Available online at www.sciencedirect.com

ScienceDirect

Discrete Applied Mathematics 156 (2008) 492–499

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Quasi-concave functions on meet-semilattices

Yulia Kempner[a], Ilya Muchnik[b]

[a]*Department of Computer Science, Holon Academic Institute of Technology, Holon, 58102, Israel*
[b]*Department of Computer Science, Rutgers University, Piscataway, NJ 08854-8018, USA*

Received 16 May 2004; received in revised form 31 October 2004; accepted 20 December 2006
Available online 27 September 2007

Abstract

This paper deals with maximization of set functions defined as minimum values of monotone linkage functions. In previous research, it has been shown that such a set function can be maximized by a greedy type algorithm over a family of all subsets of a finite set. In this paper, we extend this finding to meet-semilattices.

We show that the class of functions defined as minimum values of monotone linkage functions coincides with the class of quasi-concave set functions. Quasi-concave functions determine a chain of upper level sets each of which is a meet-semilattice. This structure allows development of a polynomial algorithm that finds a minimal set on which the value of a quasi-concave function is maximum. One of the critical steps of this algorithm is a set closure. Some examples of closure computation, in particular, a closure operator for convex geometries, are considered.

Keywords: Closure operator; Greedy algorithm; Monotone linkage function; Quasi-concave function; Semilattice

1. Introduction

Many combinatorial optimization problems can be formulated as: for a given *set system* over E (i.e., for a pair $(E, \mathscr{F})$ where $\mathscr{F} \subseteq 2^E$ is a family of feasible subsets of finite set E), and for a given function $F : \mathscr{F} \to \mathbf{R}$, find an element of $\mathscr{F}$ for which the value of the function F is minimum or maximum. In general, this optimization problem is NP-hard, but for some specific functions and for some specific set systems polynomial algorithms are known. The famous examples are modular cost functions that can be optimized over matroids by a polynomial greedy algorithm [6] and bottleneck functions that can be maximized over greedoids [8]. Another example is set functions defined as minimum values of monotone linkage functions. Such a set function can be maximized by a greedy type algorithm over the family of all subsets of E [10,14,19], over antimatroids and convex geometries [9,11,15], and join-semilattices [16]. In this paper, we extend these results to meet-semilattices.

Meet-semilattices are present in many areas of mathematics. For example, we can cite conceptual clustering [7,13] which deals with discovering conceptual structures of data. These structures allow the analysis of dependencies among complex data. In many cases, the data is given as an object-attribute table. Subsets of objects corresponding to all

E-mail address: yuliak@hit.ac.il (Y. Kempner).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2006.12.005

possible combinations of attributes form a meet-semilattice. This phenomenon is currently a subject of interest in conceptual clustering.

Meet-semilattices are also used in investigations of relational databases. One of the most important branches of the theory of relational databases is the design of database schemes based on the theory of functional dependencies. It has been shown [3] that functional dependencies may be represented as a closure semilattice.

This paper is organized as follows. Section 2 gives basic information about monotone linkage functions. We show that on meet-semilattices, the class of functions defined as the minimum values of monotone linkage functions coincides with the class of quasi-concave set functions. Section 3 deals with the construction of an efficient algorithm for maximizing quasi-concave functions which are associated with monotone linkage functions. A quasi-concave function determines in a meet-semilattice a chain of included upper level sets, each of which is itself a meet-semilattice. This structure allows the construction of a polynomial algorithm that finds a minimal set on which the value of a quasi-concave function is maximum. One of the critical steps of this algorithm is a set closure. In Section 4, we consider algorithms of closure construction for a number of different meet-semilattices.

2. Preliminaries

Monotone linkage functions were introduced by Mullat [17]. Here we will give some necessary definitions.

A function $\pi : E \times 2^E \to \mathbf{R}$ is called a *monotone linkage function* if

$$\text{for each } X, Y \subseteq E \text{ and } x \in E, \quad X \subseteq Y \text{ implies } \pi(x, X) \geqslant \pi(x, Y). \tag{1}$$

Consider $F : (2^E - \{E\}) \to \mathbf{R}$ defined for each $X \subset E$

$$F(X) = \min_{x \in E-X} \pi(x, X). \tag{2}$$

The function F satisfies the following condition:

$$\text{for each } X, Y \subset E, \quad F(X \cap Y) \geqslant \min\{F(X), F(Y)\}. \tag{3}$$

A function $F : 2^E \to \mathbf{R}$ satisfying (3) is called a *quasi-concave set function* [12].

