Review Of Unique Key Horn Functions

Timothy Deligero

Abstract

Horn functions have been in many research papers under various topics, some of which discuss them in the topic of database theory, such as relational databases. Keys have also been important data items used in relational databases in order to uniquely assign values to a set of attributes, also known as the keys, allowing users to retrieve data based on given certain rules regarding the database and their keys. A pure Horn function can serve as functional dependencies of a given database, where given a Horn function h, a value for attribute v, a set of attributes *A*, and a key K of the database, $A \rightarrow v$ is an implicate of h and $K \rightarrow v$ is an implicate of h for all attributes *v*. This research paper will be a review and tutorial on the use of pure Horn functions, also defined as unique key Horn functions, in Sperner hypergraphs, also defined as unique key hypergraphs, to generate minimal keys within a database represented by the pure Horn functions proposed by researchers. The research paper also goes into detail, regarding minimal key generation in Sperner hypergraphs, of proof of why the hypergraphs with all edges, also known as hyperedges, as a size of two is co-NP-complete, several cases of where hypergraphs can be solved in polynomial time, and an algorithm that generates a minimal key set from a pure Horn function with polynomial delay along with proofs of how this problem is as hard as the MINIMAL TARGET SET SELECTION problems with polynomial bounds, creating similar algorithms. These concepts presented by the reviewed research paper will then be discussed of their major contributions and strengths, any weaknesses or questions to its validity, and potential extensions that could be made of the presented research.

Index Terms

Horn function, Key, Key Horn function, Pure Horn function, Unique key Horn function, Unique Key graph, Sperner family, Sperner hypergraph, Minimal key generation, Minimal Target Set Selection

1 Introduction

Horn functions were defined in the the theory of special functions in mathematics as 34 distinct hypergeometric series of order two. This definition was enumerated by Horn, hence its name, in 1931 and then was later corrected by Borngässer in 1933. Overtime, the use of Horn functions has been presented and defined in many research papers ranging in various topics. The variety of research papers discussing the use of Horn functions can be attributed to its interesting structural and computational properties when viewed as form of subclass of Boolean functions. Pure Horn functions have even been used in databases as a form of functional dependencies, where unique relationships are established between two attributes within a database as a way of forming distinction between groups of attributes, which can consist of anything depending on the purpose of the database. More of these concepts and definitions will be discussed when going into detail of the established properties of Horn functions in terms of Boolean functions.

On the topic of databases, *keys* have been a major aspect used in this topic, especially in relational databases. This data item is used as a set of attributes/columns to be assigned to a set of values/records, that can be represented as rows. With each key, it creates unique items and relationships between the keys and the records/values it holds. This allow users to fetch or retrieve from the database based on certain conditions and requirements regarding its keys. Some of these conditions include not holding duplicate records or values and not storing NULL values within a key. The types of keys used in databases fall into many different categories, one of which is the *unique* category. *Unique keys* can contain NULL values and contain a non-clustered index, which means that they can be stored multiple times across a database separately.

The authors Bérczi, Boros, Čepek, Kučera, and Makino have proposed the use of *unique key Horn functions* $h_{\mathcal{B}}$ as a form of pure Horn functions to represent a minimal key set formed by a Sperner hypergraph $\mathcal{B}$, otherwise defined as a *unique key hypergraph* by the research paper. [1] Along with defining the characterizations of unique key hypergraphs and unique key Horn functions, the research paper [1] proves that unique key graphs are co-NP-complete when all of its edges are of size two and certain cases of hypergraphs that can be solved in polynomial time. The research also provides connections between both problems of generating minimum key sets and minimal targets sets in terms of similar algorithms and hardness in terms of computational complexity.

2 Background And Related Work

Before going into the main concepts the research paper [1], some of the terminology and definitions used for Horn functions should be explained to give context to its intended use from the authors. It is denoted that V is a set of n variables, where its members are considered *positive literals* and its negations as *negative literals*. [1] *Boolean functions* can be mapped as $f : \{0,1\}^V \rightarrow \{0,1\}$, where its values are either 0 or 1 when it comes to its variables, and can be represented in a *conjunctive normal form* (CNF), where its CNF is defined as $\Phi = C_1 \wedge ... \wedge C_q = \{C_1, ..., C_q\}$, where q is the number of clauses. As mentioned previously, Horn functions can be represented as Boolean functions represented as CNFs, where a CNF Φ is considered to be a *Horn* if its clauses contain at most one positive literal variables and a *pure Horn* if each of its clauses contains exactly one positive literal. However, it should be noted that every CNF defines a Boolean function, but a Boolean function might have different CNF representations. [1] For example, if given a pure Horn CNF $\Phi = (\overline{a} \vee b) \wedge (\overline{a} \vee \overline{c} \vee d) \wedge (\overline{a} \vee \overline{c} \vee e)$, the Boolean function would consist of Boolean variables a, b, c, d, e, but if given the Boolean function of a pure Horn function with the same given Boolean variables, then the CNF can be defined as $\Psi = (\overline{a} \vee b) \wedge (\overline{a} \vee \overline{c} \vee d) \wedge (\overline{a} \vee \overline{c} \vee e)$. Pure Horn functions can also be represented with implications, especially within their CNF counterparts. An implication consists of a body A and a head v, where the implication comes in the form of $A \rightarrow v$ and in a clause of a CNF as $(A \vee v)$, where A consists of none or at least one negative literals and v consists of exactly one positive literal, according to the rules established of a pure Horn CNF. Examples of this are $a \rightarrow b$ and $b, d \rightarrow e$ of pure Horn implications would be represented as $(\overline{a} \vee b)$ and $(\overline{b} \vee \overline{d} \vee e)$ in their CNF representations, respectively.

