Theoretical Computer Science 922 (2022) 170–178

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Unique key Horn functions

Kristóf Bérczi[a,*], Endre Boros[b], Ondřej Čepek[c], Petr Kučera[c], Kazuhisa Makino[d]

[a] *MTA-ELTE Momentum Matroid Optimization Research Group and MTA-ELTE Egerváry Research Group, Dept. of Operations Research, Eötvös University, Budapest, Hungary*
[b] *MSIS Department and RUTCOR, Rutgers University, New Jersey, USA*
[c] *Charles University, Faculty of Mathematics and Physics, Department of Theoretical Computer Science and Mathematical Logic, Praha, Czech Republic*
[d] *Research Institute for Mathematical Sciences (RIMS), Kyoto University, Kyoto, Japan*

ARTICLE INFO

Article history:
Received 14 October 2021
Received in revised form 10 April 2022
Accepted 21 April 2022
Available online 4 May 2022
Communicated by M.J. Golin

Keywords:
Minimal key
Pure Horn function
Sperner hypergraph
Target set selection

ABSTRACT

Given a relational database, a key is a set of attributes such that a value assignment to this set uniquely determines the values of all other attributes. The database uniquely defines a pure Horn function h, representing the functional dependencies. If the knowledge of the attribute values in set A determines the value for attribute v, then $A \to v$ is an implicate of h. If K is a key of the database, then $K \to v$ is an implicate of h for all attributes v.

Keys of small sizes play a crucial role in various problems. We present structural and complexity results on the set of minimal keys of pure Horn functions. We characterize Sperner hypergraphs for which there is a unique pure Horn function with the given hypergraph as the set of minimal keys. Furthermore, we show that recognizing such hypergraphs is co-NP-complete already when every hyperedge has size two. On the positive side, we identify several classes of graphs for which the recognition problem can be decided in polynomial time.

We also present an algorithm that generates the minimal keys of a pure Horn function with polynomial delay, improving on earlier results. By establishing a connection between keys and target sets, our approach can be used to generate all minimal target sets with polynomial delay when the thresholds are bounded by a constant. As a byproduct, our proof shows that the MINIMUM KEY problem is at least as hard as the MINIMUM TARGET SET SELECTION problem with bounded thresholds.

1. Introduction

Throughout the paper, we denote by V the set of n Boolean variables. We will refer to the members of V as *positive literals* and to their negations as *negative literals*, respectively. A *Boolean function* is a mapping $f : \{0,1\}^V \to \{0,1\}$. A *conjunctive normal form* (CNF) is the conjunction of clauses, where each clause is a disjunction of literals. The CNF $\Phi = C_1 \wedge \cdots \wedge C_q$ is also viewed as a set of clauses $\Phi = \{C_1, \ldots, C_q\}$.

[*] Corresponding author.
E-mail addresses: berkri@cs.elte.hu (K. Bérczi), endre.boros@rutgers.edu (E. Boros), cepek@ktiml.mff.cuni.cz (O. Čepek), kucerap@ktiml.mff.cuni.cz (P. Kučera), makino@kurims.kyoto.ac.jp (K. Makino).

https://doi.org/10.1016/j.tcs.2022.04.022

A CNF Φ is called *Horn* if each of its clauses contains at most one positive literal, and *pure Horn* if every clause contains exactly one positive literal. A Boolean function h is called *(pure) Horn* if it has a (pure) Horn CNF representation. Note that every CNF defines a Boolean function, but a Boolean function may have many different CNF representations. For instance, given the pure Horn CNF $\Phi = (\overline{a} \vee b) \wedge (\overline{b} \vee a) \wedge (\overline{a} \vee \overline{c} \vee d) \wedge (\overline{a} \vee \overline{c} \vee e)$ on variable set $V = \{a, b, c, d, e\}$, we can also represent it by the pure Horn CNF $\Psi = (\overline{a} \vee b) \wedge (\overline{b} \vee a) \wedge (\overline{b} \vee \overline{c} \vee d) \wedge (\overline{b} \vee \overline{c} \vee e)$. Note that a pure Horn clause can also be viewed as an implication. For instance, $C = \overline{b} \vee \overline{c} \vee e$ is equivalent to the implication $bc \to e$. Thus, we can view a pure Horn CNF as an implication system, e.g., we shall write Φ equivalently, as $a \to b, b \to a, ac \to de$. For an implication $A \to v$ we call A the *body* and v the *head*. We say that $A \to v$ is an *implicate* of the Horn function h if any assignment $x \in \{0,1\}^V$ that falsifies $A \to v$ also falsifies h. In particular, if h is represented by a pure Horn CNF then the clauses of this CNF are all implicates of h.

The concept of Horn functions has been widely studied under different names, such as directed hypergraphs in graph theory and combinatorics [1], as implication systems in machine learning [2,3], database theory [4,5], and as lattices and closure systems in algebra and concept lattice analysis [6,7]. Horn functions form a fundamental subclass of Boolean functions endowed with interesting structural and computational properties. The satisfiability problem can be solved for Horn functions in linear time and the equivalence of such formulas can be decided in polynomial time [8]. Horn functions are strongly related to relational databases [4] and many interesting algorithmic problems arise from that context. Given a database, we associate the set V of Boolean variables to the set of attributes of the database. For $A \subseteq V$ and $v \in V$ we write $A \to v$ if the knowledge of the attribute values in A uniquely determines the value of v (in the database records). Such a relation is called a functional dependency in the database. The set of all functional dependencies define a unique pure Horn function associated to the given database [4,5]. One of the important notions that arise from databases is the concept of a key. A key in a relational database is a set of attributes the values of which determine uniquely the values of all other attributes. Accordingly, a subset K of the variables is a *key* of a Horn function h if $K \to v$ is an implicate of h for all $v \in V \setminus K$.

We call a pure Horn function *key Horn* if the body of any of its implicates is a key of the function. Key Horn functions generalize the well studied class of *hydra functions* introduced in [9], where all the bodies are of size 2. Finding a shortest CNF representation of a given Horn function with respect to multiple relevant measures (number of clauses, number of literals, etc.) is a computationally hard problem [1,5,10]. For general pure Horn functions not even non-trivial approximation algorithms are known. For hydra functions a 2-approximation algorithm was given in [9], while [11] proved that the minimization remains NP-hard even in this special case. In [12], the authors provided logarithmic factor approximation algorithms for general key Horn functions with respect to all of the above mentioned measures.