These functions were studied in [10,19]. In particular, a simple polynomial algorithm for maximizing quasi-concave functions of the form (2) was developed. This algorithm finds a minimal set $X \subset E$ that maximizes the function F.

In this paper, we extend these results to a meet-semilattice. The family $\mathcal{S} \subseteq 2^E$ is a *meet-semilattice* if $A \cap B \in \mathcal{S}$ for each $A, B \in \mathcal{S}$. Further we use the term "semilattice" to mean "meet-semilattice".

A set function F defined on a semilattice $\mathcal{S}$ is a quasi-concave function if

$$\text{for each } A, B \in \mathcal{S} - \{E\}, \quad F(A \cap B) \geqslant \min\{F(A), F(B)\}. \tag{4}$$

In the following theorem we show that on semilattices, the class of set functions defined as the minimum values of monotone linkage functions coincides with the class of quasi-concave set functions.

Theorem 1. *A set function F defined on a semilattice $\mathcal{S}$ is a quasi-concave function if and only if there exists a monotone linkage function π such that $F(X) = \min_{x \in E-X} \pi(x, X)$ for each $X \in \mathcal{S} - \{E\}$.*

Proof. If a monotone linkage function π is given, then $F(X \cap Y) = \pi(x^*, X \cap Y)$, where

$x^* \in \arg\min_{x \in E-(X \cap Y)} \pi(x, X \cap Y)$.[1]

Without loss of generality, assume that $x^* \in E - X$. Thus,

$$F(X \cap Y) \geqslant \pi(x^*, X) \geqslant F(X) \geqslant \min\{F(X), F(Y)\}.$$

[1] arg min $f(x)$ denote the set of arguments that minimize the function f.

Conversely, if we have a quasi-concave set function F, then we can define the function

$$\pi_F(x, X) = \begin{cases} \max\limits_{A \in [X, E-x]_{\mathscr{S}}} F(A), & x \notin X \text{ and } [X, E-x]_{\mathscr{S}} \neq \varnothing, \\ \min\limits_{A \in \mathscr{S}} F(A) & \text{otherwise}, \end{cases} \tag{5}$$

where $[X, Y]_{\mathscr{S}} = \{Z \in \mathscr{S} : X \subseteq Z \subseteq Y\}$.

The function π_F is monotone. Indeed, if $x \in E - Y$ and $[Y, E - x]_{\mathscr{S}} \neq \varnothing$, then $X \subseteq Y$ implies

$$\pi_F(x, X) = \max_{A \in [X, E-x]_{\mathscr{S}}} F(A) \geqslant \max_{A \in [Y, E-x]_{\mathscr{S}}} F(A) = \pi_F(x, Y).$$

It is easy to verify the remaining cases.

Let us denote $G(X) = \min_{x \in E-X} \pi_F(x, X)$, and prove that $F = G$ on $\mathscr{S} - \{E\}$.

Now, for each $X \in \mathscr{S} - \{E\}$

$$G(X) = \min_{x \in E-X} \pi_F(x, X) = \pi_F(x^*, X) = \max_{A \in [X, E-x^*]_{\mathscr{S}}} F(A) \geqslant F(X),$$

where $x^* \in \arg\min_{x \in E-X} \pi_F(x, X)$.

On the other hand,

$$G(X) = \min_{x \in E-X} \pi_F(x, X) = \min_{x \in E-X} F(A^x),$$

where A^x is a set from $[X, E - x]_{\mathscr{S}}$ on which the value of the function F is maximal, i.e.,

$$A^x \in \arg\max_{A \in [X, E-x]_{\mathscr{S}}} F(A).$$

From quasi-concavity of F it follows that

$$\min_{x \in E-X} F(A^x) \leqslant F\left(\bigcap_{x \in E-X} A^x \right) = F(X).$$

Therefore, $G(X) \leqslant F(X)$, and, hence, $F = G$, i.e., $F(X) = \min_{x \in E-X} \pi_F(x, X)$. □

Consider the following optimization problem: given a semilattice $\mathscr{S} \subseteq 2^E$ and a monotone linkage function $\pi : E \times 2^E \to \mathbf{R}$, find a feasible set $A \in \mathscr{S} - \{E\}$, such that $F(A) = \max\{F(B) : B \in \mathscr{S} - \{E\}\}$, where the function F is defined by (2). From quasi-concavity it follows that the set of optimal solutions is also a semilattice with a unique minimal element. Our goal in this paper is to find this minimal element, which we call the *minimal maximizer*.

