On the Complexity of Designing Optimal Partial-Match Retrieval Systems

SHLOMO MORAN
The Technion—Israel Institute of Technology, Haifa

We consider the problem of designing an information retrieval system on which partial match queries have to be answered. Each record in the system consists of a list of *attributes*, and a partial match query specifies the values of some of the attributes. The records are stored in *buckets* in a secondary memory, and in order to answer a partial match query all the buckets that may contain a record satisfying the specifications of that query must be retrieved. The bucket in which a given record is stored is found by a multiple key hashing function, which maps each attribute to a string of a fixed number of bits. The address of that bucket is then represented by the string obtained by concatenating the strings on which the various attributes were mapped. A partial match query may specify only part of the bits in the string representing the address, and the larger the number of bits specified, the smaller the number of buckets that have to be retrieved in order to answer the query.

The optimization problem considered in this paper is that of deciding to how many bits each attribute should be mapped by the bashing function above, so that the expected number of buckets retrieved per query is minimized. Efficient solutions for special cases of this problem have been obtained in [1], [12], and [14]. It is shown that in general the problem is NP-hard, and that if $P \neq$ NP, it is also not fully approximable. Two heuristic algorithms for the problem are also given and compared.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval.

General Terms:

Additional Key Words and Phrases: Partial match retrieval, hashing, searching, file organization, NP-hard problems, approximation algorithms

1. INTRODUCTION

We consider the problem of storing a file F of multiattribute records on which partial match queries are to be answered. Each record is a list of *attributes* $(x_1, \ldots, x_t)$, where for $i = 1, \ldots, t$, each x_i can take one out of d_i values ($d_i \geq 2$). Let a_i be the "name" or the "field" of the ith attribute. Then a partial match query is a query of the form, "Retrieve all records for which $a_{i_1} = x_{i_1}, \ldots, a_{i_k} = x_{i_k}$"$(1 \leq i_1 < \cdots < i_k \leq t)$. This query is said to *specify* fields $a_{i_1}, \ldots, a_{i_k}$

This research was supported in part by the National Science Foundation under Grant MCS78-01736.

Author's address: Computer Science Department, The Technion, Israel Institute of Technology, Haifa, Israel.

Example. Let F be a file on passengers booked on a given airline. Then each record in F may consist of the following fields

a_1 = First name
a_2 = Initial
a_3 =Last name
a_4 =Address
a_5 =Phone number
a_6 =Flight number
a_7 = Flight date

A partial match query for this file may be "list all records of passengers booked on flight number 409 on 4–3–81" or "list all records of passengers whose first name is John, last name is Smith, and who are booked on a flight on 4–10–81." The first query specifies fields a_6 and a_7 while the second specifies fields a_1, a_3, and a_7.

We consider a system in which the file is stored in a secondary memory, which is organized in *buckets* of fixed capacity. In order to answer a partial match query, all the buckets that may contain records satisfying the specification of the query have to be transferred to the main memory, where they are searched for the desired records. The time required to answer a given query is roughly proportional to the number of buckets that have to be searched [14, 15].

Let $A = \{a_1, \ldots, a_t\}$ be the set of fields, and suppose we know for each subset S of A the probability that a query which specifies exactly the fields in S occurs (this probability may be 0). The optimization problem discussed in this paper involves the minimization of the average number of buckets searched per query. One strategy, which was suggested in [12, 14], is to use a multiple key hashing function to compute the address of a bucket containing a given record, as described below.

Suppose that the total number of buckets is 2^B for some integer B. Then each bucket may be represented by a sequence of B bits, and the address of the bucket may be computed from this representation.[1] Given a record $R = (x_1, \ldots, x_t)$, the binary sequence that represents the bucket in which R is stored is given by $\bar{h}(x_1, \ldots, x_t)$, where $\bar{h}$ is a multiple key hashing function that maps the set of all possible records onto $\{0, 1, \ldots, 2^B - 1\}$. We shall assume that $\bar{h}$ is a *product*, that is, that for $i = 1, \ldots, t$, there are single key hashing functions h_i and integers $b(i)$ such that

(a) $\sum_{i=1}^{t} b(i) = B$,
(b) h_i maps x_i to a sequence of $b(i)$ bits, and
(c) $\bar{h}(x_1, \ldots, x_t) = h(x_1) \cdot h(x_2) \cdot \ldots \cdot h(x_t)$ (here $\cdot$ represents concatenation).