Our results The present paper focuses on the structure of the set of minimal keys of a pure Horn function. In particular, we are interested in finding Sperner hypergraphs $\mathcal{B}$ that form the set of minimal keys of a unique pure Horn function $h_{\mathcal{B}}$. We call such a $\mathcal{B}$ a *unique key hypergraph*, and the corresponding Horn function $h_{\mathcal{B}}$ a *unique key Horn function*.

Section 3 gives a characterization of unique key hypergraphs and unique key Horn functions. In particular, we show that cuts of a matroid form a unique key hypergraph. The special case when every hyperedge has size two is discussed in Section 4, where we show that recognizing unique key graphs is co-NP-complete. Subsequently, we identify several classes of graphs for which the recognition problem can be decided in polynomial time. Section 5 provides an algorithm that generates all minimal keys of a pure Horn function with polynomial delay. The algorithm can be used to generate all candidate keys of a relation improving on results in [13,14]. Furthermore, we show that the problems of finding a minimum key of a pure Horn function and of finding a minimum target set of a graph are closely related. Using this connection, our algorithm can be used to generate all minimal target sets with polynomial delay when the thresholds are bounded by a constant.

2. Definitions and notation

We start with additional definitions and notation. We view the set of variables V as a ground set. A hypergraph $\mathcal{B} \subseteq 2^V$ is called a *Sperner hypergraph* if none of its hyperedges contains another one. Given a Sperner hypergraph $\mathcal{B} \subseteq 2^V$, we say that $T \subseteq V$ is a *transversal* of $\mathcal{B}$, if $T \cap B \neq \emptyset$ for all $B \in \mathcal{B}$. We say that S is an *independent set* of $\mathcal{B}$ if $T = V \setminus S$ is a transversal of $\mathcal{B}$. We denote by $\mathcal{B}^d$ the set of *minimal transversversals* of $\mathcal{B}$, and by $\mathcal{B}^*$ the *family of its independent sets*.

For a hypergraph $\mathcal{B} \subseteq 2^V$ and subset $S \subseteq V$ we denote by $\mathcal{B}_S = \{B \in \mathcal{B} \mid B \subseteq S\}$ the *subhypergraph of $\mathcal{B}$ induced by S*. In particular, if $S \in \mathcal{B}^*$ then $\mathcal{B}_S = \emptyset$. Furthermore, we denote by $\mathcal{B}^S = \text{min'l}\,\{S \cap B \mid B \in \mathcal{B}\}$ the *projection of $\mathcal{B}$ to S* where $\text{min'l}\,\{\mathcal{H}\}$ denotes the family consisting of the inclusionwise minimal members of $\mathcal{H}$. Clearly, if S is not a transversal of $\mathcal{B}$ then we have $\mathcal{B}^S = \{\emptyset\}$. We will use the following well-known lemma.

Lemma 1 *(Seymour [15]). For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$ and subset $S \subseteq V$ we have $(\mathcal{B}_S)^d = (\mathcal{B}^d)^S$ and $(\mathcal{B}^S)^d = (\mathcal{B}^d)_S$.*

For a Boolean function h, we denote by $T(h)$ the set of true vectors of h, i.e., $T(h) = \{\mathbf{x} \in \{0,1\}^V \mid h(\mathbf{x}) = 1\}$. For two functions h and h' we write $h \le h'$ if for all $\mathbf{x} \in \{0,1\}^V$ we have $h(\mathbf{x}) \le h'(\mathbf{x})$, in other words, if $T(h) \subseteq T(h')$. We say that a clause $A \to v = v \vee \bigvee_{a \in A} \bar{a}$ is an *implicate* of h if $(A \to v)(\mathbf{x}) \ge h(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^V$. For a subset $S \subseteq V$ we define the *forward chaining closure* of S by $F_h(S) = \{u \in V \mid S \to u \text{ is an implicate of } h\}$. It is a well known fact that if h is a pure Horn function represented with a pure Horn CNF Φ, then the forward chaining closure $F_h(S)$ of a given set of variables S can be computed in time linear in the length of Φ (see e.g. [16]). Note that if $h' \leq h$, then $F_{h'}(S) \supseteq F_h(S)$, since any implicate of h is also an implicate of h'. For a CNF Φ we use the same terminology and notation as it defines a unique Boolean function. For example, $\Phi \subseteq \Psi$ implies $\Phi \geq \Psi$.

Let us remark here that the mapping $S \to F_h(S)$ associates a so called *closure system* to every pure Horn function h, and such a closure system determines h uniquely. As we mentioned in the introduction, pure Horn functions appear to be the subjects of study not only in closure systems, but in fact in several areas, including the theories of directed hypergraphs, relational databases, concept lattices, certain convex geometries, etc. The mathematical equivalence of some of these terminologies motivated e.g., the recent workshop [17]. It turns out that despite of this mathematical equivalence, the "natural" representation of the underlying system (input) may vary substantially from area to area. Accordingly, the related algorithmic problems arise in different contexts and may have very different complexities. For instance, numerous CNF-s of very different sizes can represent a pure Horn function in the context of Boolean functions/implication systems, and none of these may appear "natural" to represent the corresponding unique closure system. In our paper we focus on both structural and algorithmic problems, and thus decided to use the terminology and notation based on hypergraphs and implications.

Our study focuses on keys of a pure Horn function h, that is on subsets $K \subseteq V$ for which $F_h(K) = V$. Clearly, keys of a pure Horn function form an upward monotone system. We denote by $\mathcal{K}(h)$ the *set of minimal keys* of h. To a Sperner hypergraph $\mathcal{B} \subseteq 2^V$ we associate the pure Horn CNF

$$\Phi_{\mathcal{B}} = \bigwedge_{B \in \mathcal{B}} \bigwedge_{v \in V \setminus B} (B \to v).$$

Note that we have $\mathcal{K}(\Phi_{\mathcal{B}}) = \mathcal{B}$. Let us observe that for a Sperner hypergraph $\mathcal{B}$ and pure Horn function h, $\mathcal{B} \subseteq \mathcal{K}(h)$ implies that $h \leq \Phi_{\mathcal{B}}$.

