Multi-Tuple Deletion Propagation: Approximations and Complexity

Benny Kimelfeld
IBM Research–Almaden
San Jose, CA 95120, USA
kimelfeld@us.ibm.com

Jan Vondrák
IBM Research–Almaden
San Jose, CA 95120, USA
jvondrak@us.ibm.com

David P. Woodruff
IBM Research–Almaden
San Jose, CA 95120, USA
dpwoodru@us.ibm.com

ABSTRACT

This paper studies the computational complexity of the classic problem of deletion propagation in a relational database, where tuples are deleted from the base relations in order to realize a desired deletion of tuples from the view. Such an operation may result in a (sometimes unavoidable) side effect: deletion of additional tuples from the view, besides the intentionally deleted ones. The goal is to minimize the side effect. The complexity of this problem has been well studied in the case where only a single tuple is deleted from the view. However, only little is known within the more realistic scenario of multi-tuple deletion, which is the topic of this paper. The class of conjunctive queries (CQs) is among the most well studied in the literature, and we focus here on views defined by CQs that are self-join free (sjf-CQs).

Our main result is a trichotomy in complexity, classifying all sjf-CQs into three categories: those for which the problem is in polynomial time, those for which the problem is NP-hard but polynomial-time approximable (by a constant-factor), and those for which even an approximation (by any factor) is NP-hard to obtain. A corollary of this trichotomy is a dichotomy in the complexity of deciding whether a side-effect-free solution exists, in the multi-tuple case. We further extend the full classification to accommodate the presence of a constant upper bound on the number of view tuples to delete, and the presence of functional dependencies. Finally, we establish (positive and negative) complexity results on approximability for the dual problem of maximizing the number of view tuples surviving (rather than minimizing the side effect incurred in) the deletion propagation.

1. INTRODUCTION

The practical need to allow for, and sometimes restrict to, database accesses through views gives rise to the *view update* problem: properly translate an update operation on the view to an update of the source relations. The core problem is that of underspecification—an update (e.g., tuple deletion) on the view can be realized by multiple, possibly very different, updates on the source relations. Therefore, a great deal of research has been devoted to designing syntax, semantics and restrictions to enable database systems to expose interfaces for updates through views. For example, Keller [15] defines a collection of (five) criteria qualifying the coherence of update translators, notions of *complement views* have been proposed for specifying update disambiguation and inversion [2,8], and the *relational lenses* [3,4] have been proposed as a language where view definitions include clear update policies to begin with.

The work reported in this paper is in another line of research within the view-update problem: underspecification is allowed, and the goal is to translate the update with as little as possible *side effect* [5–7, 9, 13, 16, 17]. A motivating scenario is in the context of database debugging: the user points out wrong tuples (false positives) or missing tuples (false negatives) in the result, and the proposed updated database serves as a suggestion of eliminating the errors while minimizing the side effect (e.g., on other results). Kimelfeld et al. [17] describe a use case of this motivation in the context of in information extraction.

Our focus here is on the special case of *deletion propagation.* There, we are given undesired tuples in the view, which is defined by a monotonic query, and the goal is to delete tuples (*facts*) from the base relations, so that the undesired tuples are no longer in the view. The resulting database is called a *solution.* The *side effect* is the set of view tuples that are not among the undesired ones, and yet, are also deleted due to the deletion from the base relations. If there is no side effect (i.e., only the undesired tuples are deleted from the view), then the solution is *side-effect free.* However, it is possible that no side-effect-free solution exists. Hence, the task is relaxed to that of *minimizing* the side effect [5–7, 17]: delete tuples from the base relations so that the undesired tuples disappear from the view and the side effect is of minimal cardinality. Such a solution is said to be *optimal.* Note that the measure of quality we adopt, the *view side effect*, is different from the *source side effect* [5–7], where an optimal solution is one with a minimal number of missing facts.

This paper continues the research on the theoretical computational complexity of deletion propagation for views defined by conjunctive queries [5–7, 16, 17]. The ultimate goal is, naturally, to gain the insights needed for designing efficient solutions with quality guarantees. With the exception of Cong et al. [6], the past research has been restricted to the special case where only a single tuple is deleted from the view. Only a little is known about the computational

Proceedings of the VLDB Endowment, Vol. 6, No. 13
Copyright 2013 VLDB Endowment 2150-8097/13/13... $ 10.00.

complexity in the (usually more realistic) case of multi-tuple deletion, which is the topic of this paper. We first shortly review the work on the single-tuple case. Later in this section, we relate the work here to the results of Cong et al. [6] on multi-tuple deletion.

Buneman et al. [5] showed NP-completeness of deciding on the existence of a side-effect-free solution (hence, NP-hardness of finding an optimal or approximately optimal solution) when the view is defined by the following self-join-free conjunctive query (abbrev. *sjf-CQ* here).

$$Q_2^{\star}(y_1, y_2) :- R_1(x, y_1), R_2(x, y_2) \tag{1}$$

Kimelfeld et al. [17] explore the space of all views defined by sjf-CQs and prove a dichotomy in the (data) complexity of single-tuple deletion propagation for views defined in that space. They do so by defining the property of *head domination* of a CQ, and show that this property fully characterizes the views with a tractable deletion propagation. More precisely, they prove the following dichotomy in complexity for the case where the view is defined by an sjf-CQ Q. If Q has head domination, then finding an optimal solution is in polynomial time; otherwise, it is NP-hard to find an optimal solution, and in fact any (finite) approximation thereof. Cong et al. [6] considered the complexity of the problem in the case of *key preserving* views. Later, Kimelfeld [16] generalized the results of Cong et al. [6], in the case of sjf-CQs, to a dichotomy in complexity that takes general functional dependencies into account.

We now proceed to describing the results we establish in this paper. We first show that the previously established dichotomy results for the single-tuple deletion [16,17] no longer hold in the multi-tuple case. Instead, we prove a *trichotomy* in complexity. More precisely, we show a classification of the sjf-CQs into the following three categories.

1. Those where finding an optimal solution is in polynomial time.
2. Those where an optimal solution is NP-hard to obtain, but for some constant k, a k-optimal solution (i.e., the side effect is at most k times the minimum) can be found in polynomial time.
3. Those where finding an α-optimal solution is NP-hard for all positive constants/functions α (since deciding on the existence of a side-effect-free solution is already NP-hard).

Although the dichotomy of [17] no longer holds in the multi-tuple case, our results do imply that, in some strong sense, the notion of head domination is robust enough to carry over to the general case. In particular, we parameterize their head domination into *level-k head domination*, for an integer $k \geq 1$ (exact definitions are in Section 2). We then show that when $k = 1$, we are in Case 1 above. And when $k > 1$, we are in Case 2. Consequently, Case 3 captures exactly those sjf-CQs that are hard (to approximate) already in the single-tuple case (namely, no head domination).

As part of our proof of the trichotomy, we define a generalization of the *minimal hitting-set* problem, the *focused hitting-set* problem, where the goal is to hit all the sets in one collection while hitting as few as possible sets in another collection. We show how to reduce deletion propagation with level-k head domination to that problem, where each set has k elements. We present a k-approximation algorithm for that variant of focused hitting set, through rounded linear programming.

Cong et al. [6] studied complexity aspects of multi-tuple deletion. Their variant of deletion propagation is different from ours—they aim at minimizing the side effect, and at the same time, do so *using the minimal number of deleted source tuples*. They show that the problem is NP-hard under combined (query-and-data) complexity when the view is defined by a CQ, even if the CQ preserves keys (i.e., the head variables include a key for each relation symbol involved in the query). Our results here show that finding an optimal solution can be NP-hard already under data complexity, and even if we do not impose any requirement on the source side effect. In fact, as mentioned earlier, our results show precisely for which sjf-CQs the problem is in polynomial time, hard but approximable, and inapproximable.

In addition to the above trichotomy in complexity, we study additional variants of the problem. In particular, we consider the *bounded deletion* variant of our problem, where the number of deleted tuples is bounded by some constant. We show that, unlike the unbounded case, the dichotomy of Kimelfeld et al. [17] continues to hold. We also generalize the results described thus far to take functional dependencies into account.

The negative part of our results indicates that, quite often, reducing the side effect in deletion propagation is theoretically intractable if we desire approximation guarantees w.r.t. the smallest possible side effect. A natural question is then whether there are efficient algorithms for other types of quality guarantees. For that, we consider the maximization variant of our problem, where the goal is to *maximize* the remaining view rather than to *minimize* the side effect. The two versions, side-effect minimization and view maximization, are the same if only optimal solutions are considered. However, the two imply different notions of approximation guarantees, and consequently, involve different complexities. A known polynomial-time approximation for sjf-CQs guarantees the factor h, which is the minimal number of atoms that cover all the head variables. In fact, there is always a solution (obtainable through a simple algorithm due to Buneman et al. [5]) that retains at least $1/h$ of the non-deleted tuples [17]. In contrast, here we show that in the case of multi-tuple deletion, such a solution does not necessarily exist. We further show that for unbounded deletion, a very general condition implies that no polynomial-time approximation algorithm guarantees any constant approximation factor, under a reasonable (yet sub-standard) complexity assumption made by Ambühl et al. [1]. Nevertheless, we establish positive results for maximization in the case of bounded deletion.

The paper is organized as follows. In Section 2 we give preliminary definitions and background. The focused hitting-set problem is introduced in Section 3, where we also give an approximation algorithm. We define leveled head domination in Section 4. In Section 5 we give the complexity results for minimizing the side effect. The maximization variant of the problem and the incorporation of functional dependencies are discussed in Section 6. Finally, we make concluding remarks in Section 7.

2. FORMAL SETTING

In this section we present the formal setting we will build upon throughout the paper.

2.1 Schemas and Instances

We fix an infinite set Const of *constants*. We denote constants by lowercase letters from the beginning of the Latin alphabet (e.g., a, b and c). A *schema* is a finite sequence $\mathbf{R} = \langle R_1, \dots, R_m \rangle$ of distinct relation symbols, where each R_i has an arity $r_i > 0$. An *instance* I (*over* $\mathbf{R}$) is a sequence $\langle R_1^I, \dots, R_m^I \rangle$, such that each R_i^I is a finite relation of arity r_i over Const (i.e., R_i^I is a finite subset of Const^{r_i}). If $\mathbf{c} \in \mathsf{Const}^{r_i}$, then $R_i(\mathbf{c})$ is called a *fact*, and it is a fact *of* the instance I if $\mathbf{c} \in R_i^I$. Notationally, we view an instance as the set of its facts. For example, we may write $R(\mathbf{c}) \in I$ to say that $\mathbf{c}$ is in R^I. As another example, $J \subseteq I$ means that J is a *subinstance* of I, that is, $R_i^J \subseteq R_i^I$ for all $i = 1, \dots, m$.

