polynomial functions by $p^\pm_{\varphi_1,\ldots,\varphi_n}$ and $p^\pm_{\varphi_1,\ldots,\varphi_n}$, Theorem 4 yields the following solution for the Pseudo-polynomial Interpolation Problem with known local utility functions.

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

1. $U = p(\varphi_1,\ldots,\varphi_n)$ interpolates $f$, i.e., $U|_C = f$;
2. $c^+_{I,\varphi_1,\ldots,\varphi_n} \leq c^+_{I,\varphi_1,\ldots,\varphi_n}$ for all $I \subseteq [n]$;
3. $p^{+}_{\varphi_1,\ldots,\varphi_n} \leq p^{+}_{\varphi_1,\ldots,\varphi_n}$.

**Remark 5.** Note that if there exist tuples $a,b \in C$ such that $f(a) \neq f(b)$ but $\ell(f(\varphi_1(a_1),\ldots,\varphi_n(a_n))) = (\varphi_1(b_1),\ldots,\varphi_n(b_n))$, 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

$$U(x) = p(\varphi(x_1),\ldots,\varphi(x_n)),$$

where the local utility function $\varphi : X \rightarrow L$ is not known. Our aim is to find all possible maps $\varphi$ 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:

$$\bigvee_{a \in C} \left( f(a) \land \bigwedge_{i \not\in I} \varphi(a_i) \right) \leq \bigwedge_{b \in C} \left( f(b) \lor \bigvee_{i \in I} \varphi(b_i) \right) \quad \text{for all } I \subseteq [n]. \quad (7)$$

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 $\Omega$. We will encode a map $\varphi : X \rightarrow B$ 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 $\varphi$ as $\varphi(a) = \{\omega \in \Omega : a \in S_\omega\}$.) For each $\omega \in \Omega$, we define a hypergraph $H_\omega = (X; E_\omega)$ with vertex set $X$ and edge set

$$E_\omega = \{E_\omega(a,b) : a,b \in C \text{ such that } \omega \in f(a) \setminus f(b)\},$$

where $E_\omega(a,b) = \{a_i : b_i \notin S_\omega\} \subseteq X$. (Here $f(a) \setminus f(b) = f(a) \land f(b)'$ is the difference of the sets $f(a), f(b) \subseteq \Omega$.) In the next theorem we will prove that (7) holds if and only if $S_\omega$ is a transversal (also called a hitting set or a vertex cover) of the hypergraph $H_\omega$, i.e., $S_\omega$ intersects every edge of $H_\omega$ 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 $a, b \in C$, $\omega \in f(a) \setminus f(b)$ 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) $S_\omega$ is a transversal of $H_\omega$ 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 right hand side is less than or equal to each meetand on the left hand side is less than or equal to each meetand on the right hand side:

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

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

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

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

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

Now (ii) is just a reformulation of the condition above using the definition of the sets $S_\omega$.

Next we prove that (ii) implies (iii). Let $E_\omega(a, b) = \{a_i : b_i \notin S_\omega\}$ be any edge of $H_\omega$, 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 $a_i \in E_\omega(a, b)$. This proves that $a_i \in E_\omega(a, b) \cap S_\omega$ for every edge $E_\omega(a, b)$ of the hypergraph $H_\omega$, hence $S_\omega$ is a transversal of $H_\omega$.

Finally, let us show that (iii) implies (ii). Let $b, a \in C$, $\omega \in f(a) \setminus f(b)$ and $I \subseteq [n]$, and assume that $\{b_i : i \in I\} \subseteq S_\omega$. Since $S_\omega$ is a transversal of $H_\omega$, there exists a vertex $v \in E_\omega(a, b) \cap S_\omega$. From $v \in E_\omega(a, b)$ 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 $\varphi$ that provide a quasi-polynomial function interpolating $f$ on $C$. We start with $S_\omega = \emptyset$, and at every iteration we extend $S_\omega$ if necessary to obtain a transversal of $H_\omega$. It suffices
Algorithm 1 Constructing all sets $S_\omega$ that satisfy the conditions of Theorem 6

