Size Bounds and Query Plans for Relational Joins

Albert Atserias
Universitat Politècnica de Catalunya
Barcelona, Spain
atserias@lsi.upc.edu

Martin Grohe
Humboldt Universität zu Berlin
Berlin, Germany
grohe@informatik.hu-berlin.de

Dániel Marx*
Budapest University of Technology and Economics
Budapest, Hungary
dmarx@cs.bme.hu

Abstract

Relational joins are at the core of relational algebra, which in turn is the core of the standard database query language SQL. As their evaluation is expensive and very often dominated by the output size, it is an important task for database query optimisers to compute estimates on the size of joins and to find good execution plans for sequences of joins. We study these problems from a theoretical perspective, both in the worst-case model, and in an average-case model where the database is chosen according to a known probability distribution. In the former case, our first key observation is that the worst-case size of a query is characterised by the fractional edge cover number of its underlying hypergraph, a combinatorial parameter previously known to provide an upper bound. We complete the picture by proving a matching lower bound, and by showing that there exist queries for which the join-project plan suggested by the fractional edge cover approach may be substantially better than any join plan that does not use intermediate projections. On the other hand, we show that in the average-case model, every join-project plan can be turned into a plan containing no projections in such a way that the expected time to evaluate the plan increases only by a constant factor independent of the size of the database. Not surprisingly, the key combinatorial parameter in this context is the maximum density of the underlying hypergraph. We show how to make effective use of this parameter to eliminate the projections.

*The third author is supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (OTKA grant 67651).

1. Introduction

The join operation is one of the core operations of relational algebra, which in turn is the core of the standard database query language SQL. The two key components of a database system executing SQL-queries are the query optimiser and the execution engine. The optimiser translates the query into several possible execution plans, which are basically terms of the relational algebra (also called operator trees) arranging the operations that have to be carried out in a tree-like order. Using statistical information about the data, the optimiser estimates the execution cost of the different plans and passes the best one on to the execution engine, which then executes the plan and computes the result of the query. See [4] for a survey of query optimisation techniques.

Among the relational algebra operations, joins are usually the most costly, simply because a join of two relations, just like a Cartesian product of two sets, may be much larger than the relations. Therefore, query optimisers pay particular attention to the execution of joins, especially to the execution order of sequences of joins, and to estimating the size of joins. In this paper, we address the very fundamental questions of how to estimate the size of a sequence of joins and how to execute the sequence best from a theoretical point of view. While these questions have been intensely studied in practice, and numerous heuristics and efficiently solvable special cases are known (see, e.g., [4, 9, 8]), the very basic theoretical results we present here and their consequences apparently have not been noticed so far. Our key starting observation is that the size of a a sequence of joins is tightly linked to two combinatorial parameters of the underlying database schema, the *fractional edge cover number*, and the *maximum density*.

To make this precise, we need to get a bit more technical:

A *join query* Q is an expression of the form

$$R_1(x_{11},\ldots,x_{1r_1}) \bowtie \cdots \bowtie R_m(x_{m1},\ldots,x_{mr_m}),$$

where the R_i are *relation names* with *attributes* $x_{i1},\ldots,x_{ir_i}$. Let A be the set of all attributes occurring in Q and $n = |A|$. A *database instance* D for Q consists of relations $R_i(D)$ of arity r_i. It is common to think of the relation $R_i(D)$ as a table whose columns are labelled by the attributes $x_{i1},\ldots,x_{ir_i}$ and whose rows are the tuples in the relation. Then *answer*, or *set of solutions*, of the query Q in D is the n-ary relation $Q(D)$ with attributes A consisting of all tuples t whose projection on the attributes of R_i belongs to the relation $R_i(D)$, for all i. Hence we are considering *natural joins* here (all of our results can easily be transferred to *equi-joins*, but not to general θ*-joins*). Now the most basic question is how large $Q(D)$ can get in terms of $|D|$. We address this question both in the worst case and the average case, and also subject to various constraints imposed on D.

An *execution plan* for a join query describes how to carry out the evaluation of the query by simple operations of the relational algebra such as joins of two relations or projections. The obvious execution plans for a join query break up the sequence of joins into pairwise joins and arrange these in a tree-like fashion. We call such execution plans *join plans*. As described in [4], most practical query engines simply arrange the joins in some linear (and not even a tree-like) order and then evaluate them in this order. However, it is also possible to use other operations, in particular projections, in an execution plan for a join query. We call execution plans that use joins and projections *join-project plans*. It is one of our main results that, even though projections are not necessary to evaluate join queries, their use may speed up the evaluation of a query super-polynomially.

1.1. Fractional covers and worst-case size. Recall that an *edge cover* of a hypergraph H is a set C of edges of H such that each vertex is contained in at least one edge in C, and the *edge cover number* $\rho(H)$ of H is the minimum size of an edge cover of H. A *fractional edge cover* of H is a feasible solution for the linear programming relaxation of the integer linear program describing edge covers, and the *fractional edge cover number* $\rho^*(H)$ of H is the cost of an optimal solution. With a join query Q of the form (1) we can associate a hypergraph $H(Q)$ whose vertex set is the set of all attributes of Q and whose edges are the attribute sets of the relations R_i. The *(fractional) edge cover number* of Q is defined by $\rho(Q) = \rho(H(Q))$ and $\rho^*(Q) = \rho^*(H(Q))$.

An often observed fact about edge covers is that for every given database D, the size of $Q(D)$ is bounded by $|D|^{\rho(Q)}$, where $|D|$ is the total number of tuples in D. Much less obvious is the fact that the size of $Q(D)$ can actually be bounded by $|D|^{\rho^*(Q)}$, as proved by the second and third author [10] in the context (and the language) of constraint satisfaction problems. This is a consequence to Shearer's Lemma [5], which is a combinatorial consequence of the submodularity of the entropy function, and is closely related to a result due to Friedgut and Kahn [7] on the number of copies of a hypergraph in another. Our first and most basic observation is that the fractional edge cover number $\rho^*(Q)$ also provides a lower bound to the worst-case answer size: we show that for every Q, there exist arbitrarily large databases D for which the size of $Q(D)$ is at least $|D|^{\rho^*(Q)} \cdot |Q|^{-1}$. The proof is a simple application of linear programming duality. Another result from [10] implies that for every join query there is a join-project plan, which can easily be obtained from the query and certainly be computed in polynomial time, that computes $Q(D)$ in time $O(|Q|^2 \cdot |D|^{\rho^*+1})$. Our lower bound shows that this is optimal up to a polynomial factor (of $|Q|^{2+\rho^*} \cdot |D|$, to be precise). In particular, we get the following equivalences giving an exact combinatorial characterisation of all classes of join queries that have polynomial size answers and can be evaluated in polynomial time.

Theorem 1. *Let $\mathcal{Q}$ be a class of join queries. Then the following statements are equivalent:*

(1) $\mathcal{Q}$ has bounded fractional edge cover number.

(2) Queries in $\mathcal{Q}$ have answers of polynomial size.

(3) Queries in $\mathcal{Q}$ can be evaluated in polynomial time.

(4) Queries in $\mathcal{Q}$ can be evaluated in polynomial time by an explicit join-project plan.

Note that it is not even obvious that the second and third statements are equivalent, that is, that for every class of queries with polynomial size answers there is a polynomial time evaluation algorithm (the converse is of course trivial).