Definition 2. For a Sperner family $\mathcal{B}$ we call $\Phi_{\mathcal{B}}$ a *key Horn CNF*. A pure Horn function h is called *key Horn* if it has a key Horn CNF representation. Furthermore, we call h *unique key Horn* if it is the unique pure Horn function with $\mathcal{K}(h)$ as the set of minimal keys.

By definition, a unique key Horn function is key Horn.

Let us also note that there may be several pure Horn functions with the same family of keys. As an example, consider the hypergraph $\mathcal{B} = \{\{a, b\}, \{b, c\}, \{c, d\}\}$ over the ground set $V = \{a, b, c, d\}$, and the pure Horn CNFs $\Psi^1 = \Phi_{\mathcal{B}} \wedge (b \to d)$, $\Psi^2 = \Phi_{\mathcal{B}} \wedge (c \to a)$, and $\Psi^3 = \Phi_{\mathcal{B}} \wedge (b \to d) \wedge (c \to a)$. It is easy to verify now that the CNFs $\Phi_{\mathcal{B}}$, and Ψ^i, $i = 1, 2, 3$ define four pairwise distinct pure Horn functions and each has $\mathcal{B}$ as its set of minimal keys.

3. Unique key Horn functions

The purpose of this section is to give an understanding of the structure of pure Horn functions that have the same set of keys and in particular the structure of unique key Horn functions.

Lemma 3. *Let $\mathcal{B} \subseteq 2^V$ be a Sperner hypergraph and $h : \{0, 1\}^V \to \{0, 1\}$ be a pure Horn function such that $h \leq \Phi_{\mathcal{B}}$. Then $\mathcal{K}(h) \neq \mathcal{B}$ if and only if there exists an implicate $A \to v$ of h and a minimal transversal $T \in \mathcal{B}^d$ such that $T \cap A = \emptyset$ and $v \in T$.*

Proof. Since $h \leq \Phi_{\mathcal{B}}$, any $B \in \mathcal{B}$ is a key of h. Thus $\mathcal{K}(h) \subseteq \mathcal{B}$ implies $\mathcal{K}(h) = \mathcal{B}$ because $\mathcal{B}$ is a Sperner hypergraph.

Assume first that $\mathcal{K}(h) \neq \mathcal{B}$, that is, there exists a minimal key $K \in \mathcal{K}(h) \setminus \mathcal{B}$. Since the sets of $\mathcal{B}$ are keys of h and K is a minimal key of h, we must have $K \in \mathcal{B}^*$. Let W denote a maximal independent set which contains K as a subset. It follows that $T = V \setminus W$ is a minimal transversal which is disjoint from K. Let v be an arbitrary node in T. Then $K \to v$ is an implicate of h because K is a key. Thus, choosing $A = K$ proves one direction of our claim.

For the reverse direction, let us assume that there exists an implicate $A \to v$ of h and a minimal transversal $T \in \mathcal{B}^d$ such that $T \cap A = \emptyset$ and $v \in T$. Since T is a minimal transversal of $\mathcal{B}$, there exists $B \in \mathcal{B}$ such that $T \cap B = \{v\}$. This implies that $F_{(A \to v) \wedge \Phi_{\mathcal{B}}}(V \setminus T) = V$. Because we have $h \leq (A \to v) \wedge \Phi_{\mathcal{B}}$ by our assumptions, $F_h(V \setminus T) = V$ follows. Therefore, there exists a minimal key $K \subseteq V \setminus T$ of h. Finally, $V \setminus T \in \mathcal{B}^*$ implies $K \in \mathcal{B}^*$, from which $K \in \mathcal{K}(h) \setminus \mathcal{B}$ follows as claimed. □

We introduce the notation $\cup\mathcal{B} = \bigcup_{B \in \mathcal{B}} B$ to denote the union of the hyperedges of $\mathcal{B}$.

Lemma 4. *Let $\mathcal{B} \subseteq 2^V$ be a Sperner hypergraph and $h : \{0, 1\}^V \to \{0, 1\}$ be a pure Horn function such that $h \leq \Phi_{\mathcal{B}}$. Then $\mathcal{K}(h) = \mathcal{B}$ if and only if for all implicates $A \to v$ of h with $A \in \mathcal{B}^*$ we have $v \in (V \setminus A) \setminus (\cup\mathcal{B}^{V \setminus A})$.*

Proof. Let us first note that for any subset $A \subseteq V$ that has a disjoint minimal transversal $T \in \mathcal{B}^d$ we must have $A \in \mathcal{B}^*$. Thus, by Lemma 3, we have $\mathcal{K}(h) = \mathcal{B}$ if and only if for all implicates $A \to v$ of h for which $A \in \mathcal{B}^*$ and for all minimal transversals $T \in \mathcal{B}^d$ with $T \cap A = \emptyset$ we have $v \notin T$. By Lemma 1, we have $(\mathcal{B}^{V \setminus A})^d = (\mathcal{B}^d)_{V \setminus A}$. Furthermore, the equality $\cup\mathcal{H} = \cup\mathcal{H}^d$ holds for all Sperner hypergraphs $\mathcal{H}$. Hence, we have $\cup(\mathcal{B}^d)_{V \setminus A} = \cup\mathcal{B}^{V \setminus A}$, implying the claim. □

Lemma 5. *Let $\mathcal{B} \subseteq 2^V$ be a Sperner hypergraph and define $\Psi = \{A \to v \mid A \in \mathcal{B}^*, v \notin \cup\mathcal{B}^{V\setminus A}\}$. Let φ be a set of clauses of the form $A \to v$ that are not implicates of $\Phi_{\mathcal{B}}$. Then $\mathcal{K}(\varphi \wedge \Phi_{\mathcal{B}}) = \mathcal{B}$ if and only if $\varphi \subseteq \Psi$.*

Proof. The claim follows by Lemma 4. □

Now we are ready to characterize unique key hypergraphs.

Theorem 6. *For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$, the pure Horn function $h = \Phi_{\mathcal{B}}$ is unique key Horn if and only if for all $T \in \mathcal{B}^d$ and $v \notin T$ there exists $T' \in \mathcal{B}^d$ such that $T' \neq T$ and $T' \subseteq T \cup \{v\}$.*

Proof. For any pure Horn function h with $\mathcal{K}(h) = \mathcal{B}$ we have $h \leq \Phi_{\mathcal{B}}$.