As discussed previously about Horn functions, there have been various topics that have discussed Horn functions in different uses, such as:

- Directed hypergraphs in graph theory and combinatorics.
- Implication systems in machine learning.
- Database theory (e.g. relational databases as functional dependencies).
- Lattices and closure systems in algebra and concept lattice analysis.
- Hydra functions, where the bodies of the Horn functions are of size two, or in a CNF representation have clauses containing two literals.

Given its established properties, Horn functions have strong relations to databases, as its algorithmic problems give the same type of context to problems arising from databases and its implications are used in a similar manner as to assigning unique values to a set of attributes, or keys, to create functional dependencies in a database. For example, if a set of attributes $A \subseteq V$ and a given value $v \in V$, then the implication $A \rightarrow v$ is created if the attribute values of A uniquely determine the value v, the database records. Accordingly, a subset K of the variables is a key of a Horn function h if $K \rightarrow v$ is an implicate of h for all $v \in V \backslash K$ (set exclusion), or all values of v that are in V and not in K. [1] Given this definition for a key, a *key Horn* is defined as a pure Horn function if any of its implicates within its body, or any of its its clauses of its CNF representation, is a key of the function, which is the equivalent of assigning a key to a set of values to create unique identifiers within a relational database. Minimizing the CNF representation of a pure Horn functions is difficult, as the process is dependent on many variables, such as number of literals, number of clauses, etc. Previous research has denoted that established algorithms, even for special cases, such as the discussed hydra functions, have been proven to be NP-hard. The research paper [1] gives algorithms for minimizing keys in a different context with using unique key Horn functions and their correlated unique key hypergraphs and give proof to their hardness and time complexities.

3 Methods and Results

This section will be devoted to the the characteristics of unique key Horn functions and unique key hypergraphs, as well as the proof for hardness of generating certain unique key graphs and the algorithms to generate minimum key sets from pure Horn functions. These concepts are established by multiple Lemmas, Theorems, and Corollaries throughout the research paper [1] by the authors along with elaborate proofs that will be summarized in this paper.

3.1 Unique Key Horn Functions

Before going into details of the unique key Horn functions, it should be established of the type graphs used to correlate to such functions, being Sperner hypergraphs. Just to get an idea of terminology used, a *Sperner family* is a family F of subsets of a finite set *E*, where none of the sets contain one another, and a *hypergraph*, otherwise used in notation as $H = (V, E)$, where V is a set of Nodes/vertices and E is a set of non-empty subset of V called hyperedges/edges, is a generalization of a graph, when an edge can join any number of vertices. In a way, the edges could be considered sets containing vertices grouped together where each edge is connected between all the vertices within its set. The size of the vertex set V would be considered the order of the hypergraph, and the number of edges E would be considered the size of the hypergraph. Given these two terms, a *Sperner hypergraph*, also known as clutters, given a notation of (V, E), or how the research paper [1] defines it as $\mathcal{B} \subseteq 2^V$, where V is already defined to be the set of Boolean variables, is where given subsets A, $B \in E$, $A \nsubseteq B$ and $A \neq B$ (i.e. no hyperedge properly contains one another, where B is a hyperedge, or none of the hyperedges and their sets of contained vertices that are connected together contain duplicates or overlapping of one another). The research paper [1] also makes additional definitions and notations for Sperner hypergraphs:

- A *transversal* $T \subseteq V$ of $\mathcal{B}$ is when $T \cap B \neq \varnothing$, where $\forall B \in \mathcal{B}$, or has a nonempty intersection with every edge.
- An *independent set* S of $\mathcal{B}$ is when $T = V \backslash S$ is a transversal of $\mathcal{B}$. Independent sets in graph theory are also referred to as a set S of vertices where there are no edges contained between a pair of vertices within the same set S. A *maximal independent set* would be a case of an independent set where it is not a subset of another independent set.
- $\mathcal{B}^d$ is denoted as a set of *minimal transversals* of $\mathcal{B}$, with $\mathcal{B}^*$ being a *family of independent sets*. A minimal transversal T contains no proper subset of itself that is a transversal. d in this case would be the edge set that consists of all minimal transversals. The set of minimal transversals $\mathcal{B}^d$ can also be considered as the complement of the family of independent sets $\mathcal{B}^*$, or more specifically the maximal independent sets.
- A *subhypergraph* of $\mathcal{B}$ *induced by* S, where $S \subseteq V$, is denoted as $\mathcal{B}_S = \{B \in \mathcal{B} \mid B \subseteq S\}$. If $S \in \mathcal{B}^*$, then $\mathcal{B}_S = \varnothing$.
- A *projection of* $\mathcal{B}$ *to* S is denoted as $\mathcal{B}^S = min'l\{S \cap B \mid B \in \mathcal{B}\}$, where $min'l\{H\}$ denotes the family consisting of inclusionwise minimal members of H. If S is not a transversal of $\mathcal{B}$, then $\mathcal{B}^S = \varnothing$.
- Notation $\cup\mathcal{B}$ is denoted as the union of hyperedges of a hypergraph (i.e. $\cup\mathcal{B} = \bigcup_{B \in \mathcal{B}} B$).