3. Minimal maximizers of quasi-concave functions

Consider the following operator:

$$\tau(X) = \begin{cases} \cap\{A \mid X \subseteq A, A \in \mathscr{S}\}, & [X, E]_{\mathscr{S}} \neq \varnothing, \\ E & \text{otherwise}. \end{cases}$$

If $\mathscr{S}$ is a semilattice, $\tau(X)$ is the smallest set in $\mathscr{S}$ containing X (if such a set exists). In other words, $\tau(X)$ is the null of the semilattice $[X, E]_{\mathscr{S}}$ if the semilattice is not empty, and it is defined as E otherwise.

Note that a semilattice $\mathscr{S}$ should not have a maximal element, and we use the element E only for the definition of the operator τ.

It is straightforward to check that the operator τ has the properties of a closure operator:

(i) $X \subseteq \tau(X)$;
(ii) $\tau(X) = \tau(\tau(X))$;
(iii) $X \subseteq Y \Rightarrow \tau(X) \subseteq \tau(Y)$.

So, $\tau(X)$ is *a closure* of X.

We assume that a procedure for finding a closure $\tau(X)$ for each $X \subset E$ is available. Later we will consider some examples of efficient closure constructors.

To find the minimal maximizer of a quasi-concave function F, consider initially the special structure that function F determines on a semilattice. It has been already noted that the family of feasible sets maximizing function F is a semilattice with a unique minimal element. Denote this family by $\mathscr{T}^0$, and let a^0 be the value of function F on the sets from $\mathscr{T}^0$. We denote by $\mathscr{T}^1$ the family of sets which maximize function F over $\mathscr{S} - \mathscr{T}^0$, and by a^1 the value of function F on these sets. Continuing this process, we have $\mathscr{S} = \bigcup_{i=0}^{t} \mathscr{T}^i$, where $t+1$ is a number of different values of function F. It is easy to see that $\mathscr{L}_j = \bigcup_{i=0}^{j} \mathscr{T}^i$ is a subsemilattice of $\mathscr{S}$, where $\mathscr{L}_j = \{X \in \mathscr{S} - \{E\} : F(X) \geqslant a^j\}$. We call these subsemilattices *upper level semilattices*. Denote by K^j the minimal element (null) of the upper level semilattice $\mathscr{L}_j$. Since $\mathscr{L}_i \subseteq \mathscr{L}_{i+1}$, we obtain $K^0 \supseteq K^1 \supseteq \cdots \supseteq K^t$, where K^t is the null of the semilattice $\mathscr{S}$.

Let $K^0 = H^0 \supset H^1 \supset \cdots \supset H^r = K^t$ be the subchain of all different nulls of the chain $K^0 \supseteq K^1 \supseteq \cdots \supseteq K^t$. Thus, to find a minimal maximizer, we have to find a null H^0. In fact, we construct an algorithm that finds the complete chain $H^0 \supset H^1 \supset \cdots \supset H^r$ of different nulls. This chain of "local maximizers"[2] has a number of interesting applications [14].

We define for any real number u the *u-level set* of a family $\mathscr{F}$ as

$$\mathscr{F}_u = \{X \in \mathscr{F} : F(X) > u\}.$$

It is obvious that the u-level set of some semilattice is also a semilattice when F is quasi-concave. The following is an algorithm, which for a given threshold u and for a given set $X \subset E$ returns the null of non-empty $(\mathscr{S} \cap [X, E])_u$. The algorithm is motivated by the main procedure from [18].

The Level-Set Algorithm (u, X).
1. Set $A = \tau(X)$
3. While $A \neq E$ do
    3.1 Set $I_u(A) = \{x \in E - A : \pi(x, A) \leqslant u\}$
    3.2 If $I_u(A) = \varnothing$ then stop and return A
    3.3 Set $A = \tau(A \cup I_u(A))$
4. Return A.