For the only if direction, take an arbitrary $T \in \mathcal{B}^d$ and $v \notin T$, and let $A = V \setminus (T \cup \{v\})$. By definition of A, we have that $\cup\mathcal{B}^{V\setminus A} \subseteq T \cup \{v\}$. Since T is a transversal, we have that $T \subseteq \cup\mathcal{B}^{V\setminus A}$ and that $A \to v$ is not an implicate of h. If $v \notin \mathcal{B}^{V\setminus A}$, then by Lemma 5 we have $\mathcal{K}(\Phi_{\mathcal{B}} \wedge (A \to v)) = \mathcal{B}$ which is a contradiction with the assumption that h is the only Horn function with this property. It follows that $v \in \cup\mathcal{B}^{V\setminus A}$ and altogether we get that $T \cup \{v\} = \cup\mathcal{B}^{V\setminus A}$. In particular, this means that there exists a $B \in \mathcal{B}$ with $B \setminus A$ being minimal and $v \in B$. Since T is a transversal of $\mathcal{B}$, we have $B \cap T \neq \emptyset$. Consider an element $u \in B \cap T$. By the minimality of $B \setminus A$, for every $B' \in \mathcal{B}$ different from B either $B' \cap (T \setminus \{u\}) \neq \emptyset$ or $v \in B'$. This means that $T' = (T \setminus \{u\}) \cup \{v\}$ is a transversal of $\mathcal{B}$.

For the opposite direction, take an arbitrary $A \in \mathcal{B}^*$ and $v \notin \cup\mathcal{B}^{V\setminus A}$. Then $A \cup \{v\} \in \mathcal{B}^*$, hence there exists $T \in \mathcal{B}^d$ disjoint from $A \cup \{v\}$. By the assumption, there exists $u \in T$ such that $T' = (T \setminus \{u\}) \cup \{v\}$ is also a minimal transversal of $\mathcal{B}$. Therefore, there exists $B \in \mathcal{B}$ for which $B \cap T' = \{v\}$. As $v \notin \cup\mathcal{B}^{V\setminus A}$, there exists $B' \in \mathcal{B}$ such that $B' \setminus A \subsetneq B \setminus A$ and $v \notin B' \setminus A$. This implies $B' \cap T' = \emptyset$, contradicting T' being a transversal. This shows that the set Ψ in Lemma 5 is empty, proving the uniqueness of h. □

We assume that the reader is familiar with the notion of a matroid [18,19].

Corollary 7. *The cuts of a loopless matroid form a unique key hypergraph.*

Proof. If $\mathcal{B}$ is the set of cuts of a matroid, then $\mathcal{B}^d$ is the set of bases. If the matroid is loopless, then $\cup\mathcal{B}^d = V$. The basis exchange axiom implies the necessary and sufficient condition of Theorem 6. □

The following example shows that not all unique key hypergraphs are related to matroids. Let $\mathcal{B} = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3,4\}\}$, where $V = \{1,2,3,4\}$. Then $\mathcal{B}^d = \mathcal{B}$ and satisfies the conditions of Theorem 6, hence $\mathcal{B}$ is unique key. Clearly, $\mathcal{B}^d$ is not the set of bases of a matroid.

Remark 8. The conditions of Theorem 6 can be checked in polynomial time if $\mathcal{B}^d$ can be generated from $\mathcal{B}$ in time polynomial in the length of the input. For example, if $\mathcal{B}$ is 2-monotonic or forms the set of bases of a matroid. For a 2-monotonic system $\mathcal{B}$ it was shown e.g., in [20–22] that $|\mathcal{B}^d| \leq |V| \cdot |\mathcal{B}|$ and that $\mathcal{B}^d$ can be generated with polynomial delay. These two facts imply that $\mathcal{B}^d$ can be generated in time polynomial in the length of the input when $\mathcal{B}$ is given as input. For the dual of the bases of a matroid $\mathcal{B}$ it is well-known and easy to check that the sets in $\mathcal{B}^d$ are complements of the hyperplanes of the matroid and $|\mathcal{B}^d| \leq rank(V) \cdot |\mathcal{B}|$. Consequently, a result of Seymour [23] implies that in fact $\mathcal{B}^d$ can be generated from $\mathcal{B}$ in time polynomial in the length of the input.

4. Unique key graphs

Let us now consider Sperner hypergraphs $\mathcal{B} \subseteq 2^V$ such that $|B| = 2$ for all $B \in \mathcal{B}$ (i.e., graphs). For the sake of simplicity, we use $G = (V, E)$ to denote such a hypergraph $\mathcal{B} = E$. We say that G is a *unique key graph* if $\mathcal{B} = E$ is a unique key hypergraph. Following standard notation in graph theory, we denote by $N(u) = \{v \in V \mid (u, v) \in E\}$ the set of neighbors of vertex $u \in V$. We consider simple graphs throughout, hence $u \notin N(u)$. For a subset $S \subseteq V$ we denote by $N(S) = \left(\bigcup_{u\in S} N(u)\right) \setminus S$ the set of neighbors of S.

4.1. Complexity of recognizing unique key graphs

Given a graph $G = (V, E)$ and a maximal independent set $I \subseteq V$ we say that $u \notin I$ is an *individual neighbor of* $v \in I$ if $N(u) \cap I = \{v\}$.

Theorem 9. *A graph $G = (V, E)$ is a unique key graph if and only if for every maximal independent set $I \subseteq V$ and vertex $v \in I$ there exists a vertex $u \notin I$ that is an individual neighbor of v.*

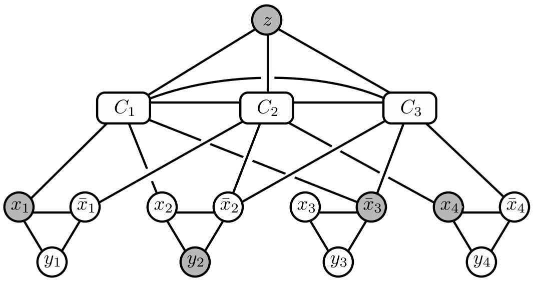


Fig. 1. The graph G_Φ corresponding to CNF formula $\Phi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x_2} \vee \bar{x}_3 \vee \bar{x}_4)$. Grey vertices form a maximal independent set corresponding to a satisfying truth assignment. Note that z has no individual neighbor.