2.2 Conjunctive Queries

We fix an infinite set Var of *variables*. We assume that Var and Const are disjoint sets. We denote variables by lowercase letters from the end of the Latin alphabet (e.g., x, y and z). We use the Datalog style for denoting a conjunctive query (abbrev. CQ); that is, a CQ over a schema $\mathbf{R}$ is an expression of the form $Q(\mathbf{y}) :\!-\ \Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$, where $\mathbf{x}$ and $\mathbf{y}$ are disjoint tuples of variables (from Var), $\mathbf{c}$ is a tuple of constants (from Const), and $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is a conjunction of atomic formulas $\varphi_i(\mathbf{x}, \mathbf{y}, \mathbf{c})$ over $\mathbf{R}$; an atomic formula is also called an *atom*. We may write just $Q(\mathbf{y})$, or even just Q, if $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ is irrelevant. We denote by $\text{atoms}(Q)$ the set of atoms of Q. We usually write $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ by simply listing the atoms of Q. We require every variable of $\mathbf{y}$ to occur at least once in $\Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$. The *arity of* Q, denoted $\text{arity}(Q)$, is the length of the tuple $\mathbf{y}$.

When we mention a CQ Q, we usually avoid specifying the underlying schema $\mathbf{R}$, and rather assume that this schema is the one that consists of the relation symbols that appear in Q (and each symbol has the arity it takes in Q). When we want to refer to that schema, we denote it by $\text{schema}(Q)$.

Let $Q(\mathbf{y}) :\!-\ \Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$ be a CQ. A variable in $\mathbf{x}$ is called an *existential* variable, and a variable in $\mathbf{y}$ is called a *head* variable. We use $\mathsf{Var}_\exists(Q)$ and $\mathsf{Var}_\mathsf{h}(Q)$ to denote the sets of existential variables and head variables of Q, respectively. Similarly, if φ is an atom of Q, then $\mathsf{Var}_\exists(\varphi)$ and $\mathsf{Var}_\mathsf{h}(\varphi)$ denote the sets of existential and head variables, respectively, that occur in φ. We denote by $\mathsf{Var}(Q)$ and $\mathsf{Var}(\varphi)$ the sets of all variables that occur in Q and φ, respectively (that is, the unions $\mathsf{Var}_\exists(Q) \cup \mathsf{Var}_\mathsf{h}(Q)$ and $\mathsf{Var}_\exists(\varphi) \cup \mathsf{Var}_\mathsf{h}(\varphi)$, respectively).

The focus of this paper is on CQs that are *self-join free*, which means that each relation symbol occurs at most once in the CQ. We use *sjf-CQ* as an abbreviation of *self-join-free CQ*. If R is a relation symbol that occurs in the sjf-CQ $Q(\mathbf{y}) :\!-\ \Psi(\mathbf{x}, \mathbf{y}, \mathbf{c})$, then the unique atom over R is denoted by $\varphi_R(\mathbf{x}, \mathbf{y}, \mathbf{c})$ or just φ_R. Similarly, R_φ denotes the relation symbol of the atom φ.

EXAMPLE 2.1. We give here examples of sjf-CQs that we later reference. First, consider the CQ $Q_2^\star$ in (1). The atoms of $Q_2^\star$ are $\varphi_1 = R_1(x, y_1)$ and $\varphi_2 = R_2(x, y_2)$. In the examples of this article, R_i and R_j are assumed to be different symbols when $i \neq j$. In particular, $\text{schema}(Q_2^\star)$ consists of two distinct binary relation symbols: R_1 and R_2. Hence, $Q_2^\star$ has no self joins (but it would have a self join if we replaced the symbol R_2 with R_1). There is only one existential variable in $Q_2^\star$, namely x, and the two head variables are y_1 and y_2. Hence $\mathsf{Var}_\exists(Q) = \{x\}$ and $\mathsf{Var}_\mathsf{h}(Q) = \{y_1, y_2\}$. Furthermore, $\mathsf{Var}_\exists(\varphi_1) = \{x\}$ and $\mathsf{Var}_\mathsf{h}(\varphi_1) = \{y_1\}$.

We will frequently refer to the following special sjf-CQ:

$$Q_2^\times(y_1, y_2) :\!-\ R_1(y_1), R_2(y_2) \tag{2}$$

More generally, for a natural number k:

$$Q_k^\times(y_1, \dots, y_k) :\!-\ R_1(y_1), \dots, R_k(y_k) \tag{3}$$

That is, $Q_k^\times$ is simply the Cartesian product of k distinct unary relations. Observe that $Q_k^\times$ has no existential variables.

Finally, our running example for this section is the CQ Q_0, defined as follows.

$$\begin{aligned} Q_0(y_1, y_2, y_3, y_4) :\!-\ & R_1(x_1, y_1), R_2(x_1, y_2), \qquad (4) \\ R_3(x_2, y_1, y_2), & R_4(x_3, y_1, y_2, y_3), R_5(x_2, x_3), R_6(y_3, y_4) \end{aligned}$$

The CQs in the examples of this paper do not have constants. Nevertheless, all the definitions and results have no restriction on constants. □

Let $Q(\mathbf{y})$ be a CQ. An *assignment* for Q is a mapping $\mu : \mathsf{Var}(Q) \to \mathsf{Const}$. For an assignment μ for Q, the tuple $\mu(\mathbf{y})$ is the one obtained from $\mathbf{y}$ by replacing every head variable y with the constant $\mu(y)$. Similarly, for an atom φ of Q, the fact $\mu(\varphi)$ is the one obtained from φ by replacing every variable z with the constant $\mu(z)$. Let I be an instance over $\text{schema}(Q)$. A *match for Q in I* is an assignment μ for Q, such that $\mu(\varphi)$ is a fact of I for all $\varphi \in \text{atoms}(Q)$. We denote by $\mathcal{M}(Q, I)$ the set of all matches for Q in I. If $\mu \in \mathcal{M}(Q, I)$, then $\mu(\mathbf{y})$ is called an *answer* (*for Q in I*). The *result* of evaluating Q over I, denoted $Q(I)$, is the set of all the answers for Q in I; that is, $Q(I)$ is the set $\{\mu(\mathbf{y}) \mid \mu \in \mathcal{M}(Q, I)\}$. Let f be a fact in R^I. We say that f is *(Q, I)-useful* if $\mu(\varphi_R) = f$ for some $\mu \in \mathcal{M}(Q, I)$. We say that f is *Q-consistent* with a tuple $\mathbf{a}$ if there is an assignment μ for Q, such that $\mu(\mathbf{y}) = \mathbf{a}$ and $\mu(\varphi_R) = f$. Note that (Q, I)-usefulness implies Q-consistency with some $\mathbf{a}$, but not vice versa.

2.3 Deletion Propagation

The focus of this paper is on (but not restricted to) the problem of minimizing the *view side effect* [5] when propagating the deletion of *multiple* answers (a.k.a. *group deletion* [6]) back to the source relations. Formally, for a CQ Q the problem $\text{MinVSE}\langle Q \rangle$ is defined as follows. The input consists of an instance I over $\text{schema}(Q)$, and a set $A \subseteq Q(I)$ of answers. A *solution* (*for I and A*) is an instance $J \subseteq I$ such that $Q(J) \cap A = \emptyset$. The *side effect* (*of J*), denoted $\mathit{seff}(J)$, is the set $Q(I) \setminus (A \cup Q(J))$, that is, the set of answers that are deleted in addition to A. (Note that the notation $\mathit{seff}(J)$ assumes that Q, I and A are known from the context, which will be the case whenever we will use this notation.) The goal is to find an *optimal* solution, which is a solution J that minimizes the side effect; that is, J is such that $|\mathit{seff}(J)| \leq |\mathit{seff}(K)|$ for all solutions K.

As we discuss later, finding an optimal solution for the problem $\text{MinVSE}\langle Q \rangle$ may be intractable. Often, though, we can settle for *approximations*, which are defined as follows. For a number (or numeric function) $\alpha \geq 1$, a solution J is said to be *α-optimal* if $|\mathit{seff}(J)| \leq \alpha \cdot |\mathit{seff}(K)|$ for all solutions K.

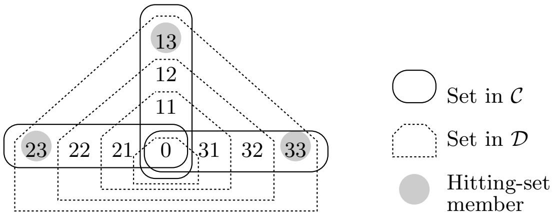

Figure 1: An example of a FcsHit instance

Comment 2.2. *It is important to note that in our previous work [16, 17], approximation is defined differently, as that for a maximization problem, where the goal is to maximize the remaining view. Under that sense, an α-optimal solution J is one that satisfies $|Q(J)| \geq |Q(J')|/\alpha$ for all solutions J'. We study approximation under the maximization notion in Section 6.1; but except for that section, we use the minimization variant, which is the same as that used by Buneman et al. [5].*

We will also study a restriction of MinVSE$\langle Q \rangle$ by imposing the constraint of a fixed bound c on the size of the input set A of tuples to delete. Formally, for a natural number c, the problem MinVSE$_c\langle Q \rangle$ is the same as MinVSE$\langle Q \rangle$, except that $|A| \leq c$ is a restriction on the input. For example, MinVSE$_1\langle Q \rangle$ is the previously studied problem of minimizing the side effect when deleting a single tuple [5,6,9,16,17].

An additional problem we consider is the decision problem FreeVSE$\langle Q \rangle$, where the input is the same as that for MinVSE$\langle Q \rangle$, and the goal is to determine whether there exists a solution J that is *side-effect free*, that is, *seff* $(J) = \emptyset$. The problem FreeVSE$_c\langle Q \rangle$ is defined similarly, except that $|A| \leq c$ is imposed.

3. FOCUSED HITTING SET

To establish our results, we introduce and analyze a combinatorial problem that we call *focused hitting set*, which generalizes the classic *minimal hitting set*.

The minimal hitting set problem, denoted here as MinHit, is the following: given a collection $\mathcal{C}$ of subsets of a ground set $\mathcal{X}$, find a *hitting set* of a minimal cardinality. Recall that a *hitting set* is a subset H of $\mathcal{X}$, such that H *hits* (i.e., has a nonempty intersection with) every set in $\mathcal{C}$. For a number $\alpha \geq 1$, a hitting set H is α-*optimal* if $|H| \leq \alpha|H'|$ for all hitting sets H'.

The *focused* hitting-set problem, denoted FcsHit, is the following: given two collections $\mathcal{C}$ and $\mathcal{D}$ of subsets of a ground set $\mathcal{X}$, find a hitting set (of any size) for $\mathcal{C}$ that hits as few as possible of the sets in $\mathcal{D}$. We use $H \sqcap \mathcal{D}$ to denote the set of all $s \in \mathcal{D}$ satisfying $s \cap H \neq \emptyset$. For a number (or numeric function) $\alpha \geq 1$, a hitting set H is α-*optimal* if $|H \sqcap \mathcal{D}| \leq \alpha|H' \sqcap \mathcal{D}|$ for all hitting sets H' for $\mathcal{C}$.