Theorem 2. *Let $\mathscr{S} \subseteq 2^E$ be a semilattice. Then, for every monotone linkage function π and the corresponding function $F(X) = \min_{x \in E-X} \pi(x, X)$ the Level-Set Algorithm (u, X) returns the null of non-empty semilattice $(\mathscr{S} \cap [X, E])_u$ and returns E when this u-level set is empty.*

Proof. At first, consider the case when the algorithm returns $A \neq E$. Since $I_u(A) = \varnothing$, $F(A) > u$, i.e., $A \in (\mathscr{S} \cap [X, E])_u$. It remains to prove that A is the null of the u-level set, i.e., that $B \in (\mathscr{S} \cap [X, E])_u$ implies $A \subseteq B$. Suppose the opposite was true, and let $\tau(X) = X_0 \subset X_1 \subset \cdots \subset X_k = A$ be a sequence of sets generated by the algorithm, where $X_{i+1} = \tau(X_i \cup I_u(X_i))$ for $0 \leqslant i < k$. Since $B \in (\mathscr{S} \cap [X, E])_u$, we have $X \subseteq B$, and thus $X_0 = \tau(X) \subseteq \tau(B) = B$. On the other hand, since $A \nsubseteq B$, there exists the least integer j for which $X_j \nsubseteq B$. Then $X_{j-1} \subseteq B$, and there exists $x_j \in I_u(X_{j-1})$ that is not in B. For if not $I_u(X_{j-1}) \subseteq B$, which implies $X_{j-1} \cup I_u(X_{j-1}) \subseteq B$, and then (from the closure operator properties) $X_j = \tau(X_{j-1} \cup I_u(X_{j-1})) \subseteq \tau(B) = B$, contradicting $X_j \nsubseteq B$. Hence

$$F(B) \leqslant \pi(x_j, B) \leqslant \pi(x_j, X_{j-1}) \leqslant u,$$

a contradiction.

[2] Indeed, for each $A \in \mathscr{S} - E$, and for each null H^l, if $A \nsupseteq H^l$ then $F(A) < F(H^l)$.

If the algorithm returns $A = E$, then $(\mathscr{S} \cap [X, E])_u = \emptyset$. Assuming the opposite, then there is a non-empty set $B \in (\mathscr{S} \cap [X, E])_u$ and $B \subset A$. Similarly to the first part of the proof, we obtain that $F(B) \leqslant u$, a contradiction. □

Note that for an empty u-level set the Level-Set Algorithm returns E rather than $\emptyset$. We do it for more convenient use of this algorithm in the following Chain Algorithm. The Chain Algorithm finds the chain of all local maximizers for a non-empty semilattice $\mathscr{S}$.

The Chain Algorithm $(E, \mathscr{S})$.
1. Set $\Gamma_0 = \tau(\emptyset)$
2. $i = 0$
3. While $\Gamma_i \neq E$ do
    3.1 $u = F(\Gamma_i)$
    3.2 Γ_{i+1}=Level-Set(u, Γ_i)
    3.3 $i = i + 1$
4. Return the chain $\Gamma_0 \subset \Gamma_1 \subset \cdots \subset \Gamma_{i-1}$.

Theorem 3. *Let $\mathscr{S} \subseteq 2^E$ be a non-empty semilattice. Then, for every monotone linkage function π and the corresponding function $F(X) = \min_{x \in E-X} \pi(x, X)$, the Chain Algorithm returns the chain $\Gamma_0 \subset \Gamma_1 \subset \cdots \subset \Gamma_p$, which coincides with $H^0 \supset H^1 \supset \cdots \supset H^r$—the chain of all different nulls of the upper level semilattices.*