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

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

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

Remark 7. In order to make sure that we find all solutions, we must try every transversal of $\tilde{H}_\omega$ 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 $H$, 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 $\varphi (\omega) = \{ \omega \in \Omega : a \in S_\omega \}$. After computing all such maps $\varphi$, 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 $\varphi$. 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

$$f(a,a) = 0,$$
$$f(0,1) = f(1,0) = a,$$
$$f(1,1) = 1.$$ 

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

$$c_{0,\varphi} = c_{0,\varphi}^+ = 0,$$
$$c_{1,\varphi} = c_{1,\varphi}^+ = c_{1,\varphi}^- = c_{1,\varphi}^+ = a,$$
$$c_{1,\varphi}^+ = c_{1,\varphi}^+ = 1.$$ 

Hence, $p = p_{\varphi} = p_{\varphi}^+ = p_{\varphi}^-$, 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 $\varphi$. 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 $\tilde{U} : X^n \to L$ defined by $\tilde{U}(x) = p(\varphi (x_1), \ldots , \varphi (x_n))$. Observe that $X_1 \times \cdots \times X_n \subseteq X^n$ and the restriction of $\tilde{U}$ 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 \( L \).
- \( X = \{P, S, I, N\} \), where \( P \) means “the test is positive”, \( S \) means “only slight abnormality is detected”, \( I \) means “the test is inconclusive” and \( N \) means “the test is negative”. We do not need an order structure on \( X \), however, it seems natural to consider the following partial order: \( N < S < I < P \).
- \( C = \{(S, P, N, S), (P, N, S, S), (N, P, I, I), (S, I, P, N), (I, I, P, S)\} \), and the function \( f : C \rightarrow L \) is given by

\[
\begin{align*}
  f(S, P, N, S) &= 0, \quad f(S, I, P, N) = 1, \\
  f(P, N, S, S) &= a, \quad f(I, I, P, S) = 1, \\
  f(N, P, I, I) &= a.
\end{align*}
\]

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 \( \Omega \):

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

Note that \( B = \{0, a, a', 1\} \), where \( a' = \{\omega_2\} \). One can interpret \( \omega_1 \) as “monitor patient” and \( \omega_2 \) 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 \( S_{\omega_i} \) that satisfy the conditions of Theorem 6. Starting with \( S_{\omega_1} = \emptyset \), we obtain \( \mathcal{E}_{\omega_1} = \{\{P, S, I, N\}, \{P, I, N\}, \{P, S, N\}, \{P, S, I\}\} \) by (3). The hypergraph \( \mathcal{H}_{\omega_1} = (X, \mathcal{E}_{\omega_1}) \) has 4 minimal transversals, namely \( \{P\}, \{S, I\}, \{S, N\}, \{I, N\} \). 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 \( S_{\omega_1} = \{P\} \), we obtain \( \mathcal{E}_{\omega_1} = \{\{I, N\}\} \), hence we must add either \( I \) or \( N \) to \( S_{\omega_1} \). In the former case we get \( \mathcal{E}_{\omega_1} = \emptyset \), which yields the solution \( S_{\omega_1} = \{P, I\} \). In the latter case we have \( S_{\omega_1} = \{P, N\} \) and \( \mathcal{E}_{\omega_1} = \{\{S, I\}\} \), hence one of \( S \) and \( I \) must be added to \( S_{\omega_1} \). The case \( S_{\omega_1} = \{P, S, N\} \) gives \( \mathcal{E}_{\omega_1} = \emptyset \), and the corresponding hypergraph has no transversals. The case \( S_{\omega_1} = \{P, I, N\} \) gives \( \mathcal{E}_{\omega_1} = \emptyset \), and this means that there are no edges that need to be covered, i.e., \( S_{\omega_1} = \{P, I, N\} \) is a transversal of \( \mathcal{H}_{\omega_1} \). The rest of the computation is shown on Fig. 2. Note that if we
had started with $S_{ω1} = \{S, N\}$ instead of $S_{ω1} = \{P\}$ 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 2 shows the computations for $S_{ω1}$, again only working with minimal transversals. Taking into account non-minimal transversals as well, one obtains all possible sets $S_{ω1}$ and $S_{ω2}$:

$S_{ω1} : \{P, I\}, \{S, I\}, \{I, N\}, \{P, S, I\}, \{P, I, N\}$;

$S_{ω2} : \{P\}, \{I\}, \{S, I\}, \{I, N\}, \{P, S, I\}, \{S, I, N\}$.