Example 3.1. Figure 1 illustrates the example involving the following sets:

- $\mathcal{X} = \{0, 11, 12, 13, 21, 22, 23, 31, 32, 33\}$
- $\mathcal{C} = \{\{0, 11, 12, 13\}, \{0, 21, 22, 23\}, \{0, 31, 32, 33\}\}$
- $\mathcal{D} = \{\{0\}, \{0, 11, 21, 31\}, \{0, 11, 21, 31, 12, 22, 32\}, \{0, 11\ 12, 13, 21, 22, 23, 31, 32, 33\}\}$

A *smallest* hitting set for $\mathcal{C}$ is $\{0\}$, but this set hits all 4 sets in $\mathcal{D}$. A most *focused* hitting set is $\{13, 23, 33\}$ (marked by the grey circles in Figure 1), which hits only one set in $\mathcal{D}$. Finally, $\{12, 22, 32\}$ is a 2-optimal hitting set, as it hits (no more than) twice the minimum number of sets in $\mathcal{D}$. □

Note that FcsHit generalizes MinHit through the following approximation-preserving reduction: from the input $(\mathcal{X}, \mathcal{C})$ for MinHit generate the input $(\mathcal{X}, \mathcal{C}, \mathcal{D})$ for FcsHit with $\mathcal{D}$ being the set of all singletons $\{x\}$ where $x \in \mathcal{X}$.

Let k be a natural number. The problems MinHit$\langle k \rangle$ and FcsHit$\langle k \rangle$ are similar to MinHit and FcsHit, respectively, except that now we assume that the size of each set in $\mathcal{C}$ or $\mathcal{D}$ is at most k.

A special case of MinHit$\langle k \rangle$ is *vertex cover in a k-uniform k-partite hypergraph*: this is a vertex cover problem where the vertex set $\mathcal{X}$ is partitioned into k pairwise-disjoint sets $s_1, \dots, s_k$ such that each set (hyperedge) in $\mathcal{C}$ has exactly one element from each s_i. The *focused* version of this problem is similarly a special case of FcsHit$\langle k \rangle$ where, again, each set in $\mathcal{C} \cup \mathcal{D}$ has exactly one element from each s_i. We denote these problems by MinUVC$\langle k \rangle$ and FcsUVC$\langle k \rangle$, respectively.

3.1 Approximation Algorithm

As well known, MinHit$\langle k \rangle$ can be k-approximated in polynomial time. Such an approximation is achieved by greedily selecting pairwise-disjoint sets from $\mathcal{C}$ until the union of these sets is a hitting set, which is taken as the solution. This simple greedy approach fails when the sets of $\mathcal{C}$ are weighted. Moreover, there does not seem to be any corresponding greedy approach to k-approximate FcsHit$\langle k \rangle$; one explanation of that is in the fact that focused hitting set can encode weighted hitting set. So, in the proof of the following lemma we adopt an approach that works for weighted vertex cover (which is a special case of weighted hitting set), namely linear programming with rounding [25]. As a result, we get that FcsHit$\langle k \rangle$ can also be k-approximated in polynomial time.

Lemma 3.2. *For* FcsHit$\langle k \rangle$, *a k-optimal hitting set can be found in polynomial time.*

Proof. Consider the input $(\mathcal{C}, \mathcal{D})$ for FcsHit$\langle k \rangle$. We define the following linear program, with a variable y_s for each $s \in \mathcal{D}$ and x_e for each $e \in \cup(\mathcal{C} \cup \mathcal{D})$.

$$\text{Minimize} \sum_{s \in \mathcal{D}} y_s \quad \text{s.t.}$$

$$\forall s \in \mathcal{C} \quad \sum_{e \in s} x_e \geq 1 \tag{5}$$

$$\forall s \in \mathcal{D}, e \in s \quad y_s \geq x_e \tag{6}$$

$$\forall e \in \cup(\mathcal{C} \cup \mathcal{D}) \quad x_e \geq 0 \tag{7}$$

Given a solution $\mathbf{x}, \mathbf{y}$, we use the following rounding to construct a hitting set H.

$$H \overset{\mathsf{def}}{=} \{e \in \cup\mathcal{C} \mid x_e \geq 1/k\}$$

Note that conditions (5) and (7), along with the fact that each $s \in \mathcal{C}$ satisfies $|s| = k$, implies that H is indeed a hitting set (that is, for each $s \in \mathcal{C}$ there is $e \in \cup\mathcal{C}$ with $x_e \geq 1/k$).

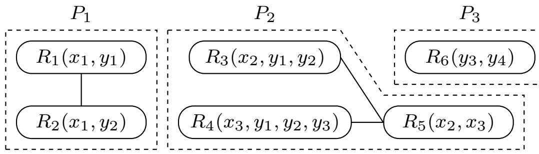

Figure 2: The graph $\mathcal{G}_\exists(Q_0)$ for the CQ Q_0 of (4)

Let H_{opt} be an optimal hitting set (i.e., a hitting set for $\mathcal{C}$ that hits as few as possible members of $\mathcal{D}$).

If H' is a hitting set for $\mathcal{C}$, then let $\mathbf{x}', \mathbf{y}'$ be the solution for the linear program assigning x'_e one if $e \in H'$ or zero otherwise, and $y'_s = 1$ if s is hit by H' or zero otherwise. It is easy to see that $\mathbf{x}', \mathbf{y}'$ is legal solution, and that $|H' \sqcap \mathcal{D}| = \sum_{s \in \mathcal{D}} y'_s$. It follows that $\sum_{s \in \mathcal{D}} y_s \leq |H_{\text{opt}} \sqcap \mathcal{D}|$. Let $\hat{\mathbf{y}}$ be the solution obtained by replacing every y_s with 1 in case s is hit by H. Observe that, due to (6) we have that $\hat{y}_s \leq k y_s$ for all $s \in \mathcal{D}$. Hence, we get the following:

$$|H \sqcap \mathcal{D}| \leq \sum_{s \in \mathcal{D}} \hat{y}_s \leq \sum_{s \in \mathcal{D}} k \cdot y_s = k \sum_{s \in \mathcal{D}} y_s \leq k |H_{\text{opt}} \sqcap \mathcal{D}|$$

Hence, our solution H is k-optimal, as claimed. □

4. LEVELED HEAD DOMINATION

In this section, we introduce the notion of *level-k head domination* (where k is a natural number). But first, we recall the definition of (ordinary) *head domination* [17].

Let Q be a CQ. The *existential-connectivity graph* of Q, denoted $\mathcal{G}_\exists(Q)$, is the undirected graph that has atoms(Q) as the set of nodes, and that has an edge $\{\varphi_1, \varphi_2\}$ whenever φ_1 and φ_2 have at least one existential variable in common (that is, $\mathsf{Var}_\exists(\varphi_1) \cap \mathsf{Var}_\exists(\varphi_2) \neq \emptyset$). If P is a connected component of $\mathcal{G}_\exists(Q)$, then we denote by $\mathsf{Var}_\mathsf{h}(P)$ the set of all the head variables that occur in the atoms of P.

Example 4.1. Figure 2 shows the graph $\mathcal{G}_\exists(Q_0)$ for the CQ Q_0 in (4). Note that $R_3(x_2, y_1, y_2)$ and $R_4(x_3, y_1, y_2, y_3)$ do not share an edge, since they do not share existential variables (they share only head variables). The graph has three connected components (surrounded by dashed-edge polygons): P_1, P_2 and P_3. Observe that $\mathsf{Var}_\mathsf{h}(P_1) = \{y_1, y_2\}$, $\mathsf{Var}_\mathsf{h}(P_2) = \{y_1, y_2, y_3\}$ and $\mathsf{Var}_\mathsf{h}(P_3) = \{y_3, y_4\}$. □

Following is the definition of the *head-domination* property of a CQ [17].

Definition 4.2. (**Head Domination**) A CQ Q has *head domination* if there is a subset Φ of atoms(Q), such that for every connected component P of $\mathcal{G}_\exists(Q)$ there is an atom $\varphi \in \Phi$ with $\mathsf{Var}_\mathsf{h}(P) \subseteq \mathsf{Var}(\varphi)$; in that case, we say that Q is *head dominated* by Φ. □

Note that in the definition, the atom φ, which satisfies $\mathsf{Var}_\mathsf{h}(P) \subseteq \mathsf{Var}(\varphi)$, is not required to be in P.

Example 4.3. Consider again the graph $\mathcal{G}_\exists(Q_0)$ in Figure 2. Recall that the connected components of $\mathcal{G}_\exists(Q_0)$ are P_1, P_2 and P_3. It holds that $\mathsf{Var}_\mathsf{h}(P_1) \subseteq \mathsf{Var}(\varphi_{R_3})$, $\mathsf{Var}_\mathsf{h}(P_2) \subseteq \mathsf{Var}(\varphi_{R_4})$ and $\mathsf{Var}_\mathsf{h}(P_3) \subseteq \mathsf{Var}(\varphi_{R_6})$. (Recall that φ_R denotes the atom with the relation symbol R.) Therefore, Q has head domination, and is head dominated by $\Phi' = \{\varphi_{R_3}, \varphi_{R_4}, \varphi_{R_6}\}$. Observe that Q is also head dominated by $\Phi = \{\varphi_{R_4}, \varphi_{R_6}\}$.

An example of a CQ without head domination is $Q_2^\star$ that we defined in (1). Indeed, $\mathcal{G}_\exists(Q_2^\star)$ has exactly one connected component, its head variables are y_1 and y_2, and no atom of Q contains both y_1 and y_2. □

The next definition parameterizes head domination by the minimal number atoms needed for domination.

Definition 4.4. (**Level-k Head Domination**) Let k be a natural number. A CQ Q with head domination has *level-k head domination* if k is the minimal cardinality of a set that head dominates Q. □

Example 4.5. Continuing Example 4.3, the CQ Q_0 is head dominated by Φ, which has two atoms. Since Q_0 is not head dominated by any single atom, we get that Q_0 has level-2 head domination. □

Kimelfeld et al. [17] proved the following dichotomy in the complexity of the problems $\text{MinVSE}_1\langle Q\rangle$ and $\text{FreeVSE}_1\langle Q\rangle$ (where the goal is to delete a single tuple). In the theorem we use "c" as the placeholder of "1" since we later refer to this theorem with an arbitrary c.

Theorem 4.6. [17] *Let Q be an sjf-CQ. The following hold for $c = 1$.*

1. *If Q has head domination, then $\text{MinVSE}_c\langle Q\rangle$ (hence, $\text{FreeVSE}_c\langle Q\rangle$) can be solved in polynomial time.*
2. *If Q has no head domination, then $\text{FreeVSE}_c\langle Q\rangle$ is NP-complete; therefore, it is NP-hard to approximate $\text{MinVSE}_c\langle Q\rangle$ by any finite factor.*

Comment 4.7. *Theorem 4.6 is weaker than the main theorem of Kimelfeld et al. [17]. Their theorem adds the following. In Case 1, the problem is solvable by an extremely simple algorithm (called "trivial" there) that was originally proposed by Buneman et al. [5]. In Case 2, the ability to approximate is limited (or more formally the problem is APX-hard) even when using the maximization notion of approximation that we mentioned in Comment 2.2 (and use in Section 6.1).*

In the next section we extend Theorem 4.6 to the deletion of multiple tuples, by studying the complexity of $\text{MinVSE}\langle Q\rangle$ and $\text{FreeVSE}\langle Q\rangle$. We will also investigate the complexity of $\text{MinVSE}_c\langle Q\rangle$ and $\text{FreeVSE}_c\langle Q\rangle$ for an arbitrary c.

