DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-2003-05

QUERYING PRICED INFORMATION IN DATABASES: THE CONJUNTIVE CASE

Eduardo Laber, Renato Carmo, and Yoshiharu Kohayakawa

Julho de 2003

QUERYING PRICED INFORMATION IN DATABASES: THE CONJUNTIVE CASE

EDUARDO LABER, RENATO CARMO, AND YOSHIHARU KOHAYAKAWA

ABSTRACT. Query optimization that involves *expensive predicates* have received considerable attention in the database community. Typically, the output to a database query is a set of tuples that satisfy certain conditions, and, with expensive predicates, these conditions may be computationally costly to verify. In the simplest case, when the query looks for the set of tuples that simultaneously satisfy k expensive predicates, the problem reduces to ordering the evaluation of the predicates so as to minimize the time to output the set of tuples comprising the answer to the query.

Here, we give a simple and fast deterministic k-approximation algorithm for this problem, and prove that k is the best possible approximation ratio for a deterministic algorithm, even if exponential time algorithms are allowed. We also propose a randomized, polynomial time algorithm with expected approximation ratio $1+\sqrt{2}/2 \approx 1.707$ for $k=2$, and prove that $3/2$ is the best possible expected approximation ratio for randomized algorithms. The approximation ratio achieved by our algorithm is not concentrated around its mean; however, we show that this limitation is *inherent to the problem*, rather than a weakness of our approach: we show that for any $0 \leq \varepsilon \leq 1$, no randomized algorithm achieves approximation ratio smaller than $1+\varepsilon$ with probability larger than $(1+\varepsilon)/2$.

1. INTRODUCTION

The main goal of *query optimization in databases* is to determine how a query over a database should be processed in order to minimize the user response time. A typical query extracts the tuples from a database relation that satisfy a set of conditions, or *predicates*, in database terminology. For example, consider the set of tuples $D = \{(a_1,b_1),(a_1,b_2),(a_1,b_3),(a_2,b_1)\}$ (see Figure 1(a)) and a conjunctive query that seeks to extract the subset of tuples (a_i,b_j) for which a_i satisfies predicate P_1 and b_j satisfies predicate P_2. Clearly, these predicates can be viewed together as a 0/1-valued function δ defined on the set of tuple elements $\{a_1,a_2,b_1,b_2,b_3\}$, with the convention that, $\delta(a_i)=1$ if and only if $P_1(a_j)$ holds and $\delta(b_j)=1$ if and only if $P_2(b_j)$ holds. The answer to the query is the set of pairs (a_i,b_j) with $\bar{\delta}(a_i,b_j)=\delta(a_i)\delta(b_j)=1$. The query optimization problem that we consider is that of determining a strategy for evaluating $\bar{\delta}$ so as to compute this set of tuples by evaluating as few values of the function δ as possible (or, more generally, with the total cost for evaluating the function $\bar{\delta}$ minimal).

It is usually the case that the cost (measured as the computational time) needed to evaluate the predicates of a query can be assumed to be bounded by a constant so that the query can be answered by just scanning through all the tuples in D while evaluating the corresponding predicates.

In the case of computationally expensive predicates, however, e.g., when the database holds complex data as images and tables, this constant may happen to be so large as to render this

Date: July 3, 2003.

laber@inf.puc-rio.br: Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro, Brazil.

renato@ime.usp.br: Departamento de Informática da Universidade Federal do Paraná and Instituto de Matemática e Estatística da Universidade de São Paulo, Brazil; Partially supported by PICDT/CAPES. .

yoshi@ime.usp.br: Instituto de Matemática e Estatística da Universidade de São Paulo, Brazil; Partially supported by MCT/CNPq through ProNEx Programme (Proc. CNPq 664107/1997–4) and by CNPq (Proc. 300334/93–1 and 468516/2000–0).

strategy impractical. In such cases, the different costs involved in evaluating each predicate must also be taken into account in order to keep user response time within reasonable bounds.

Among several proposals to model and solve this problem, (see, for example, [3, 5, 8]), we focus on the improvement of the approach proposed in [11] where, differing from the others, the query evaluation problem is reduced to an optimization problem in a hypergraph (see Figure 1).

1.1. **Problem statement.** A *hypergraph* is a pair $G = (V(G), E(G))$ where $V(G)$, the set of *vertices* of G, is a finite set and each *edge* $e \in E(G)$ is a non-empty subset of $V(G)$.

The size of the largest edge in G is called the *rank* of G and is denoted $r(G)$. A hypergraph G is said to be *uniform* if each edge has size $r(G)$, and is said to be k-partite if there is a partition $\{V_1, \ldots, V_k\}$ of $V(G)$ such that no edge contains two vertices in the same partition.

Given a hypergraph G and a function $\delta : V(G) \to \{0, 1\}$ we define an *evaluation* of (G, δ) as a set $\mathsf{E} \subseteq V(G)$ such that, knowing the value of $\delta(v)$ for each $v \in \mathsf{E}$, one may determine the value of

$$\bar{\delta}(e) = \prod_{v \in e} \delta(v),$$

for each $e \in E(G)$.

Given a hypergraph G and a function $\gamma : V(G) \to \mathbb{R}$ we define the *cost* of a set $X \subseteq V(G)$ by

$$\gamma(X) = \sum_{v \in X} \gamma(v).$$

An instance to the Dynamic Multipartite Ordering problem (DMO) is an $r(G)$-partite, uniform hypergraph G, together with functions δ and γ as above. The objective in DMO is to determine an evaluation of minimum cost for (G, δ, γ). Observe that the function δ is 'unknown to us at first'. More precisely, the value of $\delta(v)$ becomes known only when $\delta(v)$ is actually evaluated, and this evaluation costs $\gamma(v)$. The restriction of DMO to instances in which $r(G) = 2$ deserves special attention and will be referred to as the Dynamic Bipartite Ordering problem (DBO).

Before we proceed, let us observe that DMO models our database problem as follows: the sets in the partition $\{V_1, \ldots, V_k\}$ of $V(G)$ correspond to the k different attributes of the relation that is being queried and each vertex of G corresponds to a distinct attribute value (tuple element). The edges correspond to tuples in the relation, $\gamma(v)$ is the time required to evaluate δ on v and $\delta(v)$ corresponds to the result of a predicate evaluated at the corresponding tuple element.

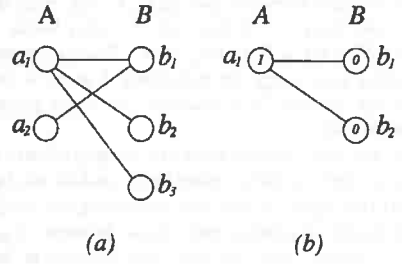

FIGURE 1. The set of tuples $\{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_2, b_1)\}$ and an instance for DBO

Figure 1(b) shows an instance of DBO. The value of $\delta(v)$ is indicated inside each vertex v. Suppose that $\gamma(a_1) = 3$ and $\gamma(b_1) = \gamma(b_2) = 2$. In this case, any strategy that starts evaluating $\delta(a_1)$ will return the evaluation $\{a_1, b_1, b_2\}$ of cost 7. However, the evaluation of minimum cost for this instance is $\{b_1, b_2\}$ of cost 4. This example underlines the point that the crux of the problem is

to devise a strategy for dynamically choosing, based on the function γ and the values of δ already revealed, the next vertex v whose δ-value should be evaluated, so as to minimize the overall cost.

Let $\mathcal{A}$ be an algorithm for DMO and let $\mathcal{I} = (G, \delta, \gamma)$ be an instance to DMO. We will denote the evaluation computed by $\mathcal{A}$ on input $\mathcal{I}$ by $\mathcal{A}(\mathcal{I})$. Establishing a measure for the performance of a given algorithm $\mathcal{A}$ for DMO is somewhat delicate: for example, a worst case analysis of $\gamma(\mathcal{A}(\mathcal{I}))$ is not suitable since any correct algorithm should output an evaluation comprising all vertices in $V(G)$ when $\delta(v) = 1$ for every $v \in V(G)$ (if G has no isolated vertices). This remark motivates the following definition.