Proof. First, prove that for each $l = 0, 1, \ldots, p$, Γ_l is the null of some upper level semilattice. It is clear that if $F(\Gamma_l) = a^j$, then $\Gamma_l \in L_j$. To prove that Γ_l is the null of L_j, we have to show that for each $A \in \mathscr{S} - \{E\}$, $A \not\supseteq \Gamma_l$ implies $F(A) < F(\Gamma_l)$. Suppose that the opposite is true, and let k be the least integer for which there exists $A \in \mathscr{S} - \{E\}$, such that $A \not\supseteq \Gamma_k$ and $F(A) \geqslant F(\Gamma_k)$. Note that $k > 0$, because $\Gamma_0 = \tau(\emptyset)$ is the null of semilattice $\mathscr{S}$, and so $A \not\supseteq \Gamma_0$ never holds. The structure of the Chain Algorithm implies $F(\Gamma_k) > F(\Gamma_{k-1})$. Hence $F(A) > F(\Gamma_{k-1})$ and, consequently, $A \supseteq \Gamma_{k-1}$. Thus $A \in (\mathscr{S} \cap [\Gamma_{k-1}, E])_u$, where $u = F(\Gamma_{k-1})$. On the other hand, from the Chain Algorithm construction it follows that Γ_k is the null of $(\mathscr{S} \cap [\Gamma_{k-1}, E])_u$, i.e., $A \supseteq \Gamma_k$, a contradiction.

It remains to show that for each null H^i there exists $l \in \{0, 1, \ldots, p\}$ such that $\Gamma_l = H^i$. Assume the opposite, and let H^j be a minimal (by inclusion) null for which the statement is not correct. Since $H^r = \tau(\emptyset) = \Gamma_0$, it follows that $j < r$, i.e., there exists $l \in \{0, 1, \ldots, p\}$ such that $H^{j+1} = \Gamma_l$. From $F(H^j) > F(H^{j+1}) = F(\Gamma_l)$ and $H^j \supset H^{j+1} = \Gamma_l$, it follows that $H^j \in (\mathscr{S} \cap [\Gamma_l, E])_u$, where $u = F(\Gamma_l)$. Thus $H^j \supseteq \Gamma_{l+1}$, where Γ_{l+1} is the null of $(\mathscr{S} \cap [\Gamma_l, E])_u$. On the other hand, since Γ_{l+1} is the null of some upper level semilattice and H^j is the closest null to H^{j+1}, this implies that $\Gamma_{l+1} \supseteq H^j \supset H^{j+1} = \Gamma_l$. Hence $H^j = \Gamma_{l+1}$, a contradiction. □

Corollary 4. *Let $\mathscr{S} \subseteq 2^E$ be a non-empty semilattice. Then, for every monotone linkage function π, the Chain Algorithm finds a minimal maximizer of the quasi-concave function $F(X) = \min_{x \in E-X} \pi(x, X)$.*

Consider the complexity of the Chain Algorithm. In the worst case in each step the algorithm finds only one element with the minimal value of the function π, i.e., $|I_u(A)| = 1$. Then the Chain Algorithm finds the minimal maximizer in $O(|E|(T + P|E|))$ time, where T is the maximum complexity of closure (τ) computation, and P is the maximum complexity of π computation. In some clustering problems [10], the complexity of π computation is $O(|E|)$, and so the complexity of the algorithm is $O(|E|(T + |E|^2))$.

4. Algorithms for closure construction

The efficiency of the closure construction depends on the representation of a semilattice. In this section we consider two forms. In the first case, a semilattice is represented by a set of irreducible elements; the intersection of these elements determines all elements of the semilattice. In the second case, a semilattice is specified by a quasi-concave function.

In addition to these two cases, we consider a convex geometry that is a specific case of a semilattice. Two algorithms of closure construction for a convex geometry—depending of its representation—are also presented.

1. Set of irreducible elements: An element $X \in \mathscr{S}$ is called *(meet)-irreducible* if $X = Y \cap Z$, where $Y, Z \in \mathscr{S}$, implies $X = Y$ or $X = Z$. The set of all irreducible elements is denoted by $M(\mathscr{S})$. Every element of $\mathscr{S}$ is an intersection of elements from $M(\mathcal{S})$. Since $\tau(X) = \cap\{A \mid X \subseteq A, A \in \mathcal{S}\}$ (for $[X, E]_{\mathcal{S}} \neq \emptyset$) and each $A \in \mathcal{S}$ is an intersection of sets from $M(\mathcal{S})$, the definition of $\tau(X)$ may be changed to $\tau(X) = \cap\{A \mid X \subseteq A, A \in M(\mathcal{S})\}$. Hence, the following algorithm finds for each set X its closure:

`Closure` $(X, M(\mathcal{S}))$
Assume $M(\mathcal{S}) = \{A_1, \ldots, A_m\}$
1. $A = E$
2. for $i = 1$ to m do
   2.1 if $X \subseteq A_i$ then $A = A \cap A_i$
3. Return A.