Proof. The minimal transversals of E are exactly the complements of the maximal independent sets of G, that is the minimal vertex covers of G. For a maximal independent set I with $v \in I$ and $u \notin I$, the set $(I \setminus \{v\}) \cup \{u\}$ is an independent set if and only if u is an individual neighbor of v. If this is the case, then $(I \setminus \{v\}) \cup \{u\}$ can be extended to a maximal independent set I' of G not containing v. Thus the statement follows from Theorem 6. □

Our next goal is to show that recognizing if $\mathcal{B}$ is the set of minimal keys of a unique key function is difficult already for hypergraphs of dimension two. Let us consider a CNF $\Phi = C_1 \wedge \cdots \wedge C_m$ over Boolean variables x_i, $i = 1, \ldots, n$. Let us associate a graph G_Φ to Φ as follows. The set of vertices is $V(G_\Phi) = \{x_i, \bar{x}_i, y_i \mid i = 1, \ldots, n\} \cup \{C_j \mid j = 1, \ldots, m\} \cup \{z\}$. The edges are defined as follows: vertices x_i, $\bar{x}_i$ and y_i form a triangle for all $i = 1, \ldots, n$. Vertices C_j, $j = 1, \ldots, m$ and z form a clique. Finally, all vertices C_j are connected to the literals they include (see Fig. 1).

Theorem 10. *A CNF Φ is not satisfiable if and only if the graph G_Φ is unique key.*

Proof. We derive this claim using Theorem 9.

Let us note first that every maximal independent set $I \subseteq V(G_\Phi)$ has exactly $n+1$ points, one from each of the following cliques: $T_i = \{x_i, \bar{x}_i, y_i\}$, $i = 1, \ldots, n$, and $K = \{C_j \mid j = 1, \ldots, m\} \cup \{z\}$. This is because an independent set I can contain at most one vertex from each of these cliques, and if it is disjoint from T_i, then $I \cup \{y_i\}$ is also independent. Similarly, if $I \cap K = \emptyset$, then $I \cup \{z\}$ is also independent. We now verify the conditions of Theorem 9.

Let I be a maximal independent set, and assume that $v = x_i \in I$ or $v = \bar{x}_i \in I$. In both cases $u = y_i$ is an individual neighbor of v. Note next that the sets $N(x_i) \cap K$ and $N(\bar{x}_i) \cap K$ are disjoint, and therefore any independent set is disjoint from at least one of these sets. Thus, if $v = y_i \in I$, then either $u = x_i$ or $u = \bar{x}_i$ (or both) is an individual neighbor of v. If $v = C_j \in I$, then $u = z$ is an individual neighbor of v.

Thus, the only claim left to show is that Φ is satisfiable if and only if there exists a maximal independent set I of G containing vertex z such that z does not have an individual neighbor. To see this let us first assume that Φ is satisfiable. Consider the set I that contains the literals that are true in a satisfying assignment and vertex z. Since every clause C_j is satisfied, it has a neighbor in I other than z, and thus z does not have an individual neighbor. For the other direction let us assume that I is a maximal independent set, containing z such that z does not have an individual neighbor. Therefore, every clause C_j must have a neighbor in I, which must be a literal. Since $(x_i, \bar{x}_i)$ is an edge of G for all $i = 1, \ldots, n$, I cannot contain a complementary pair of literals, and thus the literals in I can be set to true simultaneously, satisfying Φ. □

Corollary 11. *Deciding if a hypergraph is unique key is co-*NP*-complete already for hypergraphs of dimension 2.*

Proof. It is easy to see that the problem belongs to co-NP, and thus the statement follows by Theorem 10. □

4.2. Bipartite graphs

Theorem 12. *A bipartite graph $G = (V, E)$ without isolated vertices is unique key if and only if E is a perfect matching.*

Proof. If E forms a perfect matching on V, then every maximal independent set I contains exactly one end vertex of every edge in E. For any vertex $v \in I$, the other end vertex u of the matching edge incident to v is an individual neighbor of v, thus G is a unique key graph by Theorem 9.

For the other direction, let A and B denote the color classes of G, that is, $V = A \cup B$. By the assumption that there are no isolated vertices in G, both A and B are maximal independent sets. By Theorem 9, every vertex $v \in V$ has an individual neighbor in the opposite color class, that is, a neighbor of degree exactly one. This implies that E is a matching as stated. □

4.3. Bounded treewidth graphs

Theorem 13. *For graphs of bounded treewidth, it is possible to decide in linear time if a graph is a unique key graph.*

Proof. We will formulate the problem in monadic second order logic (MSO), the result then follows by Courcelle's theorem [24]. Assume that a graph $G = (V, E)$ is described with a set of vertices V and an adjacency relation $\mathrm{adj}(u, v)$ which represents the set of edges. The unique key property can then be described as the predicate

$$\mathrm{UniqKey}(G) = (\forall I \subseteq V)(\forall v \in I)(\exists u \in V)[\mathrm{IndSet}(I) \to \mathrm{IndNeigh}(I, v, u)]$$

where IndSet(I) is a predicate satisfied if I is an independent set of G and $\mathrm{IndNeigh}(I, v, u)$ is satisfied if $v \in I$ and u is its individual neighbor. These predicates can be defined in the following way.

$$\mathrm{IndSet}(I) = (\forall u \in I)(\forall v \in I)[\neg\,\mathrm{adj}(u, v)]$$

$$\mathrm{IndNeigh}(I, v, u) = (\forall w \in I)[\mathrm{adj}(w, u) \to w = v]$$

This concludes the proof. □

Since the formulation of UniqKey(G) uses only quantification over a set of vertices I and not over any set of edges, we can use it to show the following corollary.

