Available online at www.sciencedirect.com
SCIENCE DIRECT®
Artificial Intelligence 164 (2005) 1–22
Artificial Intelligence
www.elsevier.com/locate/artint

A formal analysis of why heuristic functions work

B. John Oommen [a,*,1], Luis G. Rueda [b,2]

[a] *Senior Member, IEEE. School of Computer Science, Carleton University, 1125 Colonel By Dr., Ottawa, ON, K1S 5B6, Canada*
[b] *School of Computer Science, University of Windsor, 401 Sunset Ave., Windsor, ON, N9B 3P4, Canada*

Received 10 May 2001

Abstract

Many optimization problems in computer science have been proven to be NP-hard, and it is unlikely that polynomial-time algorithms that solve these problems exist unless P = NP. Alternatively, they are solved using *heuristics algorithms*, which provide a *sub-optimal* solution that, hopefully, is arbitrarily close to the *optimal*. Such problems are found in a wide range of applications, including artificial intelligence, game theory, graph partitioning, database query optimization, etc. Consider a heuristic algorithm, A. Suppose that A could invoke one of two possible heuristic functions. The question of determining which heuristic function is superior, has typically demanded a yes/no answer—one which is often substantiated by empirical evidence. In this paper, by using Pattern Classification Techniques (PCT), we propose a formal, rigorous theoretical model that provides a *stochastic answer* to this problem. We prove that given a heuristic algorithm, A, that could utilize either of two heuristic functions H_1 or H_2 used to find the solution to a particular problem, if the accuracy of evaluating the cost of the *optimal* solution by using H_1 is greater than the accuracy of evaluating the cost using H_2, then H_1 has a higher probability than H_2 of leading to the optimal solution. *This unproven conjecture has been the basis for designing numerous algorithms such as the A* algorithm, and its variants*. Apart from formally proving the result, we also address the corresponding database query optimization problem that has been open for at least two decades. To validate our proofs, we report empirical results on database query optimization techniques involving a few well-known histogram estimation methods.
© 2005 Elsevier B.V. All rights reserved.

* Corresponding author.
E-mail addresses: oommen@scs.carleton.ca (B.J. Oommen), lrueda@scs.carleton.ca (L.G. Rueda).

[1] Partially supported by NSERC, the Natural Science and Engineering Research Council of Canada. Fellow of the IEEE.

[2] This work was partially supported by Departamento de Informática, Universidad Nacional de San Juan, Argentina, and by NSERC. Member of the IEEE.

0004-3702/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.artint.2002.02.001

Keywords: A* algorithms; Heuristic algorithms; Pattern recognition; Optimization

1. Introduction

1.1. Overview

The area of computer science has still quite a few open, unsolved problems. In this paper, we are concerned with one such problems, namely that of using *heuristics* to solve optimization problems.

Any arbitrary optimization problem[3] is typically defined in terms of instances which are drawn from a (finite) set, $\mathcal{X}$, an objective function, and some feasibility functions. The aim is to find an (and hopefully, the unique) instance of $\mathcal{X}$, which leads to the maximum (or the minimum) value of the objective function subject to the feasibility constraints. A formal definition of an optimization problem can be found in [10]. But to be more specific, consider the well-known Traveling Salesman Problem (TSP), in which the cities are numbered from 1 to n, and the salesman starts from city 1, visits every other city once, and returns to city 1. An instance of $\mathcal{X}$ is a permutation of the cities, for example, 14325, if we are considering a world consisting of five cities. The objective function for that instance, $f(14325)$ is obtained by performing the summation of the inter-city distances: $1 \rightarrow 4$, $4 \rightarrow 3$, $3 \rightarrow 2$, $2 \rightarrow 5$, and $5 \rightarrow 1$. The optimal solution is the instance that minimizes the value of f.

A *heuristic algorithm* is an algorithm that attempts to find a certain instance of $\mathcal{X}$ that maximizes f (or the *profit*) by iteratively invoking a *heuristic* function. The instance that maximizes f will be the *optimal solution*[4] to the optimization problem. A *heuristic* is a method that performs one or more modifications to a given solution or instance, in order to obtain a different solution which is either superior, or which leads to a superior solution. The heuristic, in turn, invokes a heuristic function, which estimates (or measures) the cost of the solution at the particular state in the search process. This is the context in which we use these terms.

Many heuristic algorithms and heuristic functions have been reported in the literature, where the former include the *alpha-beta search* [11], *backtracking*, *hill-climbing* [10], *simulated annealing* [1], *genetic algorithms* [13], *tabu search* [7], *learning automata* [15], etc. The issue of how heuristic functions are used in such heuristic algorithms in searching, game playing, etc., can be found in [16,24] and is, indeed, an enormous field of study in itself. This question is not addressed here.

To clarify issues, let us consider the classical n-puzzle problem [16]. This problem consists of a square board containing n square tiles and an empty position called the "blank". The aim is to rearrange the tiles from some pre-defined (usually random) initial configuration into a pre-determined goal configuration, by sliding any tile adjacent to the blank into

[3] Every optimization problem can also be formulated as a decision problem [6].

[4] We use the term "solution" to refer to an element $x \in \mathcal{X}$, and the term "profit" to refer to the value of $f(x)$. In minimization problems, $f(.)$ will be a cost function.

the blank position. A heuristic algorithm solves this problem by examining, using a heuristic function, *some* of the possible *valid* movements. Viewed from the perspective of the underlying state graph, the possible states encountered at the next level form the children nodes of the current node in the search structure. Other variants of heuristic algorithms involve the examination of lower levels as well. The breadth-first search and depth-first search schemes are examples of heuristic algorithms, useful in any such problem solving strategy. An example of a heuristic function, however, is the measurement (or estimate) of the number of tiles that are out of place. Another measure is the sum of the depth of the node and the number of tiles that are out of place.

One of the better-known solutions to the n-puzzle problem is the A* algorithm. This algorithm is a graph search algorithm that is used to find the path of minimum cost between two nodes, the start node and the goal node. The A* maintains a tree which stores the paths that are already explored. Using these paths, a measure, f, of the potential advantage of choosing each path is calculated. The value of f, which is the cost of traversing the graph between two nodes, can be calculated by using different heuristic functions. A heuristic is said to be *admissible*, and the A* converges to the correct result, if the heuristic function is an upper bound of the true cost from all nodes to the goal node.

In general, for any arbitrary problem, the question of how useful a heuristic function is, in determining the cost of traversing from one node to another, has no known analytic solution—it has traditionally been empirically analyzed. In this paper, we present a formal analysis that provides a *stochastically positive* answer to the question of comparing the relative advantages of potential heuristic functions.

The A* algorithm and its variants (like the A+ algorithm) have also been successfully applied to other problems, such as object recognition using deformable templates [16,26,28]. Various solutions to optimization problems using different heuristic functions are found in [28]; we shall use this paper, [28], to highlight the difference between the heuristic algorithms, and the effect of the *same* algorithm using various potential heuristic functions. The authors of [28] address the problem of tracking roads in satellite images using the *twenty-question* search paradigm, and the A+ algorithm, a "cousin" of the A* algorithm. Using these algorithms the roads can be represented in terms of straight-line segments. The various paths are expanded by the application of an ensemble of heuristic functions. One such heuristic function is the one based on the conditional entropy measurements of the branches, which are used to choose the most "promising" path. While the paper discusses other heuristic functions, the question of how one can compare the solutions obtained using the various heuristic functions is achieved by comparing the empirical simulation results. We hope that our formal analysis can be a tool to achieve a more rigorous comparison of these heuristic functions in [28], and other similar scenarios.[5]

The tools we propose to use are drawn from the well-established theory of Pattern Recognition (PR) [5,27]—a prominent field of machine intelligence. Broadly speaking, PR involves decision-making, based on *a priori* and learned knowledge of the classes and objects being recognized. More specifically, the system learns information about the features

[5] The model presented here has some limitations when investigating the quality of solutions yielded by an A*-like algorithm. These limitations will be discussed in a later sub-section.

of a set of classes. Subsequently, given an object of unknown identity, and this information, the system attempts to recognize the unknown object as belonging to one of the known classes with some arbitrary accuracy. Necessarily, our overview of PR is brief!