Fig. 2. Computing $S_{ω1}$.

---

$S_{ω1} = \emptyset$, $\bar{E}_{ω1} = \{\{P, I, N\}, \{P, S, N\}, \{P, S, I\}, \{P, S, I, N\}\}$


Fig. 3. Computing $S_{ω2}$. 

$S_{ω1} = \emptyset$, $\bar{E}_{ω1} = \{\{P, S, I, N\}\}$


$S_{ω1} = \emptyset$, $\bar{E}_{ω1} = \emptyset → output \{P\}$


$S_{ω1} = \emptyset$, $\bar{E}_{ω1} = \emptyset → output \{I\}$
There are 30 possibilities for the systems of sets $S_ω(ω ∈ Ω)$, hence there are 30 maps $ϕ : X → B$ for which an interpolating polynomial exists over $B$. However, if there is an element $u ∈ S_ω \setminus S_ω^*$, then $ϕ(u) = a^* \notin L$. Therefore, it suffices to consider the cases where $S_ω^* ≤ S_ω$, giving 13 local utility functions $ϕ : X → L$.

If we consider the partial ordering $n < S, I < P$ 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 ∧ and use juxtaposition instead):

- $S_ω = \{P, S, I\}, S_ω^* = \{P, S, I\}$: In this case we have
  \[
  ϕ(P) = 1, \quad ϕ(S) = 1, \quad ϕ(I) = 1, \quad ϕ(N) = 0;
  \]
  \[
  p^-_ϕ = y_1y_2y_3 ∨ ay_1y_3y_4 ∨ ay_2y_3y_4, \quad p^+_ϕ = ay_3 ∨ y_1y_2y_3.
  \]

- $S_ω = \{P, I\}, S_ω^* = \{P\}$: In this case we have
  \[
  ϕ(P) = 1, \quad ϕ(S) = 0, \quad ϕ(I) = a, \quad ϕ(N) = 0;
  \]
  \[
  p^-_ϕ = ay_1 ∨ a^*y_3 ∨ y_1y_3 ∨ y_2y_3, \quad p^+_ϕ = ay_1 ∨ y_3 ∨ y_4 ∨ y_1y_2.
  \]

Here $p^-_ϕ$ involves $a^*$ as a coefficient, hence it is not a polynomial over $L$. The least polynomial $p$ over $L$ satisfying $p^-_ϕ ≤ p$ is obtained by replacing $a^*$ by 1:

\[
\begin{align*}
  p &= ay_1 ∨ 1y_3 ∨ y_1y_3 ∨ y_2y_3 = ay_1 ∨ y_3.
\end{align*}
\]

Probably this is the simplest polynomial over $L$ that lies between $p^-_ϕ$ and $p^+_ϕ$; the corresponding quasi-polynomial $U(x) = aϕ(x_1) ∨ ϕ(x_3)$ 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.

- $S_ω = \{P, S, I\}, S_ω^* = \{P\}$: In this case we have
  \[
  ϕ(P) = 1, \quad ϕ(S) = a, \quad ϕ(I) = a, \quad ϕ(N) = 0;
  \]
  \[
  p^-_ϕ = a^*y_3 ∨ y_1y_2y_3 ∨ y_1y_3y_4 ∨ y_2y_3y_4, \quad p^+_ϕ = y_3 ∨ a^*y_4 ∨ a^*y_1y_2.
  \]

Again $a^*$ appears in the polynomials; we need to replace it by 1 in $p^-_ϕ$ and by 0 in $p^+_ϕ$ to all find polynomials $p$ over $L$ such that $p^-_ϕ ≤ p ≤ p^+_ϕ$. 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: $U(x) = ϕ(x_3)$; 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 $f : C → L (C ⊆ X^n)$. 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 $f : C \rightarrow L$, decide whether there exist an $n$-ary lattice polynomial operation $p$ on $L$ and maps