Let $\bar{h}$ be given, and let $S = \{a_{i_1}, \ldots, a_{i_j}\} \subseteq A$. Define $b(S) = b(i_1) + \cdots + b(i_j)$. Then a query that specifies fields $a_{i_1}, \ldots, a_{i_j}$ specifies $b(S)$ bits of the binary sequences which represent the buckets that have to be searched in order to answer this query. Hence, the number of buckets which have to be searched is

[1] The restriction of the number of buckets to be a power of two is not necessary for most of our results; however, it simplifies the discussion and fits with binary computers, where information is stored by sequences of bits.

$2^{B-b(S)}$, which is $2^{-b(S)}$ times the size of the entire file. Let $Q = \{S_1, \ldots, S_n\}$, where for each i $S_i \subseteq A$, and let $p_i > 0$ be the probability that a query specifies exactly the fields in $S_i (\sum_{i=1}^{n} p_i = 1)$. Then, on the average, the portion of the entire file that has to be searched on each query is given by $\sum_{j=1}^{n} p_j 2^{-b(S_j)}$. The optimization problem considered in this paper is to find a distribution of the B bits among the various fields such that this average is minimized. In the next section we give a formal definition of the problem, together with some known results. In Section 3 we show that the problem is NP-hard. In Section 4 we extend the result to show that even good approximation algorithms that guarantee a worst-case relative error smaller than ϵ in time which is polynomial in both the length of the input and $1/\epsilon$ do not exist for this problem unless $P = \text{NP}$. In Section 5 we describe and compare two heuristic algorithms for the problem.

2. PRELIMINARIES

The partial match retrieval optimization problem (PMR) is the following.

Input. (A, Q, P, B), where $A = \{a_1, \ldots, a_t\}$ is a set of *fields*; $Q = \{S_1, \ldots, S_n\}$, where, for each i, $S_i \subseteq A$, is a set of "query-specifications;" $P = \{p_1, \ldots, p_n\}$ is the set of the corresponding probabilities $(\sum_{i=1}^{n} p_i = 1)$ and B is the number of bits (2^B is the number of buckets).

Output. A function $b: A \rightarrow Z^+$ satisfying

(1) $\sum_{i=1}^{t} b(a_i) = B$

(2) $w(b) = \sum_{j=1}^{n} p_j 2^{-b(S_j)}$ is minimized, where $b(S_j)$ is defined as follows: Let $\delta_{ij} = 0$ if $a_i \notin S_j$, 1 otherwise. Then $b(S_j) = \sum_{i=1}^{t} \delta_{ij} b(a_i)$. $w(b)$ will be denoted as the *weight of* b.

We shall sometimes assume that the range of b is the set $\mathbf{R}^+$ of nonnegative real numbers—and in this case we shall refer to the problem as the "continuous PMR."

Efficient algorithms to solve the PMR have been found in some special cases: In [12] an efficient solution is given for the case where all the subsets of A of a given cardinality have the same probability to be specified (see also [2, 15]). In [14] a solution is given to the case where each query specifies only one field, and in [1] a solution is given to the case where the fields are specified independently. None of these cases seems to be general enough to reflect realistic models, like the example given at the beginning of this paper. (Among the queries that specify exactly two fields, *last name* and *first name* are more likely to be specified than *initial* and *telephone number*. It is also likely that the event that the *first name* is specified is dependent on the event that the *last name* is specified.) However, in the next section we shall show that in the general case the problem is NP-hard. This should also be compared with a result in [2], which shows that the continuous PMR *can* be solved efficiently in the general case.[2]

[2] Since an optimal solution to the continuous PMR may consist of irrational numbers (even if the input numbers are rational), we define an *efficient algorithm* for this problem to be an algorithm which efficiently finds (i.e., in polynomial time) an optimal solution rounded to some fixed (but arbitrarily large) number of digits.

Note 1. In general, we should have for each field a_i an upper bound $M_i \geq 1$ on $b(a_i)$. M_i is determined by the number d_i of the possible values which a_i can attain. (e.g., if a_i specifies sex, then only 2 values are possible, and hence $M_i = 1$). All the results of this paper apply to both the case where the M_is are not specified and to the case where the M_is are specified and take any value ≥ 1.

Note 2. An important requirement for an optimal design for the PMR is the existence of *balanced* hashing functions which distribute the d_i possible values of field a_i evenly among the $2^{b(a_i)}$ buckets. We shall not discuss this requirement in this paper, but shall take the existence of such hashing functions for granted.