5. COMPLEXITY RESULTS

In this section we give our main complexity results for the problems $\text{MinVSE}\langle Q\rangle$ and $\text{FreeVSE}\langle Q\rangle$. We begin with upper bounds.

5.1 Upper Bounds

To obtain positive complexity results (upper bounds), we will show how $\text{MinVSE}\langle Q\rangle$ reduces to focused hitting set when Q has head domination. Later, we will give complexity results based on this reduction.

Let Q be an sjf-CQ with level-k head domination. We now show how $\text{MinVSE}\langle Q\rangle$ reduces to $\text{FcsUVC}\langle k\rangle$. We begin with a specific sjf-CQ Q, namely $Q_k^\times$ (specified in (3)), where this reduction is straightforward. Given an instance I over schema($Q_k^\times$) and a set A of answers to delete, the reduction is as follows. The ground set $\mathcal{X}$ is (the set of all facts of) I.

Algorithm $\mathsf{ToFcsHit}\langle Q\rangle(I, A)$

```
1: Let $\Phi = \{\varphi_1, \dots, \varphi_k\}$ head dominate $Q$
2: for $i = 1, \dots, k$ do
3:     $s_i \leftarrow \{\mu_{\mathsf{h}}[\varphi_i] \mid \mu \in \mathcal{M}(Q, I)\}$
4: $\mathcal{C} \leftarrow \{\mu_{\mathsf{h}}[\Phi] \mid \mu \in \mathcal{M}(Q, I) \wedge \mu(\mathbf{y}) \in A\}$
5: $\mathcal{D} \leftarrow \{\mu_{\mathsf{h}}[\Phi] \mid \mu \in \mathcal{M}(Q, I) \wedge \mu(\mathbf{y}) \notin A\}$
6: $H \leftarrow \mathsf{FcsUVC}(s_1, \dots, s_k, \mathcal{C}, \mathcal{D})$
7: $J \leftarrow I$
8: for all $\mu_{\mathsf{h}}[\varphi] \in H$ do
9:     Delete from $R^J_\varphi$ every fact consistent with $\mu_{\mathsf{h}}[\varphi]$
10: return $J$
```

Figure 3: Reducing MinVSE$\langle Q\rangle$**, for an sjf-CQ** Q **with level-**k **head domination, to** FcsUVC$\langle k\rangle$

The pairwise-disjoint sets $s_1, \dots, s_k$ are simply the relations (i.e., fact sets) $R^I_1, \dots, R^I_k$, respectively. We then define:

$$\mathcal{C} \overset{\mathsf{def}}{=} \{\{R_1(a_1), \dots, R_k(a_k)\} \mid (a_1, \dots, a_k) \in A\}$$

$$\mathcal{D} \overset{\mathsf{def}}{=} \{\{R_1(b_1), \dots, R_k(b_k)\} \mid (b_1, \dots, b_k) \in Q_k(I) \setminus A\}$$

It is then easy to see that from a hitting set H for $\mathcal{C}$ we obtain a solution J by deleting from I every fact in H; conversely, from a solution J we obtain a hitting set for $\mathcal{C}$ by taking $H = I \setminus J$. Most importantly, H and J have the same "quality": $|H \sqcap \mathcal{D}| = |\mathit{seff}(J)|$. Hence, we get an approximation-preserving reduction.

The sjf-CQ $Q = Q^\times_k$ gives an extremely simple case of reducing MinVSE$\langle Q\rangle$ to FcsUVC$\langle k\rangle$. In the remainder of this section, we show that a reduction with similar guarantees exists for every sjf-CQ with level-k head domination.

We begin with some notation. Let $Q(\mathbf{y})$ be a CQ, let I be an instance of schema(Q), and let μ be a match for Q in I. If $\varphi \in$ atoms(Q), then $\mu_{\mathsf{h}}[\varphi]$ denotes the restriction of μ to the head variables of φ. If Φ is a subset of atoms(Q), then $\mu_{\mathsf{h}}[\Phi]$ denotes the set $\{\mu_{\mathsf{h}}[\varphi] \mid \varphi \in \Phi\}$. A fact f over the relation symbol R_φ is *consistent with* $\mu_{\mathsf{h}}[\varphi]$ if f can be obtained from φ by *(1)* replacing each existential variable with some constant, *and (2)* replacing each head variable y with $\mu_{\mathsf{h}}[\varphi](y)$.

Let Q be an sjf-CQ with level-k head domination, and suppose that Q is head dominated by $\Phi = \{\varphi_1, \dots, \varphi_k\}$. The reduction is depicted in Figure 3 as an algorithm named $\mathsf{ToFcsHit}\langle Q\rangle$. The ground set $\mathcal{X}$ consists of all mappings $\mu_{\mathsf{h}}[\varphi_i]$ where $\mu \in \mathcal{M}(Q, I)$ and $1 \le i \le k$. The pairwise-disjoint sets $s_1, \dots, s_k$ are defined by taking as s_i the set of all $\mu_{\mathsf{h}}[\varphi_i]$. Note that the s_i are indeed pairwise disjoint, since different φ_i have different sets of head variables (otherwise, Q does not have level-k head domination). The collection $\mathcal{C}$ consists of all the sets $\mu_{\mathsf{h}}[\Phi]$ where $\mu \in \mathcal{M}(Q, I)$ and $\mu(\mathbf{y}) \in A$, and the collection $\mathcal{D}$ consists of all the sets $\mu_{\mathsf{h}}[\Phi]$ where $\mu \in \mathcal{M}(Q, I)$ and $\mu(\mathbf{y}) \notin A$. A solution (hitting set) H for the instance $(s_1, \dots, s_k, \mathcal{C}, \mathcal{D})$ of FcsUVC$\langle k\rangle$ is constructed in line 6. From H the reduction constructs a solution J for MinVSE$\langle Q\rangle$, as follows. Starting with $J = I$, for each mapping $\mu_{\mathsf{h}}[\varphi_i] \in H$ the algorithm deletes from J *every* fact f over φ_i such that f is consistent with $\mu_{\mathsf{h}}[\varphi]$.

Next, we prove the correctness $\mathsf{ToFcsHit}\langle Q\rangle(I, A)$. For that, we need the following lemma, which is proved similarly to the optimality of the "unirelation" algorithm for MinVSE$_1\langle Q\rangle$ in the case of head domination [17].

Lemma 5.1. *Let Q be an m-ary sjf-CQ that is head dominated by a set $\Phi \subseteq$ atoms(Q). Let J be an instance over* schema(Q) *and $\mathbf{a} \in \mathsf{Const}^m$. The following are equivalent.*

- $\mathbf{a} \in Q(J)$
- *For all $\varphi \in \Phi$ there is a fact f_φ over R_φ, such that f_φ is both (Q, J)-useful and Q-consistent with $\mathbf{a}$.*

Next, we prove the following theorem.

Theorem 5.2. *Let Q be an sjf-CQ with level-k head domination. Consider an execution of $\mathsf{ToFcsHit}\langle Q\rangle(I, A)$. If $\mathsf{FcsUVC}(s_1, \dots, s_k, \mathcal{C}, \mathcal{D})$ returns an α-optimal hitting set, then the returned instance J is an α-optimal solution for I and A.*

Proof. Observe that every tuple $\mathbf{a} \in Q(I)$ corresponds to a unique set in $\mathcal{C} \cup \mathcal{D}$; we denote this set by $s_{\mathbf{a}}$. Suppose, w.l.o.g., that every fact in I is (Q, I)-useful (since we can delete every fact that is not (Q, I)-useful without affecting the problem). With that assumption, the following are equivalent for a fact f_φ over a relation symbol R_φ (where $\varphi \in$ atoms(Q)) and $\mathbf{a} \in Q(I)$.

- f_φ is Q-consistent with $\mathbf{a}$.
- f_φ is consistent with $\mu_{\mathsf{h}}[\varphi]$ for some $\mu \in \mathcal{M}(Q, I)$ that satisfies $\mu(\mathbf{y}) = \mathbf{a}$.

Then, Lemma 5.1 implies that for a tuple $\mathbf{a} \in Q(I)$, we have $\mathbf{a} \in Q(J)$ if and only if for all $i = 1, \dots, k$ there is a match $\mu \in \mathcal{M}(Q, J)$ such that $\mu_{\mathsf{h}}[\varphi_i]$ agrees with $\mathbf{a}$. In other words, $\mathbf{a} \in Q(J)$ if and only if we can find in $\mathcal{M}(Q, J)$ matches to cover every member of $s_{\mathbf{a}}$. This immediately implies that J is indeed a solution for I and A, since by hitting $\mathcal{C}$ the algorithm eliminates at least one such match for each $\mathbf{a} \in A$. By the same arguments we get that the solution J is such that $|H \sqcap \mathcal{D}| = |\mathit{seff}(J)|$.

We complete the proof by showing that for every solution J for I and A there is a hitting set H such $|H \sqcap \mathcal{D}| = |\mathit{seff}(J)|$. By using Lemma 5.1, we obtain the hitting set H by taking every mapping $\mu_{\mathsf{h}}[\varphi_i]$ (where $\mu \in \mathcal{M}(Q, I)$) that is consistent with none of the tuples in $Q(J)$. Then H hits exactly those sets s that correspond to tuples in $Q(I) \setminus Q(J)$, which implies both that H is a hitting set and that $|H \sqcap \mathcal{D}| = |\mathit{seff}(J)|$, as claimed. □

Next, we draw immediate corollaries from Lemma 3.2, in combination with Theorem 5.2. The first one states that MinVSE$\langle Q\rangle$ is k-approximable whenever Q has level-k head domination.

Corollary 5.3. *If Q is an sjf-CQ with level-k head domination, then* MinVSE$\langle Q\rangle$ *is k-approximable in polynomial time.*

As described above, Corollary 5.3 is obtained by reducing MinVSE$\langle Q\rangle$ to FcsUVC$\langle k\rangle$. Recall that our solution for FcsHit$\langle k\rangle$ (which generalizes FcsUVC$\langle k\rangle$) is through the linear program we described in the proof of Lemma 3.2. In that program, each variable y_s corresponds to a unique tuple in $Q(I) \setminus A$, and vice versa. An important observation is that we could associate *preferences* to the tuples in $Q(I) \setminus A$ (stating that survival of some tuples is more important than those of others) by associating with each y_s a weight w_s, thereby setting the goal of minimizing $\sum_{s \in \mathcal{D}} w_s y_s$. Hence, we get a natural extension of the space of supported quality measures for solutions.