Corollary 14. *For graphs of bounded clique-width, it is possible to decide in linear time if a graph is a unique key graph.*

Proof. Follows by using a version of Courcelle's theorem for clique-width [25] on the formulation of predicate $\mathrm{UniqKey}(G)$ in the proof of Theorem 13. □

4.4. Graphs with small induced matchings

Theorem 15. *Let $G = (V, E)$ be a graph, and assume that the size of the largest induced matching of G is bounded by a constant. Then there is an efficient algorithm to decide if G is a unique key graph.*

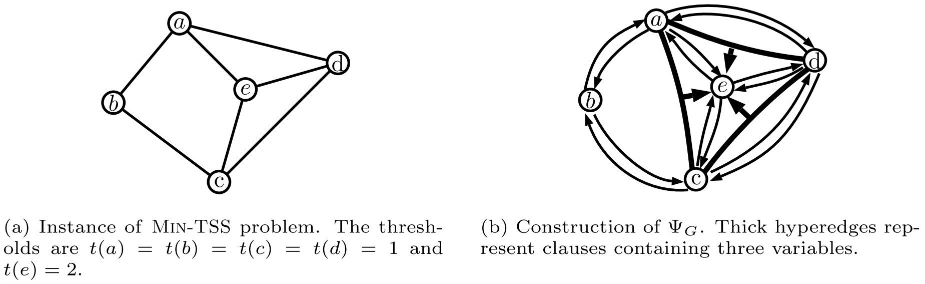
Proof. If $\mathcal{B} = E$ then $\mathcal{B}^d$ is the family of minimal vertex covers that are exactly the complements of maximal independent sets. It is known that if the largest induced matching in G has size at most p, then it has at most n^{2p} maximal independent sets [26]. Thus if p is a constant, then all of them can be generated in polynomial time [27]. This in turn implies that the conditions of Theorem 9 can be checked in polynomial time. □

5. Generating minimal keys

We shift the focus from unique key hypergraphs to the problem of generating all possible minimal keys of a given pure Horn function. The proposed approach can be applied for various problems, for example for generating all *minimal target sets* of a graph. Note that the number of minimal keys can be exponential in the size of the input CNF, hence the efficiency of generating them is measured by the time spent between outputting two of them. A *generation algorithm* outputs the objects in question one by one without repetition. Such a procedure is called *polynomial delay* if the computing time between any two consecutive outputs is bounded by a polynomial of the input size.

For the generation of minimal keys of a pure Horn function we employ the technique based on *supergraphs*. A supergraph is a directed graph in which the vertices are the objects to be generated, and arcs represent a neighborhood relation such that (i) the out-neighborhood of any vertex can be generated in time polynomial in the length of the input, and (ii) the supergraph is strongly connected. Then, a carefully executed breadth-first search in this graph, starting from an arbitrary vertex, will reach and output all vertices with polynomial delay; see e.g., [28,29]. We provide a detailed description below. Let us remark that this approach is also very similar to the methods used in [30,31] for generating all prime implicates and all abductive explanations of a Horn CNF.

Given a pure Horn CNF Φ, we associate to it a directed graph $D_\Phi = (\mathcal{K}(\Phi), E)$ as follows. For a minimal key $K \in \mathcal{K}(\Phi)$, an arbitrary variable $v \in K$, and a clause $A \to v \in \Phi$, we define the set $S = (K - v) \cup A$. Note that S is a key of Φ, hence there exists $K' \in \mathcal{K}(\Phi)$ with $K' \subseteq S$. We find such a K' using a greedy procedure by dropping variables from S one-by-one, in some order, and checking at each step if the remaining set is a key by using forward chaining with respect to Φ. We include a directed edge (K, K') into E for all possible choices $v \in K$ and $A \to v \in \Phi$. For some $v \in K$ we might not have a clause $A \to v$ in which case we do not generate the corresponding K'. Note that every vertex in D_Φ has at most m outgoing edges, where m denotes the number of clauses in Φ. Let us remark that the final graph D_Φ is not uniquely defined as its edge set depends on the choices of the K' sets in the above procedure. (We could make this definition unique by always using the same order of the elements of V when dropping them one-by-one from S.)



(a) Instance of MIN-TSS problem. The thresholds are $t(a) = t(b) = t(c) = t(d) = 1$ and $t(e) = 2$.

(b) Construction of Ψ_G. Thick hyperedges represent clauses containing three variables.

Fig. 2. Illustration of Theorem 18. The CNF associated to G is $\Psi_G = (b \to a) \wedge (e \to a) \wedge (d \to a) \wedge (a \to b) \wedge (c \to b) \wedge (b \to c) \wedge (d \to c) \wedge (e \to c) \wedge (a \to d) \wedge (c \to d) \wedge (e \to d) \wedge (\{a, c\} \to e) \wedge (\{a, d\} \to e) \wedge (\{c, d\} \to e)$.

Lemma 16. *D_Φ is strongly connected.*

Proof. First we introduce a measure between minimal keys. Let $K_1, K_2 \in \mathcal{K}(\Phi)$ be two minimal keys. We know that the forward chaining closure of K_2 is equal to V. Let us partition V into layers $L_0, L_1, \ldots, L_t$ where $L_0 := K_2$, define $L_{i+1} := \{v \in V \setminus L_i \mid \text{there exists } A \to v \in \Phi \text{ s.t. } A \subseteq \bigcup_{j=0}^{i} L_j\}$, and t is the largest index such that $L_t \neq \emptyset$. Let $\varrho(K_1, K_2) := (\varrho_0, \varrho_1, \ldots, \varrho_t)$ where $\varrho_i = |L_i \cap K_1|$ for $i = 0, \ldots, t$.

We claim that there exists an out-neighbor K_3 of K_1 in D_Φ such that $\varrho(K_3, K_2)$ is strictly smaller in the reverse lexicographic order than $\varrho(K_1, K_2)$. To see this, let i be the largest index such that $\varrho_i \neq 0$, and let v be in $K_1 \cap L_i$. Since $v \in L_i$, there exists an $A \to v \in \Phi$ such that $A \subseteq \bigcup_{j=0}^{i-1} L_j$. For the set $S = (K_1 - v) \cup A$ we have that $|L_i \cap S| < |L_i \cap K_1|$ and $|L_j \cap S| = 0$ for $j > i$. Thus the out-neighbor $K_3 \subseteq S$ satisfies the claim. By induction in the reverse lexicographic order of the possible ϱ vectors, there exists a directed path in D_Φ from K_3 to K_2. As $K_1K_3 \in E$, the same holds for K_1, thus finishing the proof of the lemma. □

Theorem 17. *Given a pure Horn CNF Φ, we can generate all minimal keys of Φ with polynomial delay.*

Proof. Consider the directed graph D_Φ. Our algorithm will perform a breadth-first search on D_Φ. As D_Φ is strongly connected according to Lemma 16, all minimal keys are visited (and hence generated) in this way.

We start with a minimal key which we generate by greedily leaving out elements one by one from V. In this process we store the generated minimal keys both in a first-in-first-out (FIFO) queue and in a binary tree. Note that insertion/deletion takes $O(1)$ time in the queue, insertion into the binary tree takes $O(n)$ time, and checking if a set is already in the binary tree also can be done in $O(n)$ time.