3. NP-HARDNESS OF THE PMR

The NP-hardness of the PMR will follow from the NP-completeness of the following 3 *Hitting Set* (3HS) problem:

Input. (A, Q), where $A = \{a_1, \ldots, a_t\}$ is a set, $Q = \{S_1, \ldots, S_{3n}\}$ is a family of subsets of A, such that for each $a_i \in A$ there are exactly 3 sets $S_j \in Q$ such that $a_i \in S_j$.

Property. There is a set $H \subseteq A$ such that $|H \cap S_j| = 1$ for $j = 1, \ldots, 3n$.

LEMMA 3.1. *The 3HS problem is NP-complete.*

PROOF. We shall show that the NP complete problem *exact cover with* 3 *element per set* (3 *X C*, [4, 9]) is polynomially reducible to 3HS.

Input (to 3 *X C*). A set $T = \{x_1, \ldots, x_{3n}\}$ and a family $\{T_1, \ldots, T_t\}$ of subsets of T, where for $1 \leq i \leq t$, $|T_i| = 3$.

Property. There are $1 \leq i_1, \ldots, i_n \leq t$ such that $\cup_{j=1}^{n} T_{ij} = T$. The reduction is $A = \{a_1, \ldots, a_t\}$, and $Q = \{S_1, \ldots, S_{3n}\}$, where $S_i = \{a_j \mid x_i \in T_j\}$. □

THEOREM 3.1. *The PMR problem is NP-hard.*

PROOF. We shall prove the NP-completeness of the corresponding recognition problem.

Input. (A, Q, P, B, r), where A, Q, P, B are as in the definition of the PMR problem, r is a rational number.

Property. There is a function $b: A \rightarrow Z^+$ such that $\sum_{i=1}^{t} b(a_i) = B$ and $w(b) = \sum_{j=1}^{n} p_j 2^{-b(S_j)} \leq r$.

We shall see that the problem remains NP-complete even if all the elements of Q have the same probability, and b is restricted by $b(a_i) \leq 1$ for $i = 1, \ldots, t$. Note that the problem is trivially in NP. To prove its completeness, we shall reduce the 3HS problem to it:

Let (A, Q) be an input to 3HS, where $A = \{a_1, \ldots, a_t\}$, $Q = \{S_1, \ldots, S_{3n}\}$. We reduce it to (A, Q, P, B, r) where: $P = \{p_1, \ldots, p_{3n}\}$, $p_i = 1/3$ *for* $i = 1, \ldots, 3n$; $B = n$; $r = .5$. Again, let δ_{ij} be 1 if $a_i \in S_j$, 0 otherwise. Then for each i, $\sum_{j=1}^{3n} \delta_{ij} = 3$. First we note that for each function $b: A \rightarrow Z^+$ which satisfies $\sum_{i=1}^{t} b(a_i) = n$ we have

$$\sum_{j=1}^{3n} b(S_j) = \sum_{j=1}^{3n} \sum_{i=1}^{t} \delta_{ij} b(a_i) = \sum_{i=1}^{t} \left[b(a_i) \sum_{j=1}^{3n} \delta_{ij} \right] = \sum_{i=1}^{t} 3b(a_i) = 3n.$$

Denote, for a given b, $x_j = 2^{-b(S_j)}$. Then in view of the equality above we can restate the recognition problem, in this specific case, by: Are there numbers $\{a_1, \ldots, a_{3n}\}$ such that

(i) $w = (\sum_{j=1}^{3n} x_j)/3n \leq 1/2$
(ii) $\prod_{j=1}^{3n} x_j = 2^{-3n}$
(iii) There exists a function $b: A \rightarrow Z^+$ such that for $j = 1, \ldots, 3n$, $x_j = 2^{-b(S_j)}$.