Lemma 1. 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 $(\mathcal{B}^S)^d = (\mathcal{B}^d)_S$. [1]

Lemma 1 is a well known established logic for hypergraphs that represents that the minimal transversals by set d of a projection of the Sperner hypergraph $\mathcal{B}$ induces by S is equivalent to the minimal transversals by set S of the minimal transversal by set d of the Sperner hypergraph $\mathcal{B}$. The same equivalence applies if the subsets were reversed. This can be understood by applying the properties of a transversal, a set of minimal transversals, and projections of hypergraphs.

The research paper [1] also establishes additional notations for pure Horn functions and their CNF and Boolean function representations:

- $T(h)$ is a set of true vectors of a Boolean function h, in other words in mathematical notation $T(h) = \{x \in \{0,1\}^V \mid h(x) = 1\}$.
- $h \leq h'$ if $\forall x \in \{0,1\}^V$, $h(x) \leq h'(x)$ or $T(h) \subseteq T'(h)$.
- A clause of a CNF in the form of $A \to v = v \ \vee \ \bigwedge\limits_{a \in A} \overline{a}$, representing the clauses of a pure Horn CNF, is an *implicate* of h if $(A \to v)(x) \geq h(x)$, where $\forall x \in \{0,1\}^V$.

- A *forward chaining closure* of a subset $S \subseteq V$ is defined to be $F_h(S) = \{u \in V \mid S \to v$ is an implicate of h. For Boolean functions, if $h' \leq h$, then $F_h(S) \subseteq F'_h(S)$, and the same logic applies to CNFs, where if $\Phi \subseteq \Psi$, then $\Phi \geq \Psi$. This also means that in terms of Sperner hypergraphs, $\mathcal{B} \subseteq \mathcal{K}(h)$ implies $h \leq \Phi_{\mathcal{B}}$.
- Given a Sperner hypergraph $\mathcal{B} \subseteq 2^V$, its associate pure Horn CNF is denoted as

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

- $\mathcal{K}(h)$ is the *set of minimal keys* of h and $\mathcal{K}(\Phi_{\mathcal{B}}) = \mathcal{B}$ represents the unique key Horn function's CNF representation and its associated unique key hypergraph. (Note: It is possible that several pure Horn functions in the form of CNFs can have the same family of keys with the same minimal key set from a Sperner hypergraph $\mathcal{B}$.)

Lemma 2. For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$ 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 $K(h) \neq \mathcal{B}$ if and only if there exists an implicate $A \to v$ of h and a minimal transversal $T \in \mathcal{B}^d$ such that $T \cap A = \varnothing$ and $v \in T$. [1]

For the side of $\mathcal{K}(h) \neq \mathcal{B}$ in the bidirectional implication, this can be proven by knowing this property would mean $\exists K \in \mathcal{K}(h) \backslash \mathcal{B}$ and that $K \in \mathcal{B}^*$, where K is a minimal key as already established by the definitions and notations of Sperner hypergraphs. Given a maximal independent set W that contains K since it's a part of a family of independent sets, a minimal transversal $T = V \backslash W$ would not include any elements from K. This would mean $K \to v$ would be an implicate of h as its already established to be a key, and given Lemma 2, $K = A$, which compliments $T \cap A = \varnothing$. The other side can be proven after assuming its properties, given an edge $B \in \mathcal{B}$, then $T \cap B = \{v\}$. Using the properties of a forward chaining closure of the pure Horn CNF, this implies $F_{(A \to v) \wedge \Phi_{\mathcal{B}}}(V \backslash T) = V$, and since $h \leq (A \to v) \wedge \Phi_{\mathcal{B}}$, then $F_h(V \backslash T) = V$ holds. This means $\exists K \subseteq V \backslash T$ of h, and since $K \in \mathcal{K}(h) \backslash \mathcal{B}$ as $K \in \mathcal{B}^*$ as previously established for the other side, then $\mathcal{K}(h) \neq \mathcal{B}$.

Lemma 3. For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$ 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 $K(h) \neq \mathcal{B}$ if and only if for all implicates $A \to v$ of h with $A \in \mathcal{B}^*$ we have $v \in (V \backslash A)(\cup \mathcal{B}^{V \backslash A})$. [1]

Lemma 4. For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$ and define $\Psi = \{A \to v \mid A \in \mathcal{B}^*, v \notin (V \backslash A)(\cup \mathcal{B}^{V \backslash A})\}$. Let φ be a set of clauses of the form $A \to v$ that are not implicates of $\Phi_{\mathcal{B}}$. Then $K(\varphi \wedge \Phi_{\mathcal{B}}) = \mathcal{B}$ if and only if $\varphi \subseteq \Psi$. [1]

Both Lemma 3 and 4 can be proven by applying the properties and established proofs of Lemmas 1 and 2. Given that all implicates are a part of the family of independent sets $\mathcal{B}^*$, this means that they are disjoint from all minimal transversals of $\mathcal{B}^d$, where $T \cap A = \varnothing$ and $v \notin T$. By applying the logic that $(\mathcal{B}_S)^d = (\mathcal{B}^d)^S$ and $(\mathcal{B}^S)^d = (\mathcal{B}^d)_S$, we have $(\mathcal{B}^{V \backslash A})^d = (\mathcal{B}^d)_{V \backslash A}$ and $\cup(\mathcal{B}^d)_{V \backslash A} = \cup \mathcal{B}^{V \backslash A}$, which compliments $v \in (V \backslash A)(\cup \mathcal{B}^{V \backslash A})$.