There are many applications of PR, including face and speech recognition, fingerprint identification, character recognition, medical diagnosis, etc. In each of these applications, the information about the classes can be *structural* or *statistical*. In the former, we deal with the field of structural and syntactic pattern recognition, and in the latter, with the field of statistical pattern recognition. Furthermore, in the latter, the statistical information, or *features*, about the classes is represented by random vectors. The procedure of obtaining the features consists of mapping the feature values of each sample to a vector. Feature values, for example, can be the width or the height of a figure, the value of a pixel of an image, etc. Statistical pattern recognition can also be subdivided into two well-defined approaches, *parametric* and *non-parametric*. In the former, the random vectors have a known probability distribution, e.g., normal (or Gaussian), exponential, multinomial, etc. No such model is assumed in a non-parametric case.

Although we are aware of the use of PR principles in real life scenarios, we are not aware of any previous results in which PR principles have been used to solve a theoretical unsolved problem in a completely different field.

Our result can be crystallized as follows: Given two heuristic functions, the question of determining which is superior, has typically demanded a yes/no answer which is often substantiated based on empirical evidence. We have solved the problem of deciding on the *superior* heuristic function by using PR techniques. It should be mentioned that there are numerous well-known techniques that have been utilized in the context of pattern classification, such as hypothesis testing, bootstrap methods, Neyman–Pearson methods, etc. A good reference for such methods can be found in [21]. However, the results derived in this paper essentially use the methods that have been traditionally applied to optimal Bayesian Classification, as described in the statistical pattern recognition literature [4]. Using these principles, we prove the following assertion: Given two heuristic functions, H_1 and H_2, used by a heuristic algorithm in finding a solution to a particular problem, if the accuracy in obtaining the *optimal* solution by using H_1 is greater than that of using H_2, then H_1 has a higher probability of leading to the optimal solution than H_2. To the best of our knowledge, this is an open problem. *However, this unproven conjecture has been the basis for designing numerous algorithms such as the A* algorithm, and its variants, in searching, game playing, and numerous other applications* [16,24,25,28].

Our strategy for achieving this analysis is as follows. The first task is to model the cost of the solution. Since the optimal "true" cost is unknown, we represent *it* in terms of its estimate, as estimated using the heuristic function. Observe that since the latter is inaccurate, this "cost" is represented in terms of a random variable. Note that by "cost", we do not mean the cost of the search process involved in determining the optimal solution, but rather the cost of the optimal solution, as estimated by the heuristic function. This difference is crucial.

Now that the modelling of the heuristic function is in place, the question of quantifying the quality of any heuristic function has to be considered. Informally speaking, we can say that this paper concerns this "heuristic-function quality assessment" problem, which is addressed, in turn, by viewing it as a pattern recognition problem. We solve *this* pat-

tern recognition problem by considering two independent random variables, the first for the optimal solution and the second for the sub-optimal, both of them being pursued by a heuristic function, H_1. We use a reasonable model for the accuracy of the heuristic function, in which the error of H_1 is a doubly-exponential random variable.[6] This distribution, which as we shall presently see, is used to approximate the Gaussian distribution, is typically used in reliability and failure models, and hence is reasonable in this scenario. In our model, the accuracy of the heuristic function is related to the variance of the random variable used to represent it. The analysis for the Gaussian distribution follows thereafter.

If we now consider another heuristic function, H_2, whose variance is greater than that of H_1, and whose mean is the same as that of H_1, we have a model by which the efficiency of heuristic functions can be compared. Indeed, using *this* model, we have theoretically proven that H_1 is more likely to succeed in obtaining the optimal solution than H_2. For this model, we have also proved the uniqueness of the result, and the conditions for which both heuristic functions lead to coincident probabilities of success.

The doubly exponential distribution is actually meant to be an approximation of the Gaussian distribution, typically used to model errors. However, the algebraic analysis for Gaussian distributions is impossible as there is no closed-form expression for integrating its probability density function. Consequently, we have extended the analysis for the doubly exponential distribution to formulate a reasonable analysis for the Gaussian distribution using numerical integration. By means of this analysis, we have corroborated the validity of our hypothesis for Gaussian distributions also.

We also provide empirical results on using a few histogram-like estimation methods in database query optimization, which demonstrate the validity of our theoretical analysis.

1.2. Applications

There are many heuristic algorithms that can be used to solve a wide variety of NP-hard problems. Such problems can be found in a wide range of applications spanning the whole spectrum of artificial intelligence, and include game playing and game theory, graph theory, database query optimization, networking, computational geometry, number theoretic problems, parallel processing, etc. The results presented in this paper are applicable to any heuristic algorithm that uses different heuristic functions to solve a particular problem. In this introductory section, we just describe a few of them.

In the area of database query optimization, when more than two tables have to be joined, intermediate join operations are performed to ultimately obtain the final relation. As a result, the same query can be performed by means of different intermediate (join) operations. A simple sequence of join operations that leads to the same final result is called a *query evaluation plan* (QEP). Each QEP has associated an internal cost, which depends on the number of operations performed in the intermediate joins. The problem of choosing the best QEP is a combinatorially explosive optimization problem. This problem is currently

[6] The reasoning used in this paper assumes that the errors are on either side of the true value. However, we believe that if the distribution is one-sided, similar arguments will be true as long as the distribution is not "heavily-tailed". We are grateful to the anonymous referee who brought this to our attention.

solved by estimating the query result sizes of the intermediate relations and selecting the most efficient QEP.

Since the analysis of selecting the best QEP must be done in "real" time, it is not possible to inspect the real data in this phase. Consequently, query result sizes are usually estimated using statistical information about the structures and the data maintained in the database catalogue. This information is used to approximate the distribution of the attribute values in a particular relation. Hence the problem of selecting the best QEP depends on how well that distribution is approximated.

In [8], it has been shown that errors in query result size estimates may increase exponentially with the number of joins. Since current databases and the associated queries increase in complexity, numerous efforts have being made to devise more efficient techniques that solve the query optimization problem.

Many techniques have been proposed to estimate query result sizes, including histograms, sampling, and parametric techniques [9,12,14,22]. Histograms are the most commonly used form of statistical information. They are incorporated in most of the commercial database systems such as Oracle, Microsoft SQL Server, Teradata, and DB2, which mainly use the Equi-depth histogram. The prominent models of histograms known in the literature are: *Equi-width* [2,9], *Equi-depth* [14,22], the *Rectangular Attribute Cardinality Map* (*R-ACM*) [18], the *Trapezoidal Attribute Cardinality Map* (*T-ACM*) [19], and the *V-Optimal Histograms* [8,23].

In this scenario, the heuristic algorithm is the actual algorithm that uses a histogram as the heuristic function, and obtains an optimal (or a sub-optimal) QEP. The heuristic function used by this algorithm is the *actual histogram* that approximates the distribution of the attribute values of the relevant tables. Thus, in our model (and using our terminology), Equi-width, Equi-depth, the R-ACM and the T-ACM are the heuristic functions.

Other areas in which our model can be used to answer open questions are in the fields of *game theory* and *game playing* [25]. In game playing, the most widely used structure used to analyze the best possible move and strategy is a *game tree*, whose root node represents the initial status of the board. All possible moves of the first player are the edges from the root to the first level, the edges of each child represent all possible moves of the second player, the opponent. Continuing in the same fashion, the game is played (or rather plans executed) until one of the players wins. The aim is to optimize the moves of the first player based on searching *all* the branches of the tree until the leaves, and perform the best move based on maximizing the reward of the first player and minimizing that of the second one.

There are many techniques used to optimize the moves of the first the player. One of them is the *minimax search algorithm*, which searches over a fixed number of levels of the entire tree, and finds the best moves at each node. This exhaustive search procedure has a complexity that grows exponentially with the number of nodes of the tree. A more efficient mechanism is the *alpha-beta search* algorithm [11], a heuristic that significantly reduces the number of nodes explored. Both of these assume that the heuristic function that they use, which typically evaluates the position of the board viewed from the perspective of the first player, is advantageous in determining a superior strategy. This is the question that we address in this paper. The model presented in this paper has important consequences in choosing such a heuristic function. Such a heuristic function could be, for example, the cost of a path from the current state to a goal state, which unfortunately is not exactly known,

but is estimated. The search scheme, such as the alpha-beta search and the minimax search algorithm, uses this heuristic function to search for a, hopefully, *optimal* path in the game tree.