Consider now (i) and (ii) alone: By the arithmetic-geometric inequality, w is minimized if and only if $x_j = (2^{-3n})^{1/3n} = .5$ for $j = 1, \ldots, 3n$, and the minimal value for w is, hence, $1/3n(\sum_{j=1}^{3n} .5) = .5$. This means that $(A, Q, P, B, .5)$ has the desired property if and only if $x_j = .5$ for $j = 1, \ldots, 3n$, which is equivalent to $b(S_j) = 1$ for $j = 1, \ldots, 3n$, which, in turn, is equivalent to the statement: The set $H = \{a_j \mid b(a_j) = 1\}$ is a Hitting Set for (A, Q) (and $b(a_j) = 0$ for $a_j \notin H$). This completes the proof of the theorem. □

4. ON THE APPROXIMABILITY OF THE PMR

The NP-hardness of the PMR indicates that probably no polynomial time algorithm can find the optimal solution to this problem. However, it does not exclude the existence of polynomial time algorithms which are guaranteed to find near optimal solutions for the problem even if $P \neq$ NP. In fact, there are some NP-hard optimization problems that have algorithms that, for a given $\epsilon > 0$, find a solution to the given problem whose relative error is guaranteed to be smaller than ϵ in time which is polynomial in both the size of the problem and $1/\epsilon$. Such a problem is said to be "fully approximable" or to have a "fully polynomial time approximation schema" (see [4, 5, 6, 7, 10, 11] for a more detailed exposition and for examples of such problems). In this section we shall show that the PMR is not fully approximable unless $P =$ NP. Some general results characterizing NP-hard problems that are not fully approximable (provided $P \neq$ NP) appear in [5, 10, 11]. Interestingly, Theorem 4.1 below does not follow directly from these general results, and it requires a different proof.

Let (A, Q, P, B) be an input to the PMR, let $\hat{b}$ be an optimal solution to it, and let b be a different solution. Then the relative error of b is the ratio $(w(b) - w(\hat{b}))/w(\hat{b})$.

THEOREM 4.1 *If $P \neq$ NP then the PMR is not fully approximable.*

PROOF. Assume that there is an algorithm Ap which finds for each input (A, Q, P, B) for the PMR and for each $\epsilon > 0$ a function b such that $(w(b) - w(\hat{b}))/w(\hat{b}) < \epsilon$, ($\hat{b}$ denotes an optimal solution), and that the running time of Ap is polynomial in both the size of the input and $1/\epsilon$. We shall derive a contradiction by showing that Ap can be used to provide a polynomial time algorithm for the 3HS problem (which would imply that $P =$ NP.)

Let (A, Q) be an input to the 3HS problem. Define an input (A, Q, P, B) to the PMR where P and B are as defined in the proof of Theorem 3.1 ($p_i = 1/3n$, $B = n$). From Theorem 3.1 we know that (A, Q) is in 3HS if and only if there is an optimal solution $\hat{b}$ to (A, Q, P, B) such that $w(\hat{b}) = .5$. The theorem will now follow from

Claim 1. If (A, Q) is not in 3HS, then the weight of an optimal solution to (A, Q, P, B) is at least $.5(1 + 1/6n)$.

Proof of Claim 1. Assume $(A, Q) \notin$ 3HS, and let $\hat{b}$ be an optimal solution to (A, Q, P, B). Since $\sum_{i=1}^{t} \hat{b}(a_i) = B = n$, the set $H = \{a \mid \hat{b}(a) > 0\}$ is of cardinality $\leq n$. Since (A, Q) is not in 3HS, there is no $H \subseteq A$ such that $|H| \leq n$ and $|H \cap S_j| > 0$ for all j in $\{1, \ldots, 3n\}$. Combining these two facts, we have that there must be a j_o such that $|H \cap S_{j_o}| = 0$, which means that $\hat{b}(S_{j_o}) = 0$. Without loss of generality, assume that $j_o = 3n$. Then we get

(i) $w(\hat{b}) = 1/3n(\sum_{j=1}^{3n-1} 2^{-\hat{b}(S_j)} + 1)$ and
(ii) $\sum_{j=1}^{3n-1} \hat{b}(S_j) = 3n$.

Since $\hat{b}(S_j)$ is an integer for all j, it is not hard to show that the minimal possible value for $w(\hat{b})$ under the constraint (ii) above is obtained when $\hat{b}(S_1) = 2$, $\hat{b}(S_j) = 1$ for $j = 2, \ldots, 3n - 1$, and that this value is $1/4 + (3n - 2)/2 + 1/3n = 1/2 + 1/12n = .5(1 + 1/6n)$. □

To show that the theorem follows from the claim, let $\epsilon = 1/6n$. Use Ap to obtain a solution b to (A, Q, P, B) such that $(w(b) - w(\hat{b}))/w(\hat{b}) < 1/6n$. This can be done in time which is a polynomial in both the size of the problem and $1/\epsilon = 6n$, and hence is a polynomial in the size of the problem. It follows from Claim 1 that $(A, Q) \in$ 3HS if and only if $w(b) < .5(1 + 1/6n)$ (which means that $w(b) = .5$.) □