Theorem 5. For a Sperner hypergraph $\mathcal{B} \subseteq 2^V$, the pure Horn function $h = \Phi_{\mathcal{B}}$ is the only one with $K(h) = \mathcal{B}$ 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 $T' \subseteq T \cup \{v\}$. [1]

Both sides of the bidirectional implication of Theorem 5 can be proven by applying Lemma 4 and taking arbitrary values for the if direction $A \in \mathcal{B}^*$ and $v \in \cup \mathcal{B}^{V \backslash A}$, and for the only if direction $T \in \mathcal{B}^d$ and

$v \notin T$, where $A = V \backslash (T \cup \{v\})$. For the if direction, by making the assumption that a minimal transversal $T' = (T \backslash \{u\}) \cup \{v\}$ exists within the Sperner hypergraph, where $u \in T$, and implying an edge B' that exists in the Sperner hypergraph such that $B' \backslash A \subsetneq B \backslash A$ and $v \notin B' \backslash A$, this results in $B' \cap T = \varnothing$, which contradicts T' being a minimal transversal, and compliments that, by Lemma 4, $\Psi = \varnothing$ and the unique key Horn function h is unique. For the only if direction, by knowing that the contradiction of Lemma 4 results in h not being a unique key Horn function, it shows that you get $v \in \cup B^{V \backslash A}$ and $T \cup \{v\} = \cup B^{V \backslash A}$. This results in an edge B that exists within the Sperner hypergraphs, where $B \backslash A$ is a minimality with $v \in B$, and every edge different from B, being $B' \in \mathcal{B}$, would be either $B' \cap (T \backslash \{u\}) \neq \varnothing$ or $v \in B'$, which compliments that T' from Theorem 5 is a transversal of of $\mathcal{B}$.

Corollary 6. The cuts of a loopless matroid form a unique key hypergraph. [1]

A matroid in mathematics is a generalization for vector spaces and graphs that is defined by independent sets, where we have a ground set E as the collection of elements, and $\mathcal{I}$ as the collection of independents sets, which follow certain rules and requirements within a matroid, such as $\varnothing \in \mathcal{I}$ and if $A \in \mathcal{I}$ and $B \subset A$, then $B \in \mathcal{I}$. For unique key hypergraphs, the ground set E would be the set of edges. The cut-sets of a matroid is where if edges where deleted from the graph, then the number of connected components increases, which can result in multiple circuits and is applicable to Sperner hypergraphs. A loopless matroid would be case of edge sets containing only circuits and no cycles to result in a selection of edges ending up at the same vertex. Corollary 6 can be proven simply by knowing that the $\mathcal{B}$ and $\mathcal{B}^d$ can represents the cuts-sets and base-sets of a matroid, and a loopless matroid would mean Sperner hypergraph $\cup \mathcal{B}^d = V$, all complimenting to Theorem 5. However, it should be kept in mind that this implication doesn't hold in the opposite direction, meaning that all unique key hypergraphs are related to matroid. An example of this is if $E = \mathcal{B} = \{12, 13, 14, 234\}$, where $V = \{1, 2, 3, 4\}$, as it is an example of a unique key hypergraph, but isn't a matroid as it violates the rules of its independent sets. Remark 7 serves a case of solving the problem presenting in the research paper [1] in polynomial time given certain conditions, one of which involves matroids.

Remark 7. The conditions of Theorem 5 can be checked in polynomial time if $\mathcal{B}^d$ can be generated in (input) polynomial time from $\mathcal{B}$. For example, if $\mathcal{B}$ is 2-monotone or forms the set of bases of a matroid.

3.2 Unique Key Graphs

This section is dedicated to giving proof on finding the minimal key set of a Sperner hypergraph with edge sizes of two, or each edge set of a hypergraph contains two connected vertices, is co-NP-complete, as well as examples of finding minimal key sets of certain hypergraphs can be solved in polynomial time, being bipartite graphs, bounded treewidth graphs, and graphs with small induced matchings under certain conditions an requirements. Bipartite graphs have been used to represent hypergraphs before, although its discussion with Sperner hypergraphs, especially in this context, seems to be more recent in this research. However, from the research paper [1], it is not made known if hypergraphs were discussed along with bounded treewidth graphs or graphs with small induced matchings, which compliments the new discussions made when concerning pure Horn functions and associating them with database theory, or graphs. First, lets assume that all of $B \in \mathcal{B}$, as they represent the edges within a Sperner hypergraph, where $G = (V, E)$ and $\mathcal{B} = E$, are of size two, meaning $|B| = 2$ and contains two connected vertices. If the Sperner hypergraph G is a unique key graph, then $\mathcal{B}$ is a unique key hypergraph. Lets also give the notation of neighboring vertices, where $N(u) = \{v \in V \mid (u, v) \in E\}$ is the set of neighbors for $u \in V$, and can be applicable to the subset $S \subseteq V$ of Boolean variables, where $N(S) = (\bigcup_{u \in S} N(u)) \backslash S$ is the set of neighbors for S.