Another application of our result is in graph theory, for example, in solving the uniform graph partitioning problem. Given a complete graph on $2n$ vertices, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, along with a cost function $f : \mathcal{E} \to \mathbb{Z}^+ \cup \{0\}$, the aim is to find a partition whose sum of costs of the individual subsets is minimized. This problem is also known to be NP-hard, and has several applications especially in VLSI design, hydrology, networks, etc. Many heuristic algorithms have been proposed to solve this problem, including simulated annealing, genetic algorithms, learning automata, etc. [10,20]. When considering a particular heuristic algorithm, we can incorporate different heuristic functions to approximate the sum of costs of the individual subsets of a particular partitioning. It is intuitive that a more accurate heuristic function is more likely to succeed in finding the optimal solution. However, this is not what happens in all cases. We rather provide a stochastic answer to this question. By means of a rigorous theoretical analysis, we prove that a particular heuristic function, which provides more accurate approximations for the sum of costs of the individual subsets, is more likely to obtain the minimal cost for a partitioning, than a *less accurate* heuristic function.

1.3. Problem statement

In this paper, we propose a theoretical model that solves this fundamental open problem in computer science, namely that of relating heuristic functions with solution optimality, using the principles of the theory of *pattern classification*. This problem has been (to our knowledge) open. In particular, the corresponding database query optimization problem has been unsolved for more than two decades.

More specifically, we prove the following: Given a heuristic algorithm, A, that invokes two heuristic functions, H_1 and H_2, used in a decision problem, if the accuracy in approximating the optimal solution by using H_1 is greater than that of using H_2, then H_1 has a higher probability of leading to the optimal solution than H_2.

The importance of the results of this paper is that we show that the answer to the accuracy/optimality question is "stochastically positive". In other words, we prove that although a superior heuristic function may not always yield a better solution, the probability that the superior heuristic function yields an optimal solution exceeds the probability that an inferior heuristic function yields an optimal solution. This paper thus justifies and gives a formal rigorous basis for why heuristic functions work.

We analytically prove that under the well-acclaimed models of inaccuracy, the better the accuracy of a heuristic function, the greater the probability of it choosing the optimal solution. We have also provided some empirical results related to the field of database query optimization. These results show the superiority of the R-ACM over the traditional histogram estimation methods, the Equi-width and the Equi-depth. The empirical results obtained by testing these properties for many of the above histogram methods in random databases show that the R-ACM is significantly superior to both the Equi-width and the Equi-depth schemes.

1.4. Restrictions of our model

As mentioned above, this paper addresses the problem of quantifying the quality of a heuristic function, and it achieves this by posing the problem in a fairly general framework. However, for the results to be applicable for a particular application domain[7] which uses a specific search strategy such as the A^* algorithm, the logistics of the search process itself have to be considered.

Informally speaking, the main result of our paper proves the following: Given two heuristic functions evaluating the same "cost", a search mechanism utilizing these functions will converge (with a higher probability) to a superior solution, when it utilizes a function with a lesser variance. However, comparing the performance of heuristic functions in the search process initiated by A^* is a more complicated issue. The reason for this can be argued as follows. In each iteration, A^* computes the values of the heuristic function (say, "$f(.)$") for all candidate nodes (the OPEN list), which represent how promising they are. A^* then selects the one with the highest value of $f(.)$, generates *its* children, computes *their* values of $f(.)$, and inserts them into the OPEN list. For an algorithm like A^*, the most we can claim is that it is more expedient to use a heuristic function which better estimates the "cost" than one which estimates it poorly. The question of how the nodes in the OPEN list lead to solutions, is really a problem-dependent question which we cannot answer here. We intend to study this problem in the database query optimization domain mentioned later, by incorporating a *search* strategy to search the set of QEPs whose costs are estimated by the various histogram methods. Note that this does not invalidate the query-optimization results presented in this paper, because, in our simulations, we exhaustively search the QEP space without using any intelligent search strategy like A^*.

2. Heuristic function accuracy vs. optimality

Consider a heuristic algorithm, A, that invokes either of two heuristic functions, H_1 and H_2. The probability of correctly estimating a cost value of a particular solution by H_1 and that of estimating a cost value by H_2 are represented by two independent random variables. In our model, we assume that these two heuristic functions are independent, and thus, the value obtained by one heuristic function should not affect the value obtained by the second.

For the analysis done below, we work with two models for the error function: the doubly exponential distribution and the normal distribution. In the former, the probability of obtaining a value that deviates from the mean (or true value) falls exponentially as a function of the deviation. The exponential distribution is more typical in reliability analysis and in failure models, and in this particular domain, the question is one of evaluating how reliable the quality of a solution is, if only an estimate of its performance is available. More importantly, it is used as an approximation to the Gaussian distribution for reasons which will be clarified momentarily. The Gaussian model is much more difficult to analyze, since

[7] We are grateful to the anonymous referee who brought this limitation to our attention.

there is no closed-form algebraic expression for integrating the probability density function. However, a formal computational proof is included, which confirms our hypothesis.

2.1. Analysis using exponential distributions

A random variable, X, is said to be *doubly exponentially* distributed with parameter λ if the density function is given by:

$$f_X(x) = \frac{1}{2}\lambda e^{-\lambda|x-c|}, \quad -\infty < x < \infty. \tag{1}$$

If X is a doubly exponential random variable, by elementary integration and straightforward algebraic steps, it can be shown that:

$$\mathrm{E}[X] = c, \quad \text{and} \tag{2}$$

$$\mathrm{Var}[X] = \frac{2}{\lambda^2}. \tag{3}$$

Without loss of generality, if the mean of the cost of the optimal solution is c_1, by shifting the origin by c_1, we can work with the assumption that the cost of the best solution is 0, which is the mean of these two random variables. The cost of the second best solution is given by another two random variables (one for H_1 and the other one for H_2) whose mean, $c_2 > 0$, is the same for both variables. An example will help to clarify this.

Example 1. Suppose that using H_1 leads to the optimal cost with a probability represented by a doubly exponential random variable, $X_1^{(opt)}$, whose mean is 0 and $\lambda_1 = 0.4$. This heuristic function also leads to another sub-optimal cost according to $X_1^{(subopt)}$ whose mean is 8 and $\lambda_1 = 0.4$.

H_2 is another heuristic function using which the optimal cost is chosen with a probability distribution given by $X_2^{(opt)}$ whose parameters are $c_1 = 0$ and $\lambda_2 = 0.2$. It leads to the second sub-optimal cost value with a probability density given by $X_2^{(subopt)}$ whose parameters are $c_2 = 8$ and $\lambda_2 = 0.2$.

The fact that $2/\lambda_1^2 < 2/\lambda_2^2$ signifies that the probability of using H_1 could lead to a sub-optimal cost is smaller than the probability of using H_2 leading to a sub-optimal cost. This scenario is depicted in Fig. 1, and is formalized presently.

The result depicted above is formalized in the following theorem, *which is the first primary result of this paper, and answers the open question referred to above*. The theorem is formulated in terms of the probabilities that the two heuristic functions lead to the wrong decision, which we show is inherently related to the probability that these heuristic functions lead to the convergence to the *sub-optimal* solutions. The formulation of the result and the proof utilize techniques typically foreign to database theory, game theory, artificial intelligence, or for that matter any computer science area in which this approach can be applied. They belong to the theory of PR.

The second theorem, extends the results of the first, and shows how the results can also be geometrically interpreted.