Given an instance $\mathcal{I} = (G, \delta, \gamma)$, let E be an evaluation for $\mathcal{I}$ and let $\gamma^*(\mathcal{I})$ denote the cost of a minimum cost evaluation for $\mathcal{I}$. We define the *deficiency of evaluation* E *(with respect to* $\mathcal{I}$*)* as the ratio $d(\mathsf{E}, \mathcal{I}) = \gamma(\mathsf{E})/\gamma^*(\mathcal{I})$. Given an algorithm $\mathcal{A}$ for DMO, we define the *deficiency of* $\mathcal{A}$ as the worst case deficiency of the evaluation $\mathcal{A}(\mathcal{I})$, where $\mathcal{I}$ ranges over all possible instances of the problem, that is,

$$d(\mathcal{A}) = \max_{\mathcal{I}} d(\mathcal{A}(\mathcal{I}), \mathcal{I}).$$

If $\mathcal{A}$ is a randomized algorithm, $d(\mathcal{A}(\mathcal{I}), \mathcal{I})$ is a random variable, and the *expected deficiency* of $\mathcal{A}$ is then defined as the maximum over all instances of the mean of this random variable, that is,

$$d(\mathcal{A}) = \max_{\mathcal{I}} \mathbb{E}[d(\mathcal{A}(\mathcal{I}), \mathcal{I})] = \max_{\mathcal{I}} \frac{\mathbb{E}[\gamma(\mathcal{A}(\mathcal{I}))]}{\gamma^*(\mathcal{I})}.$$

Clearly, we wish to devise fast algorithms whose (expected) deficiency is as close to 1 as possible. In this paper, we will be concerned with designing algorithms for DMO, analyzing them and establishing bounds for their deficiency.

1.2. **Statement of results.** In Section 2, we briefly argue that $d(\mathcal{A}) \geq r(G)$ for any deterministic algorithm $\mathcal{A}$ and $d(\mathcal{B}) \geq (r(G)+1)/2$ for any randomized algorithm $\mathcal{B}$. It is worth noting that such bounds apply even allowing exponential time algorithms. We then present an optimal deterministic algorithm for DMO with time complexity $O(|E(G)| \log r(G))$, developed with the primal-dual approach. As an aside, we remark that this algorithm does not need to know the whole hypergraph in advance in order to solve the problem, since it scans the edges (tuples), evaluating each of them as soon as they become available. This is a most convenient feature for the database application that motivates this work.

In Section 3, for any given $0 \leq \varepsilon \leq 1 - \sqrt{2}/2$, we present a *randomized*, polynomial time algorithm $\mathcal{R}_\varepsilon$ for DBO whose expected deficiency is at most $2 - \varepsilon$. The best expected deficiency is achieved when $\varepsilon = 1 - \sqrt{2}/2$. However, the smaller the ε, the smaller is the probability that a particular execution of $\mathcal{R}_\varepsilon$ will return a truly poor result: we show that the probability that $d(\mathcal{R}_\varepsilon(\mathcal{I}), \mathcal{I}) \leq 1 + 1/(1-\varepsilon)$ holds is 1.

The deficiency of R_ε is not assured to be highly concentrated around the expectation. In Section 3.2, we show that this limitation is *inherent to the problem*, rather than a weakness of our approach: for any $0 \leq \varepsilon \leq 1$, no randomized algorithm can have deficiency smaller than $1 + \varepsilon$ with probability larger than $(1+\varepsilon)/2$. The proof of this fact makes a use of *Yao's Minimax Principle* [13].

For both, theoretical and applied reasons, *property testing* and sub-linear time algorithms in general have attracted some attention recently (see, e.g., [12]). In Section 4 we consider a problem related to DMO in which we wish to estimate *in sub-linear time* the *number of tuples* that a given query will return.

1.3. **Related work.** The problem of optimizing queries with expensive predicates has gained some attention in the database community [1, 3, 5, 8, 10, 11]. However, most of the proposed approaches [3, 5, 8] do not take into account the fact that an attribute value may appear in different tuples in order to decide how to execute the query. In this sense, they do not view the input relation as a general hypergraph, but as a set of tuples without any relation between them (*i.e.*, as a matching

hypergraph). The Predicate Migration algorithm proposed in [8], the main reference in this subject, may be viewed as an optimal algorithm for a variant of DMO, in which the input graph is always a matching, the probability p_i of a vertex from V_i (ith attribute) evaluating to true ($\delta(v) = 1$) is known, and the objective is to minimize the expected cost of the computed evaluation (we omit the details).

The idea of processing the hypergraph induced by the input relation appears first in [11], where a greedy algorithm is proposed with no theoretical analysis. The distributed case of DBO, in which there are two available processors, say P_A and P_B, responsible for evaluating δ on the nodes of the vertex classes A and B of the input bipartite graphs is studied in [10]. The following results are presented in [10]: a lower bound of $3/2$ on the deficiency of any randomized algorithm, a randomized polynomial time algorithm of expected deficiency $8/3$, and a linear time algorithm of deficiency 2 for the particular case of DBO with constant γ. We do not go into details, but the approach here allows one to improve some of these results.

In this extended abstract, we restrict our attention to *conjunctive* queries. However, much more general queries may happen. For example, $\bar{\delta}\colon E(G) \to \{0,1\}$ could be any formula in the first order propositional calculus involving the predicates represented by δ. In [4], Charikar et al. considered the problem of querying priced information. In particular, they considered the problem of evaluating a query that can be represented by an "AND/OR tree" over a set of variables, where the cost of probing each variable may be different. The framework for querying priced information proposed in that paper can be viewed as a restricted version of the problem described in this paragraph, where the input graph has one single edge.

1.4. **Preliminaries.** Let $\mathcal{I} = (G, \delta, \gamma)$ be an instance to DMO. The *neighbourhood* of $v \in V(G)$ is the set $\Gamma(v) = \{u \in V(G) \setminus \{v\}\colon \{u,v\} \subseteq e \text{ for some } e \in E(G)\}$. An *isolated vertex* in G is a vertex contained in no edge of G. For any $X \subseteq V(G)$, we let $V_0(X) = \{v \in X\colon \delta(v) = 0\}$, $V_1(X) = \{v \in X\colon \delta(v) = 1\}$, and $\Gamma_1(X) = \Gamma(V_1(X))$. Note that any evaluation for $\mathcal{I}$ must contain $\Gamma_1(X)$.

A *cover* for G is a set $C \subseteq V(G)$ such that every edge of G has at least one vertex in C. A *minimum cover* for (G,γ) is a cover C for G such that $\gamma(C)$ is minimum. Observe that any evaluation for $\mathcal{I}$ must contain a cover for G as a subset, otherwise the $\bar{\delta}$-value of at least one edge cannot be determined.

For a cover C for G, we let

$$\mathsf{E}(C) = C \cup \Gamma_1(C)$$

be the *C-evaluation for $\mathcal{I} = (G,\delta,\gamma)$*. It is not difficult to see that a C-evaluation for $\mathcal{I}$ is indeed an evaluation for $\mathcal{I}$. Moreover, since any evaluation for (G,δ) must contain some cover for G and $\Gamma_1(V(G))$, it is not difficult to conclude that that the deficiency of a C-evaluation for an instance to DBO has deficiency at most 2, whenever C is an minimum cover for (G,γ). This obsevation appears in [10] for the distributed version of DBO.

An optimal cover C for (G,γ), and as a consequence $\mathsf{E}(C)$, may be computed in polynomial time if G is a bipartite graph [7, 9]. Let us use COVER to denote the algorithm that outputs $\mathsf{E}(C)$ for some minimum cover C. Since 2 is a lower bound for the deficiency of any deterministic algorithm for DBO (See Section 2), we have that COVER is a polynomial time, optimal deterministic algorithm for DBO. This algorithm plays an important role on the design of the randomized algorithm proposed in Section 3.