3.2.1 Complexity

Before going into the complexity of unique key graphs, it should be noted that, given a maximal independent set $I \subseteq V$ and a graph $G = (V, E)$, $u \notin I$ is a neighbor of v if $N(u) \cap I = \{v\}$. To give an idea of the type of complexity used to describe these types of Sperner hypergraphs with edge sizes of two, co-NP, in terms of com basically means that there exists a polynomial time algorithm to solve the problem in terms of ”NO” instances, considering they are complement to NP problems where they solve the ”YES” instances in polynomial time. The co-NP-complete problems are considered the hardest out of the co-NP, in the sense that any co-NP problem can be reformulated into a co-NP-complete problem with polynomial overhead.

Theorem 8. A graph $G = (V, E)$ is unique key 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. [1]

Theorem 8 can simply be proven after knowing that its already established that independent sets, including the maximal independent sets, are the complements of minimal transversals, which means that the minimal transversals also represent the minimum vertex covers of the unique key graph G. Given the established property of neighboring vertices for independent sets, the set $(I \backslash \{v\}) \cup \{u\}$ can only be independent if u is an individual neighbor of v, which results in a maximal independent set I' of G that doesn't contain v, complimenting Theorem 5.

In order to prove that recognizing that $\mathcal{B}$ has a set of minimal keys from a unique key Horn function with hypergraphs containing edges of size two, the established properties for graphs and their CNF representation is needed. Along with these characterizations, the research paper [1] establishes a set of vertices $V(G_\Phi) = \{x_i, \overline{x}_i, y_i \mid i = 1, ..., n\} \cup \{C_j \mid j = 1, ..., m\}$, where C_j vertices represent the clauses of the CNF formula with edges formed by vertices $x_i, \overline{x}_i$, and y_i into a triangle, all vertices C_j in the range of $j = \{1, m\}$ and z forming a clique, and vertices C_j are connected to literals. An example of this graph is shown in Figure 1.

Theorem 9. A CNF Φ is not satisfiable if and only if the graph G_Φ is unique key. [1]

For a CNF, Boolean satisfiability, otherwise known as the SAT problem, would mean that an interpretation of the Boolean variables to be replaced with a value of TRUE or FALSE will result in the CNF formula evaluating to TRUE. If not, then CNF formula would be evaluated to FALSE and would regarded as not satisfiable. The basis for proof of Theorem 9 revolves around the maximal independent sets within $V(G_\Phi)$, where they contain exactly $n + 1$ vertices as they can contain at most one vertex from the cliques represented in the graph, and even if it is disjoint from the set of vertices or creates an empty set when intersected with the set of keys, then $I \cup \{y_i\}$ and $I \cup \{z\}$ are considered to be independent as well, respectively. The proof shown by the research paper [1] basically proves the opposite of the statement, meaning that a CNF Φ is satisfiable if and only if the

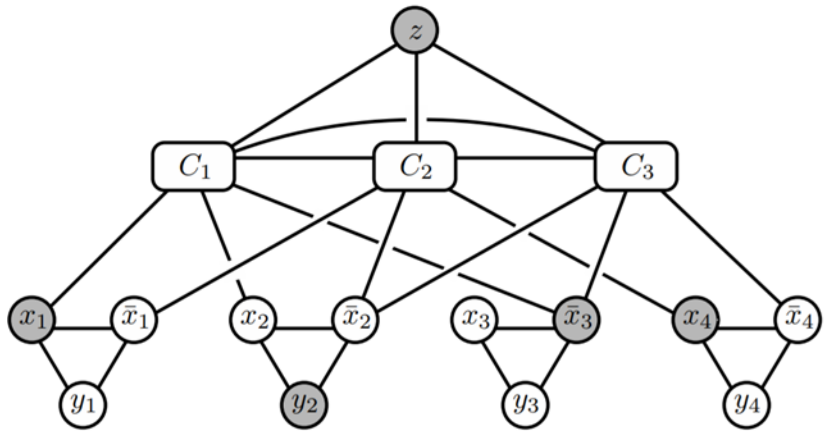

Fig. 1. The graph G_Φ corresponding to CNF formula $\Phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)$. Grey vertices form a maximal independent set corresponding to a satisfying truth assignment.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

graph G_Φ is not a unique key, which also means that there exists a maximal independent set I that contains vertex z with no individual neighbor. Given that Φ is satisfied, that would mean the every clause in C_j would be satisfied, resulting in neighboring vertices with I rather than z, leaving z with no individual neighbor. Then, lets assume that z has no individual neighbor and there exists a maximal independent set I. This would mean that every clause in C_j must have a neighbor with I, which must be a literal set to TRUE, given that edges do exist between a complementary pair of vertices and cannot be contained within I. This results in a satisfiable CNF Φ, and proves the statement for both directions of the implication. This type of proof also leads into Corollary 10 where it states recognizing a unique key hypergraph with edges of size two is co-NP-complete, as a polynomial algorithm is given that solves the ”NO” instances for unique key hypergraphs.