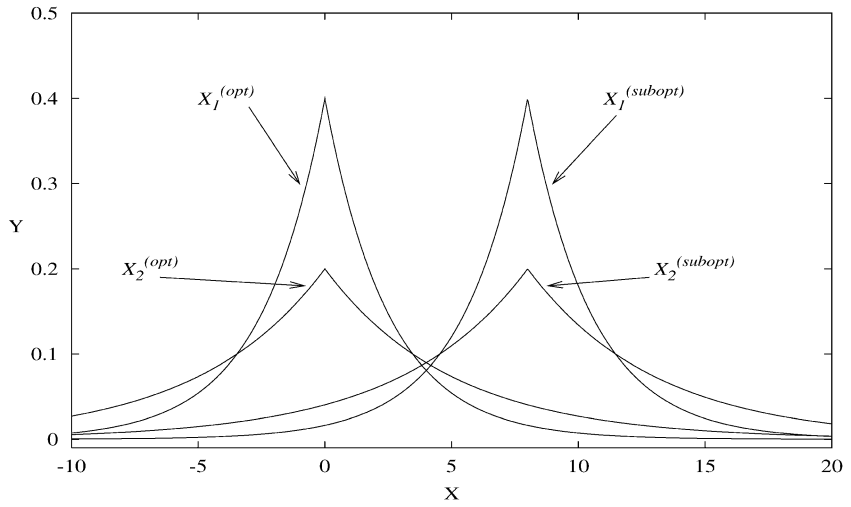

Fig. 1. An example of doubly exponential distributions for the random variables $X_1^{(opt)}$, $X_2^{(opt)}$, $X_1^{(subopt)}$ and $X_2^{(subopt)}$, whose parameters are $\lambda_1 = 0.4$ and $\lambda_2 = 0.2$.

Theorem 1. *Suppose that A is a heuristic algorithm that can potentially utilize either of two heuristic functions, H_1 and H_2. Let*:

- *X_1 and X_2 be two doubly exponential random variables that represent the estimated costs of the optimal solutions obtained by using H_1 and H_2 respectively.*
- *X_1' and X_2' be two other doubly exponential random variables representing the estimated costs of non-optimal solutions obtained by using H_1 and H_2 respectively.*
- *$0 = \mathrm{E}[X_1] = \mathrm{E}[X_2] \leqslant \mathrm{E}[X_1'] = \mathrm{E}[X_2'] = c$.*
- *p_1 and p_2 be the probabilities that H_1 and H_2 respectively lead to the wrong decision.*

Then,

$$\textit{if } \mathrm{Var}[X_1] = \mathrm{Var}[X_1'] = \frac{2}{\lambda_1^2} \leqslant \frac{2}{\lambda_2^2} = \mathrm{Var}[X_2] = \mathrm{Var}[X_2'], \quad p_1 \leqslant p_2.$$

Proof. Consider a particular cost, x. The probability that x leads to a wrong decision when A uses H_1 is that of incorrectly classifying x as being obtained from the non-optimal solution. This is, indeed, the error in classification, and is the area under the curve of the pdf function of X_1' or the cumulative probability of x under the pdf of H_1 when it refers to the sub-optimal solution. Because of the discontinuity of the doubly exponential function at c, this area is decomposed into the following two integrals:

$$\begin{aligned} I_{11} &= \int_{-\infty}^{x} \frac{1}{2}\lambda_1 \mathrm{e}^{\lambda_1(u-c)}\,\mathrm{d}u, \quad \text{if } x \leqslant c, \quad \text{and} \\ I_{12} &= \int_{-\infty}^{c} \frac{1}{2}\lambda_1 \mathrm{e}^{\lambda_1(u-c)}\,\mathrm{d}u + \int_{c}^{x} \frac{1}{2}\lambda_1 \mathrm{e}^{-\lambda_1(u-c)}\,\mathrm{d}u \quad \text{if } x > c. \end{aligned} \tag{4}$$

Solving the integrals, (4) results in:

$$\begin{aligned} I_{11} &= \frac{1}{2}e^{\lambda_1(x-c)} - \lim_{u\to-\infty}\frac{1}{2}e^{-\lambda_1(u-c)} = \frac{1}{2}e^{-\lambda_1(x-c)}, \quad \text{and} \\ I_{12} &= \lim_{u\to-\infty}\frac{1}{2}e^{-\lambda_1(-u+c)} + \frac{1}{2} - \frac{1}{2}e^{-\lambda_1(x-c)} + \frac{1}{2} = 1 - \frac{1}{2}e^{-\lambda_1(x-c)}. \end{aligned} \tag{5}$$

The probability that using H_1 leads to the wrong decision for *all* the values of x is the following function of λ_1 and c:

$$p_1 = I(\lambda_1, c) = \int_{-\infty}^{0} I_{11}\frac{1}{2}\lambda_1 e^{\lambda_1 x}\,dx + \int_{0}^{c} I_{11}\frac{1}{2}\lambda_1 e^{-\lambda_1 x}\,dx + \int_{c}^{\infty} I_{12}\frac{1}{2}\lambda_1 e^{-\lambda_1 x}\,dx, \tag{6}$$

which, after applying the distributive law and substituting the values of I_{11} and I_{12}, can be written as:

$$\int_{-\infty}^{0}\frac{\lambda_1}{4}e^{2\lambda_1 x-\lambda_1 c}\,dx - \int_{0}^{c}\frac{\lambda_1}{4}e^{-\lambda_1 c}\,dx + \int_{c}^{\infty}\left[\frac{\lambda_1}{2}e^{-\lambda_1 x} - \frac{\lambda_1}{4}e^{-2\lambda_1 x+\lambda_1 c}\right]dx. \tag{7}$$

After solving the integrals, (7) is transformed into:

$$\frac{1}{8}e^{-\lambda_1 c} + \frac{1}{4}\lambda_1 c e^{-\lambda_1 c} + \frac{3}{8}e^{-\lambda_1 c} = \frac{1}{2}e^{-\lambda_1 c} + \frac{1}{4}\lambda_1 c e^{-\lambda_1 c}. \tag{8}$$

Similarly, we do the same analysis for p_2, which is a function of λ_2 and c:

$$p_2 = I(\lambda_2, c) = \frac{1}{2}e^{-\lambda_2 c} + \frac{1}{4}\lambda_2 c e^{-\lambda_2 c}. \tag{9}$$

We have to prove that:

$$p_1 = \frac{1}{2}e^{-\lambda_1 c} + \frac{1}{4}\lambda_1 c e^{-\lambda_1 c} \leqslant \frac{1}{2}e^{-\lambda_2 c} + \frac{1}{4}\lambda_2 c e^{-\lambda_2 c} = p_2. \tag{10}$$

Multiplying both sides by 2, and substituting $\lambda_1 c$ for α_1 and $\lambda_2 c$ for α_2, (10) can be written as follows:

$$e^{-\alpha_1} + \frac{1}{2}\alpha_1 e^{-\alpha_1} \leqslant e^{-\alpha_2} + \frac{1}{2}\alpha_2 e^{-\alpha_2}. \tag{11}$$

Substituting α_2 for $k\alpha_1$, $\alpha_1 \geqslant 0$ and $0 < k \leqslant 1$, (11) results in:

$$q_1 = e^{-\alpha_1} + \frac{1}{2}\alpha_1 e^{-\alpha_1} \leqslant e^{-k\alpha_1} + \frac{1}{2}k\alpha_1 e^{-k\alpha_1} = q_2. \tag{12}$$

We now prove that $q_1 - q_2 \leqslant 0$. After applying natural logarithm to both sides of (12) and some algebraic manipulations, $q_1 - q_2 \leqslant 0$ implies:

$$F(\alpha_1, k) = k\alpha_1 - \alpha_1 + \ln\left(1 + \frac{1}{2}\alpha_1\right) - \ln\left(1 + \frac{1}{2}k\alpha_1\right) \leqslant 0. \tag{13}$$

To prove that $F(\alpha_1, k) \leqslant 0$, we use the fact that $\ln x \leqslant x - 1$. Hence, we have:

$$F(\alpha_1, k) = \alpha_1(k-1) + \ln\left(\frac{1+\frac{1}{2}\alpha_1}{1+\frac{1}{2}k\alpha_1}\right) \tag{14}$$

$$\leqslant \alpha_1(k-1) + \frac{1+\frac{1}{2}\alpha_1}{1+\frac{1}{2}k\alpha_1} - 1 \tag{15}$$

$$= \alpha_1(k-1) + \frac{\alpha_1 - k\alpha_1}{2+k\alpha_1} \tag{16}$$

$$= \frac{k\alpha_1 + k^2\alpha_1^2 - \alpha_1 - k\alpha_1^2}{2+k\alpha_1} \tag{17}$$

$$= \frac{\alpha_1(k-1)(k\alpha_1+1)}{2+k\alpha_1} \leqslant 0, \tag{18}$$

because:

(i) $0 < k \leqslant 1$ and $\alpha_1 \geqslant 0 \Rightarrow \alpha_1(k-1) \leqslant 0$ and $k\alpha_1 + 1 > 0$. Hence $\alpha_1(k-1)(k\alpha_1+1) \leqslant 0$, and
(ii) $0 < k \leqslant 1$ and $\alpha_1 \geqslant 0 \Rightarrow 0 < k\alpha_1 \leqslant \alpha_1 \Rightarrow k\alpha_1 + 2 > 2 > 0$.