Since the step 2.1 needs $O(|E|)$ time, the algorithm finds the closure of X in $O(m|E|)$ time.

This algorithm may be applied to conceptual clustering [7,13]. In many cases the data are represented as a boolean matrix (data table). Every row corresponds to an object and every column to an attribute. These attributes, in turn, extract their own specific subsets of objects which we call *single-attribute sets.* Considering a cluster as a specific subset of objects which share a commonality of attributes, we focus on those semilattices whose elements are subsets of objects which correspond to all possible combinations of attributes. That is, our object of focus is a semilattice where the set of all irreducible elements corresponds exactly to the family of all single-attribute sets.

2. *Inequality constraint*: Assume that the set Ω of feasible solutions is determined by the following inequality constraint: for each $H \in \Omega$, $\widehat{F}(H) > \alpha$, where $\widehat{F}$ is defined by a monotone linkage function $\widehat{\pi}$.

It is easy to see that the set Ω is a α-level set of the family of all subsets E, i.e., $\Omega = \{X \subset E : \widehat{F}(X) > \alpha\}$. Since $\widehat{F}$ is a quasi-concave function, the set Ω is a semilattice. The problem is to find closure $\tau(X)$ for some set $X \subset E$ over Ω, i.e., to find the null of non-empty semilattice $\Omega \cap [X, E]$. Note that the Level-Set Algorithm(α, X) enables us to find the null of non-empty semilattice $(2^E \cap [X, E])_\alpha$, i.e., $\tau(X)$ over Ω. The modified Level-Set Algorithm is as follows:

`Quasi-concave Closure` (α, X)
1. Set $A = X$
3. While $A \neq E$ do
   3.1 Set $I_\alpha(A) = \{x \in E - A : \widehat{\pi}(x, A) \leqslant \alpha\}$
   3.2 If $I_\alpha(A) = \emptyset$ then stop and return A
   3.3 Set $A = A \cup I_\alpha(A)$
4. Return A.

Let P be the maximum complexity of $\widehat{\pi}$ computation, then the quasi-concave closure algorithm finds the closure $\tau(X)$ over Ω in $O(P|E|^2)$ time. If the complexity of $\widehat{\pi}$ computation is $O(|E|)$, then the complexity of the algorithm is $O(|E|^3)$.

3. *Convex geometry*: A set system $(E, \mathcal{S})$ forms *a convex geometry* if it satisfies the following properties:

(i) if $X \in \mathcal{S}$ and $Y \in \mathcal{S}$, then $X \cap Y \in \mathcal{S}$;
(ii) if $X \in \mathcal{S} - \{E\}$, there exists some $x \in E - X$ such that $X \cup x \in \mathcal{S}$.

Note that by property (ii), $E \in \mathcal{S}$. In original works [4,8] the definition of a convex geometry includes also the requirement that $\emptyset \in \mathcal{S}$, whereas other authors [2] do not involve this property in the definition.

Therefore a convex geometry is a semilattice $\mathcal{S}$ that includes the set E and satisfies property (ii). Hence, the closure operator τ, introduced in the previous section, is defined in a convex geometry. To construct an algorithm for closure detection, consider at first the following proposition:

Proposition 5. *Let $(E, \mathcal{S})$ be a convex geometry, then for each $A, B \in \mathcal{S}$ if $A \subset B$ then there exists some $x \in B - A$ such that $B - x \in \mathcal{S}$.*

Proof. Property (ii) means that we can find a sequence $x_1 x_2 \ldots x_k$ such that $A = X_0 \subset X_1 \subset \cdots \subset X_k = E$, where $X_i = X_{i-1} \cup x_i$, and $X_i \in \mathcal{S}$ for $0 \leqslant i \leqslant k$. Let j be the least integer for which $B \subseteq X_j$. Then $B \not\subseteq X_{j-1}$, and $x_j \in B$. Hence, $B - x_j = B \cap X_{j-1} \in \mathcal{S}$, because $\mathcal{S}$ is a semilattice. On the other hand, $x_j \notin A$, then $x_j \in B - A$. □