Corollary 10. Deciding if a hypergraph is unique key is co-NP-complete already for hypergraphs of dimension 2. [1]

3.2.2 Bipartite Graphs

Bipartite graphs consist of two disjoint and independent vertex sets where every edge connects a pair of vertices from one set to another. This also means that no edges exists between a pair of vertices within the same set. Hypergraphs can be represented as bipartite graph, also known as incidence graphs in this type of context, if it contains fixed parts and no unconnected vertices.

Theorem 11. A bipartite graph $G = (V, E)$ without isolated vertices is unique key if and only if G is a perfect matching. [1]

A perfect matching edge set, also known as the independent edge set, is a case where each vertex has only one edge coming and out of it. The largest number of possible edges in a perfect matching graph is $V/2$, meaning only graphs with an even number of vertices is possible to create a matching edge set. A perfect matching bipartite graph is considered to be a unique key as every maximal independent set I contains exactly one end vertex for every edge in E, and all vertices have an individual neighbor. A bipartite graph would be considered a unique key if there are no isolated vertices and the two disjoint and independent sets are considered maximal independent sets of a hypergraph, results in vertices have exactly one neighbor. This matches the exact definition of a perfect matching E.

3.2.3 Bounded Treewidth Graphs

The treewidth graphs consist of undirected edges with numbers associated with. This type of association is used for many applications commonly for the purpose of parameterized complexity analysis in graph algorithms. For unique key graphs and their CNF representations, the number associated with bounded treewidth graph should be clique-width size in this type of context.

Theorem 12. For graphs of bounded treewidth, it is possible to decide in linear time if a graph is a unique key graph. [1]

Theorem 12 can be proven knowing that unique key graphs require independent sets I with pairs of vertices u and v connected by a single edge with both their individual neighbors. The choice of whether or not a bounded treewidth graph is a unique key can be solved if the following predicates are satisfied defined by the research paper [1]:

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

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

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

This also leads into Corollary 13, as bounded treewidth graphs can also be defined with clique-width as mentioned before.

Corollary 13. For graphs of bounded clique-width, it is possible to decide in linear time if a graph is a unique key graph. [1]

3.2.4 Small Induced Matchings Graphs

An induced matching, or strong matching, in graphs is where a subset of edges of an undirected graph do not share any vertices other than its single pair of vertices connected in between within its subset. This definition matches the independent sets used for unique key graphs which compliments the topic.

Theorem 14. 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. [1]

Given that the size of the induced matching is proportional to the size of the graph p, where p is the number of edges, then there is at most n^{2p} independent sets. If the size of p is bounded by constant, then all the independent sets can be checked in polynomial time, which compliments the properties of Theorem 8 in finding if the graph is a unique key graph.

3.3 Minimal Key Generation

This section will be dedicated to the algorithms and solutions of the MIN-KEY and MIN-TSS problems of unique key Horn functions and their corresponding unique key graphs. The time complexity of generating a minimal key set can be potentially exponential due to many factors of the CNF representation of the pure Horn function, such as number of literals, clauses, etc., so the efficiency measured is depending on time spent between generating two keys. This type of algorithm presented by the research paper [1] is a *generated algorithm* that outputs the questions one by one without repetition, in a similar manner to *polynomial delay* where the time spent between the two consecutive outputs is computed and bounded by a polynomial based on the input size.

3.3.1 MIN-KEY

Lets first associate the pure Horn CNF Φ with a directed graph $D_\Phi = (\mathcal{K}(\Phi), E)$. Lets define a minimal key as $K \in \mathcal{K}(\Phi)$, an arbitrary value as $v \in K$, and a clause as $A \rightarrow v \in \Phi$. We define a key of S of Φ as $S = (K - v) \cup A$, where there also exists a key $K' \in \mathcal{K}(\Phi)$ and $K' \subseteq S$. K' can be found using a greedy procedure by removing values from S one-by-one, while also checking if the remaining set results in a key that exists within Φ. The research paper [1] makes a few notes regarding this greedy procedure:

- The edge $KK' \in E$ is included as part of the greedy procedure.
- The clause $A \rightarrow v$, where $v \in K$, may not exist to produce the key K'.
- Each vertex v within the directed graph D_Φ has at most m outgoing edges.
- The final graph D_Φ is determined to be a unique key graph based on the K' keys found in the greedy procedure.

Lemma 15. D_Φ is strongly connected. [1]

A directed graph is considered to be strongly connected if there exists between each pair of vertices within a graph. Lemma 15 can be proven by first an average measurement, in terms of vertices, between two minimal keys $K_1, K_2 \in \mathcal{K}(\Phi)$. This is computed by dividing the path into layers L_i, where $i = 0, 1, ..., t$, then calculating the average distance between the two keys in terms of the sum of similar values visited in each current layer and the minimal key K_1. A claim is made that there exists a minimal key K_3 that is an out-neighbor of K_1, where the average distance between K_3 and K_2 is smaller than K_2 and K_1. It is proven that $K_3 \subseteq S$ by the properties given to the directed graphs of pure Horn CNFs. This results in a directed path from K_3 to K_2 in D_Φ, and since there is an edge between K_1 and K_3, then the same logic applies to K_1.