Hence the theorem. □

The above theorem can be viewed as a "sufficiency result". In other words, we have shown that $q_1 - q_2 \leqslant 0$ or that $p_1 \leqslant p_2$. We now show a "necessity result" stated as a uniqueness result. This result states that the function $p_1 \leqslant p_2$ has its equality ONLY at the boundary condition where the two distributions are exactly identical.

To prove the necessity result, we consider $q_2 - q_1$ which, derived from (12), can be written, as a function of α_1 and k, as:

$$G(\alpha_1, k) = \mathrm{e}^{-k\alpha_1} + \frac{1}{2}k\alpha_1\mathrm{e}^{-k\alpha_1} - \mathrm{e}^{-\alpha_1} - \frac{1}{2}\alpha_1\mathrm{e}^{-\alpha_1}. \tag{19}$$

By examining its partial derivatives, we shall show that there are two solutions for equality. Furthermore, when $\alpha_1 \geqslant 0$ and $0 < k \leqslant 1$, we shall see that for a given k, there is only one solution, namely $\alpha_1 = 0$ and k, $0 < k \leqslant 1$, proving the uniqueness.

Theorem 2. *Suppose that* $\alpha_1 \geqslant 0$, $0 < k \leqslant 1$. *Let* $G(\alpha_1, k)$ *be*:

$$G(\alpha_1, k) = \mathrm{e}^{-k\alpha_1} + \frac{1}{2}k\alpha_1\mathrm{e}^{-k\alpha_1} - \mathrm{e}^{-\alpha_1} - \frac{1}{2}\alpha_1\mathrm{e}^{-\alpha_1}. \tag{20}$$

Then $G(\alpha_1, k) \geqslant 0$, *and there are exactly two solutions for* $G(\alpha_1, k) = 0$, *being*: $\{\alpha_1 = -1, k = 1\}$ *and* $\{\alpha_1 = 0, k\}$.

Proof. We must prove that, as defined in the theorem statement, $G(\alpha_1, k) \geqslant 0$.

We shall prove that this is satisfied by determining the local minima for $G(.,.)$, where $\alpha_1 \geqslant 0$ and $0 < k \leqslant 1$. We first find the partial derivatives of (19) with respect to α_1 and k:

$$\frac{\partial G}{\partial \alpha_1} = -\frac{1}{2}k\mathrm{e}^{-k\alpha_1} - \frac{1}{2}k^2\alpha_1\mathrm{e}^{-k\alpha_1} + \frac{1}{2}\mathrm{e}^{-\alpha_1} + \frac{1}{2}\alpha_1\mathrm{e}^{-\alpha_1} = 0, \quad \text{and} \tag{21}$$

$$\frac{\partial G}{\partial k} = -\frac{1}{2}\alpha_1 e^{-k\alpha_1} - \frac{1}{2}k\alpha_1^2 e^{-k\alpha_1} = 0. \tag{22}$$

We now solve (21) and (22) for α_1 and k. Eq. (22) can be written as follows:

$$-\frac{1}{2}\alpha_1 e^{-k\alpha_1} = \frac{1}{2}k\alpha_1^2 e^{-k\alpha_1}, \tag{23}$$

which, after canceling some terms results in $k\alpha_1^2 + \alpha_1 = 0$. Solving this equation for α_1, we have: $\alpha_1 = -\frac{1}{k}$ and $\alpha_1 = 0$. Substituting $\alpha_1 = -\frac{1}{k}$ in (21), and canceling some terms, we obtain:

$$\frac{1}{2}e^{-\alpha_1} + \frac{1}{2}\alpha_1 e^{-\alpha_1} = 0, \tag{24}$$

which results in the solution to be $\alpha_1 = -1$, and consequently, $k = 1$.

The second root, $\alpha_1 = 0$, indicates that the minimum is achieved for any value of k.

We have thus found two solutions for (21) and (22), $\{\alpha_1 = 0, k\}$ and $\{\alpha_1 = -1, k = 1\}$. Since $\alpha_1 \geqslant 0$, it means that α_1 can have at least a value of 0, and hence the local minima is in $\{\alpha_1 = 0, k\}$. Substituting these two values in G, we see that $G(\alpha_1, k) = 0$, which is the minimum. Therefore, $G(\alpha_1, k) \geqslant 0$ for $\alpha_1 \geqslant 0$ and $0 < k \leqslant 1$.

Hence the theorem. □

To get a physical perspective of these results, let us analyze the geometric relation of the function G and the heuristic functions. G is a positive function in the region $\alpha_1 \geqslant 0$, $0 < k \leqslant 1$. When $\alpha_1 \to 0$, $G \to 0$. This means that for small values of α_1, G is also small. Since $\alpha_1 = \lambda_1 c$, the value of α_1 depends on λ_1 and c. When c is small, G is very close to its minimum, 0, and hence both probabilities, p_1 and p_2, are very close. This behavior can be noticed in Fig. 2, and the phenomenon is observed if the heuristic functions are both comparable and almost equally efficient.

In terms of histogram methods and in database query optimization, when c is small, the optimal and the sub-optimal QEP are very close. Since histogram methods such as Equi-

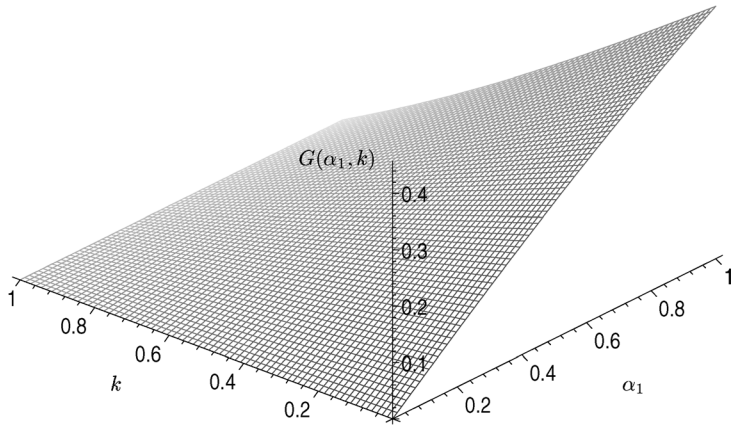

Fig. 2. Function $G(\alpha_1, k)$ plotted in the ranges $0 \leqslant \alpha_1 \leqslant 1$ and $0 \leqslant k \leqslant 1$.

width and Equi-depth produce a larger error than the R-ACM and the T-ACM, the former are less likely to find the optimal QEP than the latter.

Interpreted alternatively, G is very small when λ_1 is close to 0. This means that $\mathrm{Var}[X_1]$ is very large. Since $\mathrm{Var}[X_1] \leqslant \mathrm{Var}[X_2]$, $\mathrm{Var}[X_2]$ is also very large, and both are close each other (in Fig. 1, we would observe almost flat curves for both distributions). Random variables for histogram methods such as Equi-width and Equi-depth yield similar error estimation distributions with large and similar variances. Hence, the probabilities p_1 and p_2 are quite close, and consequently, similar results are expected for these estimation methods. However, when the heuristic functions yield widely different estimated costs (as in the case when the new histogram methods, the R-ACM and the T-ACM, are compared to the traditional methods), these effectively imply random variables with smaller variances being compared to random variables with larger variances. In such a case, the value of G is very high—implying that the former would yield superior solutions.

2.2. Analysis considering normal distributions

For the analysis done in this section, we consider that we are given two heuristic functions, H_1 and H_2, for which the probabilities of choosing optimal or suboptimal solutions are represented by two normally distributed random variables, X_1 and X_2, whose means are μ_1 and μ_2, and whose variances are σ_1^2 and σ_2^2 respectively.