Corollary 5.3 gives, as a special case, a class of sjf-CQs for which MinVSE$\langle Q \rangle$ is solvable in polynomial time—those with level-1 head domination (i.e., at least one atom contains all the head variables). Note that this upper bound could also be obtained through a straightforward argument that does not require Corollary 5.3.

Corollary 5.4. *If Q is an sjf-CQ with level-1 head domination, then* MinVSE$\langle Q \rangle$ *is in polynomial time.*

We later give a complementary lower bound, showing that the class of sjf-CQs with level-1 head domination is *precisely* that of sjf-CQs for which MinVSE$\langle Q \rangle$ is solvable in polynomial time (assuming P $\neq$ NP).

An algorithm for MinVSE$\langle Q \rangle$, with any upper-bound guarantee on the approximation ratio, can be used for solving FreeVSE$\langle Q \rangle$: if there is a side-effect-free solution, the algorithm will necessarily find one. Hence, we get the following corollary, generalizing the corresponding part of Theorem 4.6 from MinVSE$_1\langle Q \rangle$ to MinVSE$\langle Q \rangle$.

Corollary 5.5. *If Q is an sjf-CQ with head domination, then* FreeVSE$\langle Q \rangle$ *is in polynomial time.*

5.2 Lower Bounds

Theorem 4.6 implies that if Q has no head domination, then it is NP-hard to solve MinVSE$\langle Q \rangle$ optimally. The following lemma states that this is also the case whenever Q has level-k head domination with $k > 1$.

Lemma 5.6. *If Q is an sjf-CQ with level-k head domination for $k > 1$, then* MinVSE$\langle Q \rangle$ *is NP-hard.*

Lemma 5.6 is a actually special case of a more general result that accounts for functional dependencies, which we discuss in Section 6.2.

Recall from Corollary 5.3 that for an sjf-CQ Q with level-k head domination, the problem MinVSE$\langle Q \rangle$ can be k-approximated in polynomial time. Whether this bound is tight for every Q is an open problem. However, we can show an infinite series of such Q where the lower bound is tight up to a constant factor (that does not depend on k). More specifically, based on a recent result by Guruswami and Saket [14], the following theorem states that for $k > 4$, the CQ $Q_k^{\times}$ is such that it is NP-hard to approximate MinVSE$\langle Q_k^{\times} \rangle$ within some ratio linear in k.

Theorem 5.7. *Let $k > 4$ be a natural number and let $\epsilon > 0$ be a number. It is NP-hard to $(k/4 - \epsilon)$-approximate* MinVSE$\langle Q_k^{\times} \rangle$.

Proof. The proof is by a reduction from MinUVC$\langle k \rangle$ (defined in Section 3). Guruswami and Saket [14] showed that for all $\epsilon > 0$, it is NP-hard to obtain a $(k/4 - \epsilon)$-optimal hitting set. So consider an instance $(s_1, \dots, s_k, \mathcal{C})$ of this problem. We define an instance I over schema$(Q_k^{\times})$ as follows. Each relation R_i contains all the values of s_i (as unary tuples) and, in addition, a set of N fresh (distinct) constants. (Note that each fresh constant occurs exactly once.) The value N will be determined later. The set A of answers to delete is obtained from $\mathcal{C}$ by straightforwardly translating each $s \in \mathcal{C}$ into a tuple of the Cartesian product $\times_{i=1}^{k} s_i$ that comprises the elements in s.

Let H_{opt} be a minimal hitting set for $(s_1, \dots, s_k, \mathcal{C})$. Let J_o be the solution that is obtained from H_{opt} by deleting every element of H_{opt} (from the proper relation). Let J be any solution for I and A (obtained through some algorithm). From J we obtain a hitting set H_J by selecting all the values of the s_i that are missing (i.e., have been deleted) from J. We will show that if J is a good approximation w.r.t. J_o, then H_J is a good approximation w.r.t. H_{opt}. More precisely, we will prove the following for the right choice of N and some $\delta > 0$ that depends only on ϵ.

$$\frac{|\mathit{seff}(J)|}{|\mathit{seff}(J_o)|} < \frac{k}{4} - \delta \quad \Rightarrow \quad \frac{|H_J|}{|H_{\text{opt}}|} < \frac{k}{4} - \epsilon \qquad (8)$$

We first estimate the size of $\mathit{seff}(J)$. Define $M = \prod_{i=1}^{k} |s_i|$. Then $\mathit{seff}(J)$ is the disjoint union of two sets:

- Tuples t that contain precisely one element in $\cup_{i=1}^{k} s_i$. There are precisely $|H_J| \cdot N^{k-1}$ such tuples.
- Tuples t that contain at least two elements in $\cup_{i=1}^{k} s_i$. There are at most MN^{k-2} such tuples (since M bounds the number of choices of the elements from $\cup_{i=1}^{k} s_i$).

We conclude the following:

$$|H_J| \cdot N^{k-1} \leq |\mathit{seff}(J)| \leq |H_J| \cdot N^{k-1} + MN^{k-2}$$

And similarly (since J is arbitrary):

$$|H_{\text{opt}}| \cdot N^{k-1} \leq |\mathit{seff}(J_o)| \leq |H_{\text{opt}}| \cdot N^{k-1} + MN^{k-2}$$

So we get the following argument.

$$\begin{aligned} \frac{k}{4} - \delta > & \frac{|\mathit{seff}(J)|}{|\mathit{seff}(J_o)|} \geq \frac{|H_J|}{|H_{\text{opt}}| + M/N} = \\ & \frac{|H_J|}{|H_{\text{opt}}|} \cdot \frac{1}{1 + M/(N|H_{\text{opt}}|)} \geq \frac{|H_J|}{|H_{\text{opt}}|} \cdot \frac{1}{1 + M/N} \\ \Rightarrow & \frac{|H_J|}{|H_{\text{opt}}|} < \frac{k}{4} - \delta + \frac{k - 4\delta}{4N/M} < \frac{k}{4} - \delta + \frac{kM}{4N} \end{aligned}$$

So we obtain (8) by choosing $\delta = 2\epsilon$ and $N = 4kM/\epsilon$. □

5.3 Trichotomy in Complexity

The main result of this paper is the following theorem, stating a trichotomy in complexity for MinVSE$\langle Q \rangle$ over the sjf-CQs Q. The theorem is obtained by combining Theorem 4.6, Corollaries 5.3, 5.4 and 5.5, and Lemma 5.6.

Theorem 5.8 (Trichotomy). *Let Q be an sjf-CQ.*

- *If Q has level-1 head domination, then* MinVSE$\langle Q \rangle$ *is solvable in polynomial time.*
- *If Q has level-k head domination with $k > 1$, then* MinVSE$\langle Q \rangle$ *is NP-hard but k-approximable in polynomial time; moreover in that case* FreeVSE$\langle Q \rangle$ *is in polynomial time.*
- *If Q has no head domination, then* FreeVSE$_1\langle Q \rangle$ *is NP-complete (and approximating* MinVSE$_1\langle Q \rangle$ *by any finite factor is NP-hard).*

Table 1: The complexity of $\textsc{MinVSE}\langle Q\rangle$ and $\textsc{FreeVSE}\langle Q\rangle$, and their bounded-deletion variants

Problem	Level-1 h. d.	Level-k h. d. for $k > 1$	No h. d.
$\textsc{MinVSE}\langle Q\rangle$ (optimal)	PTime	NP-hard	NP-hard
$\textsc{MinVSE}\langle Q\rangle$ (approx.)	PTime	PTime	NP-hard
$\textsc{FreeVSE}\langle Q\rangle$	PTime	PTime	NP-complete
$\textsc{MinVSE}_c\langle Q\rangle$ (optimal)	PTime	PTime	NP-hard
$\textsc{MinVSE}_c\langle Q\rangle$ (approx.)	PTime	PTime	NP-hard
$\textsc{FreeVSE}_c\langle Q\rangle$	PTime	PTime	NP-complete

Example 5.9. Consider the following three sjf-CQs.

$$Q_1(y_1, y_2, y_3) :- R(y_1, y_2), S(y_2, y_3), T(y_1, y_2, y_3)$$
$$Q_2(y_1, y_2, y_3) :- R(y_1, y_2), S(x, y_3), T(y_1, x, y_3)$$
$$Q_3(y_1, y_2, y_3) :- R(y_1, y_2), S(x, y_3), T(y_1, y_2, x)$$

Observe that Q_1 has level-1 head domination, Q_2 has level-2 head domination, and Q_3 has no head domination. From Theorem 5.8 we then conclude the following. First, the problem $\textsc{MinVSE}\langle Q_1\rangle$ is solvable in polynomial time. Second, $\textsc{MinVSE}\langle Q_2\rangle$ is NP-hard but 2-approximable in polynomial time; moreover, $\textsc{FreeVSE}\langle Q_2\rangle$ is solvable in polynomial time. Third, the problem $\textsc{FreeVSE}\langle Q_3\rangle$ is NP-complete, and hence, $\textsc{MinVSE}\langle Q_3\rangle$ cannot be approximated in polynomial time for any finite factor, unless P = NP; moreover, these results hold already for the single-tuple variants: $\textsc{FreeVSE}_1\langle Q_3\rangle$ and $\textsc{MinVSE}_1\langle Q_3\rangle$. □

5.4 Bounded Deletion

We now consider the complexity of $\textsc{MinVSE}_c\langle Q\rangle$, and give an immediate corollary of Theorem 5.2: in the case of head domination, $\textsc{MinVSE}_c\langle Q\rangle$ is solvable in polynomial time, and is even FPT for the parameter c. Recall that *FPT* stands for *fixed-parameter tractability* [12], which means that when a parameter c is involved, the running time is bounded by a function of the form $f(c) \cdot p(n)$, where f is a function and p is a polynomial over the size n of the input. Hence, this corollary extends the positive result of Theorem 4.6 for $\textsc{MinVSE}_c\langle Q\rangle$ from $c = 1$ to an arbitrary c.

Corollary 5.10. *If Q is an sjf-CQ with head domination and c is a natural number, then $\textsc{MinVSE}_c\langle Q\rangle$ is solvable (optimally) in polynomial time, and is even FPT for the parameter c.*

Proof. When calling $\mathsf{ToFcsHit}\langle Q\rangle(I, A)$ with $|A| = c$, the generated instance of $\textsc{FcsUVC}\langle k\rangle$ has a set $\mathcal{C}$ with $|\mathcal{C}| \leq c$, where each member $s \in \mathcal{C}$ is of a constant size (i.e., k if Q has level-k head domination). But this instance of $\textsc{FcsUVC}\langle k\rangle$ has a straightforward polynomial-time solution: consider every possible set H' obtained by selecting one member of each set in $\mathcal{C}$, compute each $|H' \sqcap \mathcal{D}|$, and then select as H the H' with a minimal $|H' \sqcap \mathcal{D}|$. Observe that, since k is fixed, the number of considered sets H' is bounded by a function of c (regardless of I and A), and hence, we get that the problem is FPT. □

Combining Theorem 4.6 and Corollary 5.10, we get the following.