2. The general case and an optimal polynomial deterministic algorithm

2.1. **Lower bounds.** We start with some lower bounds for the deficiency of algorithms for DMO. It is worth noting that *these bounds apply even to algorithms of exponential time/space complexity.*

A *matching* in a hypergraph G is a set $M \subseteq E(G)$ with no two edges in M sharing a common vertex. A hypergraph G is said to be a *matching* if $E(G)$ is a matching. It will be useful to single out the following class of instances. We will call an instance (G, δ, γ) to DMO an *extremal instance* if (i) G is a matching, (ii) each edge $e \in E(G)$ has exactly one vertex $v \in e$ such that $\delta(v) = 0$, and (iii) $\gamma(v) = 1$ for all $v \in V(G)$.

Theorem 1. *Let G be a hypergraph of rank $r(G)$ and let $\mathcal{A}$ be a deterministic algorithm for* DMO. *There is an extremal instance $\mathcal{I} = (G, \delta, \gamma)$ to* DMO *for which the deficiency of the output of $\mathcal{A}$ when run $\mathcal{I}$ is at least $r(G)$.*

Proof. Let $\mathcal{I} = (G, \delta, \gamma)$ be the extremal instance in which, for each edge $e \in E(G_{\mathcal{A}})$, the vertex $v \in e$ having $\delta(v) = 0$ is the last vertex evaluated by $\mathcal{A}$ in this edge. It is not difficult to see that $\gamma(\mathcal{A}(\mathcal{I}_{\mathcal{A}})) = |V(G_{\mathcal{A}})|$, while $V_0(V(G_{\mathcal{A}}))$ is an evaluation for $\mathcal{I}_{\mathcal{A}}$ with cost $|E(G_{\mathcal{A}})|$. Therefore $d(\mathcal{A}) = \max_{\mathcal{I}} d(\mathcal{A}(\mathcal{I}), \mathcal{I}) \geq d(\mathcal{A}(\mathcal{I}_{\mathcal{A}}), \mathcal{I}_{\mathcal{A}}) = \gamma(\mathcal{A}(\mathcal{I}_{\mathcal{A}}))/\gamma^*(\mathcal{I}_{\mathcal{A}}) \geq |V(G_{\mathcal{A}})|/|E(G_{\mathcal{A}})| = r(G_{\mathcal{A}})$. □

We now state without proof a lower bound on the expected deficiency of any randomized algorithm for DMO. We only mention that the proof of the result below is based on Yao's minimax principle (a more interesting application of this principle occurs in Section 3.2).

Theorem 2. *Let G be a hypergraph of rank $r(G)$ and let $\mathcal{B}$ be a randomized algorithm for* DMO. *There is an extremal instance $\mathcal{I} = (G, \delta, \gamma)$ to* DMO *for which the expected deficiency of the output of $\mathcal{B}$ when run $\mathcal{I}$ is at least $(r(G) + 1)/2$.*

2.2. An optimal polynomial deterministic algorithm for DMO. We will now introduce a polynomial time, deterministic algorithm for DMO that has deficiency at most r on instances with G of rank $r(G)$ at most r. In view of Theorem 1, this algorithm has the best possible deficiency for a deterministic algorithm.

Let (G, δ, γ) be a fixed instance to DMO, and let $E_i = \{e \in E(G) \colon \bar{\delta}(e) = i\}$ and

$$W_i = \bigcup_{e \in E_i} e \quad (i \in \{0, 1\})$$

We let $G[E_i]$ the hypergraph with vertex set W_i and edge set E_i. Let γ_0^* be the cost of a minimum cover for $(G[E_0], \gamma)$, among all covers for $(G[E_0], \gamma)$ that contain vertices in $V_0 = V_0(V(G)) = \{v \in V(G) \colon \delta(v) = 0\}$ only. Then $\gamma^*(G, \delta, \gamma) = \gamma_0^* + \gamma(W_1)$.

Let us look at γ_0^* as the optimal solution of the following Integer Programming problem, which we will denote by $L_I(G, \delta, \gamma)$:

$$\min \left\{ \sum_{v \in V_0} \gamma(v) x_v \colon \sum_{v \in e \cap V_0} x_v \geq 1 \text{ for all } e \in E_0 \text{ and } x_v \in \{0, 1\} \text{ for all } v \in V_0 \right\}.$$

Let us denote by $L(G, \delta, \gamma)$ the linear relaxation of $L_I(G, \delta, \gamma)$, where the restrictions $x_v \in \{0, 1\}$ are replaced by $x_v \geq 0$ for all $v \in V_0$. The dual $L(G, \delta, \gamma)^D$ of $L(G, \delta, \gamma)$ is

$$\max \left\{ \sum_{e \in E_0} y_e \colon \sum_{e \colon v \in e} y_e \leq \gamma(v) \text{ for all } v \in V_0 \text{ and } y_e \geq 0 \text{ for all } e \in E_0 \right\}.$$

Theorem 3. *Let (G, δ, γ) be an instance to* DMO *and let $\bar{y} : E_0 \to \mathbb{R}$ be a feasible solution of $L(G, \delta, \gamma)^D$. Any evaluation* E *of (G, δ) satisfying*

$$\gamma(v) \leq \sum_{e \colon v \in e} \bar{y}_e \quad \textit{for all } v \in \mathsf{E} - W_1, \tag{1}$$

has deficiency at most $r(G)$.

Proof. Duality between $L(G,\delta,\gamma)$ and $L(G,\delta,\gamma)^D$ gives $\sum_{e\in E_0}\bar{y}_e \le \gamma_0^*$. Any evaluation for (G,δ) must include W_1, so that $\gamma(\mathsf{E}) = \gamma(\mathsf{E}-W_1)+\gamma(\mathsf{E}\cap W_1) = \gamma(\mathsf{E}-W_1)+\gamma(W_1)$, and

$$\gamma(\mathsf{E}-W_1) = \sum_{v\in\mathsf{E}-W_1}\gamma(v) \le \sum_{v\in\mathsf{E}-W_1}\sum_{e:\,v\in e}\bar{y}_e \le r(G)\sum_{e\in E_0}\bar{y}_e,$$

where the last inequality follows from the fact that each edge contributes with at most $r(G)$ terms to the sum. Therefore

$$\gamma^*(G,\delta,\gamma) \le \gamma(\mathsf{E}) \le r(G)\sum_{e\in E_0}\bar{y}_e + \gamma(W_1) \le r(G)(\gamma_0^* + \gamma(W_1)) = r(G)\gamma^*(G,\delta,\gamma),$$

and hence $d(\mathsf{E},(G,\delta,\gamma)) = \gamma(\mathsf{E})/\gamma^*(G,\delta,\gamma) \le r(G)$. □

The algorithm presented below uses a primal-dual approach to construct a vector $y\colon E\to\mathbb{R}$ and an evaluation E such that both the restriction of y to E_0 and E satisfy the the conditions of Theorem 3. This algorithm resembles the one presented by Bar-Yehuda [2] for the minimum vertex cover problem. The main difference is that $L(G,\delta,\gamma)^D$ it is not known beforehand, because the set V_0 is not known.

Our algorithm maintains for each $e\in E(G)$ a value y_e and for every $v\in V(G)$ the value $r_v = \sum_{e:\,v\in e} y_e$. At each step, the algorithm selects an unevaluated edge e and increases the corresponding dual variable y_e until it "saturates" the next non-evaluated vertex v (r_v becomes equal to $\gamma(v)$). The values of r_u ($u\in e$) are updated and the vertex v is then evaluated. If $\delta(v)=0$, then the edge e is added to E_0 along with all other edges that contain v, and the algorithm proceeds to the next edge. Otherwise the algorithm increases the value of the dual variable y_e until it "saturates" another unevaluated vertex in e and executes the same steps until either e is put into E_0 or there are no more unevaluated vertices in e, in which case e is put in E_1.