Although the model using normal distributions is more realistic in real life problems, the analysis becomes impossible because there is no closed-form algebraic expression for integrating the normal probability density function. Alternatively, we have used numerical integration and we have obtained rather representative values for which the implication between efficiency and optimality is again corroborated.

Without loss of generality, if the mean cost of the optimal solution is μ_1, by shifting the origin by μ_1, we again assume that the cost of the best solution is 0, which is the mean of these two random variables. The cost of the second best solution is given by another two random variables (one for using the heuristic function H_1, and the other one for using the heuristic function H_2) whose mean, $\mu_2 > 0$, is the same for both variables. We also assume that, by scaling both distributions,[8] the variance of using H_1 and leading to the optimal solution is unity. An example will help to clarify this.

Example 2. Suppose that using H_1 leads to the optimal cost with probability represented by the normal random variable $X_1^{(opt)}$ whose mean is 0 and standard deviation is $\sigma_1 = 1$. This heuristic function also estimates another sub-optimal cost according to $X_1^{(subopt)}$ whose mean is 4 and $\sigma_1 = 1$.

H_2 is another heuristic function that is used to estimate the optimal cost with probability given by $X_2^{(opt)}$ whose parameters are $\mu_1 = 0$ and $\sigma_2 = 1.4$. The other corresponding sub-optimal cost given by the heuristic function H_2 is obtained with probability given by $X_2^{(subopt)}$ whose parameters are $\mu_2 = 4$ and $\sigma_2 = 1.4$.

[8] This can be done by multiplying σ_1^2 and σ_2^2 by σ_1^{-2}, and μ_1 and μ_2 by σ_1^{-1}. This is a particular case of the simultaneous diagonalization between d-dimensional normal random vectors for which $d = 1$ [5].

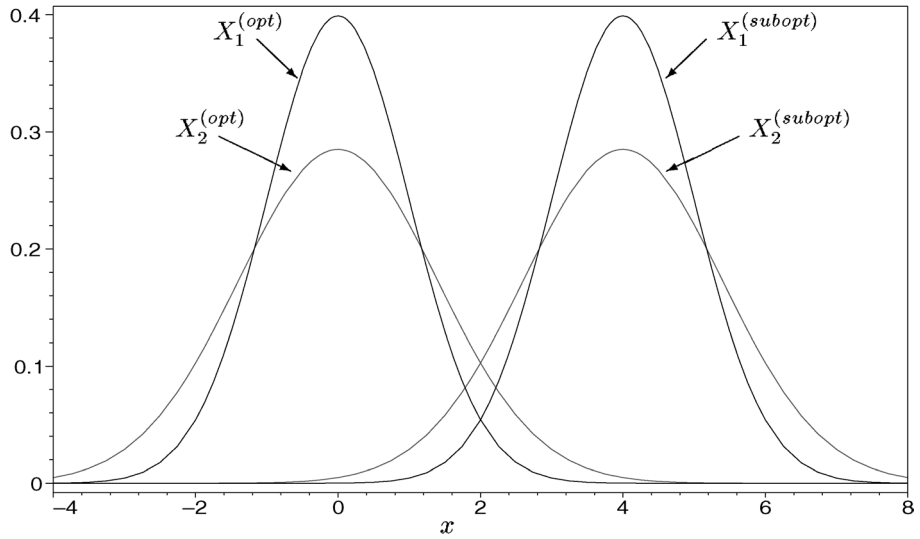

Fig. 3. An example showing the probability density function of four normal random variables whose parameters are $\sigma_1 = 1$, $\sigma_2 = 1.4$, $\mu_1 = 0$, and $\mu_2 = 4$.

Observe that $\sigma_1 < \sigma_2$, and hence we are expecting that the probability of using H_1 and leading to a wrong decision is smaller than that of using H_2. The probability density functions for these four random variables are depicted in Fig. 3. Note that, as in the doubly exponential distribution, given a particular value of x, if its probability under $X_1^{(opt)}$ is high, then the area for which using H_1 leads to the wrong decision (i.e., its cumulative probability under $X_1^{(subopt)}$) is small. Since these two quantities are multiplied and integrated, the final value is smaller than that of using H_2, since σ_2 is greater than $\sigma_1 = 1$. This is what we formally show below.

Result 1. [9] *Suppose that A is a heuristic algorithm that can potentially utilize either of two heuristic functions, H_1 and H_2. Let*:

- *X_1 and X_2 be two normally distributed random variables that represent the costs of the optimal solutions obtained by H_1 and H_2 respectively.*
- *X_1' and X_2' be two other normally distributed random variables that represent the costs of non-optimal solutions obtained by using H_1 and H_2 respectively.*
- *$0 = \mathrm{E}[X_1] = \mathrm{E}[X_2] \leqslant \mathrm{E}[X_1'] = \mathrm{E}[X_2'] = \mu$.*
- *p_1 and p_2 be the probabilities that using H_1 and H_2 respectively lead to the wrong decision.*

[9] We cannot claim this result as a theorem, since the formal analytic proof is impossible. This is because there is no closed-form expression for integrating the Gaussian probability density function. However, the computational proof that we present renders this to be more than a conjecture.

Then,

$$\textit{if } \mathrm{Var}[X_1] = \mathrm{Var}[X_1'] = \sigma_1^2 \leqslant \sigma_2^2 = \mathrm{Var}[X_2] = \mathrm{Var}[X_2'], \quad p_1 \leqslant p_2.$$

Computational Proof. To achieve this proof, we proceed by doing the same analysis that we did for the doubly exponential distributions (Theorem 1). If we consider a particular cost x, the probability that x leads to a wrong decision made by using H_1, is given by:

$$I_1 = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma_1} \mathrm{e}^{-\frac{(u-\mu)^2}{2\sigma_1^2}} \,\mathrm{d}u. \tag{25}$$

The probability that using H_1 leads to the wrong decision for *all* values of x is obtained by integrating the function resulting from multiplying every value of I_1 for each x with the respective probability density function of $X_1^{(opt)}$, which results in:

$$p_1 = \int_{-\infty}^{\infty} I_1 \frac{1}{\sqrt{2\pi}\sigma_1} \mathrm{e}^{-\frac{x^2}{2\sigma_1^2}} \,\mathrm{d}x. \tag{26}$$

Similarly, p_2 can also be expressed as follows:

$$p_2 = \int_{-\infty}^{\infty} I_2 \frac{1}{\sqrt{2\pi}\sigma_2} \mathrm{e}^{-\frac{x^2}{2\sigma_2^2}} \,\mathrm{d}x, \tag{27}$$

where I_2 is obtained in the same way as in (25) for the distribution with variance σ_2^2.

Since there is no closed-form algebraic expression for integrating the normal probability density function, no analytical solution for proving that $p_1 \leqslant p_2$ can be formalized.

Alternatively, we have invoked a computational analysis by calculating these integral for various representative values of σ_1 and σ_2 by using the trapezoidal rule. The values of $G = p_2/p_1 \geqslant 1$ (i.e., for $1 \leqslant \sigma_1 \leqslant 10$ and $1 \leqslant \sigma_2 \leqslant 10$, where $\sigma_1 \leqslant \sigma_2$) are depicted in Table 1 in the form of a *lower-diagonal* matrix. All the values of the *upper-diagonal* matrix (not shown here) are less than unity. Note that by making the value of $\sigma_1 = 1$, the analysis reduces to the first and second columns of this table. For example, if $\sigma_1 = 1$ and $\sigma_2 = 2$, $p_2/p_1 \approx 33.6276$. For more neighboring values of σ_1 and σ_2, e.g., $\sigma_1 = 9$ and $\sigma_2 = 10$ ($\sigma_1 = 1$ and $\sigma_2 \approx 1.2345$ after scaling), $p_2/p_1 \approx 1.0318$, which is very close to unity. The ratio for $\sigma_1 = 1$ and $\sigma_2 = 10$ is much bigger, i.e., more than one hundred times. □

In order to get a better perspective of the computational analysis, we study the behavior of the function $G = p_2/p_1$. Using the values of G given in Table 1, we have plotted this function in the three-dimensional space as $G(\sigma_1, \alpha_1)$, where $\alpha_1 = k\sigma_1$, $1 \leqslant k \leqslant 10$. The plot is depicted in Fig. 4.