Hence with regards to worst-case complexity, join-project plans are optimal (up to a polynomial factor) for the evaluation of join queries. Our next result is that join plans are not: We prove that there are arbitrarily large join queries Q and database instances D such that our generic join-project plan computes $Q(D)$ in at most cubic time, whereas any join plan requires time $|D|^{\Omega(\log|Q|)}$ to compute $Q(D)$. We also observe that this bound is tight. Hence incorporating projections into a query plan may lead to a super-polynomial speed-up even if the projections are completely irrelevant for the query answer.

1.2. Maximum density and average-case size. Consider the model $\mathcal{D}(N,(p_R))$ of random databases where the tuples in each relation R are chosen randomly and independently with probability $p_R = p_R(N)$ from a domain of size N. This is the analogue of the Erdős-Rényi model of random graphs adapted to our context. It is easy to see that for D from $\mathcal{D}(N,(p_R))$, the expected size of the query answer $Q(D)$ is $N^n \cdot \prod_R p_R$, where n is the number of attributes and the product ranges over all relation names R in Q. The

question is whether $|Q(D)|$ will be concentrated around the expected value. This is governed by the *maximum density* $\overline{\delta}(Q,(p_R))$ of the query, a combinatorial parameter depending on the hypergraph of the query and the weights p_R. An application of the second moment method shows that if $\overline{\delta} = \log N - \omega(1)$, then $|Q(D)|$ is concentrated around its expected value, and if $\overline{\delta} = \log N + \omega(1)$, then $|Q(D)| = 0$ almost surely. We observe that the maximum density $\overline{\delta}$ can be computed in polynomial time using max-flow min-cut techniques. Interestingly, maximum density is closely related to the edge cover number of a matroid associated with the query Q (a generalisation of the bicircular matroid of a graph) — hence edge covers also show up in the average case scenario. Due to space limitation, we have to defer the details of this connection to the full version of this paper [3].

In view of the results about worst-case, it is a natural question whether join-project plans are more powerful than join plans in the average case setting as well. It turns out that this is not the case: We show that every join-project plan φ for Q can be turned into a join plan φ' for which the expected execution time increases only by a constant factor independent of the database. This may be viewed as our main technical result. The transformation of φ' into φ depends on a careful balance between delaying certain joins in order to reduce the number of attributes considered in each subquery occurring in the plan and keeping as many joins as possible in order to increase the density of the subquery. The choice of which subqueries to delay and which to keep is governed by a certain submodular function related to the density of the subqueries.

1.3. Size constraints. So far, we considered worst-case bounds which make no assumptions on the database, and average-case bounds which assume a known distribution on the database. However, practical query optimisers usually make use of known information about the databases when computing their size estimates. We consider the simplest such setting where the size of the relations is known, and we want to get a (worst case) estimate on the size of $Q(D)$ subject to the constraint that the relations in D have the given sizes. By suitably modifying the objective function of the linear program for edge covers, we obtain results analogous to those obtained for the unconstrained setting. A notable difference between the two results is that here the gap between upper and lower bound becomes 2^{-n}, where n is the number of attributes, instead of $|Q|^{-\rho^*}$. We give an example showing that the gap between upper and lower bound is essentially tight. However, this is not an inadequacy of our approach through fractional edge covers, but due to the inherent complexity of the problem: by a reduction from the maximum independent-set problem on graphs, we show that, unless NP = ZPP, there is no polynomial time algorithm that approximates the worst case answer size $|Q(D)|$ for given Q and relation sizes N_R by a better-than-exponential factor.

The structure of the paper follows this introduction. Omitted proofs are provided in full version of the paper [3].

2. Preliminaries

For integers $m \le n$, by $[m,n]$ we denote the set $\{m, m+1, \ldots, n\}$ and by $[n]$ we denote $[1,n]$.

Our terminology is similar to that used in [1]: An *attribute* is a symbol a with an associated *domain* $\mathrm{dom}(a)$. If not specified otherwise, we assume $\mathrm{dom}(a)$ to be an arbitrary countably infinite set, say, $\mathbb{N}$. Sometimes, we will impose restrictions on the size of the domains. A *relation name* is a symbol R with an associated finite set of attributes A. For a set $A = \{a_1, \ldots, a_n\}$ of attributes, we write $R(A)$ or $R(a_1, \ldots, a_n)$ to denote that A is the set of attributes of R. The *arity* of $R(A)$ is $|A|$. A *schema* is a finite set of relation names.

For a set A of attributes, an *A-tuple* is a mapping t that associates an element $t(a)$ from $\mathrm{dom}(a)$ with each $a \in A$. Occasionally, we denote A-tuples in the form $t = (t_a : a \in A)$, with the obvious meaning that t is the A-tuple with $t(a) = t_a$. The set of all A-tuples is denoted by $\mathrm{tup}(A)$. An *A-relation* is a set of A-tuples. The *active domain* of an A-relation R is the set $\{t(a) : t \in R, a \in A\}$. The *projection* of an A-tuple t to a subset $B \subseteq A$ is the restriction $\pi_B(t)$ of t to B, and the *projection* of an A-relation R is the set $\pi_B(R) = \{\pi_B(t) : t \in R\}$.

A *database instance* D of schema σ, or a *σ-instance*, consists of an A-relation $R(D)$ for every relation name R in σ with set of attributes A. The *active domain* of D is the union of active domains of all its relations. The *size* of a σ-instance D is $|D| := \sum_{R \in \sigma} |R(D)|$.

A *join query* is an expression

$$Q := R_1(A_1) \bowtie \cdots \bowtie R_m(A_m),$$

where R_i is a relation name with attributes A_i. The *schema* of Q is the set $\{R_1, \ldots, R_m\}$, and the *set of attributes* of Q is $\bigcup_i A_i$. We often denote the set of attributes of a join query Q by A_Q, and we write $\mathrm{tup}(Q)$ instead of $\mathrm{tup}(A_Q)$. We write $Q(A)$ to denote that A is the set of attributes of Q. The *size* of Q is $|Q| := \sum_i |A_i|$. We write $H(Q)$ for the (multi-)hypergraph that has vertex-set A and edge-(multi-)set $\{A_1, \ldots, A_m\}$. If D is a $\{R_1, \ldots, R_m\}$-instance, the *answer* of Q on D is the A-relation

$$Q(D) = \big\{ t \in \mathrm{tup}(A) : \pi_{A_i}(t) \in R_i(D) \text{ for every } i \in [m] \big\}.$$

A *join plan* is a term built from relation names and binary join operators. For example, $(R_1 \bowtie R_2) \bowtie (R_3 \bowtie R_4)$ and $((R_1 \bowtie R_2) \bowtie R_3) \bowtie (R_1 \bowtie R_4)$ are two join plans corresponding to the same join query $R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4$. A *join-project plan* is a term built from relation names, binary

join operators, and unary project operators. For example, $(\pi_A(R_1) \bowtie R_2) \bowtie \pi_B(R_1)$ is a join-project plan. Join-project plans have a natural representation as labelled binary trees, where the leaves are labelled by relation names, the unary nodes are labelled by projections π_A, and the binary nodes by joins. Evaluating a join plan or join-project plan φ in a database instance D means substituting the relation names by the actual relations from D and carrying out the operations in the expression. We denote the resulting relation by $\varphi(D)$. A join(-project) plan φ is a plan *for* a query Q if $\varphi(D) = Q(D)$ for every database D. The *subplans* of a join(-project) plan are defined in the obvious way. For example, the subplans of $(R_1 \bowtie R_2) \bowtie \pi_A(R_3 \bowtie R_4)$ are R_1, R_2, R_3, R_4, $R_1 \bowtie R_2$, $R_3 \bowtie R_4$, $\pi_A(R_3 \bowtie R_4)$, $(R_1 \bowtie R_2) \bowtie \pi_A(R_3 \bowtie R_4)$. If φ is a join project plan, then we often use A_φ to denote the set of attributes of the query computed by φ (this only includes "free" attributes and not those projected away by some projection in φ), and we write $\mathrm{tup}(\varphi)$ instead of $\mathrm{tup}(A_\varphi)$.