Algorithm $\mathcal{PD}(G,\delta,\gamma)$

(1) Start with E_0, E_1 and E as empty sets, $r_v = 0$ for all $v\in V(G)$ and $y_e = 0$ for all $e\in E(G)$
(2) While $E(G)\neq E_1\cup E_0$
  (a) Select an edge $e\in E(G)-(E_1\cup E_0)$
  (b) While $e\not\subseteq\mathsf{E}$ and $e\notin E_0$
    (i) select a vertex $v\in e-\mathsf{E}$ such that $\gamma(v)-r_v$ is minimum
    (ii) add $\gamma(v)-r_v$ to y_e and to each r_u such that $u\in e$
    (iii) insert v in E
    (iv) If $\delta(v)=0$, insert in E_0 every edge $e'\in E(G)$ such that $v\in e'$
  (c) If $e\notin E_0$, insert e in E_1
(3) Return E

Lemma 4. *Let (G,δ,γ) be an instance to* DMO. *At the end of the execution of $\mathcal{PD}(G,\delta,\gamma)$, the restriction of y to E_0 is a feasible solution to $L(G,\delta,\gamma)^D$ and E is an evaluation of (G,δ) satisfying* (1). *Algorithm $\mathcal{PD}(G,\delta,\gamma)$ runs in time $O(|E(G)|\log r(G))$.*

Proof. The restriction of y to E_0 is a feasible solution to $L(G[E_0],\delta,\gamma)^D$ at the beginning of the algorithm, at which point we also have that $r_v\le\gamma(v)$ for every $v\in V(G)$.

Consider any $v\in V(G)$. The value of each r_v is increased along the execution of the algorithm every time an edge e containing v is evaluated. This value only reaches $\gamma(v)$ at the point in the computation where an edge e containing v is to be evaluated. If $\delta(v)=0$ then e is included in E_0 along with any other edge containing v. From then on, neither the dual variables relative to these edges nor r_v is modified.

The asserted time complexity may be achieved by using a suitable priority queue for each edge in $E(G)$. □

Corollary 5. *Algorithm $\mathcal{PD}$ is a polynomial time, optimal deterministic algorithm for* DMO. □

3. The bipartite case and a polynomial randomized algorithm

Let $0 \leq \varepsilon \leq 1 - \sqrt{2}/2$. In this section, we present $\mathcal{R}_\varepsilon$, a polynomial time randomized algorithm for DBO with the following properties: for every instance $\mathcal{I}$,

$$\mathbb{E}[d(\mathcal{R}_\varepsilon(\mathcal{I}))] \leq 2 - \varepsilon \tag{2}$$

and

$$\mathbb{P}\left(d(\mathcal{R}_\varepsilon(\mathcal{I})) \leq 1 + \frac{1}{1-\varepsilon}\right) = 1. \tag{3}$$

Thus, $\mathcal{R}_\varepsilon$ provides a trade-off between expected deficiency and worst case deficiency. At one extreme, when $\varepsilon = 1 - \sqrt{2}/2$, we have expected deficiency 1.707 and worst case deficiency up to 2.41 for some particular execution. At the other extreme ($\varepsilon = 0$), we have a deterministic algorithm with deficiency 2.

The key idea in $\mathcal{R}_\varepsilon$'s design is trying to understand under which conditions the COVER algorithm explained in Section 1.4 does not perform well. More exactly, given an instance $\mathcal{I}$ to DBO, a minimum cover C for (G, δ), and $\varepsilon > 0$, we turn our attention to the instances $\mathcal{I}$ having $d(\mathsf{E}(C), \mathcal{I}) \geq 2 - \varepsilon$.

One family of such instances can be constructed as follows. Consider an instance (G, δ, γ) to DBO where G is a matching of n edges, the vertex classes of G are A and B, and $\delta(v) = 1$ for every $v \in A$ and $\delta(v) = 0$ for every $v \in B$. Clearly, B is an optimum evaluation for $\mathcal{I}$, with cost n. On the other hand, note that the deficiency of the evaluation $\mathsf{E}(C)$ which is output by COVER depends on which of the 2^n distinct minimum covers of G is picked. In the particular case in which $C = A$, we have $d(\mathsf{E}(C), \mathcal{I}) = 2n/n = 2$.

This example suggests the following idea. If C is a minimum cover for (G, γ) and nonetheless $\mathsf{E}(C)$ is not a "good evaluation" for $\mathcal{I} = (G, \delta, \gamma)$, then there must be another cover C' of G whose intersection with C is "small" and still C' is not "far from being" a minimum cover for G. The following lemmas formalize this idea.

Lemma 6. *If $\mathcal{I} = (G, \delta, \gamma)$ is an instance to* DBO, *$T \subseteq \Gamma_1(V(G))$, and C_T is a minimum cover of $G - T$, then $\gamma(C_T) + \gamma(T) \leq \gamma^*(\mathcal{I})$.*

Proof. Let E^* be a minimum cost evaluation for $\mathcal{I}$. Since $T \subseteq \Gamma_1(V(G))$ it follows that $T \subset \mathsf{E}^*$. Furthermore, $\mathsf{E}^* - T$ is a cover for $G - T$, for otherwise $\bar{\delta}(uv)$ cannot be determined for some $uv \in G - T$. Since C_T is a minimum cover for $G - T$, it follows that $\gamma(C_T) \leq \gamma(\mathsf{E}^* - T)$. Therefore, $\gamma(C_T) + \gamma(T) \leq \gamma(\mathsf{E}^* - T) + \gamma(T) = \gamma(\mathsf{E}^*) = \gamma^*(\mathcal{I})$. □

Lemma 7. *Let $\mathcal{I} = (G, \delta, \gamma)$ be an instance of* DBO, *let C be a minimum cover for (G, δ) and let $0 < \varepsilon < 1$. If $d(\mathsf{E}(C)) \geq 2 - \varepsilon$, then there is a vertex cover C_ε for G such that $\gamma(C_\varepsilon) \leq (\gamma(C - C_\varepsilon))/(1 - \varepsilon)$.*

Proof. Let $T = \Gamma_1(C) - C$. Since $d(\mathsf{E}(C), \mathcal{I}) \geq 2 - \varepsilon$, it follows that $2 - \varepsilon \leq (\gamma(C) + \gamma(T))/(\gamma^*(\mathcal{I}))$.

Now, let C_T be a minimum cover for $G - T$. Since $T \subseteq \Gamma_1(V(G))$, it follows from lemma 6 that $\gamma(C_T) + \gamma(T) \leq \gamma^*(\mathcal{I})$. Simple calculations give $\gamma(T) \leq (\gamma(C) - (2 - \varepsilon)\gamma(C_T))/(1 - \varepsilon)$.

Take $C_\varepsilon = C_T \cup T$ and note that C_ε is a cover for G with $(C_\varepsilon \cap C) \subset C_T$.

$$\gamma(C_\varepsilon) \leq \gamma(C_T) + \gamma(T) \leq \frac{\gamma(C) - \gamma(C_T)}{1 - \varepsilon} \leq \frac{\gamma(C) - \gamma(C_\varepsilon \cap C)}{1 - \varepsilon} = \frac{\gamma(C - C_\varepsilon)}{1 - \varepsilon}.$$

□

Let $\mathcal{I} = (G, \delta, \gamma)$, C and ε be as in the statement of Lemma 7, consider the cost function $\gamma_{C,\varepsilon}$ given by

$$\gamma_{C,\varepsilon}(v) = \begin{cases} (1-\varepsilon)\gamma(v), & \text{if } v \notin C; \\ (2-\varepsilon)\gamma(v), & \text{otherwise,} \end{cases}$$

and let C' be a minimum cover for $(G, \gamma_{C,\varepsilon})$.

We can formulate the problem of finding a cover C_ε satisfying $\gamma(C_\varepsilon) \leq \gamma(C - C_\varepsilon)/(1-\varepsilon)$ as a linear program in order to conclude that such a cover, actually, exists if and only if $\gamma_{C,\varepsilon}(C') \leq \gamma(C)$. Furthermore, if $\gamma_{C,\varepsilon}(C') \leq \gamma(C)$ then $\gamma(C') \leq \gamma(C - C')/(1-\varepsilon)$.

This last remark, together with Lemma 7, provides an efficient way to verify whether or not a particular minimum cover C is going to give a good evaluation for (G, δ, γ).

As the cover C', as above, can be calculated in polynomial time in those cases where G is bipartite, we can devise the following randomized algorithm for DBO, which proceeds in two steps.