Theorem 5.11. *Theorem 4.6 holds true for every $c \in \mathbb{N}$ (not just $c = 1$).*

Table 1 summarizes the complexity results of Theorems 5.8 and 5.11. Each entry corresponds to a problem (row) and a class of sjf-CQs (column). In the second and fifth rows, “PTime” refers to k-approximation, while “NP-hard” refers to every finite (constant or function) approximation.

6. EXTENSIONS

In this section we discuss two extensions of our complexity results. In the first extension we study the maximization variants of our problems, and in the second we consider the effect of functional dependencies.

6.1 Maximizing the View

In this section, we consider a deletion-propagation problem dual to $\textsc{MinVSE}\langle Q\rangle$, namely, *maximizing* the remaining view $Q(J)$ rather than *minimizing* the side effect. Of course, in terms of seeking optimal solutions these problems are identical. However, in the case of a single-tuple deletion, the two problems have different complexities in terms of approximation [17].

Formally, we consider the problem $\textsc{MaxView}\langle Q\rangle$ that has the same input as $\textsc{MinVSE}\langle Q\rangle$, namely I and A, except that the goal is to find a solution J (i.e., $J \subseteq I$ satisfying $Q(J) \cap A = \emptyset$) with a maximal $|Q(J)|$. In particular, given a number (or numeric function) $\alpha \geq 1$, a solution J is α-*optimal* if $|Q(J)| \geq |Q(K)|/\alpha$ for all solutions K. For a natural number c, the problem $\textsc{MaxView}_c\langle Q\rangle$ is the same as $\textsc{MaxView}\langle Q\rangle$, except that the input is restricted to $|A| \leq c$.

Let Q be a CQ. For a natural number h, we say that Q has h-*coverage* if Q has h atoms $\varphi_1, \ldots, \varphi_h$ that cover all the head variables, that is, $\mathsf{Var_h}(Q) = \cup_{i=1}^{h} \mathsf{Var_h}(\varphi_i)$. Kimelfled et al. [17] proved the following.

Proposition 6.1. [17] *If Q is an sjf-CQ with h-coverage, then one can find in polynomial time a solution J such that $|Q(J)| \geq (|Q(I)| - 1)/h$. In particular, $\textsc{MaxView}_1\langle Q\rangle$ is h-approximable in polynomial time.*

The question we explore here is whether Proposition 6.1 extends to $\textsc{MaxView}\langle Q\rangle$ and to $\textsc{MaxView}_c\langle Q\rangle$. In the reminder of this section we address the complexity aspect this question and show the following.

- Under a fairly general assumption on the sjf-CQ Q (in addition to a reasonable complexity assumption), it is intractable to approximate $\textsc{MaxView}\langle Q\rangle$ within any constant factor (Section 6.1.1).
- If Q is an sjf-CQ with h-coverage, then $\textsc{MaxView}_c\langle Q\rangle$ is (almost) h-approximable in polynomial time (Section 6.1.2).

But first, a complexity-independent question is whether there always exists a solution that retains at least $1/h$ of the non-deleted answers, or even any constant factor thereof. The following example shows a negative answer.

Example 6.2. Consider the CQ $Q_2^{\times}$ defined in (2), and let $I^{\times}$ be an instance over schema($Q_2^{\times}$), such that each of the relations R_1 and R_2 contains the tuples consisting of the constants $1, \dots, n$. Then $Q_2^\times(I^\times)$ is the set $\{1, \dots, n\} \times \{1, \dots, n\}$. Now suppose that A is the set of all pairs $(i, j) \in Q_2^\times(I^\times)$ such that $i \neq j$. Then there are $n^2 - n$ answers to delete and n remaining answers. Now, let J be a solution. Then J cannot contain any $R_1(i)$ and $R_2(j)$ with $i \neq j$, and hence, $Q_2^\times(J)$ contains at most one answer. In particular, at most $1/n$ of the non-deleted answers can survive in any solution. □

6.1.1 Hardness of Approximation

The next theorem states a general condition (though not a dichotomy) on an sjf-CQ Q, such that MaxView$\langle Q\rangle$ (without any restriction on $|A|$) does not have a constant-factor approximation in polynomial time, under reasonable (yet somewhat non-standard) complexity assumptions. The condition is that Q has two head variables that are not atomic neighbors, where two variables y_1 and y_2 are *atomic neighbors* if some atom of Q contains both y_1 and y_2. The complexity assumptions are due to Ambühl et al. [1] whose result we use to prove our theorem.

Theorem 6.3. *Let Q be an sjf-CQ with a pair of head variables that are not atomic neighbors. Moreover, assume that for some $\epsilon > 0$, SAT does not have a probabilistic algorithm that decides in $O(2^{n^\epsilon})$ time whether a given instance of size n is satisfiable. Then no (randomized) polynomial-time algorithm approximates* MaxView$\langle Q\rangle$ *within any constant factor.*

Next, we prove Theorem 6.3. The simplest example of an sjf-CQ with a pair of head variables that are not atomic neighbors is $Q_2^\times$, and we begin by proving the theorem for that CQ.

Lemma 6.4. *The following hold.*

1. MaxView$\langle Q_2^\times\rangle$ *(hence,* MinVSE$\langle Q_2^\times\rangle$*) is NP-hard.*
2. *Theorem 6.3 is true for $Q = Q_2^\times$.*

Proof. We will show that MaxView$\langle Q_2^\times\rangle$ is actually the *maximum edge biclique* problem, which in turn is the following problem: given a bipartite graph G, find a complete bipartite subgraph (biclique) of G with a maximal number of edges. Note that in a biclique of the number of edges is simply the product of the sizes of the two sides. We encode this problem as MaxView$\langle Q_2^\times\rangle$, as follows. Suppose that the input consists of a bipartite graph $G = (V_1, V_2, E)$. In the instance I, the relations R_1 and R_2 contain the nodes of V_1 and V_2 (as unary tuples), respectively. The set A contains every *non-edge* $(v_1, v_2) \in V_1 \times V_2$. Observe that a solution J necessarily induces a biclique B_J; moreover, $Q_2^\times(J)$ is precisely the set of edges in B_J. The other direction is also true: every biclique B represents a solution J_B such that $Q_2^\times(J_B)$ consists of the edges of B.

Peeters [23] proved that maximum edge biclique is NP-hard; hence, we get Part 1. Ambühl et al. [1] showed inapproximability by any constant factor for maximum edge biclique, under the complexity assumption of Theorem 6.3; consequently, we get Part 2. □

We can now prove Theorem 6.3.

Proof. (Theorem 6.3) We show a fairly straightforward approximation-preserving reduction from MaxView$\langle Q_2^\times\rangle$ to MaxView$\langle Q\rangle$. Let $I^\times$ and $A^\times$ be input for MaxView$\langle Q_2^\times\rangle$. We construct the input I and A for MaxView$\langle Q\rangle$, as follows. Let y_1 and y_2 be head variables of Q that are not atomic neighbors. For each pair $(a_1, a_2) \in Q_2^\times(I^\times)$, we define the mapping $\lambda_{a_1,a_2} : \mathsf{Var}(Q) \to \mathsf{Const}$, as follows.

$$\lambda_{a_1,a_2}(z) = \begin{cases} a_1 & \text{if } z = y_1; \\ a_2 & \text{if } z = y_2; \\ c & \text{otherwise.} \end{cases}$$

Next, for each $(a_1, a_2) \in Q_2^\times(I^\times)$ and atom φ of Q we add to I every fact of the form $\lambda_{a_1,a_2}(\varphi)$. We also add to A the tuple that is obtained from $\mathbf{y}$ by applying λ_{a_1,a_2}. This completes the reduction, and it is straightforward to show that this reduction is indeed approximation preserving (building on the fact that y_1 and y_2 are not atomic neighbors). In particular, for each solution $J^\times$ for $I^\times$ and $A^\times$ we can construct a solution J for I and A with $|Q(J)| = |Q_2^\times(J^\times)|$ and vice versa. □

Theorem 5.8 states that MaxView$\langle Q\rangle$ is NP-hard if Q has no level-1 head domination. It is known that a constant lower bound (in addition to a constant upper bound) exists on the polynomial-time approximability of MaxView$_1\langle Q\rangle$ in case Q has no head domination (see Comment 2.2). But it remains open whether MaxView$\langle Q\rangle$ has a constant-factor approximation for *any* sjf-CQs Q without level-1 head domination, such that Q is not among those of Theorem 6.3.

6.1.2 Approximation for Bounded Deletion

In this section we consider the complexity of approximating MaxView$_c\langle Q\rangle$. In that case, one can delete the undesired answers one by one and, by applying Proposition 6.1, still get a constant-factor approximation. Hence, we get the following corollary.

Corollary 6.5. *If Q is an sjf-CQ with h-coverage, then* MaxView$_c\langle Q\rangle$ *is h^c-approximable in polynomial time.*

In the remainder of this section, we reduce the approximation factor h^c of Corollary 6.5 to almost h (specifically, $h + \epsilon$ for all $\epsilon > 0$) by using further insights into the problem MaxView$_c\langle Q\rangle$. We begin by describing a particularly naive algorithm for MaxView$\langle Q\rangle$, which we call the *unirelation algorithm*, and which extends the algorithm originally proposed by Buneman et al. [5].

Consider an sjf-CQ $Q(\mathbf{y})$, and let I and A be input for MaxView$\langle Q\rangle$. For each atom φ of Q, the solution J_φ for I and A is obtained by removing from the relation R_φ^I every fact that is Q-consistent with at least one of the tuples in A. Recall that a fact f of R_φ^I is Q-consistent with a tuple $\mathbf{a} \in A$ if there is some mapping μ for $\mathsf{Var}(Q)$, where μ is not necessarily a match for Q in I, such that $\mu(\varphi) = f$ and $\mu(\mathbf{y}) = \mathbf{a}$. The unirelation algorithm simply constructs the solution J_φ for each $\varphi \in \text{atoms}(Q)$, and returns the *best* J_φ, that is, the one with the maximal $|Q(J_\varphi)|$. Observe that the unirelation algorithm is indeed highly naive, since it does not even consider simultaneous deletions of facts from different relations. Moreover, this algorithm does not guarantee any multiplicative-factor approximation; as evidence, observe that the solution J it returns for the input $I^\times$ and A of MaxView$\langle Q_2^\times\rangle$ that we defined in Example 6.2 is such that $Q_2^\times(J) = \emptyset$. However, this algorithm does provide an approximation guarantee that we will use for analyzing the complexity of approximating $\textsc{MaxView}_c\langle Q\rangle$. This guarantee is stated by the following lemma.