3. Worst-case model

3.1. Size bounds. Let Q be a join query with schema σ. For every $R \in \sigma$, let A_R be the set of attributes of R, and $A = \bigcup_R A_R$. Then fractional edge covers are precisely the feasible solutions $(x_R : R \in \sigma)$ for the following linear program L_Q, and the fractional edge cover number $\rho^*(Q)$ is the cost of an optimal solution.

$$\begin{array}{lllll} L_Q: & \text{minimise} & \sum_R x_R & & \\ & \text{subject to} & \sum_{R:\, a \in A_R} x_R \geq 1 & \text{for all } a \in A, \\ & & x_R \geq 0 & \text{for all } R \in \sigma. \end{array} \tag{3.1}$$

By standard arguments, there always is an optimal fractional edge cover whose values are rational and of bit-length polynomial in $|Q|$.

Lemma 2 ([10]). *Let Q be a join query with schema σ and let D be a σ-instance. Then for every fractional edge cover $(x_R : R \in \sigma)$ of Q we have*

$$|Q(D)| \leq \prod_{R \in \sigma} |R(D)|^{x_R}.$$

For the reader's convenience, we give a proof of this lemma, which is actually a simplification of the proof in [10], in the full version of the paper [3]. Note that the fractional edge cover in the statement of the lemma is not necessarily one of minimum cost. The next lemma shows that the upper bound of the previous lemma is tight:

Lemma 3. *Let Q be a join query with schema σ, and let $(x_R : R \in \sigma)$ be an optimal fractional edge cover of Q. Then for every $N_0 \in \mathbb{N}$ there is a σ-instance D such that $|D| \geq N_0$ and*

$$|Q(D)| \geq \prod_{R \in \sigma} |R(D)|^{x_R}.$$

Furthermore, we can choose D in such a way that $|R(D)| = |R'(D)|$ for all $R, R' \in \sigma$ with $x_R, x_{R'} > 0$.

Proof. Let A_R be the set of attributes of $R \in \sigma$ and $A = \bigcup_R A_R$. Recall $(x_R : R \in \sigma)$ is an optimal solution for the linear program (3.1). By LP-duality, there is a solution $(y_a : a \in A)$ for the dual linear program

$$\begin{array}{lllll} \text{maximise} & \sum_a y_a & & \\ \text{subject to} & \sum_{a \in A_R} y_a \leq 1 & \text{for all } R \in \sigma, \\ & y_a \geq 0 & \text{for all } a \in A \end{array} \tag{3.2}$$

such that $\sum_a y_a = \sum_R x_R$. There even exists such a solution with rational values.

We take an optimal solution $(y_a : a \in A)$ with $y_a = p_a/q$, where $q \geq 1$ and $p_a \geq 0$ are integers. Let $N_0 \in \mathbb{N}$, and let $N = N_0^q$. We define a σ-instance D by letting

$$R(D) := \left\{ t \in \mathrm{tup}(A_R) : t(a) \in [N^{p_a/q}] \text{ for all } a \in A_R \right\}$$

for all $R \in \sigma$. Here we assume that $\mathrm{dom}(a) = \mathbb{N}$ for all attributes a. As there is at least one a with $y_a > 0$ and hence $p_a \geq 1$, we have $|D| \geq N^{1/q} = N_0$. Observe that

$$|R(D)| = \prod_{a \in A_R} N^{p_a/q} = N^{\sum_{a \in A_R} y_a}$$

for all $R \in \sigma$. Furthermore, $Q(D)$ is the set of all tuples $t \in \mathrm{tup}(A)$ with $t(a) \in [N^{p_a/q}]$ for every $a \in A$. Hence

$$|Q(D)| = \prod_{a \in A} N^{p_a/q} = N^{\sum_{a \in A} y_a} = N^{\sum_{R \in \sigma} x_R} = \prod_{R \in \sigma} N^{x_R}.$$

By complementary slackness of linear programming we have

$$\sum_{a \in A_R} y_a = 1 \quad \text{for all } R \in \sigma \text{ with } x_R = 0.$$

Thus $|R(D)| = N$ for all $R \in \sigma$ with $x_R > 0$ and

$$|Q(D)| = \prod_{R \in \sigma} N^{x_R} = \prod_{R \in \sigma} |R(D)|^{x_R}.$$

□

Combining Lemmas 2 and 3 we obtain that, for every join query Q, every instance D gives $|Q(D)| \leq |D|^{\rho^*(Q)}$, and that there exist arbitrarily large instances D such that $|Q(D)| \geq |D|^{\rho^*(Q)} \cdot |Q|^{-1}$. This yields the equivalence between statements (1) and (4) of Theorem 1

3.2. Execution plans. It was proved in [10] that there is an algorithm for evaluating a join query Q in a database D that runs in time $O\big(|Q|^2 \cdot |D|^{\rho^*(Q)+1}\big)$. An analysis of the proof shows that the algorithms can actually be casted as the evaluation of an explicit (and simple) join-project plan. Combined with the bounds obtained in the previous section, this yields Theorem 1

We shall prove next that join plans perform significantly worse than join-project plans. Note that to evaluate a join plan one has to evaluate all its subplans. Hence for every subplan ψ of φ and every instance D, the size $|\psi(D)|$ is a lower bound for the time required to evaluate φ in D.

Theorem 4. *For every $m,N \in \mathbb{N}$ there are a join query Q and an instance D with $|Q| \geq m$ and $|D| \geq N$, and:*

(1) $\rho^*(Q) = 2$ and hence $|Q(D)| \leq |D|^2$ (actually, $|Q(D)| \leq |D|$).

(2) Every join plan φ for Q has a subplan ψ such that $|\psi(D)| \geq |D|^{\frac{1}{5}\log|Q|}$.

Proof. Let $n = \binom{2m}{m}$. For every $s \subseteq [2m]$ with $|s| = m$, let a_s be an attribute with domain $\mathbb{N}$. For every $i \in [2m]$, let R_i be a relation name having as attributes all a_s such that $i \in s$. Let A_i be the set of attributes of R_i and $A = \bigcup_{i\in[2m]} A_i$. The arity of R_i is $|A_i| = \binom{2m-1}{m-1} = \frac{m}{2m} \cdot \binom{2m}{m} = \frac{n}{2}$. Let $Q := R_1 \bowtie \cdots \bowtie R_{2m}$. Then $|Q| = 2m \cdot n/2 = m \cdot n$. Furthermore, $\rho^*(Q) \leq 2$. To see this, let $x_{R_i} = 1/m$ for every $i \in [2m]$. This forms a fractional edge cover of Q, because for every $s \subseteq [2m]$ with $|s| = m$, the attribute a_s appears in the m atoms R_i with $i \in s$.