Theorem 16. Given a pure Horn CNF Φ, we can generate all minimal keys of Φ with polynomial delay. [1]

Using Lemma 15, an algorithm can be produced that generated all minimal keys of a pure Horn CNF with polynomial delay according to the variables and clauses of the CNF using these steps:

1) Given that D_Φ is strongly connected, then all out-neighbors will be generated from the minimal keys that are already generated, starting from a minimal key which is generated by greedily leaving out elements from V. [1]
2) Store the minimal keys in a LIFO queue.

Fig. 2. Illustration of Theorem 17. The CNF associated to G is $\Psi_G = (b \to a) \wedge (e \to a) \wedge (d \to a) \wedge (a \to b) \wedge (c \to b) \wedge (b \to c) \wedge (d \to c) \wedge (e \to c) \wedge (a \to d) \wedge (c \to d) \wedge (e \to d) \wedge (\{a, c\} \to e) \wedge (\{a, d\} \to e) \wedge (\{c, d\} \to e)$. The left graph is an instance of the MIN-TSS problem, where the threshold are $t(a) = t(b) = t(c) = t(d) = 1$ and $t(e) = 2$, and the right graph is a construction of Φ_G with thick hyperedges representing clauses containing three variables. [1]

3) Generate the out-neighboring vertices/values of the top element of the queue and add the new ones to the queue.
4) Output the top element of the queue.
5) Repeat Steps 3) and 4) until all minimal keys are generated.

3.3.2 MIN-TSS

This section will be dedicated to showing the strong relations between the MIN-KEY and MIN-TSS problems, including their time complexities and similar algorithms. The minimum target set selection problem involves finding a minimum size initial set of Nodes/vertices S, also known as the *target set*, that will eventually activate the entire graph. The given graph is defined to be an undirected graph $G = (V, E)$ and we are also given a threshold function $t : V \to \mathbb{Z}_+$. Initially, the subset $S \subseteq V$ will be activated before activating the rest of the neighboring vertices. A vertex v becomes activated if at least $t(v)$ of its neighbors are already active.

Theorem 17. The MIN-TSS problem with constant thresholds is polynomial-time reducible to the MIN-KEY problem. [1]

To prove Theorem 17, lets assume the given properties of the undirected graph G and its threshold function $t(v)$ as a part of the MIN-TSS problem. Lets also denote all neighboring vertices as $N(v) \subseteq V$. A Horn CNF can be constructed by the following equation and can be solved in polynomial time as shown in Figure 2:

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

By applying the same definitions and generation algorithm for the MIN-KEY problem from Theorem 15 for Φ_G, it matches the activation process of the undirected graph G for Ψ_G. This results in that the key $K \subseteq V$ is a target set of G if and only if it is a unique key graph.

Theorem 18. The MIN-KEY problem with constant thresholds is polynomial-time reducible to the MIN-TSS problem. [1]

Fig. 3. Illustration of Theorem 18. Note that the size of the graph G is polynomial in the length of the input. The left graph shows a pure Horn clause $C = A \to v$, where $A = \{a, b, c\}$. and the right graph shows the gadget and threshold values corresponding to $C = A \to v$. [1]

Theorem 18 can be proven by constructing a graph $G = (V', E)$ based on a given pure Horn CNF Φ with its variables V to have all minimal key sets be also minimal target sets of G, while also having the minimal target sets of G be able to transform into minimal keys in a CNF Φ without any changes to its size. First, the graph G is constructed with the V values as vertices $v \in V$, where all vertices from the CNF have threshold values of $t(v) = 1$. The constructed graph will include gadgets for every clause, where p^C will be added as a vertex corresponding to an individual clause $C = A \to v$ with a threshold value of $|A|$. Then, gadgets will be connected between the vertices within A and v from p^C with edges, which consist of vertices of x_a^C, y_a^C, and z_a^C with threshold values of one and w_a^C with a threshold value of two. An example of this process is shown in Figure 3

Lets assume that a target set S exists within graph G. It is not guaranteed that its a key of a CNF Φ, but if the value of from the equation

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

results in $|K| \leq |S|$, then K is a key of Φ. Also, given the CNF Φ and its associated constructed graph G, its key $K \subseteq V$ forms the same target set in G. Finding the minimal keys set from G also follows the same procedure as the activation process of the minimal target set with the use of the gadgets. After proving both Theorems 16 and 17, it leads into Corollary 19, where both MIN-KEY and MIN-TSS problems can be solved with similar algorithms with polynomial delay.