In a general step we take the first element K of the queue, and generate all its $O(m)$ out-neighbors in D_Φ, as described above. For each out-neighbor we test first if it is in the binary tree of already generated minimal keys, and if not, we insert it there and add it to the end of the queue. Once we processed all out-neighbors of K, we delete it from the queue, and output it. According to the above, all these can be done in $O(poly(n, m))$ time. Thus, this procedure has a polynomial delay. □

5.1. Minimum target set selection

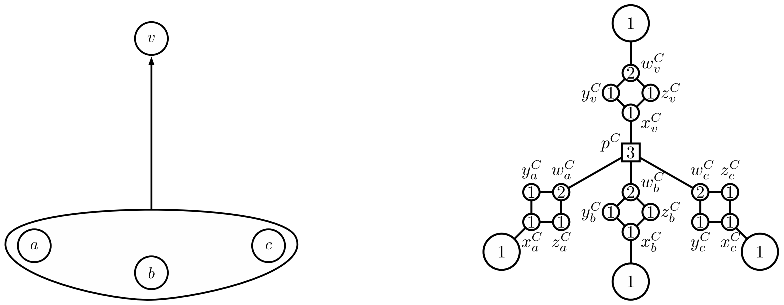
In the MINIMUM TARGET SET SELECTION problem, we are given an undirected graph $G = (V, E)$ and a threshold function $t : V \to \mathbb{Z}_+$. As a starting step, we can activate a subset $S \subseteq V$ of vertices. In every subsequent round, a vertex v becomes activated if at least $t(v)$ of its neighbors are already active. The goal is to find a minimum sized initial set S of active nodes (called a *target set*) so that the activation spreads to the entire graph.

Finding a minimum sized target set is rather difficult. Chen [32] showed that the problem is difficult to approximate within a $O(\text{poly}\log(n))$ factor even when all thresholds are 2 and the graph has a constant degree. Charikar et al. [33] proved that, assuming that the Planted Dense Subgraph conjecture is true, MINIMUM TARGET SET SELECTION is in fact difficult to approximate within a factor of $O(n^{1/2-\varepsilon})$ for every $\varepsilon > 0$ even for constant thresholds.

The aim of this section is to show that the problems of finding a minimum target set in a graph (MIN-TSS) and of finding a minimum key of a pure Horn function (MIN-KEY) are closely related.

Theorem 18. *The* MIN-TSS *problem with constant thresholds is polynomial-time reducible to the* MIN-KEY *problem.*

Proof. Let $G = (V, E)$, $t : V \to \mathbb{Z}_+$ be an instance of the MIN-TSS problem. For a vertex $v \in V$, we denote the set of its neighbors by $N(v) \subseteq V$. We construct a Horn CNF as follows (see Fig. 2):



(a) A pure Horn clause $C = A \to v$, where $A = \{a, b, c\}$.

(b) The gadget and threshold values corresponding to $C = A \to v$.

Fig. 3. Illustration of Theorem 19. Note that the size of the graph G is polynomial in the length of the input.

$$\Psi_G := \bigwedge_{v \in V} \bigwedge_{\substack{A \subseteq N(v) \\ |A| = t(v)}} A \to v.$$

Note that Ψ_G can be determined in polynomial time as the thresholds are assumed to be constants. By the definition of Φ_G, the activation process in G is equivalent to the forward chaining process in Ψ_G. This means that $K \subseteq V$ is a target set of G if and only if it is a key of Ψ_G, concluding the proof of the theorem. □

The reduction in the proof of Theorem 18, together with the hardness result of [33] implies that Min-Key is difficult to approximate within a factor of $O(n^{1/2-\varepsilon})$ for every $\varepsilon > 0$, assuming that the Planted Dense Subgraph conjecture holds.

Based on a construction previously used in [33] for establishing a connection between the directed and undirected variants of the target set selection problem, we show that Min-TSS includes Min-Key as a special case.

Theorem 19. *The* Min-Key *problem is polynomial-time reducible to the* Min-TSS *problem.*

Proof. Let Φ be a pure Horn CNF on variables V. We construct a graph $G = (V', E)$ together with a threshold function $t : V \to \mathbb{Z}_+$ such that every key of Φ is a target set of G, while every target set of G can be transformed to a key of Φ without increasing the size of the set.

We add the set of variables V to the vertices of G, and define $t(v) = 1$ for $v \in V$. For every clause $C = A \to v$ of Φ, we construct a gadget as follows. We add a vertex p^C that corresponds to the clause and set $t(p^C) = |A|$. For every variable $a \in A$, we add four new vertices x_a^C, y_a^C, z_a^C and w_a^C with thresholds $t(x_a^C) = t(y_a^C) = t(z_a^C) = 1$ and $t(w_a^C) = 2$, together with the edges $ax_a^C, x_a^C y_a^C, x_a^C z_a^C, y_a^C w_a^C, z_a^C w_a^C$ and $w_a^C p^C$. Finally, we add four new vertices x_v^C, y_v^C, z_v^C and w_v^C with thresholds $t(x_v^C) = t(y_v^C) = t(z_v^C) = 1$ and $t(w_v^C) = 2$, together with the edges $p^C x_v^C, x_v^C y_v^C, x_v^C z_v^C, y_v^C w_v^C, z_v^C w_v^C$ and $w_v^C v$ (see Fig. 3).

If $K \subseteq V$ is a key of Φ, then the same set of vertices in G form a target set. Indeed, when the forward chaining procedure uses a clause $C = A \to v$ to reach a variable v, then this step corresponds to the activation of v through the gadget associated to C in G.

Now assume that S is a target set of G. We cannot directly say that S is a key of Φ as S might contain vertices from $V' \setminus V$. However, it is not difficult to see that

$$\begin{aligned} K := {} & (V \cap S) \\ & \cup \{v \in V \mid \text{there exists } C \in \Phi \text{ with } v \in C, S \cap \{x_v^C, y_v^C, z_v^C, w_v^C\} \neq \emptyset\} \\ & \cup \{v \in V \mid \text{there exists } C = A \to v \in \Phi \text{ with } p^C \in S\} \end{aligned}$$

is a key of Φ with $|K| \leq |S|$, concluding the proof of the theorem. □

We have seen that finding a minimum sized target set is difficult already for constant thresholds. However, by combining the reductions in the proofs of Theorems 17 and 18, we get the following result.