Next, we define an instance D by letting $R_i(D)$ be the set of all A_i-tuples that have an arbitrary value from $[N]$ in one coordinate and 1 in all other coordinates. Observe that $|R_i(D)| = (N-1)n/2 + 1$ for all $i \in [2m]$ and thus

$$|D| = (N-1)mn + 2m \geq N.$$

Furthermore, $Q(D)$ is the set of all A-tuples that have an arbitrary value from $[N]$ in one coordinate and 1 in all other coordinates. Hence $|Q(D)| = (N-1)n+1 \leq |D|$. This completes the proof of (1).

To prove (2), we shall use the following simple (and well-known) combinatorial lemma:

Lemma 5. *Let T be a binary tree whose leaves are coloured with $2m$ colours, for some $m \geq 1$. Then there exists a node t of T such that at least $(m+2)/2$ and at most $m+1$ of the colours appear at leaves that are descendants of t.*

Proof. For every node t of T, let $c(t)$ be the number of colours that appear at descendants of T. The *height* of a node t is the length of the longest path from t to a leaf.

Let t be a node of minimum height such that $c(t) \geq m+2$, and let u_1, u_2 be the children of t. (Note that t cannot be a leaf because $c(t) \geq 2$.) Then $c(u_i) \leq m+1$ for $i = 1,2$. Furthermore, $c(u_1) + c(u_2) \geq c(t)$, hence $c(u_i) \geq (m+2)/2$ for at least one i. □

Continuing the proof of the theorem, we let φ be a join plan for Q. We view the term φ as a binary tree T whose leaves are labelled by atoms R_i. We view the atoms as colours. Applying the lemma, we find a node t of T such that at least $(m+2)/2$ and at most $m+1$ of the colours appear at leaves that are descendants of t. Every inner node of the tree corresponds to a subplan of φ. We let ψ be the subplan corresponding to t. Then at least $(m+2)/2$ and at most $m+1$ atoms R_i appear in ψ. By symmetry, we may assume without loss of generality that the atoms of ψ are $R_1, \ldots, R_\ell$ for some $\ell \in \left[\lceil (m+2)/2 \rceil, m+1\right]$. Hence ψ is a plan for the join query

$$R_1 \bowtie \cdots \bowtie R_\ell.$$

Let $B := \bigcup_{i=1}^{\ell} A_i$ be the set of all attributes occurring in ψ. For $i \in [m+1]$, let $s_i = \{i\} \cup [m+2, 2m]$. Then for all $i, j \in [\ell]$ we have $a_{s_i} \in A_j$ if and only if $i = j$. Hence all tuples $t \in \mathrm{tup}(B)$ with $t(a_{s_i}) \in [N]$ for all $i \in [\ell]$ and $t(b) = 1$ for all $b \in B \setminus \{a_{s_1}, \ldots, a_{s_\ell}\}$ are contained in $\psi(D)$. As there are N^ℓ such tuples, it follows that

$$|\psi(Q)| \geq N^\ell \geq N^{(m+2)/2}.$$

Statement (2) of the lemma follows, because

$$\log|Q| = \log m + \log n \leq \log m + \log 2^{2m} \leq 5 \cdot (m+2)/2,$$

provided m is large enough, which we may assume without loss of generality. □

Note that statement (2) of the theorem implies that any evaluation algorithm for the query Q based on evaluating join plans, which may even depend on the database instance, has a running time $O(|D|^{\Omega(\log|Q|)})$, in contrast with the running time $O(|D|^3)$ achieved by evaluating the join-project plan of Theorem 18. Using the well-known fact that the integrality gap of the linear program for edge covers is logarithmic in the number of vertices of the hypergraph (that is, attributes of the join query), it can be proved that for every query Q there is a join plan φ that can be evaluated in time $O(|D|^{2\rho^*(Q)\cdot\log|Q|})$, hence the lower bound is tight up to a small constant factor. Furthermore, this proofs shows that, for every join query Q, there is a join plan that can be evaluated in time $|D|^{\rho(Q)}$, where $\rho(Q)$ denotes the edge cover number of Q. However, we note that not only $|D|^{\rho(Q)}$ is potentially superpolynomial over $|D|^{\rho^*(Q)}$, but finding this plan is in general NP-hard. Compare this with the fact that the join-project plan given by [10] can be found efficiently.

4. Average-case model

In this section we assume that database is randomly generated according to the following model. Let σ be a schema and let A_R be the set of attributes of $R \in \sigma$. For every $R \in \sigma$, let $p_R : \mathbb{N} \to (0,1)$ be a function of N, and let $p(N) = (p_R(N) : R \in \sigma)$. We denote by $\mathscr{D}(N, p(N))$ the probability space on σ-instances with domain $[N]$ defined by placing each tuple $t \in [N]^{A_R}$ in $R(D)$ with probability

$p_R(N)$, independently for each tuple t and each $R \in \sigma$. Typical p's of interest are $p_R(N) = 1/2$, $p_R(N) = C \cdot N^{1-|A_R|}$, or $p_R(N) = N^{1-|A_R|}\log N$. When each $p_R = 1/2$, this is the uniform distribution over σ-instances with domain $[N]$.

4.1Size bounds and concentration. Let Q be a join query with schema σ, let n be the number of attributes of Q, and let m be the number of relation names in σ. Let X denote the size of the query answer $Q(D)$ when D is taken from $\mathscr{D}(N, p(N))$. The expectation of X is, trivially,

$$\mathrm{E}[X] = N^n \prod_{R\in\sigma} p_R(N). \tag{4.1}$$

We want to determine under what circumstances is $|Q(D)|$ concentrated around this value. For this we need to compute the variance of X, which depends on a parameter of Q defined next.

For every $R \in \sigma$, let w_R be a positive real weight, and let $w = (w_R : R \in \sigma)$. The *density* of Q with respect to w is defined as $\delta(Q,w) = \frac{1}{n}\sum_{R\in\sigma} w_R$. Note that if $w_R = 1$ for every R, then the density is m/n. For every $B \subseteq A$, let $Q[B]$ denote the subquery induced by B; that is, $Q[B]$ is the subquery formed by all the atoms $R \in \sigma$ that have all attributes in B. The *maximum density* of Q with respect to w is $\overline{\delta}(Q,w) = \max\{\delta(Q[B],w) : B \subseteq A, B \neq \emptyset\}$.

In applications to random instances, we typically fix $w_R(N)$ to $\log_2(1/p_R(N))$ and write $\delta(Q[B])$ and $\overline{\delta}(Q)$ instead of $\delta(Q[B],w)$ and $\overline{\delta}(Q,w)$. For this choice of weights a crucial distinction is made according to whether $\overline{\delta}$ is larger or smaller than $\log_2(N)$. In the first case, there exists subquery $Q[B]$ whose expected number of solutions is smaller than 1 and therefore, by Markov's inequality, Q itself has no solutions at all with probability bounded away from 0. In the second case, every subquery has at least one solution in expectation, and we can bound the variance of X as a function of $\overline{\delta}$. Since this will be of use later on, we derive it in detail. We show that, whenever $\overline{\delta} \le \log_2(N)$, we have the following bound for the variance:

$$\mathrm{V}[X] \le \mathrm{E}[X]^2 \cdot (2^n - 1)2^{\overline{\delta} - \log_2 N}. \tag{4.2}$$