In order to enhance the visualization of G, we have approximated it by using the regression utilities of the symbolic mathematical software package Maple V [3]. When $k = 1$, the surface lies on the $z = 0$ plane, in the form of a straight line $x = y$ (labeled "$k = 1$ or $\sigma_1 = \sigma_2$" in the figure). This is the place in which G reaches its minimum, when both heuristic functions have identical variances. When k is larger (i.e., $k = 10$), the function G

Table 1
Ratio between the probability of making the wrong decision for two normally distributed random variables whose standard deviations are σ_1 and σ_2

σ_2	σ_1									
	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
1.00	1.0000									
2.00	33.6276	1.0000								
3.00	73.9210	2.1982	1.0000							
4.00	102.5081	3.0483	1.3867	1.0000						
5.00	122.1988	3.6339	1.6531	1.1921	1.0000					
6.00	136.2472	4.0516	1.8431	1.3291	1.1150	1.0000				
7.00	146.6138	4.3599	1.9834	1.4303	1.1998	1.0761	1.0000			
8.00	154.7078	4.6006	2.0929	1.5092	1.2660	1.1355	1.0552	1.0000		
9.00	161.0448	4.7891	2.1786	1.5710	1.3179	1.1820	1.0984	1.0410	1.0000	
10.00	166.1716	4.9415	2.2480	1.6211	1.3598	1.2196	1.1334	1.0741	1.0318	1.0000

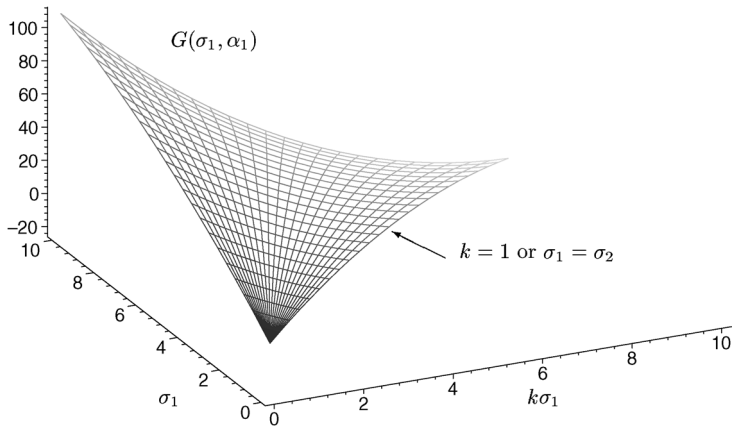

Fig. 4. Function $G(\sigma_1, k\sigma_1)$ plotted in the ranges $1 \leqslant \sigma_1 \leqslant 10$ and $1 \leqslant k\sigma_1 \leqslant 10$, where $\sigma_2 = k\sigma_1$.

becomes much larger (up to 166.1716 in Table 1). This clearly shows the importance of minimizing the variance in deciding on a heuristic function.

When it concerns histograms in database query optimization, when k is small, it implies that the optimal and sub-optimal QEP are very close. Therefore, histogram methods like the Equi-width and the Equi-depth are less likely to find the optimal QEP, since they produce larger errors than histogram approximation methods such as the R-ACM and the T-ACM. The latter produce very small errors, and hence, when comparing any of them with the Equi-width or the Equi-depth, we will have a much larger value of k. This will be reflected in our empirical results presented in the next section.

3. Simulation results for database heuristic functions

3.1. Empirical results

In order to provide practical evidence of the theoretical results presented above,[10] we have performed some simulations in database query optimization. In the experiments, we have conducted four independent runs. In each run, 100 random databases were generated. Each database was composed of six relations, each of them having six attributes. Each relation was populated with 100 tuples.

For each database, a random query including the six relations and arbitrary attributes was performed. The cost of executing the query using the estimates of the histograms obtained from the Equi-width, the Equi-depth, and the R-ACM was evaluated. This cost is calculated by counting the number of tuples of the intermediate relations involved in the query processing tree. More details of the simulations can be found in [17].

The efficiency of the R-ACM was compared with that of the Equi-width and the Equi-depth after performing these simulations using 50 values per attribute. We set the number of bins for the Equi-width and the Equi-depth to be 22. In order to be impartial with the evaluation, we set the number of bins for the R-ACM to be *approximately half* of that of the Equi-width and the Equi-depth, because the former needs twice as much storage as that of the latter.

The simulation results obtained from 400 independent runs, used to compare the efficiency of the R-ACM with that of the Equi-width and that of the Equi-depth, are given in Table 2. The column labeled "R > W" is the number of times that the R-ACM obtains a better solution than that of the Equi-width. The column labeled "W > R" indicates the number of times in which the Equi-width leads to a better QEP than the one determined by the R-ACM. Similarly, the column labeled "R > D" represents the number of times that the R-ACM yields a better solution than the Equi-depth, and the column labeled "D > R" is the

Table 2
Simulation results for the R-ACM, the Equi-width, and the Equi-depth, after optimizing queries on 400 randomly generated databases. The column labeled "R > W" contains the number of times in which R-ACM obtained a better solution than the Equi-width on 100 randomly generated databases. The information contained in the other columns has a similar interpretation, where "R", "W" and "D" stand for the R-ACM, the Equi-width and the Equi-depth respectively. The last row contains the sum of the values in each column

Simulation	R > W	W > R	R > D	D > R
1	26	12	35	12
2	24	15	42	13
3	35	11	46	8
4	29	15	46	8
Total	**114**	**53**	**169**	**41**

[10] The empirical results presented in this paper are not intended to compare the various histogram methods: Equi-width, Equi-depth, R-ACM, T-ACM, V-optimal, etc. The experimental results submitted are merely included to demonstrate that the theoretically proven results can be experimentally justified.

number of times in which the Equi-depth is superior to the R-ACM. The last row, the total of each column, gives us the evidence that the superiority of the R-ACM over the Equi-width is demonstrated more than twice as often. The same factor relating the superiority of the R-ACM over the Equi-depth is about four.

3.2. Geometric justification of the rationale

We now present a different perspective for the formulation of the QEP model that has been used earlier. Indeed, we shall analyze the suitability of using the doubly exponential distribution for the query optimization problem. To demonstrate this suitability, we examined 200 randomly selected queries. Since the cost of each query is different for each database, we computed the difference between the actual cost of executing the query and the estimated cost. For each of the histogram methods, namely the Equi-width, the Equi-depth and the R-ACM, we obtained two hundred points.[11] Using these points (or samples) we estimated the parameters of the doubly exponential distribution, λ, for each histogram method, using a Maximum Likelihood Estimate (MLE) method [4].

Given N samples, $\{x_1, \ldots, x_N\}$, obeying a doubly exponential distribution, it is easy (almost purely algebraic) to see that the maximum likelihood parameter, $\hat{\lambda}$, satisfying the distribution obeys:

$$\hat{\lambda} = \frac{N}{\sum_{i=1}^{N} |x_i|}. \tag{28}$$

Using the estimate of (28), we computed the parameters for the doubly exponential distribution for the Equi-width, the Equi-depth, and the R-ACM, which resulted in 0.6399, 0.6120, and 0.7089 respectively. We have also calculated their variances as in (3) – they are 4.8834, 5.3401, and 3.9791 for the Equi-width, the Equi-depth and the R-ACM respectively. As expected, the variance for the R-ACM is smaller than that of the Equi-width and the Equi-depth. This can also be observed in Fig. 5, in which the corresponding doubly exponential probability distribution functions are plotted for the three histograms. This slight difference between the R-ACM, the Equi-width and the Equi-depth schemes reflects in the corresponding results leading to superior QEPs as shown in Table 2. Clearly, the R-ACM, whose variance is smaller than that of the Equi-width and the Equi-depth, is a superior heuristic function.

In order to observe the similarities between the doubly exponential distribution and the distribution of the actual cost of executing a query, we have plotted the expected values of the doubly exponential distribution and the actual costs obtained when optimizing queries using the R-ACM histogram. The plot depicted in Fig. 6 was obtained by grouping the data in bins of width two, for the values in the ranges $[x_1, x_2)$, where $x_2 = x_1 + 2$, and $x_2 = 2i$ for $i = -4, \ldots, 5$. In the figure, "R-ACM" (in light gray) represents the actual cost values of the queries, and "d-exp" (in dark gray) represents the expected population in each bin when the random variable is doubly exponential with a value of λ being determined by

[11] Since these histograms always tend to under-estimate the costs of the queries, we have shifted all the points so that the estimated mean of these samples is zero. In this way, we could work with zero-mean random variables.