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

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 $f : C \rightarrow L$, decide whether there exist an $n$-ary lattice polynomial operation $p$ on $L$ and a map $\varphi : X \rightarrow L$ 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 $\Omega$, 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 \( a_1, a_2 \in L^n, a_1 \in \alpha \) and \( a_1 \leq a_2 \) (in the componentwise ordering) imply \( a_2 \in \alpha \). We will denote the complement of \( \alpha \) by \( \beta \), i.e., \( \beta = L^n \setminus \alpha \). Observe that \( \beta \) is a downset (order ideal): \( b_1 \in \beta \) and \( b_1 \geq b_2 \) imply \( b_2 \in \beta \) for all \( b_1, b_2 \in L^n \). For every upset \( \alpha \subseteq L^n \) we define a problem \( P(\alpha) \) as follows.

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

Note that \( P(\alpha) \) 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 \( f : C \rightarrow L \), where \( C \subseteq X^n \). There exists a quasi-polynomial function interpolating \( f \) if and only if \( P(\alpha) \) has a solution for some upset \( \alpha \subseteq L^n \) with \( V = X \) and

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

**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 \( \varphi : X \rightarrow L \) can be given by a single set \( S_\omega = \{a \in X : \varphi(a) = 1\} \), and the hyperedges of the hypergraph \( H_\omega \) are of the form \( E_\omega(a, b) = \{a_i : b_i \not\in S_\omega\} = \{a_i : \varphi(b_i) = 0\} \) for \( a, b \in C \) with \( \omega \in f(a) \setminus f(b) \). Since both \( f(a) \) and \( f(b) \) are either \( \emptyset \) or \( \{\omega\} \), the condition \( \omega \in f(a) \setminus f(b) \) is satisfied if and only if \( f(a) = \{\omega\} = 1 \) and \( f(b) = \emptyset = 0 \). Thus we have

\[
E_\omega = \{\{a_i : \varphi(b_i) = 0\} : a, b \in C \text{ such that } f(a) = 1 \text{ and } f(b) = 0\}.
\]

If \( v \in E_\omega(a, b) \cap S_\omega \), then \( v = a_i \) with \( \varphi(b_i) = 0 \) (since \( v \in E_\omega(a, b) \)) and \( \varphi(a_i) = 1 \) (since \( v \in S_\omega \)). Therefore, the intersection \( E_\omega(a, b) \cap S_\omega \) is nonempty if there is an \( i \in [n] \) such that \( \varphi(a_i) = 1 \) and \( \varphi(b_i) = 0 \). Note that the latter condition means that \( \varphi(a) \not\leq \varphi(b) \) in the componentwise ordering of \( n \)-tuples over \( L = \{0, 1\} \). (We use the shorthand notation \( \varphi(a) = (\varphi(a_1), \ldots, \varphi(a_n)) \).) We conclude that \( S_\omega \) is a transversal of \( H_\omega \) if and only if

\[
\forall a, b \in C : f(a) = 1 \text{ and } f(b) = 0 \Rightarrow \varphi(a) \not\leq \varphi(b), \quad (10)
\]

and by Theorem 6 this is equivalent to the existence of an interpolating quasi-polynomial function with the local utility function \( \varphi \).

(Note that the implication in 10 can be reformulated as \( \varphi(a) \leq \varphi(b) \Rightarrow f(a) \leq f(b) \). This gives an alternative way of proving that 10 is equivalent to the existence of a polynomial \( p \) such that \( f(c) = p(\varphi(c)) \) for all \( c \in C \), since lattice polynomial functions coincide with nondecreasing functions over the two-element lattice.)
Assume that $\varphi$ satisfies (10), and let $\alpha$ be the least upset containing $\varphi(a)$ for all $a \in A$:

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

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

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

According to Lemma 10 we can split Quasi$(n, L)$ into finitely many subproblems $P(\alpha)$ with $\alpha$ 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 Quasi$(n, L)$, and hence 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 $P(\alpha)$ 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 $\alpha$ we give a polymorphism $h$ of the constraint language $\{\alpha, \beta\}$ that shows that $P(\alpha)$ belongs to P by Schaefer’s dichotomy theorem. (For better readability we write elements of $L^3$ as words.)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${111}$</td>
<td>$h = x \land y$</td>
</tr>
<tr>
<td>${101, 111}$</td>
<td>$h = x \land y$</td>
</tr>
<tr>
<td>${101, 110, 111}$</td>
<td>$h = (x \land y) \lor (x \land z) \lor (y \land z)$</td>
</tr>
<tr>
<td>${100, 101, 110, 111}$</td>
<td>$h = x \land y$</td>
</tr>
<tr>
<td>${011, 101, 110, 111}$</td>
<td>$h = (x \land y) \lor (x \land z) \lor (y \land z)$</td>
</tr>
<tr>
<td>${011, 100, 101, 110, 111}$</td>
<td>$h = (x \land y) \lor (x \land z) \lor (y \land z)$</td>
</tr>
<tr>
<td>${010, 011, 100, 101, 110, 111}$</td>
<td>$h = x \lor y$</td>
</tr>
<tr>
<td>${001, 010, 011, 100, 101, 110, 111}$</td>
<td>$h = x \lor y$</td>
</tr>
</tbody>
</table>

For $n \geq 4$ one can find upsets $\alpha \subseteq L^n$ such that $P(\alpha)$ 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 Quasi$(n, L)$ is indeed NP-complete for $n \geq 4$.

**Theorem 12.** If $n \geq 4$, then problem 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 $P(\alpha)$ is NP-complete, by Schaefer’s dichotomy theorem.
For every instance of $P(\alpha)$ we construct an instance of the quasi-polynomial interpolation problem with $L = \{0,1\}$ and $n = 4$ such that the solutions $\psi$ of the former are in a one-to-one correspondence with the local utility functions $\varphi$ that solve the latter. So assume that $V$ and $A, B \subseteq V^4$ are given, as in $P(\alpha)$. Let $X = V \cup \{0,1\}, C = A \cup B \cup \{0,1\}^4$ (where $\cup$ denotes disjoint union) and $f : C \to L^4$ such that

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

(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 $\varphi$ to $V$ is a solution of $P(\alpha)$.

First suppose that $\varphi$ satisfies \(^{[10]}\). This immediately implies that $\varphi(a) \not\in \varphi(b)$ for all $a \in \alpha$ and $b \in \beta$, and it easy to see that this holds if and only if $\varphi(0) = 0$ and $\varphi(1) = 1$. Now applying \(^{[10]}\) with $a \in A$, $b \in \beta$, we get $\varphi(a) \not\in \varphi(b) = b$; in particular, $\varphi(a) \not\in b$. Since this holds for all $b \in \beta$, we have that $\varphi(a) \not\in \beta$, i.e., $\varphi(a) \in \alpha$. A similar argument shows that $\varphi(b) \in \beta$ for all $b \in B$, and this proves that $\psi = \varphi|_V$ is indeed a solution to $P(\alpha)$.

Next assume that $\psi$ is a solution of $P(\alpha)$, and let $\varphi : X \to L$ coincide with $\psi$ on $V$, and let $\varphi(0) = 0$, $\varphi(1) = 1$. Then we have $\varphi(a) \in \alpha$ for all $a \in A \cup \alpha$ (if $a \in A$ then by the constraints of $P(\alpha)$, if $a \in \alpha$ then by the fact that $\varphi(a) = a$), and similarly, $\varphi(b) \in \beta$ for all $b \in B \cup \beta$. Therefore, if $f(a) = 1$ and $f(b) = 0$, then $\varphi(a) \in \alpha$ and $\varphi(b) \in \beta$, and this implies that $\varphi(a) \not\in \varphi(b)$, hence \(^{[10]}\) holds.

This proves that $P(\alpha)$ reduces to 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 Pseudo($n, L$), one could study the following analogue of problem $P(\alpha)$ (here $\alpha \subseteq \{0,1\}^n$ is an upset, as before):

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

The difference compared to $P(\alpha)$ is that the set $V$ is partitioned into $n$ disjoint sets, and in the constraints $a \in A$ and $b \in B$ the $i$-th coordinate must come from $V_i$. Thus the above problem is a subproblem of $P(\alpha)$, 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 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 interpolating 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