Corollary 6. *If $A \subset B$, $A, B \in \mathcal{S}$, then there exists a sequence $x_1x_2\ldots x_l$ that determines a chain $B = X_0 \supset X_1 \supset \cdots \supset X_l = A$, where $X_i = X_{i-1} - x_i$, and $X_i \in \mathcal{S}$ for $0 \leqslant i \leqslant l$.*

So, to find $\tau(X)$, one can build the chain $E = X_0 \supset X_1 \supset \cdots \supset X_m = \tau(X)$.

`Convex Geometry Closure` $(X, \mathcal{S})$
1. $A = E$
2. Find $x \in A - X$, such that $A - x \in \mathcal{S}$
   if no such x exists, then stop and return A
3. Set $A = A - x$ and go to 2.

Let a convex geometry $(E, \mathcal{S})$ be given by a membership oracle which for each set $A \subseteq E$ decides whether $A \in \mathcal{S}$ or not. Then convex geometry closure algorithm finds the closure of a set in at most $k(k+1)/2$ oracle calls, where $k = |E - X|$. Thus the complexity of closure construction is $O(|E|^2\theta)$, where θ is the complexity of the membership oracle.

Consider another way to define convex geometries. Let $P = \{x_1 < x_2 < \cdots < x_n\}$ be a linear order on E. Define

$$D_P = \{X_i : X_i = \{x_1, x_2, \ldots, x_i\}, \quad 1 \leqslant i \leqslant n\} \cup \{\emptyset\}.$$

It is easy to see that (E, D_P) is a convex geometry.

Let $(E, \mathcal{S}_1)$ and $(E, \mathcal{S}_2)$ be two convex geometries. Define

$$\mathcal{S}_1 \vee \mathcal{S}_2 = \{X \cap Y : X \in \mathcal{S}_1, Y \in \mathcal{S}_2\}.$$

Then [4] $(E, \mathcal{S}_1 \vee \mathcal{S}_2)$ is also a convex geometry.

From this point we consider only convex geometries including $\emptyset$, according to the original definition. Hence, each convex geometry may be defined by a set T of linear orders as

$$\mathcal{S} = \bigvee_{P \in T} D_P. \tag{6}$$

The set T is called a convex realizer [5]. Thus, if $\{P_1, P_2, \ldots, P_k\}$ is a convex realizer of $(E, \mathcal{S})$, then each element of $\mathcal{S}$ is a meet of elements in $D_{P_1}, \ldots, D_{P_k}$. Note that each $D_{P_i} \subseteq \mathcal{S}$.

Since each (E, D_{P_i}) is a convex geometry, there are k closure operators τ_{P_i}, where $\tau_P(X) = \{y \in E : y \leqslant_P \max X\}$, i.e., let $P = \{x_1 < x_2 < \cdots < x_n\}$ and a maximal element of X with respect to the order P be $x^* = \max X$, then $\tau_P(X) = \{x_1, x_2, \ldots, x^*\}$.

Proposition 7. $\tau(X) = \cap_{P \in T} \tau_P(X)$.

Proof. Let $A = \cap_{P \in T} \tau_P(X)$. Since for each $P \in T$, $X \subseteq \tau_P(X)$ and $\tau_P(X) \in \mathcal{S}$, it follows that $\tau(X) \subseteq \tau_P(X)$, which implies $\tau(X) \subseteq A$. Conversely, from (6) $\tau(X) = \cap_{P \in T} X_P$, where $X_P \in D_P$. Since $X \subseteq \tau(X)$ implies $X \subseteq X_P$ for all $P \in T$, we have $\tau_P(X) \subseteq X_P$ and so $A \subseteq \tau(X)$. □

Let a convex geometry $(E, \mathcal{S})$ be given by a convex realizer $T = \{P_1, P_2, \ldots, P_k\}$, then the following algorithm builds the closure set using Proposition 7.

`Ordering Closure` $(X, \mathcal{S})$
1. For $i = 1$ to k do
   1.1 build $\tau_{p_i}(X)$
2. Return $\tau(X) = \cap_{i=1}^{k} \tau_{P_i}(X)$.