Let A be the set of attributes of Q. For every $R \in \sigma$, let A_R be the set of attributes of R and for every $t \in [N]^{A_R}$, let $X(R,t)$ be the indicator for the event $t \in R(D)$. These are mutually independent random variables and the expectation of $X(R,t)$ is $p_R(N)$. For every $t \in [N]^A$, let $X(t)$ be the indicator for the event $t \in Q(D)$. Note that $X(t) = \prod_{R\in\sigma} X(R,t_R)$, where t_R denotes the projection of t to the attributes of R. Also $X = \sum_t X(t)$. Towards proving (4.2), let us bound

$$\mathrm{E}\left[X^2\right] = \sum_{s,t} \mathrm{E}\left[X(s)X(t)\right].$$

For every fixed $B \subseteq A$, let F_B be the set of pairs $(s,t) \in [N]^A \times [N]^A$ such that $s(a) = t(a)$ for every $a \in B$ and $s(a) \neq t(a)$ for every $a \in A - B$. Clearly, $(F_B)_{B\subseteq A}$ is a partition of $[N]^A \times [N]^A$ and therefore

$$\sum_{s,t} \mathrm{E}\left[X(s)X(t)\right] = \sum_{B\subseteq A} \sum_{(s,t)\in F_B} \mathrm{E}\left[X(s)X(t)\right].$$

Fix now some $B \subseteq A$ and $(s,t) \in F_B$, and let σ_B be the relations appearing in $Q[B]$. Observe that since s and t agree on B we have $t_R = s_R$ for every $R \in \sigma_B$ and therefore $X(R,s_R)X(R,t_R) = X(R,s_R)$ for every such R. Hence:

$$\begin{aligned} X(s)X(t) &= \prod_{R\in\sigma} X(R,s_R) \prod_{R\in\sigma} X(R,t_R) \\ &= \prod_{R\in\sigma\setminus\sigma_B} X(R,s_R)X(R,t_R) \prod_{R\in\sigma_B} X(R,s_R). \end{aligned}$$

All variables in the right-hand side product are mutually independent because either they involve different relations or different tuples. Therefore,

$$\mathrm{E}\left[X(s)X(t)\right] = \prod_{R\in\sigma\setminus\sigma_B} p_R^2 \prod_{R\in\sigma_B} p_R = \prod_{R\in\sigma} p_R^2 \prod_{R\in\sigma_B} p_R^{-1}.$$

The number of pairs (s,t) in F_B is bounded by $N^{2|A|-|B|}$. Therefore,

$$\begin{aligned} \sum_{(s,t)\in F_B} \mathrm{E}\left[X(s)X(t)\right] &\le N^{2|A|-|B|} \prod_{R\in\sigma} p_R^2 \prod_{R\in\sigma_B} p_R^{-1} \\ &= \mathrm{E}\left[X\right]^2 \cdot N^{-|B|} \prod_{R\in\sigma_B} p_R^{-1}. \end{aligned}$$

For $B = \emptyset$, the second term in the last line is 1 and we get $\mathrm{E}\left[X\right]^2$. For $B \neq \emptyset$, we have

$$N^{-|B|} \prod_{R\in\sigma_B} p_R^{-1} = N^{-|B|} 2^{|B|\,\delta(Q[B])} \le \left(N^{-1}2^{\overline{\delta}}\right)^{|B|} \le 2^{\overline{\delta} - \log_2 N}$$

where the first inequality holds because $\delta(Q[B]) \le \overline{\delta}$, and the second inequality holds because $|B| > 0$ and $\overline{\delta} \le \log_2(N)$. Putting it all together we get

$$\begin{aligned} \mathrm{E}\left[X^2\right] &= \sum_{B\subseteq A} \sum_{(s,t)\in F_B} \mathrm{E}\left[X(s)X(t)\right] \\ &\le \mathrm{E}\left[X\right]^2 + (2^n - 1)\,\mathrm{E}\left[X\right]^2 2^{\overline{\delta} - \log_2 N}. \end{aligned}$$

Since $\mathrm{V}\left[X\right] = \mathrm{E}\left[X^2\right] - \mathrm{E}\left[X\right]^2$, this proves (4.2).

In the following, if X is a random variable defined on the probability space $\mathscr{D}(N, p(N))$, the expression "$X \sim A$ a.s." means that for every $\varepsilon > 0$ and $\delta > 0$, there exists N_0 such that, for every $N \ge N_0$, we have $\Pr[\,|X - A| \le \varepsilon A\,] \ge 1 - \delta$. With all this notation, we obtain the following threshold behaviour as an immediate consequence to Markov's and Chebyshev's inequalities:

Theorem 6. *Let Q be a join query with schema σ and n attributes. For every $R \in \sigma$, let $p_R : \mathbb{N} \to (0,1)$, $p(N) = (p_R(N) : R \in \sigma)$, and $\overline{\delta}(N) = \overline{\delta}(Q, w_R(N))$ for $w_R(N) = \log_2(1/p_R(N))$. Let D be drawn from $\mathscr{D}(N, p(N))$ and let X denote the size of $Q(D)$.*

(1) If $\overline{\delta}(N) = \log N - \omega(1)$*, then* $X \sim N^n \prod_{R\in\sigma} p_R(N)$ *a.s.*
(2) If $\overline{\delta}(N) = \log N + \omega(1)$*, then* $X = 0$ *a.s.*

In certain occasions, the concentration defined by "$X \sim A$ almost surely" is not enough. For example, it may sometimes be necessary to conclude that $\Pr[\,|X-A| \le \varepsilon A\,] \ge 1 - N^{-d}$ for every $\varepsilon > 0$ and $d > 0$ in order to apply a union bound that involves a number of cases that grows polynomially with N. It turns out that such a strong concentration can also be guaranteed at the expense of a wider *threshold width* in Theorem 6: instead of $\log_2(N) - \omega(1)$ vs $\log_2(N) + \omega(1)$, we require $\log_2(N) - \omega(\log\log(N))$ vs $\log_2(N) + \omega(1)$. This does not follow from Chebyshev's inequality, and for the proof, which can be found in the full version [3], we use the polynomial concentration inequality from [12].

We conclude this section with the max-flow construction to compute $\overline{\delta}(Q,w)$. For every real number $\delta > 0$, we build a network $N(\delta)$ as follows. The network has a source s, a target t, and $|A| + |\sigma|$ intermediate nodes. There is a link of capacity δ between s and each $a \in A$. Each $a \in A$ has a link of infinite capacity to each $R \in \sigma$ with $a \in A_R$. And each $R \in \sigma$ is linked to t with capacity w_R. Recall that a cut in the network is a set of links that disconnects the target from the source. The capacity of the cut is the sum of the capacities of the links in it. Let $\gamma(Q,w,\delta)$ be the minimum capacity of all cuts of $N(\delta)$. It holds that $\gamma(Q,W,\delta) < \sum_{R\in\sigma} w_R$ if and only if $\overline{\delta}(Q,w) > \delta$. The proof of this is rather easy and can be found in Lemma 24 in [3]. By the max-flow min-cut algorithm, it follows that $\overline{\delta}(Q,w)$ is computable in polynomial time.

4.2Execution plans. Theorem 4 shows that certain queries admit a join-project plan that cannot be converted into a join plan without causing a superpolynomial increase in the worst-case running time. The following result shows that when we are considering average-case running time, projections may be eliminated at a very small expected cost.