5. HEURISTIC ALGORITHMS

By the previous two sections, no efficient algorithm can solve the PMR (unless $P = NP$), neither can an arbitrarily good approximation to PMR be obtained at a relatively low cost (unless $P = NP$). An alternative way to attack this problem is to use heuristic algorithms, based on some simple and/or local searching arguments. Such an approach has been proved useful for some other NP-hard optimization problems, such as the traveling salesman and the bin packing problems [3, 8, 13].

In this section we shall describe two such algorithms for the PMR. Both are natural generalizations of algorithms which yield optimal solutions in those special cases of the PMR that are known to have efficient solutions [1, 14, 15]. The first of these, called BALANCE, is based on the algorithm in [14] and involves a step-by-step distribution of the B bits among the t fields. Initially, all the fields are assigned 0 bits, and at each step the number of bits of a field which has a maximal "weight" is increased by one. The second algorithm, called ROUND, is based on the technique in [1, 2] and involves computing an optimal solution to the continuous PMR (which, by [2], can be done efficiently), and then rounding up the results to integers according to certain rules. We shall also compare the performances of these algorithms on two specific examples.

Definition 5.1. Let $b: A \rightarrow R^+$ be a given function, and let $a_i \in A$. Then the weight of a_i with respect to b, $w(b, a_i)$, is given by $w(b, a_i) = \sum_{j=1}^{n} \delta_{ij} p_j 2^{-b(S_j)}$.

Our first heuristic algorithm, BALANCE, is based on the following observation:

Let $b{:}A \rightarrow Z^+$ be a partial assignment of bits to the fields, such that $\sum_{i=1}^{t} b(a_i) < B$. Let b' be defined by: For some i_o, $b'(a_{io}) = b(a_{i_o}) + 1$, and $b'(a_i) = b(a_i)$ for $i \neq i_o$. Then $w(b') = w(b) - w(b, a_{io})/2$. It follows that $w(b')$ is minimized when $w(b, a_{io})$ is maximized.

Algorithm BALANCE. Input: (A, Q, P, B). Output: $b{:}A \rightarrow Z^+$.

1. // *Initialization*// *set* $b(a_i) \leftarrow 0$ for $i = 1, \ldots, t$; $N \leftarrow 0$
2. // Terminate?// *if* $N = B$ *then stop* and *return* b.
3. // Select a field of maximum weight// find $a^* \epsilon A$ such that $w(b, a^*) \geq w(b, a)$ for all $a \epsilon A$.
4. // *Increase* $b(a^*)$// set $b(a^*) \leftarrow b(a^*) + 1$, $N \leftarrow N + 1$.
5. Go to 2.

Our second algorithm is based on the rounding technique of [1] (see also [15]).

Algorithm ROUND. Input and output: the same as for BALANCE.

1. Compute an optimal solution b^* to the continuous PMR. [For $i = 1, \ldots, t$, let $b^*(a_i) = N_i + f_i$, where $N_i \epsilon Z^+$, $0 \leq f_i < 1$]
2. Let $D = \sum_{i=1}^{t} f_i$ (D must be an integer).
3. Let $f_{i1} \geq f_{i2}, \ldots, \geq f_{it}$. Set $b(a_{i1}) \leftarrow N_{i1} + 1, \ldots, b(a_{i_D}) \leftarrow N_{iD} + 1$, $b(a_{iD+1}) \leftarrow N_{iD+1}, \ldots, b(a_{it}) \leftarrow N_{it}$.

In order to analyze the performance of ROUND, we need some facts about the properties of optimal solutions to the continuous PMR.

LEMMA 5.1 *Let* b^* *be an optimal solution to the continuous PMR and assume that* $b^*(a_i) > 0$ *for* $i = 1, \ldots, t$. Then for all i, j, $w(b^*, a_i) = w(b^*, a_j)$.