At the first step, the algorithm determines a minimum cover C for (G, γ) and a minimum cover C' for $(G, \gamma_{C,\varepsilon})$. If $\gamma_{C,\varepsilon}(C') > \gamma(C)$, then $\mathsf{E}(C)$ will be a good evaluation for (G, δ, γ) and the algorithm returns $\mathsf{E}(C)$. Otherwise, the algorithm enters the second step, where it returns $\mathsf{E}(C)$ with probability $p = p(\varepsilon)$ (see the pseudo-code below) or $\mathsf{E}(C')$ with probability $1-p$ as the solution.

Algorithm $\mathcal{R}_\varepsilon(G, \delta, \gamma)$

(1) $C \leftarrow$ a minimum cover for (G, γ)
(2) $C' \leftarrow$ a minimum cover for $(G, \gamma_{C,\varepsilon})$
(3) If $\gamma_{C,\varepsilon}(C') > \gamma(C)$, then return $\mathsf{E}(C)$
(4) Let $p = (1 - 3\varepsilon + \varepsilon^2)/(1-\varepsilon)$
(5) Let x be a real number randomly selected from interval [0,1]
    If $x < p$, then return $\mathsf{E}(C)$,
    otherwise return $\mathsf{E}(C')$.

Since $\gamma(C) \leq \gamma(C')$ it is reasonable to select C with probability higher than $1/2$, that is $p \geq 0.5$. This condition together with the definition of p in terms of ε force $0 \leq \varepsilon \leq 1 - \sqrt{2}/2$.

3.1. Algorithm analysis. The correctness of algorithm $\mathcal{R}_\varepsilon$ follows from the fact that R_ε always outputs a cover evaluation (recall Section 1.4). Hence, what remains to be shown is the properties of the evaluation calculated by $\mathcal{R}_\varepsilon$, which are claimed at the beginning of this section.

Let $\mathcal{I} = (G, \delta, \gamma)$, C, C' and ε be as in the statement of algorithm $\mathcal{R}$. It will be convenient to define

$$\begin{aligned} H &= \Gamma_1(C \cap C') - (C \cup C'), \\ H_C &= \Gamma_1(C') \cap (C - C'), \\ H_{C'} &= \Gamma_1(C) \cap (C' - C), \end{aligned}$$

so that

$$\begin{aligned} H \cup H_{C'} &= \Gamma_1(C) - C = \Gamma_1(C) - (C \cap \Gamma_1(C)), \\ H \cup H_C &= \Gamma_1(C') - C' = \Gamma_1(C') - (C' \cap \Gamma_1(C')), \end{aligned}$$

and then, as H, $H_C, H_{C'} \subseteq \Gamma_1(V(G))$ are disjoint sets:

$$\gamma(E(C)) = \gamma(C) + \gamma(H_C) + \gamma(H), \tag{4}$$

$$\gamma(E(C')) = \gamma(C') + \gamma(H_{C'}) + \gamma(H), \tag{5}$$

$$\gamma^*(\mathcal{I}) \geq \gamma(H) + \gamma(H_C) + \gamma(H_{C'}). \tag{6}$$

Figure 2 shows a schematic representation of $V(G)$.

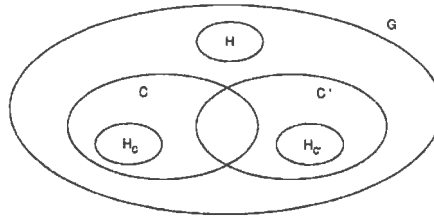

FIGURE 2. A representation for graph G

The following simple result will be useful in our analysis.

Lemma 8. *If C and C' are two covers for G, then $\Gamma(V(G) - (C \cup C')) \subseteq C \cap C'$.*

Proof. Let $v \in V(G) - (C \cup C')$ and $u \in \Gamma(v)$. We must have $u \in C$, otherwise C would not cover uv. The same argument allows to conclude $u \in C'$ and hence the result. □

In the context of Figure 2, this lemma tells us that any edge having an endpoint in U must have its other endpoint in R, since $R = C \cap C'$.

Theorem 9. *Let $0 \leq \varepsilon \leq 1 - \sqrt{2}/2$. For any instance $\mathcal{I} = (G, \delta, \gamma)$ we have $\mathbb{E}[d(\mathcal{R}_\varepsilon(\mathcal{I}))] \leq 2 - \varepsilon$ and $\mathbb{P}\left(d(\mathcal{R}_\varepsilon(\mathcal{I})) \leq (2-\varepsilon)/(1-\varepsilon)\right) = 1$.*

Proof. If $\gamma_{C,\varepsilon}(C') > \gamma(C)$ at step 4, it follows from Lemma 7 that $d(\mathcal{R}) \leq 2 - \varepsilon$. Hence, we assume that $\gamma_{C,\varepsilon}(C') \leq \gamma(C)$. First, we argue about the expectation of $d(\mathcal{R}_\varepsilon(\mathcal{I})$. We have that

$$\mathbb{E}[d(\mathcal{R}_\varepsilon)] = \max_{\mathcal{I}} \frac{\mathbb{E}[\gamma(\mathcal{R}_\varepsilon(\mathcal{I}))]}{\gamma^*(\mathcal{I})}.$$

On the one hand, from equations (4) and (5) we have

$$\mathbb{E}[\gamma(\mathcal{R}_\varepsilon(\mathcal{I}))] \leq p\gamma(\mathsf{E}(C)) + (1-p)\gamma(\mathsf{E}(C')) = p\gamma(C) + (1-p)\gamma(C') + (1-p)\gamma(H_C) + p\gamma(H_{C'}) + \gamma(H) \tag{7}$$

Now, let C_H be a minimum cover for $(G - H, \gamma)$. Applying Lemma 8 to $V(H)$ we have that $C_H \cup (C \cap C')$ is a cover for G, since every edge in $G - H$ is covered by C_H and every edge incident to H is covered by a vertex in $C \cap C'$.

Since C is a minimum cover for G, we have that

$$\gamma(C) \leq \gamma(C_H \cup (C \cap C')) \leq \gamma(C_H) + \gamma(C \cap C'),$$

or, equivalently,

$$\gamma(C_H) \geq \gamma(C) - \gamma(C \cap C') \geq \gamma(C - C').$$

Since $H \subseteq \Gamma_1(V(G))$, it follows from Lemma 6 that

$$\gamma^*(\mathcal{I}) \geq \gamma(H) + \gamma(C - C'),$$

so that, from equation (6) and the above we have,

$$\gamma^*(\mathcal{I}) \geq \max\{\gamma(C), \gamma(H) + \gamma(H_C) + \gamma(H_{C'}), \gamma(H) + \gamma(C - C')\} \tag{8}$$

Since $p \geq 0.5$, it follows from (7) and (8) that

$$\mathbf{E}[d(\mathcal{R}_\varepsilon)] \leq 2p + (1-p)(\gamma(C') + \gamma(H))/(\gamma(H) + \gamma(C - C')).$$

Replacing p by its definition in terms of ε and using the fact that

$$\gamma(C') \leq (\gamma(C) - \gamma(C \cap C'))/(1-\varepsilon),$$

we get that

$$\mathbf{E}[d(\mathcal{R}_\varepsilon)] \leq 2 - \varepsilon.$$

If the cover C is selected by $\mathcal{R}_\varepsilon$, then $d(\mathcal{R}_\varepsilon(\mathcal{I})) \leq 2$, since C is a minimum cost cover. On the other hand, if C' is selected, then

$$d(\mathcal{R}_\varepsilon(\mathcal{I})) \leq \frac{\gamma(C') + \gamma(H_C) + \gamma(H)}{\max\{\gamma(C), \gamma(H_C) + \gamma(H)\}}$$

Since $\gamma(C') \leq (\gamma(C) - \gamma(C \cap C'))/(1-\varepsilon)$, we haev $d(\mathcal{R}_\varepsilon(\mathcal{I})) \leq (2-\varepsilon)/(1-\varepsilon)$. Therefore (3) holds and we are done. □

The above analysis for the expected case is tight when $\varepsilon = 1 - \sqrt{2}/2$. Indeed, consider the instance $\mathcal{I} = (G, \delta, \gamma)$, where G is a complete bipartite graph with bipartition $\{A, B\}$ where $|B| = \sqrt{2}|A| \approx 1.41|A|$, $\delta(a) = 0$ for every $a \in A$, $\delta(b) = 1$ for every $b \in B$, and $\gamma(v) = 1$ for every $v \in V(G)$. Clearly, A is an evaluation of cost $|A|$ since it only checks the vertices in A. The set B, however, is a minimum cover for $(G, \gamma_{C,\varepsilon})$ and $\gamma_{C,\varepsilon}(B) \leq \gamma(A)$. Hence, $\mathcal{R}(\mathcal{I})$ returns $\mathsf{E}(A)$ with probability $1/2$ and $\mathsf{E}(B)$ with probability $1/2$, so that the expected deficiency is close to $1 + \sqrt{2}/2$.