Theorem 7. *Let Q be a join query with schema σ and n attributes. For every $R \in \sigma$, let $p_R : \mathbb{N} \to (0,1)$, $p(N) = (p_R(N) : R \in \sigma)$. Let D be drawn from $\mathscr{D}(N, p(N))$. If there is a join-project plan φ for Q such that $E(|\varphi'(D)|) \le T$ for every subplan φ' of φ, then there is a join plan ψ for Q such that $E(|\psi'(D)|) \le c_\varphi T$ for every subplan ψ' of ψ, where c_φ is a constant depending only on φ.*

The join plan ψ is obtained by iteratively using a procedure that is capable of reducing the number of projections by one in such a way that the expected size of each subplan increases only by a factor depending only on the query. In each iteration, the procedure selects a subplan $\pi_X(\varphi_0)$ of φ such that φ_0 contains no projections, i.e. this projection π_X is lowest in the tree representation of φ. In the first step of the procedure, we replace φ_0 with a join plan φ_0' that contains only those relations appearing in φ_0 whose attributes are completely contained in X^*, where $X^* \supseteq X$ is an appropriate subset of attributes. In the second step, the projection π_X is removed (or, in other words, π_X is replaced by π_{X^*}, making it redundant). The key step of the algorithm is choosing the right X^*. If X^* is too small, then φ_0' is much less restrictive than φ_0, hence $|\pi_X(\varphi_0'(D))|$ can be much larger than $|\pi_X(\varphi_0(D))|$. On the other hand, if X^* is too large, then $|\pi_{X^*}(\varphi_0'(D))|$ can be much larger than $|\pi_X(\varphi_0'(D))|$.

Let $S \subseteq \sigma$ be a set of relations over the attributes A and denote by A_R the attributes of a relation R. For a subset $X \subseteq A$, let

$$f_S(X) := |X|(\log N - n - 1) - \sum_{R\in S[X]} w_R(N),$$

where the set $S[X]$ contains those relations $R \in S$ whose attributes are contained in X. It is easy to see that $f_S(X)$ is submodular (we assume that N is sufficiently large such that $\log N - n - 1$ is positive). It follows that X has a unique minimum-value extension:

Proposition 8. *For every $X \subseteq A$ and $S \subseteq \sigma$, there is a unique $C_S(X) \supseteq X$ such that $f_S(C_S(X))$ is minimal and, among such sets, $|C_S(X)|$ is maximal.*

The following two lemmas explain why $X^* = C_S(X)$ is the right choice. The first lemma shows that we do not get many additional tuples if we take the join of only those relations whose attributes are in $C_S(X)$.

Lemma 9. *Let $S \subseteq \sigma$ be a set of relation names and X a set of attributes. Let $X^* = C_S(X)$ and let $S' = S[X^*]$. For every X^*-tuple t,*

$$\Pr(t \in \pi_{X^*}(\bowtie_{R\in S} R(D)) \mid t \in \bowtie_{R\in S'} R(D)) \ge 1/2.$$

Proof. By definition, $t \in \pi_{X^*}(\bowtie_{R\in S} R(D))$ if and only if there is a $t' \in \bowtie_{R\in S} R(D)$ with $\pi_{X^*}(t') = t$. Note that if $t \in \bowtie_{R\in S'} R(D)$ and $\pi_{X^*}(t') = t$, then t' satisfies all the relations in S', hence the probability that such a t' is in $\bowtie_{R\in S} R(D)$ (assuming $t \in \bowtie_{R\in S'} R(D)$) depends only on the relations in $S \setminus S'$. We claim that this conditional probability is equal to the probability that a certain query Q' with schema σ' has at least one solution. The query Q' is over the attributes $A \setminus X^*$. The schema σ' contains a relational symbol R' for each $R \in S \setminus S'$; the set of attributes of R' is $A_R \setminus X^*$. We define the probability of placing a tuple into R' as $p_{R'}(N) = p_R(N)$ for every $R' \in \sigma'$. It is not difficult to see that $\Pr(t \in \pi_{X^*}(\bowtie_{R\in S} R(D)) \mid t \in \bowtie_{R\in S'} R(D))$ is equal to the probability that Q' has at least one solution.

Observe that if A' is a subset of the attributes in Q', then the relations in $\sigma'[A']$ were obtained from the relations in $S[X^* \cup A'] \setminus S[X^*]$, which means that the weight of these relations is counted in $f_S(X^* \cup A')$ but not in

$f_S(X^*)$. If the weight of the relations in $\sigma'[A']$ is greater than $|A'|(\log N - n - 1)$, then $f_S(X^* \cup A') < f_S(X^*)$ would follow, contradicting the minimality of $X^* = C_S(X)$. This means that the maximum density $\overline{\delta}$ of Q' is at most $\log N - n - 1$. Thus by (4.2), the variance of the number of solutions is at most $(2^{n'} - 1) \cdot 2^{\overline{\delta} - \log N} < (2^{n'} - 1)2^{-(n+1)} \leq 1/2$ times the square of the expected number of solutions. Therefore, by Chebyshev's Inequality, the probability that there is no solution is at most $1/2$. □

The second lemma shows that extending the projection from X to X^* does not increase the number of tuples too much: a tuple in the projection to X do not have too many extensions to X^*.

Lemma 10. *Let $S \subseteq \sigma$ be a set of relation names and X be a set of attributes. Let $X^* = C_S(X)$ and let $S' = S[X^*]$. For an X-tuple t, let L_t be the set of those X^*-tuples $t' \in \bowtie_{R \in S'} R(D)$ that have $\pi_X(t') = t$.*

(1) For every X-tuple t, $\mathrm{E}(|L_t| \mid t \in \pi_X(\bowtie_{R \in S'} R(D))) \leq 2^{n(n+2)}$.

(2) For every X^*-tuple t', $\Pr(t' \in \bowtie_{R \in S'} R(D) \mid \pi_X(t') \in \pi_X(\bowtie_{R \in S'} R(D))) \leq 2^{n(n+2)} N^{-|X^* \setminus X|}$.

Proof. Let t', t'' be two X^*-tuples with $\pi_X(t') = \pi_X(t'') = t$. The conditional probability $\Pr(t' \in L_t \mid t'' \in L_t)$ depends on the set $X \subseteq Y \subseteq X^*$ of attributes where t' and t'' are the same. Let w be the total weight of the relations in S' and let w_Y be the total weight of the relations in $S'[Y]$. Observe that $w - w_Y \geq |X^* \setminus Y|(\log N - n - 1)$: otherwise $f_S(Y)$ would be strictly less than $f_S(X^*)$, contradicting the minimality of $X^* = C_S(X)$. If $t'' \in L_t$, then every relation $R \in S'[Y]$ is satisfied by t' as well. Thus $\Pr(t' \in L_t \mid t'' \in L_t) = 2^{-(w - w_Y)}$. The number of X^*-tuples that agree with t'' exactly on the attributes in Y is $(N-1)^{|X^* \setminus Y|}$, hence the expected number of such tuples in L_t, on the condition that $t'' \in L_t$, is

$$\begin{aligned}(N-1)^{|X^* \setminus Y|} \cdot 2^{-(w - w_Y)} &\leq N^{|X^* \setminus Y|} \cdot 2^{-|X^* \setminus Y|(\log N - n - 1)} \\ &\leq 2^{|X^* \setminus Y|(n+1)} \leq 2^{n(n+1)}.\end{aligned}$$

Summing for every Y, we get that $\mathrm{E}(|L_t| \mid t'' \in L_t) \leq 2^n 2^{n(n+1)} = 2^{n(n+2)}$.