Lemma 6.6. *Let Q be an sjf-CQ with h-coverage. Let I and A be input for* $\textsc{MaxView}\langle Q\rangle$*. The unirelation algorithm returns a solution J with*

$$|Q(J)| \geq |Q(I)|/h - |A|^h .$$

Proof. Suppose that the atoms $\varphi_1, \dots, \varphi_h$ cover the head variables of Q. For $i = 1, \dots, h$, let C_i be the set of tuples $\mathbf{b}$ in $Q(I)$ such that for some $\mathbf{a} \in A$, the projection of $\mathbf{b}$ to $\mathsf{Var}_\mathsf{h}(\varphi_i)$ is the same as that of $\mathbf{a}$. Let C be the set $C_1 \cap \cdots \cap C_h$. An easy observation is that $|C| \leq |A|^h$, since $\varphi_1, \dots, \varphi_h$ cover $\mathsf{Var}_\mathsf{h}(Q)$. By definition, each tuple $\mathbf{a} \in Q(I) \setminus C$ is excluded from some C_i. Now, a simple counting argument implies that there is some j such that C_j excludes at least $|Q(I) \setminus C|/h$ answers. The solution J_{φ_j} is such that $Q(J_{\varphi_j})$ contains all those excluded answers, and hence, $|Q(J)| \geq |Q(I)|/h - |A|^h$. So, we get the correctness of the lemma, since J_{φ_j} is of the solutions considered by the unirelation algorithm. □

The next lemma implies that $\textsc{MaxView}\langle Q\rangle$ can be solved optimally, in polynomial time, when $Q(I)$ is of a bounded cardinality.

Lemma 6.7. *Let Q be an sjf-CQ, and let d be a fixed number. Given the input I and A for* $\textsc{MaxView}\langle Q\rangle$*, if we assume that $|Q(I) \setminus A| \leq d$, then* $\textsc{MaxView}\langle Q\rangle$ *can be solved in polynomial time.*

Proof. Let $D = Q(I) \setminus A$. If J is any solution, then $Q(J) \subseteq D$. For each $\mathbf{a} \in Q(J)$, fix an arbitrary match $\mu_\mathbf{a}$ for Q in J, such that $\mu_\mathbf{a}(\mathbf{y}) = \mathbf{a}$. Let J' be the subinstance of J that consists of all the facts $\mu_\mathbf{a}(\varphi)$ where $\mathbf{a} \in Q(J)$ and $\varphi \in \text{atoms}(Q)$. Note that $Q(J) = Q(J')$. Moreover, observe that $|Q(J)| \leq d \cdot |\text{atoms}(Q)|$. It thus suffices to search for an optimal solution among all the subinstances of I of a cardinality bounded by the fixed number $d \cdot |\text{atoms}(Q)|$. And of course, searching all those subinstances can be done in polynomial time. □

We can now prove our upper bound for approximating $\textsc{MaxView}_c\langle Q\rangle$.

Theorem 6.8. *Let Q be an sjf-CQ with h-coverage, and let $\epsilon > 0$. The problem* $\textsc{MaxView}_c\langle Q\rangle$ *is $(h+\epsilon)$-approximable in polynomial time.*

Proof. Let I and A be input for $\textsc{MaxView}_c\langle Q\rangle$. We fix a numeric function $f(h, c, \epsilon)$ that we determine later. Note that $f(h, c, \epsilon)$ does not depend on the input I and A. In particular, if $|Q(I) \setminus A|$ has at most $f(h, c, \epsilon)$ answers, then we apply Lemma 6.7 to get an optimal solution in polynomial time. So assume that $|Q(I) \setminus A| \geq f(h, c, \epsilon)$. We will define $f(h, c, \epsilon)$ to be such that Lemma 6.6 gives a $(h+\epsilon)$-approximation. Specifically, we need the following to hold.

$$(h+\epsilon)(|Q(I)|/h - |A|^h) \geq |Q(I) \setminus A|$$

Hence, it suffices to require the following:

$$|Q(I)| \left(\frac{h+\epsilon}{h} - 1\right) \geq |A| + (h+\epsilon)|A|^h$$

Then, using the assumption that $|A| \leq c$ we get that it suffices to choose $f(h, c, \epsilon)$ as:

$$f(h, c, \epsilon) = 2h(h+\epsilon)c^h/\epsilon$$

That completes the proof. □

6.2 Functional Dependencies

In this section, we generalize the complexity results of Section 5 to incorporate functional dependencies.

6.2.1 Notation

We begin with some notation. Recall that a schema is a finite sequence $\mathbf{R}$ of relation symbols. We extend the definition of a schema to include functional dependencies. A functional dependency (over $\mathbf{R}$), abbreviated *fd*, has the form $R_i : A \to B$, where $R_i \in \mathbf{R}$ and A and B are subsets of $\{1, \dots, \text{arity}(R_i)\}$. So, in this section a schema has the form $\mathbf{S} = (\mathbf{R}, \Delta)$, where Δ is a set of functional dependencies. An instance over $\mathbf{S}$ satisfies every fd in Δ, which means that for every fd $R_i : A \to B$ in Δ and tuples $\mathbf{t}$ and $\mathbf{u}$ in R_i^I, if $\mathbf{t}$ and $\mathbf{u}$ agree on (i.e., have the same values for) the indices of A, then they also agree on the indices of B.

Let $\mathbf{S} = (\mathbf{R}, \Delta)$ be a schema, and let Q be a CQ over $\mathbf{S}$. If μ is an assignment for Q and Z is a subset of $\mathsf{Var}(Q)$, then $\mu[Z]$ denotes the restriction of μ to the variables in Z (i.e., Z is the domain of $\mu[Z]$, and $\mu[Z](z) = \mu(z)$ for all $z \in Z$). Let Z_1 and Z_2 be subsets of $\mathsf{Var}(Q)$. We denote by $Q : Z_1 \to Z_2$ the functional dependency stating that for every instance I over $\mathbf{S}$ and matches $\mu, \mu' \in \mathcal{M}(Q, I)$ we have:

$$\mu[Z_1] = \mu'[Z_1] \Rightarrow \mu[Z_2] = \mu'[Z_2]$$

Note that when Δ is empty, $Q : Z_1 \to Z_2$ is equivalent to $Z_1 \supseteq Z_2$. We may write $Q : Z \to z$ instead of $Q : Z \to \{z\}$. The *image* of Z, denoted *img*(Z), is the set of all the variables $z \in \mathsf{Var}(Q)$ with $Q : Z \to z$. Note that $Z \subseteq$ *img*(Z). For $\varphi \in \text{atoms}(Q)$, we use *img*$(\varphi)$ as a shorthand notation of *img*$(\mathsf{Var}(\varphi))$.

Example 6.9. The schema $\mathbf{S}$ in our running example for this section has 4 relation symbols, ternary R, binary S, quaternary T and binary U, and the following fds:

$$R : 1 \to 2 \quad R : 2 \to 3 \quad T : 1 \to 2 \quad T : \{2, 3\} \to 4$$

Let Q_0 be the following sjf-CQ:

$$Q_0(y_1, y_2, y_3) :- R(x_1, y_1, x_3), S(x_1, x_2), T(x_2, y_2, x_3, x_4), U(x_4, y_3)$$

We have $Q_0 : \{x_1, x_2\} \to y_1$ due to the fd $R : 1 \to 2$. As a result, we also have $Q_0 : \{x_1, x_2\} \to x_3$ due to the fd $R : 2 \to 3$. Let φ_S be the atom $S(x_1, x_2)$. The reader can verify that *img*$(\varphi_S) = \{x_1, x_2, y_1, x_3, y_2, x_4\}$. □

Functional head domination [16] is a generalization of head domination that incorporates functional dependencies. Here we refine that generalized definition with the corresponding level-k parameterization.

Definition 6.10. (**Functional Head Domination**) A CQ Q over a schema $\mathbf{S}$ has *functional head-domination* if there is a subset Φ of atoms(Q) such that for every connected component P of $\mathcal{G}_\exists(Q)$ there is $\varphi \in \Phi$ with $\mathsf{Var}_\mathsf{h}(P) \subseteq$ *img*(φ); in that case, Q has *level-k* functional head domination if k is the minimal cardinality of such a subset Φ. □

Example 6.11. We continue our running example over the schema $\mathbf{S}$ and sjf-CQ Q_0 defined in Example 6.9. The graph $\mathcal{G}_\exists(Q_0)$ is shown in the top of Figure 4. Observe that $\mathcal{G}_\exists(Q_0)$ has one connected component, with the head

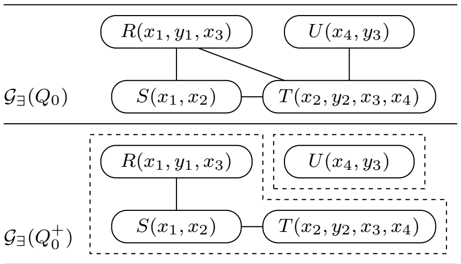

Figure 4: $\mathcal{G}_\exists(Q_0)$ for the CQ Q_0 of Example 6.9 (top), and $\mathcal{G}_\exists(Q_0^+)$ for the CQ Q_0^+ of Example 6.12 (bottom)

variables y_1, y_2 and y_3. The reader can verify that no atom φ of Q_0 satisfies $\{y_1, y_2, y_3\} \subseteq img(\varphi)$. Therefore, Q_0 does not have functional head domination. □

For a CQ Q over a schema $\mathbf{S}$, we denote by Q^+ the CQ that is obtained by appending to the head of Q (in some order) all the existential variables that are functionally determined by the head variables; so, the set $\mathsf{Var}_\mathsf{h}(Q^+)$ of head variables of Q^+ is not $\mathsf{Var}_\mathsf{h}(Q)$, but rather the set $img(\mathsf{Var}_\mathsf{h}(Q))$.

Example 6.12. For the schema $\mathbf{S}$ and sjf-CQ Q_0 of Example 6.9, we have:

$$img(\{y_1, y_2, y_3\}) = \{y_1, y_2, y_3, x_3, x_4\}$$

Therefore, the CQ Q_0^+ is the same as Q_0, except that the head of Q_0^+ is $Q_0^+(y_1, y_2, y_3, x_3, x_4)$. The graph $\mathcal{G}_\exists(Q_0^+)$ is shown in the middle part of Figure 4. Observe that there is no edge between the atoms $R(x_1, y_1, x_3)$ and $T(x_2, y_2, x_3, x_4)$ since x_3 is now a head variable. Similarly, there is no edge between $T(x_2, y_2, x_3, x_4)$ and $U(x_4, y_3)$ since x_4 is a head variable. As shown in the figure, $\mathcal{G}_\exists(Q_0^+)$ has two connected components, one with the set of head variables $Y_1 = \{y_1, y_2, x_3, x_4\}$, and one with the set of head variables $Y_2 = \{x_4, y_3\}$. Let φ_S and φ_U be the atoms $S(x_1, x_2)$ and $U(x_4, y_3)$, respectively. Then we have $Y_1 \subseteq img(\varphi_S)$ and $Y_2 \subseteq img(\varphi_U)$. Therefore, Q_0^+ has functional head domination. And since Q_0^+ does not have an atom φ with all the head variables in $img(\varphi)$ (hence Q_0^+ does not have level-1 functional head domination), we conclude that Q_0^+ has level-2 functional head domination. □

6.2.2 Generalization

As we will see, functional dependencies affect the complexity of the problems we discussed in Section 5. We parameterize these problems with the schema $\mathbf{S}$ and denote them by MinVSE$\langle \mathbf{S}, Q\rangle$, MinVSE$_k\langle \mathbf{S}, Q\rangle$, FreeVSE$\langle \mathbf{S}, Q\rangle$ and FreeVSE$_k\langle \mathbf{S}, Q\rangle$. We may continue to avoid mentioning $\mathbf{S}$ in the case where there are no functional dependencies. Kimelfeld [16] generalized Theorem 4.6 to incorporate functional dependencies.