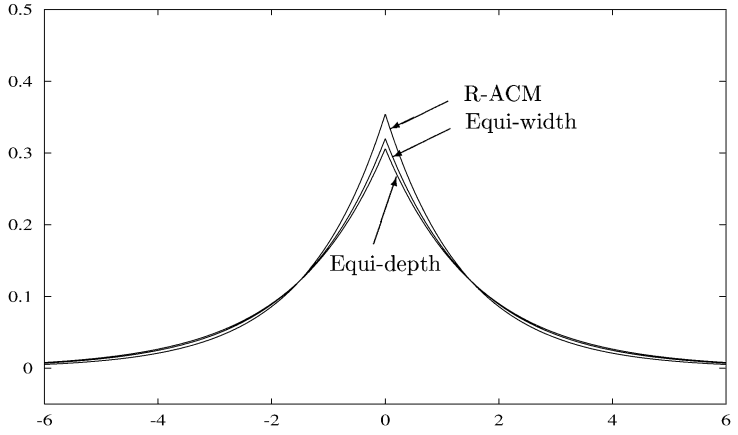

Fig. 5. Estimated probability density function for three doubly exponential random variables that represent the error in estimation for the Equi-width, the Equi-depth and the R-ACM.

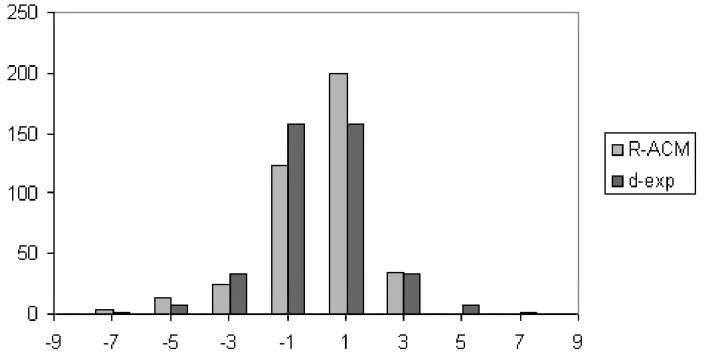

Fig. 6. Expected values for a doubly exponential random variable, and the actual costs obtained after optimizing queries on 400 random databases using the R-ACM histogram.

using (28). Observe the similarity between both histograms. We further corroborate the validity of our model for the database query optimization problem.

4. Conclusions

The theory of PR is quite developed, and has many applications. In this paper, we have applied pattern classification techniques to solve a fundamental open problem in computer science that relates heuristic function accuracy and solution optimality. More specifically, in this paper, we have discussed the efficiency of using heuristic functions for optimization problems and resolved an open problem, which has been (to our knowledge) open for at least twenty years. The problem involves how the accuracy of a heuristic function relates to the quality of the corresponding solution obtained. The efficiency has been quantified by means of the probability of the heuristic function leading to the optimal solution. We have

shown analytically (using a reasonable model of accuracy, namely the doubly exponential distribution for errors) that as the accuracy of a heuristic function increases, the probability of it leading to a superior solution also increases.

Due to the constraints involved in deriving a closed-form expression for integrating the normal probability density function, we have presented a computational analysis of the accuracy/optimality result for the Gaussian distribution. Again, our analysis corroborates the result that heuristic functions producing smaller errors lead more often to optimal solutions.

For the field of database query optimization, we have highlighted that for histogram methods that produce errors with similar variances (the Equi-width and the Equi-depth), the query processing results are also quite similar. However, we have also shown that the R-ACM and the T-ACM, which produce errors with smaller variances than the traditional methods, yield better query optimization plans more often. This result, earlier shown theoretically, has been experimentally verified. Thus, our empirical results on database query optimization show that the R-ACM provides superior solutions more than twice as many times as the Equi-width, and more than four times as often as the Equi-depth. More detailed empirical results including the design of random databases and random queries in these random databases can be found in [17].

We have also estimated the parameters of the doubly exponential distributions representing the Equi-width, the Equi-depth and the R-ACM, and shown graphically how our experiments relate to the theoretical model presented in this paper.

Acknowledgments

The authors are very grateful to the anonymous referee for his/her suggestions. These suggestions have allowed us to significantly enhance the quality of the paper. In particular, we would like to thank him/her for his/her critical remarks which enabled us to clearly elucidate the motivation for the paper in terms of examples, and an explanation of the difference between “heuristic algorithms” and the “heuristic functions” that they utilize.

References

[1] E. Aarts, J. Korst, Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, Wiley, New York, 1989.
[2] S. Christodoulakis, Estimating selectivities in data bases, Technical Report CSRG-136, Computer Science Department, University of Toronto, 1981.
[3] E. Deeba, A. Gunawardena, Interactive Linear Algebra with MAPLE V, Springer, Berlin, 1997.
[4] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley, New York, 2000.
[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1990.
[6] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[7] F. Glover, M. Laguna, Tabu Search, Kluwer Academic, Dordrecht, 1997.
[8] Y. Ioannidis, S. Christodoulakis, On the propagation of errors in the size of join results, in: Proceedings of the ACM-SIGMOD Conference, 1991, pp. 268–277.
[9] R.P. Kooi, The optimization of queries in relational databases, PhD thesis, Case Western Reserve University, 1980.

[10] D. Kreher, D. Stinson, Combinatorial Algorithms: Generation, Enumeration, and Search, CRC Press, Boca Raton, FL, 1998.
[11] D. Levy, How Computers Play Chess, Computer Science Press, New York, 1991.
[12] M.V. Mannino, P. Chu, T. Sager, Statistical profile estimation in database systems, in: ACM Computing Surveys, vol. 20, 1988, pp. 192–221.
[13] M. Michell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1998.
[14] M. Muralikrishna, D. Dewitt, Equi-depth histograms for estimating selectivity factors for multi-dimensional queries, in: Proceedings of ACM-SIGMOD Conference, 1988, pp. 28–36.
[15] Kumpati Narendra, M.A.L. Thathachar, Learning Automata: An Introduction, Prentice Hall, Englewood Cliffs, NJ, 1989.
[16] N. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, San Mateo, CA, 1998.
[17] B.J. Oommen, L. Rueda, The efficiency of modern-day histogram-like techniques for query optimization, Comput. J. 45 (5) (2002) 494–510.
[18] B.J. Oommen, M. Thiyagarajah, The Rectangular Attribute Cardinality Map: A New Histogram-like Technique for Query Optimization, in: Proceedings of the International Database Engineering and Applications Symposium, IDEAS'99, Montreal, Canada, 1999, pp. 3–15.
[19] B.J. Oommen, M. Thiyagarajah, On the use of the trapezoidal attribute cardinality map for query result size estimation, in: Proceedings of the 2000 International Database Engineering and Applications Symposium, Yokohama, Japan, 2000, pp. 236–242.
[20] B.J. Oommen, T. De St. Croix, Graph partitioning using learning automata, IEEE Trans. Comput. 45 (2) (1995) 195–208.
[21] F. Peracchi, Econometrics, Wiley, New York, 2001.
[22] G. Piatetsky-Shapiro, C. Connell, Accurate estimation of the number of tuples satisfying a condition, in: Proceedings of ACM-SIGMOD Conference, 1984, pp. 256–276.
[23] W. Poosala, Histogram based estimation techniques in databases, PhD thesis, University of Wisconsin-Madison, 1997.
[24] E. Rich, K. Knight, Artificial Intelligence, second ed., McGraw Hill, New York, 1991.
[25] G. Romp, Game Theory: Introduction & Applications, Oxford University Press, Oxford, 1997.
[26] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, second edition, Prentice-Hall, New York, 2002.
[27] A. Webb, Statistical Pattern Recognition, Oxford University Press, New York, 1999.
[28] A. Yuille, M. Coughlan, An A* perspective on deterministic optimization for deformable templates, Pattern Recognition 33 (2000) 603–616.