Let $t_1, \ldots, t_k$ be an ordering of the X^*-tuples whose projection to X is t; clearly, $k = N^{|X^* \setminus X|}$. Let X_i be the event $t_i \in \bowtie_{R \in S'} R(D)$ and Y_i be the event $\bigwedge_{j=1}^{i-1} t_j \notin \bowtie_{R \in S'} R(D)$. The event $t \in \pi_X(\bowtie_{R \in S'} R(D))$ is the disjoint union of the events $X_1 Y_1, X_2 Y_2, \ldots, X_k Y_k$. Now we bound

$$\mathrm{E}(|L_t| \mid t \in \pi_X(\bowtie_{R \in S'} R(D))) = \frac{\sum_{i=1}^k \Pr(X_i Y_i)\, \mathrm{E}(|L_t| \mid X_i Y_i)}{\sum_{i=1}^k \Pr(X_i Y_i)}$$

by

$$\frac{\sum_{i=1}^k \Pr(X_i Y_i)\, \mathrm{E}(|L_t| \mid X_i)}{\sum_{i=1}^k \Pr(X_i Y_i)} \leq 2^{n(n+2)}.$$

using the FKG Inequality (see Lemma 27 in the Appendix [3]): $|L_t|$ is a monotone function of the random variables, the event X_i is the product of some random variables, and Y_i is an antimonotone function of the random variables.

To prove the second statement, observe first that if we fix a tuple t, then by symmetry, $\Pr(t' \in \bowtie_{R \in S'} R(D) \mid t \in \pi_X(\bowtie_{R \in S'} R(D)))$ has the same value for every X^*-tuple with $\pi_X(t') = t$. There are $N^{|X^* \setminus Y|}$ such tuples t' and the size of L_t is the sum of the indicator variables corresponding to these tuples. It follows that $\Pr(t' \in \bowtie_{R \in S'} R(D) \mid t \in \pi_X(\bowtie_{R \in S'} R(D))) = \mathrm{E}(|L_t| \mid t \in \pi_X(\bowtie_{R \in S'} R(D))) N^{-|X^* \setminus Y|} \leq 2^{n(n+2)} N^{-|X^* \setminus Y|}$. □

While the statements of the previous two lemmas give the main intuition underlying the construction, the proof of Theorem 7 still requires a substantial amount of work, which we defer to Appendix D.2 [3].

5. Size constraints

To estimate the size of joins, practical query optimisers use statistical information about the database instance such as the sizes of the relations, the sizes of some of their projections, or histograms. We consider the simplest such setting where the size of the relations is known, and we prove a (worst case) estimate on the size of $Q(D)$ subject to the constraint that the relations in D have the given sizes.

Let Q be a join query with schema σ. For every $R \in \sigma$, let A_R be the set of attributes of R, and $A = \bigcup_R A_R$. For every $R \in \sigma$, let N_R be a natural number, and let $L_Q(N_R : R \in \sigma)$ be the following linear program:

$$\begin{array}{lll} \text{minimise} & \sum_R x_R \cdot \log N_R & \\ \text{subject to} & \sum_{R : a \in A_R} x_R \geq 1 & \text{for all } a \in A, \\ & x_R \geq 0 & \text{for all } R \in \sigma. \end{array} \tag{5.1}$$

Note that the only difference with L_Q as defined in (3.1) is the objective function. This implies that every feasible solution of $L_Q(N_R : R \in \sigma)$ is also a fractional edge cover of Q.

Theorem 11. *Let Q be a join query with schema σ and let $N_R \in \mathbb{N}$ for all $R \in \sigma$. Let n be the number of attributes of Q, and let $(x_R : R \in \sigma)$ be an optimal solution of the linear program $L_Q(N_R : R \in \sigma)$.*

(1) For every σ-instance D with $|R(D)| = N_R$ for all R it holds that $|Q(D)| \leq \prod_R N_R^{x_R}$

(2) There is a σ-instance D such that $|R(D)| = N_R$ for all $R \in \sigma$ and $|Q(D)| \geq 2^{-n} \prod_R N_R^{x_R}$.

Proof sketch. Statement (1) is an immediate consequence of Lemma 2. Statement (2) is proved similarly to Lemma 3. The larger gap between upper and lower bounds is due to fact that we cannot choose the size N of the relations freely and thus in particular cannot guarantee that $N^{p_i/q}$ is integral.

This causes a rounding error. The full proof can be found in Appendix E [3]. □

In the next example we show that we cannot replace the lower bound of Theorem 11(2) by $2^{-(1-\varepsilon)n}\prod_R N_R^{x_R}$ for any $\varepsilon > 0$.

Example 12. Let $n \in \mathbb{N}$ be an integer, $0 < \varepsilon < 1$ a fixed constant, and $A = \{a_1, \ldots, a_n\}$ a set of attributes with domain $\mathbb{N}$. Let $r := \lfloor \varepsilon n / \log n \rfloor$. We assume that n is sufficiently large that $2^r > n$ holds. For every $B \in \binom{[n]}{r}$, let R_B be an r-ary relation with attributes B. Furthermore, for every $a \in A$, let R_a be a unary relation with the only attribute a. Let Q be the join of all these relations and let σ be the resulting schema.

For every $B \in \binom{[n]}{r}$, let $N_{R_B} = 2^r - 1$ and for every $a \in A$, let $N_{R_a} = 2$. Consider the linear program $L_Q(N_R : R \in \sigma)$. We obtain an optimal solution for this linear program by letting $x_{R_B} := n/\left(r\binom{n}{r}\right)$ and $x_{R_a} := 0$; optimality can easily be proved by considering the dual linear program.

We prove next that $\prod_R N_R^{x_R} = 2^n(1-o(1))$:

$$\prod_{R\in\sigma} N_R^{x_R} = \left((2^r-1)^{n/\left(r\binom{n}{r}\right)}\right)^{\binom{n}{r}} \geq (2^r-1)^{n/r} \geq$$
$$\geq (2^r(1-1/n))^{n/r} = 2^n(1-o(1)).$$

The second inequality follows from $2^r > n$ and the last equality follows from the fact if n tends to infinity, then $(1-1/n)^n$ goes to $1/e$ and r goes to infinity as well.

To complete the example, we prove that $|Q(D)| \leq 2^{\varepsilon n}$ for every instance D respecting the constraints N_R. Let D be a σ-instance with $|R(D)| = N_R$ for every $R \in \sigma$. From $N_{R_a} = 2$ it follows that in $Q(D)$ each attribute has at most two values, hence we can assume that $Q(D) \subseteq \{0,1\}^n$. Thus each tuple in $t \in Q(D)$ can be viewed as a subset $A_t = \{a \in A : t(a) = 1\}$ of A. For every $B \in \binom{[n]}{r}$, it holds $\pi_B(Q(D)) \leq N_{R_B} = 2^r - 1$, hence the Vapnik-Chervonenkis dimension of $Q(D)$ is less than r. Thus by Sauer's Lemma, we have

$$|Q(D)| \leq n^r \leq n^{\varepsilon n/\log n} = 2^{\varepsilon n},$$

as claimed.

Example 12 seems to indicate that maybe the approach to estimating the size of joins through fractional edge covers is no longer appropriate in the setting where the size of the relations is fixed. However, we shall see that, in some sense, there is no better approach. In Theorem 15, we shall prove that there is no polynomial time algorithm that, given a query Q and relation sizes N_R, for $R \in \sigma$, approximates the worst case size of the query answer to a factor better than $2^{n^{1-\varepsilon}}$. The proof is based on a hardness result of Håstad for maximum independent set (Theorem 13) and on a connection between independent sets and the number of solutions (Lemma 14). We denote by $\alpha(G)$ the size of the maximum independent set in the graph G.