3.2. **Lower bound for randomized algorithms.** We have proved so far that algorithm $\mathcal{R}_\varepsilon$, for $\varepsilon = 1 - \sqrt{2}/2$, has expected deficiency 1.707. However, $\mathcal{R}_\varepsilon$ does not achieve this small deficiency with high probability. For the instance which shows that its analysis is tight, it attains deficiency 2.41 with probability $1/2$ and deficiency 1 with probability $1/2$. One can speculate if a more dynamic algorithm would not have smaller (closer to 1.5) deficiency with high probability. In this section, we prove that this is not possible, that is, no randomized algorithm for DBO can have deficiency smaller than μ for any given $1 \leq \mu \leq 2$ with probability close to 1 (see Theorem 11). *We shall prove this considering instances $\mathcal{I} = (G, \delta, \gamma)$ with G a balanced, complete bipartite graph on n vertices and with $\gamma \equiv 1$ only.* All instances in this section are assumed to be of this form.

Let $\mathcal{A}$ be a randomized algorithm for DBO and let $1/2 \leq \lambda \leq 1$. Given an instance $\mathcal{I} = (G, \delta, \gamma)$ where $|V(G)| = n$, let $P(\mathcal{A}, \mathcal{I}, \lambda n) = \mathbb{P}\big(\gamma(\mathcal{A}(\mathcal{I})) \geq \lambda n\big)$ and let

$$P(\mathcal{A}, \lambda n) = \max_{\mathcal{I}} P(\mathcal{A}, \mathcal{I}, \lambda n).$$

Given a deterministic algorithm $\mathcal{B}$ and an instance $\mathcal{I}$ for DBO, we define the *payoff of $\mathcal{B}$ with respect to $\mathcal{I}$* as

$$g(\mathcal{B}, \mathcal{I}) = \begin{cases} 1, & \text{if } \gamma(\mathcal{B}(\mathcal{I})) \geq \lambda n; \\ 0, & \text{otherwise.} \end{cases}$$

One may deduce from Yao's minimax principle [13] that, for any randomized algorithm $\mathcal{A}$, we have

$$\max_{\mathcal{I}} \mathbf{E}[g(\mathcal{A}, \mathcal{I})] \geq \max_{p} \mathbf{E}[g(\mathsf{opt}, \mathcal{I}_p)], \tag{9}$$

where opt is an optimal deterministic algorithm, in the average case sense, for the probability distribution p over the set of possible instances for DBO. (In (9), the expectation is taken with respect to the coin flips of $\mathcal{A}$ on the left-hand side and with respect to p on the right-hand side; we write $\mathcal{I}_p$ for an instance generated according to p.)

Since a randomized algorithm can be viewed as a distribution probability over the set of deterministic algorithms, we have $\mathbf{E}[g(\mathcal{A},\mathcal{I})] = P(\mathcal{A},\mathcal{I},\lambda n)$ and hence

$$\max_{\mathcal{I}} \mathbf{E}[g(\mathcal{A},\mathcal{I})] = P(\mathcal{A},\lambda n).$$

Moreover, $\mathbf{E}[g(\mathsf{opt},\mathcal{I}_p)]$ is the probability that the cost of the evaluation computed by the optimal algorithm for the distribution p is at least λn. Thus, if we are able to define a probability distribution p over the set of possible instances and analyze the optimal algorithm for such a distribution, we obtain a lower bound for $P(\mathcal{A},\lambda n)$.

Let n be an even positive integer and let G be a complete bipartite graph with $V(G) = \{1,\ldots,n\}$. Let the vertex classes of G be $\{1,\ldots,n/2\}$ and $\{n/2+1,\ldots,n\}$. Let $\gamma(v) = 1$ for all $v \in V(G)$. For $1 \le i \le n$, define the function $\delta_i\colon V(G) \to \{0,1\}$ putting

$$\delta_i(v) = \begin{cases} 1, & \text{if } i = v; \\ 0, & \text{otherwise.} \end{cases}$$

Consider the probability distribution p where the only instances with positive probability are $\mathcal{I}_i = (G,\delta_i,\gamma)$ $(1 \le i \le n)$ and all these instances are equiprobable, with probability $1/n$ each. A key property of these instances is that the cost of the optimum evaluation for all of them is $n/2$, since all the vertices of the vertex class of the graph that does not contain the vertex with δ-value 1 must be evaluated in order to determine the value of all edges. We have the following lemma.

Lemma 10. *Let* opt *be an optimal algorithm for the distribution probability p. Then*

$$\mathbf{E}[g(\mathsf{opt},\mathcal{I}_p)] \ge 1 - \lambda.$$

Proof. First of all we observe that if opt finds a node v with $\delta(v) = 1$ during its execution, then it must evaluate all nodes from the opposite side of v before it finishes (G is complete bipartite).

Let j be the integer such that opt evaluates the node j at the moment it completes the evaluation of *all* nodes from one vertex class of the graph, say X, having obtained $\delta(x) = 0$ *for all vertices* $x \in X$ *before* node j. (To determine j simply run opt and reply $\delta(v) = 0$ for all vertices until it is about to evaluate all vertices from one vertex class.)

Let $v_1, v_2, \ldots, v_{n/2} = j$ be the nodes of X in the order that they were evaluated by opt. Let $t = \lambda - 1/2$. Consider the instances $\mathcal{I}_{v_{\lceil tn\rceil}}, \mathcal{I}_{v_{\lceil tn\rceil+1}}, \ldots, \mathcal{I}_{v_{n/2}}$. In all these instances, opt evaluates at least $\lceil tn\rceil$ nodes from X and all nodes from the other side. Thus, for at least $n/2 - \lceil tn\rceil + 1$ instances opt costs at least $n/2 + \lceil tn\rceil = \lceil \lambda n\rceil$. Hence,

$$\mathbf{E}[g(\mathsf{opt},\mathcal{I}_p)] \ge \frac{n/2 - \lceil tn\rceil + 1}{n} \ge 1 - \lambda,$$

and we are done. □

Since $\gamma^*(\mathcal{I}_j) = n/2$ for $1 \le j \le n$, we have the following result.

Theorem 11. *Let $\mathcal{A}$ be a randomized algorithm for* DBO *and let $1 \le \mu \le 2$ a real number. Then there is an instance $\mathcal{I}$ for which $\mathbf{P}(\mathcal{D}(\mathcal{A}(\mathcal{I}),\mathcal{I}) \ge \mu) \ge 1 - \mu/2$.*

4. Fast Estimation of Qualifying Tuples

In this brief section, we consider the 'numeric version' of DMO, as discussed at the end of Section 1.2. Suppose we have a k-attributed database in which we would like to estimate very quickly how many tuples are likely to satisfy a certain conjunctive query. We think of a scenario where the database is a fixed set of k-tuples over which many different queries are posed for us to answer.

More precisely, we consider the following formalization of the problem. Suppose we have a k-uniform hypergraph G. An instance for our problem is the same as an instance for DMO, and we

wish to determine or estimate very quickly the value of $m(G,\delta) = |\bar{\delta}^{-1}(1)|$. In fact, we suppose that we shall have a large family δ_λ ($\lambda \in \Lambda$) of queries and we wish to estimate $m(G,\delta_\lambda)$ for each $\lambda \in \Lambda$.

Our computation model is as follows. We may preprocess G as long as we spend time polynomial in $|V(G)|$, and we suppose that we may sample elements from $V(G)$ uniformly at random. We further suppose that, given a k-tuple $g \subset V(G)$, we may check whether or not $g \in E(G)$. Finally, we suppose that the operations of sampling a random vertex v from $V(G)$, computing $\delta(v)$, and checking whether $g \in G$ all have unit cost. We may prove the following result.