Theorem 6.13. [16] *Let $\mathbf{S}$ be a schema, and let Q be an sjf-CQ over $\mathbf{S}$.*

1. *If Q^+ has functional head domination, then the problem* MinVSE$_1\langle \mathbf{S}, Q\rangle$ *(hence,* FreeVSE$_1\langle \mathbf{S}, Q\rangle$*) can be solved in polynomial time.*

2. *If Q^+ does not have functional head domination, then* FreeVSE$_c\langle \mathbf{S}, Q\rangle$ *is NP-complete (and, hence, approximating* MinVSE$_c\langle \mathbf{S}, Q\rangle$ *is NP-hard).*

Comment 6.14. *Theorem 6.13 is a weakening of the main result of [16], similarly to the way Theorem 4.6 is a weakening of the main theorem of [17] (see Comment 4.7).*

Example 6.15. We continue our running example with the sjf-CQ Q_0 of Example 6.9. Recall from Example 6.12 that Q_0^+ has functional head domination. Therefore, Theorem 6.13 states that MinVSE$_1\langle \mathbf{S}, Q_0\rangle$ is solvable (optimally) in polynomial time. □

Note that Theorem 6.13 generalizes Theorem 4.6 by replacing "head domination" with "functional head domination" and "Q" with "Q^+." The next theorem similarly generalizes Theorems 5.11 and 5.8, showing that Table 1 generalizes to account for functional dependencies (up to the needed replacement of terms).

Theorem 6.16. *Let $\mathbf{S}$ be a schema, and let Q be an sjf-CQ over $\mathbf{S}$, and let c be a fixed natural number.*

1. *If Q^+ has level-*1 *functional head domination, then* MinVSE$\langle \mathbf{S}, Q\rangle$ *is solvable in polynomial time.*

2. *If Q^+ has level-k functional head domination for $k >$ 1, then* MinVSE$\langle \mathbf{S}, Q\rangle$ *is NP-hard but k-approximable in polynomial time; moreover,* FreeVSE$\langle \mathbf{S}, Q\rangle$ *and* MinVSE$_c\langle \mathbf{S}, Q\rangle$ *are then solvable in polynomial time.*

3. *If Q^+ does not have functional head domination, then* FreeVSE$_1\langle \mathbf{S}, Q\rangle$ *is NP-complete; hence, it is NP-hard to approximate* MinVSE$_1\langle \mathbf{S}, Q\rangle$ *by any finite factor.*

Example 6.17. To complete our running example, recall from Example 6.15 that MinVSE$_1\langle \mathbf{S}, Q_0\rangle$ is solvable in polynomial time. Recall from Example 6.12 that Q_0^+ has level-2 functional head domination. Then Part 2 of Theorem 6.16 states that MinVSE$\langle \mathbf{S}, Q\rangle$ is NP-hard, but 2-approximable in polynomial time. It also says that FreeVSE$\langle \mathbf{S}, Q_0\rangle$ and MinVSE$_c\langle \mathbf{S}, Q_0\rangle$ are both solvable in polynomial time (since Q_0^+ has functional head domination). □

The proofs of parts 1 and 3 of Theorem 6.16 are similar to those we described for their correspondents in Theorems 5.11 and 5.8. The more involved proof is for part 2, and specifically the generalization of Lemma 5.6 to incorporate fds. That proof, which we do not provide here for lack of space, will appear in the full version of this paper.

7. CONCLUSIONS AND DISCUSSION

We studied the computational complexity of deletion propagation when the view is defined by an sjf-CQ, and multiple tuples are to be deleted. In particular, we investigated the problem MinVSE$\langle Q\rangle$ and showed a trichotomy in complexity, classifying the sjf-CQs into those where the problem is in polynomial time, NP-hard but approximable, and inapproximable. We extended these results to MinVSE$_c\langle Q\rangle$, where at most c tuples are deleted, and generalized them to accommodate functional dependencies. We also studied the problems MaxView$\langle Q\rangle$ and MaxView$_c\langle Q\rangle$, where the goal is to approximate an optimal solution by aiming to maximize the remaining view (rather than to minimize the side effect). There, we showed that known results on the approximability of MaxView$_1\langle Q\rangle$ extend to MaxView$_c\langle Q\rangle$, but not to MaxView$\langle Q\rangle$; the latter problem is inapproximable by any constant factor (under the complexity assumption of Ambühl et al. [1]) whenever some pair of head variables does not possess atomic neighboring.

Many questions regarding the complexity of deletion propagation remain unanswered. Does any of the known dichotomy or trichotomy results, for bounded or unbounded number of tuples, extend to CQs with self joins? It is known that head domination is not a sufficient condition for the tractability of finding an optimal solution if self joins are allowed [17]. Can $\textsc{MaxView}_c\langle Q\rangle$ be approximated better than shown here? Can our established inapproximability for $\textsc{MaxView}\langle Q\rangle$ be generalized to a classification over the entire class of (sjf-)CQs Q? How do our results extend to handle multiple views (i.e., deleted tuples come from multiple views and/or the goal is to *simultaneously* minimize multiple views); we note that our reduction to focused hitting set extends to multiple views, but it not at all clear how and whether the trichotomy so extends. Another question relates to privacy: if the source relations contain sensitive information that is hidden from the users of the view, how do we avoid leaking that information through the side effect?

The work reported in this paper relates to the concept of *causality in databases* that has lately gained interest in the database research community [20,21,24]. In fact, the debugging motivation from the beginning of the introduction can be viewed as a special case thereof. In the work on causality, the goal is to detect source facts that hold high *responsibility* for the membership of given answers in the view. The facts deleted in an (approximately) optimal solution of our setting could serve as candidates of such owners of responsibility. Validity is according to the semantics of Meliou et al. [20]—eliminating the facts causes the elimination of the answers; but here, the extent of responsibility is based on the *focus* on the answers. As an example, suppose that "Alin lives in USA" is the result of joining two facts: "Alin lives in CA" and "CA is in USA." The former fact would be assigned a higher responsibility, since eliminating the latter would incur a larger side effect. This notion of causality is different from the one of Meliou et al. [20] that, interestingly, corresponds to deletion propagation with a minimal *source* side effect (and, in particular, assigns the same responsibility to both facts in the above example).

Our complexity results in this paper add to the dichotomy results established by the theoretical database research. Besides the ones on deletion propagation we referenced in the body of this paper, various dichotomies in the complexity of operations involving CQs (and sjf-CQs) have been proved. Dalvi and Suciu [10,11] studied query evaluation on probabilistic databases with independent tuples, and classified the CQs into those that can be evaluated in polynomial time and those that are #P-hard. Meliou et al. [22] showed a dichotomy (polynomial time vs. NP-completeness) in the complexity of computing the degree of responsibility of source facts to the tuples of an sjf-CQ. Kolaitis and Pema [18] proved a dichotomy (polynomial time vs. coNP-hardness) in the complexity of computing the consistent answers of a Boolean sjf-CQ with exactly two atoms. Finally, Maslowski and Wijsen [19] showed a dichotomy (polynomial time vs. #P-hardness) in the complexity of counting the database repairs that satisfy a Boolean sjf-CQ.

8. REFERENCES

[1] C. Ambühl, M. Mastrolilli, and O. Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM J. Comput.*, 40(2):567–596, 2011.

[2] F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Trans. Database Syst.*, 6(4):557–575, 1981.

[3] D. M. J. Barbosa, J. Cretin, N. Foster, M. Greenberg, and B. C. Pierce. Matching lenses: alignment and view update. In *ICFP*, pages 193–204. ACM, 2010.

[4] A. Bohannon, B. C. Pierce, and J. A. Vaughan. Relational lenses: a language for updatable views. In *PODS*, pages 338–347. ACM, 2006.

[5] P. Buneman, S. Khanna, and W. C. Tan. On propagation of deletions and annotations through views. In *PODS*, pages 150–158, 2002.

[6] G. Cong, W. Fan, and F. Geerts. Annotation propagation revisited for key preserving views. In *CIKM*, pages 632–641, 2006.

[7] G. Cong, W. Fan, F. Geerts, J. Li, and J. Luo. On the complexity of view update analysis and its application to annotation propagation. *IEEE Trans. Knowl. Data Eng.*, 24(3):506–519, 2012.

[8] S. S. Cosmadakis and C. H. Papadimitriou. Updates of relational views. *J. ACM*, 31(4):742–760, 1984.

[9] Y. Cui and J. Widom. Run-time translation of view tuple deletions using data lineage. Technical report, Stanford University, 2001. http://dbpubs.stanford.edu:8090/pub/2001-24.

[10] N. N. Dalvi, K. Schnaitter, and D. Suciu. Computing query probability with incidence algebras. In *PODS*, pages 203–214, 2010.

[11] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, 1999.

[13] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *PODS*, pages 352–365. ACM, 1983.

[14] V. Guruswami and R. Saket. On the inapproximability of vertex cover on k-partite k-uniform hypergraphs. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2010.

[15] A. M. Keller. Algorithms for translating view updates to database updates for views involving selections, projections, and joins. In *PODS*, pages 154–163. ACM, 1985.

[16] B. Kimelfeld. A dichotomy in the complexity of deletion propagation with functional dependencies. In *PODS*, pages 191–202. ACM, 2012.

[17] B. Kimelfeld, J. Vondrák, and R. Williams. Maximizing conjunctive views in deletion propagation. *ACM Trans. Database Syst.*, 37(4):24, 2012.

[18] P. G. Kolaitis and E. Pema. A dichotomy in the complexity of consistent query answering for queries with two atoms. In press, 2011.

[19] D. Maslowski and J. Wijsen. On counting database repairs. In *LID*, pages 15–22, 2011.

[20] A. Meliou, W. Gatterbauer, J. Y. Halpern, C. Koch, K. F. Moore, and D. Suciu. Causality in databases. *IEEE Data Eng. Bull.*, 33(3):59–67, 2010.

[21] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. The complexity of causality and responsibility for query answers and non-answers. *PVLDB*, 4(1):34–45, 2010.

[22] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. The complexity of causality and responsibility for query answers and non-answers. *PVLDB*, 4(1):34–45, 2010.

[23] R. Peeters. The maximum edge biclique problem is np-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.

[24] B. Qin, S. Wang, and X. Du. Efficient responsibility analysis for query answers. In *Database Systems for Advanced Applications*, volume 7825 of *LNCS*, pages 239–253. Springer, 2013.

[25] V. V. Vazirani. *Approximation Algorithms.* Springer, 2003.