Theorem 13 ([11]). *If for some $\varepsilon_0 > 0$ there is a polynomial-time algorithm that, given an n-vertex graph G, can distinguish between the cases $\alpha(G) \leq n^{\varepsilon_0}$ and $\alpha(G) \geq n^{1-\varepsilon_0}$, then* NP = ZPP.

Lemma 14. *Let Q be a join query with schema σ and let $N_R := 2$ for all $R \in \sigma$. Let G be the primal graph of Q and let $\alpha(G)$ be the size of the maximum independent set in G. The maximum of $|Q(D)|$, taken over database instances satisfying $|R(D)| = N_R$ for every $R \in \sigma$, is exactly $2^{\alpha(G)}$.*

Proof. Let A_R be the attributes of $R \in \sigma$, and let A be the attributes of Q. First we give a database D with $|Q(D)| \geq 2^{\alpha(G)}$. Let $I \subseteq A$ be an independent set of size $\alpha(G)$. Since I is independent, $|A_R \cap I|$ is either 0 or 1 for every $R \in \sigma$. If $|A_R \cap I| = 0$, then we define $R(D)$ to contain a tuple that is 0 on every attribute. If $A_R \cap I = \{a\}$, then we define $R(D)$ to contain a tuple that is 0 on every attribute and a tuple that is 1 on a and 0 on every attribute in $A_R \setminus \{a\}$. We claim that

$$Q(D) = \{t \in \mathrm{tup}(A) : t(a) \in \{0,1\} \text{ for all } a \in I, t(a) = 0 \text{ for all } a \in A \setminus I\}.$$

Clearly, the value of an attribute in I is either 0 or 1, and every attribute in $A \setminus I$ is forced to 0. Furthermore, any combination of 0 and 1 on the attributes of I is allowed as long as all the other attributes are 0. Thus $|Q(D)| = 2^{\alpha(G)}$. Note that a relation R with $|A_R \cap I| = 0$ contains only one tuple in the definition above. To satisfy the requirement $|R(D)| = N_R = 2$, we can add an arbitrary tuple to each such relation R; this cannot decrease $|Q(D)|$.

Next we show that if $|R(D)| = 2$ for every relation $R \in \sigma$, then $|Q(D)| \leq 2^{\alpha(G)}$. Since $|R(D)| = 2$ for every relation, every attribute in A can have at most two values in $Q(D)$; without loss of generality it can be assumed that $Q(D) \subseteq \{0,1\}^{|A|}$. Furthermore, it can be assumed (by a mapping of the domain of the attributes) that the all-0 tuple is in $Q(D)$ (if $Q(D)$ is empty, then there is nothing to prove).

Let S be the set of those attributes that have two values in $Q(D)$, i.e.,

$$S = \{a \in A : |\pi_{\{a\}}(Q(D))| = 2\}.$$

For every $a \in S$, let S_a be the set of those attributes that are the same as a in every tuple of $Q(D)$, i.e.,

$$S_a = \{b \in S : t(a) = t(b) \text{ for every } t \in Q(D)\}.$$

We define a sequence $a_1, a_2, \ldots$ of attributes by letting a_i be an arbitrary attribute in $S \setminus \bigcup_{j<i} S_{a_i}$. Let a_t be the last element in this sequence, which means that $\bigcup_{i=1}^{t} S_{a_i} = S$. We claim that $a_1, \ldots, a_t$ are independent in G, implying $t \leq \alpha(G)$. Assume that a_i and a_j $(i < j)$ are adjacent in G; this means that there is an $R \in \sigma$ with $a_i, a_j \in A_R$. By assumption, the all-0 tuple is in $R(D)$. As $a_i, a_j \in S$, there

has to be a $t_1 \in R(D)$ with $t_1(a_i) = 1$ and a $t_2 \in R(D)$ with $t_2(a_j) = 1$. Since $|R(D)| = 2$ and the all-0 tuple is in $R(D)$, we have $t_1 = t_2$. But this means that $t(a_i) = t(a_j)$ for both tuples in $R(D)$, implying $a_j \in S_{a_i}$. However, this contradicts the way the sequence was defined.

Now it is easy to see that $|Q(D)| \leq 2^t \leq 2^{\alpha(G)}$: by setting the value of $a_1, \ldots, a_t$, the value of every attribute in S is uniquely determined and the attributes in $A \setminus S$ are the same in every tuple of $Q(D)$. □

Theorem 15. *For a given query Q with schema σ and a given set of size constraints $(N_R : R \in \sigma)$, denote by M the maximum of $|Q(D)|$ over databases satisfying $|R(D)| = N_R$ for every $R \in \sigma$. If for some $\varepsilon > 0$, there is a polynomial-time algorithm that, given a query Q with n attributes and size constraints N_R, computes two values M_L and M_U with $M_L \leq M \leq M_U$ and $M_U \leq M_L 2^{n^{1-\varepsilon}}$, then* NP = ZPP.

Proof. We show that if such M_L and M_U could be determined in polynomial time, then we would be able to distinguish between the two cases of Theorem 13. Given an n-vertex graph $G = (V, E)$, we construct a query Q with attributes V and schema $\sigma = E$. For each edge $uv \in E$, there is a relation R_{uv} with attributes $\{u, v\}$. We set $N_R = 2$ for every relation $R \in \sigma$. Observe that the primal graph of Q is G. Thus by Lemma 14, $M = 2^{\alpha(G)}$.

Set $\varepsilon_0 := \varepsilon/2$. In case (1) of Theorem 13, $\alpha(G) \leq n^{\varepsilon_0}$, hence $M_L \leq M \leq 2^{n^{\varepsilon_0}}$ and

$$M_U \leq M_L 2^{n^{1-\varepsilon}} \leq 2^{n^{\varepsilon_0} + n^{1-\varepsilon}} < 2^{n^{1-\varepsilon_0}}$$

(if n is sufficiently large). On the other hand, in case (2) we have $\alpha(G) \geq M \geq n^{1-\varepsilon_0}$, which implies $M_U \geq 2^{n^{1-\varepsilon_0}}$. Thus we can distinguish between the two cases by comparing M_U with $2^{n^{1-\varepsilon_0}}$. □

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley, second edition, 1992.

[3] A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. Available from the author's webpages, 2008.

[4] S. Chaudhuri. An overview of query optimization in relational systems. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 34–43, 1998.

[5] F. R. K. Chung, R. L. Graham, P. Frankl, and J. B. Shearer. Some intersection theorems for ordered sets and graphs. *J. Combin. Theory Ser. A*, 43(1):23–37, 1986.

[6] J. Flum, M. Frick, and M. Grohe. Query evaluation via tree-decompositions. *Journal of the ACM*, 49(6):716–752, 2002.

[7] E. Friedgut and J. Kahn. On the number of copies of a hypergraph in another. *Israel Journal of Mathematics*, 105:251–256, 1998.

[8] H. Garcia-Molina, J. Widom, and J. Ullman. *Database System Implementation*. Prentice-Hall, 1999.

[9] G. Graefe. Query evaluation techniques for large databases. *ACM Computing Surveys*, 25, 1993.

[10] M. Grohe and D. Marx. Constraint solving via fractional edge covers. In *Proceedings of the of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 289–298, 2006.

[11] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Math.*, 182(1):105–142, 1999.

[12] J. H. Kim and V. H. Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.

[13] J. Rhadakrishnan. Entropy and counting. Available at http://www.tcs.tifr.res.in/∼jaikumar/mypage.html.

[14] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.