Theorem 12. *There exist a randomized algorithm $\mathcal{A}$ and a deterministic, polynomial time algorithm $\mathcal{R}$ as follows. For any $\varepsilon > 0$, there is a constant $C = C(\varepsilon)$ for which the following holds. Let a k-uniform hypergraph G be given. We may preprocess G using algorithm $\mathcal{R}$ to obtain a structure $\mathcal{R}(G)$ that may be given to $\mathcal{A}$ as an auxiliary input so that, given any $\delta\colon V(G) \to \{0,1\}$, algorithm $\mathcal{A}$ returns an integer m so that $\mathbb{P}\left(|m - m(G,\delta)| \le \varepsilon |V(G)|^k\right) \ge 1 - \varepsilon$.*

Moreover, $\mathcal{A}$ samples at most C vertices from $V(G)$. In fact, if we let $S \subset V(G)$ be the set of vertices sampled by $\mathcal{A}$, then $\mathcal{A}$ only evaluates δ on S and only examines those edges $e \in E(G)$ where $e \subseteq S$. In particular, $\mathcal{A}(G,\delta)$ returns m in constant time (that is, independent on the size of G or the particular δ).

In view of Theorem 12, we may process a family δ_λ ($\lambda \in \Lambda$) of queries in time linear in $|\Lambda|$, because the cost of processing each δ_λ is constant. The overhead in our procedure is the cost of preprocessing G, which is polynomial in $|V(G)|$. We do not go into the details of the proof of Theorem 12. We only observe that we may use for $\mathcal{R}$ the algorithm for constructing regular partitions for hypergraphs given in [6].

References

[1] R. Avnur and J. M. Hellerstein. Eddies: continuously adaptive query processing. In W. Chen, J. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD*, vol. 29(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pp. 261–272, New York, NY 10036, USA, 2000. ACM Press.

[2] R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198–203, June 1981.

[3] L. Bouganim, F. Fabret, F. Porto, and P. Valduriez. Processing queries with expensive functions and large objects in distributed mediator systems. In *Proc. 17th Intl. Conf. on Data Engineering, April 2-6, 2001, Heidelberg, Germany*, pages 91–98, 2001.

[4] M. Charikar, R. Fagin, V. Guruswami, J. Kleinberg, P. Raghavan, and A. Sahai. Query strategies for priced information (extended abstract0. In ACM, editor, *Proceedings of the 32nd annual ACM Symposium on Theory of Computing: Portland, Oregon, May 21–23, [2000]*, pages 582–591, New York, NY, USA, 2000. ACM Press.

[5] S. Chaudhuri and K. Shim. Query optimization in the presence of foreign functions. In *Proc. 19th Intl. Conf. on Very Large Data Bases, August 24-27, 1993, Dublin, Ireland*, pages 529–542, 1993.

[6] A. Czygrinow and V. Rödl. An algorithmic regularity lemma for hypergraphs. *SIAM J. Comput.*, 30(4):1041–1066, 2000.

[7] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 136–146, Berkeley, California, 28–30 May 1986.

[8] J. M. Hellerstein. Optimization techniques for queries with expensive methods. *ACM Transactions on Database Systems*, 23(2):113–157, June 1998.

[9] D. S. Hochbaum. Approximating clique and biclique problems. *Journal of Algorithms*, 29(1):174–200, Oct. 1998.

[10] E. S. Laber, O. Parekh, and R. Ravi. Randomized approximation algorithms for query optimization problems on two processors. In *Proceedings of ESA 2002*, pages 136–146, Rome, Italy, September 2002.

[11] F. Porto. *Estratégias para a Execução Paralela de Consultas em Bases de Dados Científicos Distribuídos*. PhD thesis, Departamento de Informática, PUC-Rio, Apr. 2001.

[12] D. Ron. Property testing. In P. M. Pardalos, S. Rajasekaran, J. Reif, and J. D. P. Rolim, editors, *Handbook of randomized algorithms*. Kluwer Academic Publishers, 2001. to appear.

[13] A. C. Yao. Probabilistic computations : Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science*, pages 222–227, Long Beach, Ca., USA, Oct. 1977. IEEE Computer Society Press.

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1999 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail (mac@ime.usp.br).

Carlos Alberto de Bragança Pereira, Fabio Nakano e Julio Michael Stern
A DYNAMIC SOFTWARE CERTIFICATION AN VERIFICATION PROCEDURE
RT-MAC-9901, março 1999, 21pp.

Carlos E. Ferreira e Dilma M. Silva
BCC DA USP: UM NOVO CURSO PARA OS DESAFIOS DO NOVO MILÊNIO
RT-MAC-9902, abril 1999, 12pp.

Ronaldo Fumio Hashimoto and Junior Barrera
A SIMPLE ALGORITHM FOR DECOMPOSING CONVEX STRUCTURING ELEMENTS
RT-MAC-9903, abril 1999, 24 pp.

Jorge Euler, Maria do Carmo Noronha e Dilma Menezes da Silva
ESTUDO DE CASO: DESEMPENHO DEFICIENTE DO SISTEMA OPERACIONAL LINUX PARA CARGA MISTA DE APLICAÇÕES.
RT-MAC-9904, maio 1999, 27 pp.

Carlos Humes Junior e Paulo José da Silva e Silva
AN INEXACT CLASSICAL PROXIMAL POINT ALGORITHM VIEWED AS DESCENT METHOD IN THE OPTIMIZATION CASE
RT-MAC-9905, maio 1999, pp.

Carlos Humes Junior and Paulo José da Silva e Silva
STRICT CONVEX REGULARIZATIONS, PROXIMAL POINTS AND AUGMENTED LAGRANGIANS
RT-MAC-9906, maio 1999, 21 pp.

Ronaldo Fumio Hashimoto, Junior Barrera, Carlos Eduardo Ferreira
A COMBINATORIAL OPTIMIZATION TECHNIQUE FOR THE SEQUENTIAL DECOMPOSITION OF EROSIONS AND DILATIONS
RT-MAC-9907, maio 1999, 30 pp.

Carlos Humes Junior and Marcelo Queiroz
ON THE PROJECTED PAIRWISE MULTICOMMODITY FLOW POLYHEDRON
RT-MAC-9908, maio 1999, 18 pp.

Carlos Humes Junior and Marcelo Queiroz
TWO HEURISTICS FOR THE CONTINUOUS CAPACITY AND FLOW ASSIGNMENT GLOBAL OPTIMIZATION
RT-MAC-9909, maio 1999, 32 pp.

Carlos Humes Junior and Paulo José da Silva e Silva
AN INEXACT CLASSICAL PROXIMAL POINT ALGORITHM VIEWED AS A DESCENT METHOD IN THE OPTIMIZATION CASE
RT-MAC-9910, julho 1999, 13 pp.

Markus Endler and Dilma M. Silva and Kunio Okuda
A RELIABLE CONNECTIONLESS PROTOCOL FOR MOBILE CLIENTS
RT-MAC-9911, setembro 1999, 17 pp.

David Robertson, Fávio S. Corrêa da Silva, Jaume Agustí and Wamberto W. Vasconcelos
A LIGHTWEIGHT CAPABILITY COMMUNICATION MECHANISM
RT-MAC-9912, novembro 1999, 14 pp.

Flávio S. Corrêa da Silva, Jaume Agustí, Roberto Cássio de Araújo and Ana Cristina V. de Melo
KNOWLEDGE SHARING BETWEEN A PROBABILISTIC LOGIC AND BAYESIAN BELIEF NETWORKS
RT-MAC-9913, novembro 1999, 13 pp.

Ronaldo F. Hashimoto, Junior Barrera and Edward R. Dougherty
FINDING SOLUTIONS FOR THE DILATION FACTORIZATION EQUATION
RT-MAC-9914, novembro 1999, 20 pp.

Marcelo Finger and WanbertoVasconcelos
SHARING RESOURCE-SENSITIVE KNOWLEDGE USING COMBINATOR LOGICS
RT- MAC-2000-01, março 2000, 13pp.