A straightforward implementation of this algorithm runs in $O(k|E|)$, where k is the cardinality of a convex realizer.

5. Conclusions

In this article, we have investigated monotone linkage functions defined on semilattices. It was shown that the class of functions defined as the minimum values of monotone linkage functions coincides with the class of quasi-concave set functions. Quasi-concave functions determine a chain of upper level sets each of which is a semilattice. This structure allows us to build a polynomial algorithm that finds a minimal set on which the value of a quasi-concave function is maximum.

The critical step of these algorithms is a set closure. If an efficient algorithm of the closure construction exists, it also makes our algorithm efficient. It would be interesting to investigate equivalence of these two problems, i.e., to prove that the existence of a polynomial algorithm for finding an optimal set leads to the existence of a polynomial algorithm for closure construction.

On the other hand, we think that the closure construction problem is interesting enough to be investigated on its own. In this paper, polynomial algorithms for closure construction were presented for two semilattice representations and for a convex geometry that is a particular instance of a semilattice.

An interesting direction for future work is the development of our methods for relational databases, in which the polynomial algorithm for closure construction is known [1].

Acknowledgments

The authors are grateful to anonymous referees for their helpful comments and correction of style.

We would like to thank Vadim E. Levit for his valuable advice and steady interest.

References

[1] C. Beeri, P.A. Bernstein, Computational problems related to the design of normal form relational schemes, ACM Trans. Database Syst. 4 (1) (1979) 30–59.

[2] A. Björner, G.M. Ziegler, Introduction to greedoids, in: N. White (Ed.), Matroid applications, Cambridge University Press, Cambridge, UK, 1992.

[3] J. Demetrovics, L. Libkin, I. Muchnik, Functional dependencies in relational databases: a lattice point of view, Discrete Appl. Math. 40 (1992) 155–185.

[4] P.H. Edelman, R.E. Jamison, The theory of convex geometries, Geom. Dedicata 19 (1985) 247–270.

[5] P.H. Edelman, M.E. Saks, Combinatorial representation and convex dimension of convex geometries, Order 5 (1) (1988) 23–32.

[6] J. Edmonds, Matroid and the greedy algorithm, Math. Programming 1 (1971) 127–136.

[7] B. Ganter, R. Wille, Formal Concept Analysis, Springer, New York, 1999.

[8] B. Korte, L. Lovász, R. Schrader, Greedoids, Springer, New York, Berlin, 1991.

[9] Y. Kempner, V.E. Levit, Correspondence between two antimatroid algorithmic characterizations, Electron. J. Combin. 10 (2003).

[10] Y. Kempner, B. Mirkin, I. Muchnik, Monotone linkage clustering and quasi-concave functions, Appl. Math. Lett. 10 (4) (1997) 19–24.

[11] Y. Kempner, I. Muchnik, Clustering on antimatroids and convex geometries, WSEAS Trans. Math. 2 (1) (2003) 54–59.

[12] A. Malishevski, Properties of ordinal set functions, in: A. Malishevski (Ed.), Qualitative Models in the Theory of Complex Systems, Nauka, Moscow, 1998(in Russian).

[13] R.S. Michalski, R. Stepp, Learning from observation: conceptual clustering, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, TIOGA Publishing Co., Palo Alto, 1983, pp. 331–363.

[14] B. Mirkin, I. Muchnik, Layered clusters of tightness set functions, Appl. Math. Lett. 15 (2002) 147–151.

[15] B. Mirkin, I. Muchnik, Induced layered clusters, Hereditary Mappings, and Convex Geometry, Appl. Math. Lett. 15 (2002) 293–298.

[16] I. Muchnik, L.V. Shvartser, Kernels of monotonic systems on a semi-lattice of sets, Automat. Remote Control 50 (8 part 2) (1989) 1095–1102.

[17] J. Mullat, Extremal subsystems of monotone systems: I, II, Automat. Remote Control 37 (1976) 758–766 1286–1294.

[18] J. Mullat, A fast algorithm for finding matching responses in survey data table, Math. Soc. Sci. 30 (1995) 195–205.

[19] Y. Zaks (Kempner), I. Muchnik, Incomplete classifications of a finite set of objects using monotone systems, Automat. Remote Control 50 (1989) 553–560.