Corollary 20. *Given a graph $G = (V, E)$ and constant thresholds $t : V \to \mathbb{Z}_+$, we can generate all minimal target sets of G with polynomial delay.*

6. Conclusions

In this paper we defined unique key hypergraphs as Sperner hypergraphs that form the set of minimal keys of a unique pure Horn function. We gave a characterization of such hypergraphs, and showed that cuts of a matroid form a natural example. As a negative result, we proved that the recognition of unique key hypergraphs is co-NP-complete already when every hyperedge has size two. On the positive side, we identified several classes of graphs for which the recognition problem can be decided in polynomial time.

We also gave an algorithm for generating all minimal keys of a pure Horn function with polynomial delay. By showing that the problems of finding a minimum key of a pure Horn function and of finding a minimum target set of a graph are closely related, we applied our algorithm to generate all minimal target sets of a graph with polynomial delay when the thresholds are bounded by a constant. It remains an open question whether all minimal target sets can be generated with polynomial delay when the thresholds are unbounded.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Kristóf Bérczi was supported by the János Bolyai Research Fellowship, by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06. Ondřej Čepek and Petr Kučera gratefully acknowledge a support by the Czech Science Foundation (Grant 19-19463S). This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

References

[1] G. Ausiello, A. D'Atri, D. Sacca, Minimal representation of directed hypergraphs, SIAM J. Comput. 15 (2) (1986) 418–431.
[2] M. Arias, J.L. Balcázar, Canonical Horn representations and query learning, in: International Conference on Algorithmic Learning Theory, Springer, 2009, pp. 156–170.
[3] M. Arias, J.L. Balcázar, Construction and learnability of canonical Horn formulas, Mach. Learn. 85 (3) (2011) 273–297.
[4] W.W. Armstrong, Dependency Structures of Database Relationships, Proc. IFIP, vol. 74, North Holland, Amsterdam, 1974, pp. 580–583.
[5] D. Maier, Minimum covers in the relational database model, in: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, ACM, 1979, pp. 330–337.
[6] N. Caspard, B. Monjardet, The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey, in: Ordinal and Symbolic Data Analysis (OSDA '98), Univ. of Massachusetts, Amherst, Sept. 28-30, 1998, Discrete Appl. Math. 127 (2) (2003) 241–269.
[7] J.-L. Guigues, V. Duquenne, Familles minimales d'implications informatives résultant d'un tableau de données binaires, Math. Sci. Hum. 95 (1986) 5–18.
[8] W.F. Dowling, J.H. Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, J. Log. Program. 1 (3) (1984) 267–284.
[9] R.H. Sloan, D. Stasi, G. Turán, Hydras: directed hypergraphs and Horn formulas, Theor. Comput. Sci. 658 (2017) 417–428.
[10] P.L. Hammer, A. Kogan, Optimal compression of propositional Horn knowledge bases: complexity and approximation, Artif. Intell. 64 (1) (1993) 131–145.
[11] P. Kučera, Hydras: complexity on general graphs and a subclass of trees, Theor. Comput. Sci. 658 (2014) 399–416.
[12] K. Bérczi, E. Boros, O. Cepek, P. Kucera, K. Makino, Approximating minimum representations of key horn functions, SIAM J. Comput. 51 (1) (2022) 116–138.
[13] C.L. Lucchesi, S.L. Osborn, Candidate keys for relations, J. Comput. Syst. Sci. 17 (2) (1978) 270–279.
[14] M. Hermann, B. Sertkaya, On the complexity of computing generators of closed sets, in: R. Medina, S. Obiedkov (Eds.), Formal Concept Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 158–168.
[15] P. Seymour, On incomparable families of sets, Mathematica 20 (1973) 208–209.
[16] P.L. Hammer, A. Kogan, Quasi-acyclic propositional Horn knowledge bases: optimal compression, IEEE Trans. Knowl. Data Eng. 7 (5) (1995) 751–762.
[17] K. Adaricheva, G.F. Italiano, H.K. Büning, G. Turán, Horn formulas, directed hypergraphs, lattices and closure systems: Related formalisms and applications, 2014.
[18] H. Nishimura, S. Kuroda, A Lost Mathematician, Takeo Nakasawa: The Forgotten Father of Matroid Theory, Springer Science & Business Media, 2009.
[19] H. Whitney, On the abstract properties of linear dependence, in: Hassler Whitney Collected Papers, Springer, 1992, pp. 147–171.
[20] P. Bertolazzi, A. Sassano, An o(mn) algorithm for regular set-covering problems, Theor. Comput. Sci. 54 (2) (1987) 237–247.
[21] Y. Crama, Dualization of regular Boolean functions, Discrete Appl. Math. 16 (1) (1987) 79–85.
[22] K. Makino, T. Ibaraki, The maximum latency and identification of positive Boolean functions, SIAM J. Comput. 26 (5) (1997) 1363–1383.
[23] P. Seymour, A note on hyperplane generation, J. Comb. Theory, Ser. A 61 (1) (1994) 88–91.
[24] B. Courcelle, The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, Inf. Comput. 85 (1) (1990) 12–75.
[25] B. Courcelle, J.A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, Theory Comput. Syst. 33 (2) (2000) 125–150.
[26] E. Balas, C.S. Yu, On graphs with polynomially solvable maximum-weight clique problem, Networks 19 (2) (1989) 247–253.
[27] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM J. Comput. 6 (3) (1977) 505–517.
[28] B. Schwikowski, E. Speckenmeyer, On enumerating all minimal solutions of feedback problems, Discrete Appl. Math. 117 (1) (2002) 253–265.
[29] E. Boros, K. Elbassioni, V. Gurvich, Algorithms for generating minimal blockers of perfect matchings in bipartite graphs and related problems, in: S. Albers, T. Radzik (Eds.), Algorithms – ESA 2004, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 122–133.
[30] E. Boros, Y. Crama, P.L. Hammer, Polynomial-time inference of all valid implications for Horn and related formulae, Ann. Math. Artif. Intell. 1 (1–4) (1990) 21–32.
[31] T. Eiter, K. Makino, On computing all abductive explanations from a propositional Horn theory, J. ACM 54 (5) (2007) 24.
[32] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (3) (2009) 1400–1415.
[33] M. Charikar, Y. Naamad, A. Wirth, On approximating target set selection, in: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.