PROOF. Consider the problem

minimize

$$w(b) = \sum_{j=1}^{n} p_j 2^{-b(S_j)}$$

subject to

$$\prod_{i=1}^{t} 2^{b(a_i)} = 2^B, \; b(a_i) \geq 0.$$

if $b^* = (b^*(a_1), \ldots, b^*(a_t))$ is an optimal solution, and $(b(a^*_1), \ldots, b^*(a_t))$ is an interior point of the constraint set, then, by the Lagrange multipliers theorem, there should exist a λ such that for $i = 1, \ldots, t$:

$$\frac{\partial\left[w(b^*) + \lambda \prod_{i=1}^{t} 2^{b^*(a_i)}\right]}{\partial(b^*(a_i))} = 0.$$

This can be shown to imply that for $i = 1, \ldots, t$, $w(b^*, a_i) = \lambda 2^B$, which implies the lemma. □

We shall compare the performances of BALANCE and ROUND on two examples. On the first one, ROUND yields a better result.

Example 1. Let (A, Q, P, B) be defined by
$A = \{a_1, a_2, a_3\}$; $Q = \{S_1, S_2, S_3, S_4\}$, where $S_1 = \{a_1\}$, $S_2 = \{a_2\}$, $S_3 = \{a_1, a_3\}$, $S_4 = \{a_2, a_3\}$; $p_1 = p_2 = 1/4 - \epsilon$, $p_3 = p_4 = 1/4 + \epsilon$ for some $1/48 > \epsilon > 0$; $B = 2$. First,

we consider the performance of BALANCE on this input: Initially, $b(a_i) = 0$ for $1 \le i \le 3$. $w(b, a_1) = w(b, a_2) = 1/2$, $w(b, a_3) = 1/2 + 2\epsilon$. Hence, at the first stage we set $b(a_3) = 1$. Now, $w(b, a_1) = w(b, a_2) = 3/8 - \epsilon/2$, $w(b, a_3) = 1/4 + \epsilon$. Hence at the second stage we set $b(a_1) = 1$. The resulting b is $b(a_1) = b(a_3) = 1$, $b(a_2) = 0$. $w(b) = 9/16 - 3\epsilon/4$.

To determine what the output of ROUND on the input above is, we first note that by Lemma 5.1 one can show that the optimal solution b^* to the corresponding continuous PMR is given by $b^*(a_3) = \log_2((1 + 4\epsilon)/(1 - 4\epsilon)) < \log_2(13/11) \cong 0.24$. $b^*(a_1) = b^*(a_2) = (2 - b^*(a_3))/2 > (2 - \log_2(13/11))/2 \cong 0.88$. Hence, the rounding will yield $b(a_1) = b(a_2) = 1$, $b(a_3) = 0$. $w(b) = .5$, and one can check that $\hat{b}$ is the optimal solution. When $\epsilon \to 0$, the relative error of the solution b, found by BALANCE, tends to 1/8.

On our second example, however, BALANCE performs much better than ROUND.

Example 2. $A = \{a_1, \ldots, a_{2n}, \bar{a}_1, \ldots, \bar{a}_{2n}\}$ for some n. $Q = \{S_1, \ldots, S_{2n+1}\}$ where $S_i = \{\bar{a}_i\}$ for $1 \le i \le 2n$, $S_{2n+1} = \{a_1, \ldots, a_{2n}\}$. $P = \{p_1, \ldots, p_{2n+1}\}$ is defined by $p_i = p$ for $i = 1, \ldots, 2n$ $p_{2n+1} = q$, where $2np + q = 1$ and $q = p \cdot 2^{(2n-1)/2-(2n+1)\epsilon}$ for some $1/6n > \epsilon > 0$; $B = 2n$. Again, using Lemma 5.1, one can check that the solution b^* given by $b^*(\bar{a}_i) = .5 + \epsilon$, $b^*(a_i) = .5 - \epsilon$ $(i = 1, \ldots, 2n)$ is an optimal solution to the continuous PMR. Using ROUND, we obtain a solution b defined by $b(a_i) = 0$, $b(\bar{a}_i) = 1$ $(i = 1, \ldots, 2n)$. $w(b) = 2np \cdot 2^{-1} + q2^{-0} = p(n + 2^{n-.5-(2n+1)\epsilon})$. When $\epsilon \to 0$, $w(b) \to p(n + 2^{n-.5})$.