Marcos Alves e Markus Endler
PARTICIONAMENTO TRANSPARENTE DE AMBIENTES VIRTUAIS DISTRIBUÍDOS
RT- MAC-2000-02, abril 2000, 21pp.

Paulo Silva, Marcelo Queiroz and Carlos Humes Junior
A NOTE ON "STABILITY OF CLEARING OPEN LOOP POLICIES IN MANUFACTURING SYSTEMS"
RT- MAC-2000-03, abril 2000, 12 pp.

Carlos Alberto de Bragança Pereira and Julio Michael Stern
FULL BAYESIAN SIGNIFICANCE TEST: THE BEHRENS-FISHER AND COEFFICIENTS OF VARIATION PROBLEMS
RT-MAC-2000-04, agosto 2000, 20 pp.

Telba Zalkind Irony, Marcelo Lauretto, Carlos Alberto de Bragança Pereira and Julio Michael Stern
A WEIBULL WEAROUT TEST: FULL BAYESIAN APPROACH
RT-MAC-2000-05, agosto 2000, 18 pp.

Carlos Alberto de Bragança Pereira and Julio Michael Stern
INTRINSIC REGULARIZATION IN MODEL SELECTION USING THE FULL BAYESIAN SIGNIFICANCE TEST
RT-MAC-2000-06, outubro 2000, 18 pp.

Douglas Moreto and Markus Endler
EVALUATING COMPOSITE EVENTS USING SHARED TREES
RT-MAC-2001-01, janeiro 2001, 26 pp.

Vera Nagamura and Markus Endler
COORDINATING MOBILE AGENTS THROUGH THE BROADCAST CHANNEL
RT-MAC-2001-02, janeiro 2001, 21 pp.

Júlio Michael Stern
THE FULLY BAYESIAN SIGNIFICANCE TEST FOR THE COVARIANCE PROBLEM
RT-MAC-2001-03, feverciro 2001, 15 pp.

Marcelo Finger and Renata Wassermann
TABLEAUX FOR APPROXIMATE REASONING
RT- MAC-2001-04, março 2001, 22 pp.

Julio Michael Stern
FULL BAYESIAN SIGNIFICANCE TESTS FOR MULTIVARIATE NORMAL STRUCTURE MODELS
RT-MAC-2001-05, junho 2001, 20 pp.

Paulo Sérgio Naddeo Dias Lopes and Hernán Astudillo
VIEWPOINTS IN REQUIREMENTS ENGINEERING
RT-MAC-2001-06, julho 2001, 19 pp.

Fabio Kon
O SOFTWARE ABERTO E A QUESTÃO SOCIAL
RT- MAC-2001-07, setembro 2001, 15 pp.

Isabel Cristina Italiano, João Eduardo Ferreira and Osvaldo Kotaro Takai

ASPECTOS CONCEITUAIS EM DATA WAREHOUSE
RT – MAC-2001-08, setembro 2001, 65 pp.

Marcelo Queiroz , Carlos Humes Junior and Joaquim Júdice
ON FINDING GLOBAL OPTIMA FOR THE HINGE FITTING PROBLEM
RT- MAC –2001-09, novembro 2001, 39 pp.

Marcelo Queiroz , Joaquim Júdice and Carlos Humes Junior
THE SYMMETRIC EIGENVALUE COMPLEMENTARITY PROBLEM
RT- MAC-2001-10, novembro 2001, 33 pp.

Marcelo Finger, and Fernando Antonio Mac Cracken Cezar
BANCO DE DADOS OBSOLESCENTES E UMA PROPOSTA DE IMPLEMENTAÇÃO.
RT- MAC - 2001-11- novembro 2001, 90 pp.

Flávio Soares Correa da Silva
TOWARDS A LOGIC OF PERISHABLE PROPOSITIONS
RT- MAC- 2001-12 - novembro 2001, 15 pp.

Alan M. Durham
O DESENVOLVIMENTO DE UM INTERPRETADOR ORIENTADO A OBJETOS PARA ENSINO DE LINGUAGENS
RT-MAC-2001-13 – dezembro 2001, 21 pp.

Alan M. Durham
A CONNECTIONLESS PROTOCOL FOR MOBILE AGENTS
RT-MAC-2001-14 – dezembro 2001, 12 pp.

Eugênio Akihiro Nassu e Marcelo Finger
O SIGNIFICADO DE "AQUI" EM SISTEMAS TRANSACIONAIS MÓVEIS
RT-MAC-2001-15 – dezembro 2001, 22 pp.

Carlos Humes Junior, Paulo J. S. Silva e Benar F. Svaiter
SOME INEXACT HYBRID PROXIMAL AUGMENTED LAGRANGIAN ALGORITHMS
RT-MAC-2002-01 – Janeiro 2002, 17 pp.

Roberto Speicys Cardoso e Fabio Kon
APLICAÇÃO DE AGENTES MÓVEIS EM AMBIENTES DE COMPUTAÇÃO UBÍQUA.
RT-MAC-2002-02 – Fevereiro 2002, 26 pp.

Julio Stern and Zacks
TESTING THE INDEPENDENCE OF POISSON VARIATES UNDER THE HOLGATE BIVARIATE DISTRIBUTION: THE POWER OF A NEW EVIDENCE TEST.
RT- MAC – 2002-03 – Abril 2002, 18 pp.

E. N. Cáceres, S. W. Song and J. L. Szwarcfiter
A PARALLEL ALGORITHM FOR TRANSITIVE CLOSURE
RT-MAC – 2002-04 – Abril 2002, 11 pp.

Regina S. Burachik, Suzana Scheimberg, and Paulo J. S. Silva
A NOTE ON THE EXISTENCE OF ZEROES OF CONVEXLY REGULARIZED SUMS OF MAXIMAL MONOTONE OPERATORS
RT- MAC 2002-05 – Maio 2002, 14 pp.

C.E.R. Alves, E.N. Cáceres, F. Dehne and S. W. Song
A PARAMETERIZED PARALLEL ALGORITHM FOR EFFICIENT BIOLOGICAL SEQUENCE COMPARISON
RT-MAC-2002-06 – Agosto 2002, 11pp.

Julio Michael Stern
SIGNIFICANCE TESTS, BELIEF CALCULI, AND BURDEN OF PROOF IN LEGAL AND SCIENTIFIC DISCOURSE
RT- MAC – 2002-07 – Setembro 2002, 20pp.

Andrei Goldchleger, Fabio Kon, Alfredo Goldman vel Lejbman, Marcelo Finger and Siang Wun Song.
INTEGRADE: RUMO A UM SISTEMA DE COMPUTAÇÃO EM GRADE PARA APROVEITAMENTO DE RECURSOS OCIOSOS EM MÁQUINAS COMPARTILHADAS.
RT-MAC – 2002-08 – Outubro 2002, 27pp.

Flávio Protasio Ribeiro
OTTERLIB – A C LIBRARY FOR THEOREM PROVING
RT- MAC – 2002-09 – Dezembro 2002 , 28pp.

Cristina G. Fernandes, Edward L. Green and Arnaldo Mandel
FROM MONOMIALS TO WORDS TO GRAPHS
RT-MAC – 2003-01 – fevereiro 2003, 33pp.

Andrei Goldchleger, Márcio Rodrigo de Freitas Carneiro e Fabio Kon
GRADE: UM PADRÃO ARQUITETURAL
RT- MAC – 2003-02 – março 2003, 19pp.

C. E. R. Alves, E. N. Cáceres and S. W. Song
SEQUENTIAL AND PARALLEL ALGORITHMS FOR THE ALL-SUBSTRINGS LONGEST COMMON SUBSEQUENCE PROBLEM
RT- MAC – 2003-03 – abril 2003, 53 pp.

Said Sadique Adi and Carlos Eduardo Ferreira
A GENE PREDICTION ALGORITHM USING THE SPLICED ALIGNMENT PROBLEM
RT- MAC – 2003-04 – maio 2003, 17pp.

Eduardo Laber, Renato Carmo, and Yoshiharu Kohayakawa
QUERYING PRICED INFORMATION IN DATABASES: THE CONJUNTIVE CASE
RT-MAC – 2003-05 – julho 2003, 19pp.