Corollary 19. 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. [1]

4 Discussion and Summary

The introduction of unique key Horn functions and unique key graphs has major contributions to database theory and graph theory research, as well as the use of Horn functions in general in terms of its CNF or Boolean function representation computational and structure properties. Given how pure Horn CNFs have been used in many topics, this approach introduces a new, simple technique when regarding Sperner hypergraphs and their representations as pure Horn CNFs. Given the nature and properties of Sperner hypergraphs, it compliments database theory as it is structured divide data into edge sets containing vertices, similar to how databases store columns and rows of database records and separate them using keys for data retrieval by users. Since the pure Horn's implicates can be used as functional dependencies in a database, it is very much possible to associate them with Sperner hypergraphs and be very efficient and effective, as shown with presented proof and algorithms. The research paper [1] is also effective on elaborating the different definitions and notations of Sperner hypergraphs, pure Horn CNFs, unique key graphs, and unique key Horn functions when applying them to multiple Lemmas, Theorems, and Corollaries and their proofs. The presented cases of different types of graphs as unique key graphs under certain conditions, such as bipartite graphs, treewidth graphs, and small induced matching graphs give new representations of Sperner hypergraphs which extends the research on different models of hypergraphs. The algorithms presented for both MIN-KEY and MIN-TSS problems are both very simple and explained really well on their similarities based on their polynomial-time reductions. The introductions of the new gadgets containing vertices and threshold values for activation, when reducing the MIN-Key problem to the MIN-TSS problem, is creative and can introduce a new type of reduction when regarding other problems that can potentially reduce to the MIN-TSS problem regarding constructed graphs, while also proving that those certain problems can be solved with polynomial delay.

Although the research paper [1] itself was well made and effective in its objectives and goals, there are few weaknesses shown and criticisms that can be made. Keep in mind that the research paper itself was published very recently as of writing this report, so these issues can be attributed to being a considerably new introduction to the research regarding database theory and graph theory, as well as pure Horn functions, and may not be as relevant once researchers delve more into the presented topics here. Although the cases for certain graphs being unique key graphs under certain conditions are definitely valid in their proofs, there is very little variation between them when trying to regard them as similar to Sperner hypergraphs. It can be noticed that the proofs presented from these cases are highly dependent on the definitions of independent sets when concerned with the context given of neighboring vertices. While this may be a strong indicator that the independent sets of Sperner hypergraphs can be used to make cases where unique key graphs can exist under given conditions from different types of graphs, it does not explore anything outside of this property, thus lacking any meaningful variety when discussing cases of the graphs that can serve as Sperner hypergraphs, with the only difference being in how each graph functions outsides of its bounded conditions. Also, while the discussion of algorithms for both MIN-KEY and MIN-TSS are valid, the researchers do not go into detail of the polynomial delay in terms of the inputted CNF between outputs despite mentioning the exponential size of minimal keys in a pure Horn CNF. Even though this is a suitable analysis for the type of problems presented here and the given algorithms are both efficient and beneficial, considering that there is no exponential time taken on startup, a measurement of time for the polynomial delay between each outputted key could give a really good perspective on the true effectiveness and efficiency of unique key graphs and their correspond unique key Horn function based certain cases of inputted CNFs that produce different polynomial delays or overall execution times for generating all minimal keys and minimal target sets. Also, providing only one example is not beneficial to prove the case that the proposed approach can be applied to various problems, even though it is highly valid to do so given the presented proof and applications of pure Horn CNFs as unique key Horn functions.

Given that the publication of this research paper [1] itself is very recent by the time of making this review, there are potential future research and extensions that can be made to the presented topics here. A full survey paper discussing the many topics that have been done in regards of pure Horn CNFs, including the research presented here, can give an idea of the versatility of these functions with its computational and structure properties. There could also be open discussion on different types of graphs that can be represent as Sperner hypergraphs, and maybe even unique key graphs given the right conditions, giving opportunities to be represented as pure Horn CNFs. The research paper [1] also mentions the potential for this approach to be associated with various problems, so there is potential to prove that certain problems can be solved with polynomial delay given the right conditions and presenting new reductions between the MIN-KEY problem and other various problems that expand upon the versatility of the unique key Horn CNFs. There is even the open question of whether or not the minimal target sets can be generated in polynomial delay when unbounded, as the research paper [1] only provides proof for bounded examples of the MIN-TSS problem. Software implementations and a thorough analysis of the algorithms presented here in terms of time complexity and space complexity can be done in extended research to get a full perspective of their effectiveness and efficiency given different inputted CNFs presented in graphed results.

5 Conclusion

In conclusion, the research paper [1] defines and characterizes both unique key graphs and their corresponding unique key Horn functions through multiple Lemmas, Theorems, and Corollaries. Along with the introduction of these new definitions and notations, it provided proofs that Sperner hypergraphs with edges of size two is co-NP-complete when finding whether or not it is a unique key graph and for other cases of graphs, represented as Sperner hypergraphs, that can be solved in polynomial time if it is a unique key graph based on certain conditions and requirements. Algorithms were also provided for both minimal key generation and minimal target set selection problems that are related to graphs and their pure Horn function CNF representations, showcasing that both problems have strong relations with each other and are polynomial-time reducible to each other. Overall, the research done provides major contributions to graph theory and database theory, as well as pure Horn functions in general, as well as opportunities and potential for future research to extend upon the presented topics as well as attribute to the versatility of pure Horn functions.

References

[1] K. Bérczi, E. Boros, O. Čepek, P. Kučera, and K. Makino, "Unique key Horn functions," Feb. 2020.