When BALANCE is applied to the same input, we note that initially $w(b, \bar{a}_i) = p$, $w(b, a_i) = q = p \cdot 2^{(2n-1)/2-(2n+1)\epsilon}$, for $i = 1, \ldots, 2n$. The algorithm thus sets $b(a_i) \leftarrow 1$ for some i, and one can check that during the first n stages BALANCE sets at each stage $b(a_j) \leftarrow b(a_j) + 1$ for some j. After the nth stage, we have $w(b, \bar{a}_i) = p$, $w(b, a_i) = p \cdot 2^{-.5-(2n+1)\epsilon} < p$ $(i = 1, \ldots, 2n)$. Hence, during the last n stages, BALANCE will set $b(\bar{a}_i) \leftarrow b(\bar{a}_i) + 1$ for n different $\bar{a}_i$s. The resulting function b', although not optimal, satisfies $w(b') = np + np.2^{-1} + q2^{-n} = p(3n/2 + 2^{-.5-(2n+1)\epsilon})$. When $\epsilon \to 0$, $w(b') \to (3n/2 + 2^{-.5})p$, compared with the $(n + 2^{n-.5})p$ weight of the solution yielded by ROUND.

A detailed analysis of the performances of BALANCE and ROUND is probably not easy, and is beyond the scope of this paper. However, the examples above indicate that neither algorithm is strictly better than the other. A possible strategy, therefore, for a heuristic solution to the PMR, is to apply both of them on the given input, and then to choose the better result.

6. CONCLUDING REMARKS

We have shown that the PMR problem is NP-hard and, in fact, not even fully approximable, unless P = NP. Moreover, these results hold even in the case where all the query specifications that can be used in a given system are assumed to be equiprobable. Two heuristic algorithms for that problem were also presented, and were shown to be incomparable, in the sense that neither of them is strictly better than the other. It was also shown that one of these algorithms, namely ROUND, may produce arbitrarily large relative errors (Example 2 in

Section 5). An interesting question is: What is the worst-case relative error of BALANCE? If it is bounded by some small constant (which is not unlikely), then, from a practical point of view, the PMR can be considered to have a reasonably good algorithm.

ACKNOWLEDGMENT

The author wishes to thank the referees for many valuable suggestions.

REFERENCES

1. AHO, A., AND ULLMAN, J.D. Optimal partial match retrieval when fields are independently specified. *ACM Trans. Database Syst. 4* (1979), 168–179.
2. BOLOUR, A. Optimality properties of multiple-key hashing functions. *J. ACM 26*, 2 (April 1979), 196–210.
3. CHRISTOFIDES, N. Worst-case analysis of a new heuristic for the traveling salesman problem. Tech. Rep., Carnegie-Mellon Univ., 1976.
4. GAREY, M.R., AND JOHNSON, D.S. *Computers and Intractability—A Guide to the Theory of* NP *Completeness.* Freeman, San Francisco, 1979.
5. GAREY, M.R., AND JOHNSON, D.S. Strong NP completeness results: motivation, examples, and implications. *J. ACM 25*, 3 (July 1978), 499–508.
6. HOROWITZ, E., AND SAHNI, S. Exact and approximate alogrithms for scheduling nonidentical processors. *J. ACM 23*, 2 (April 1976), 317–327.
7. IBARRA, O.H., AND KIM, C.E. Fast approximation algorithms for the knapsack and sum of subsets problems. *J. ACM 22*, 4 (Oct. 1975), 463–468.
8. JOHNSON, D., DEMERS, A., ULLMAN, J.D., GAREY, M.R., AND GRAHAM, R. Worst-case performance bounds for simple one dimensional packing algorithms. *SIAM J. Comput. 3* (1974), 299–325.
9. KARP, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and W. Thatcher, Eds., Plenum Press, New York, 1972, pp. 85–103.
10. MORAN, S. General approximation algorithms for some arithmetical combinatorial problems. *Theor. Comput. Sci. 14* (1981), 289–303.
11. PAZ, A., AND MORAN, S. Nondeterministic polynomial optimization problems and their approximation. *Theor. Comput. Sci. 15* (1981), 251–277.
12. RIVEST, R. Partial match retrieval algorithms. *SIAM J. Comput. 5* (1976), 19–50.
13. ROSENKRANTZ, D., STEARNS, R., AND LEWIS, P. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput. 6* (1977), 563–581.
14. ROTHNIE, J., AND LOZANO, T. Attribute based file organization in a paged memory environment. *Commun. ACM 17*, 2 (Feb. 1974), 63–69.
15. ULLMAN, J.D. *Principles of Database Systems*, Computer Science Press, Potomac, Md., 1980.

Received December 1981; revised February 1983; accepted March 1983