# Partial match queries in relaxed multidimensional search trees*

Conrado Martínez† Alois Panholzer‡ Helmut Prodinger‡

November 10, 1998

**Abstract**

Partial match queries arise frequently in the context of large databases, where each record contains a distinct multidimensional key, that is, each key is a $K$-tuple whose components are traditionally called coordinates or attributes. In a partial match query we specify the value of $s$ attributes, $0 < s < K$, and leave the remaining $K - s$ attributes unspecified. The goal is to retrieve all the records in the database that match the specified attributes. In this paper we present several results about the average performance and variance of partial matches in relaxed $K$-dimensional trees (search trees and digital tries). These data structures are variants of the well known $K$d-trees and $K$d-tries where the sequence of attributes used to guide a query is explicitly stored at the nodes of the tree and randomly generated and, in general, will be different for different search paths. In the standard variants, the sequence is fixed and identical for all search paths: $1, 2, 3, \ldots, K-1, K, 1, 2, 3, \ldots$. We show that the probabilistic analysis of the relaxed multidimensional trees is very similar to that of standard $K$d-trees and $K$d-tries, and also to the analysis of quadtrees. In fact, besides the average cost and variance of partial match in relaxed $K$d-trees and $K$d-tries, we also obtain the variance of partial matches in 2-dimensional quadtrees. We also compute the average cost of partial matches in other relaxed multidimensional digital tries, namely, relaxed $K$d-Patricia and relaxed $K$d-digital search trees.

## 1 Introduction

In many applications such as Geographical Information Systems, databases, computer graphics and statistical data analysis, it frequently arises the need to dynamically maintain a set $F$ (also called a *file*) of *multidimensional records* [16]. Such a file should support insertions of new multidimensional records, deletions, and several *associative queries*, i.e. different types of operations that retrieve information about the data stored in the file.

*This research was supported by Acciones Integradas Hispano-Austríacas HU1997-0016 (Austrian-Spanish Scientific Exchange Program). The first author was also supported by ESPRIT LTR 20244 (ALCOM IT), CICYT TIC97-1475-CE, DGES PB95-0787 (KOALA) and CIRIT 1997SGR-00366 (SGR). The second author was also supported by the Austrian Research Society (FWF) under project P12599-MAT.

†Departament de Llenguatges i Sistemes Informàtics. Polytechnic University of Catalonia. Jordi Girona 1–3, E-08034 Barcelona, Spain. E-mail: `Conrado.Martinez@lsi.upc.es`

‡Institut für Algebra und Diskrete Mathematik. Technical University of Vienna. Wiedner Hauptstraße 8–10, A-1040 Vienna, Austria. E-mail: `e9125354@fbma.tuwien.ac.at (A. Panholzer)`, `Helmut.Prodinger@tuwien.ac.at`

Each multidimensional record in the file contains a distinct *multidimensional key* and associated data. For the sake of simplicity, we will identify each record with its multidimensional key, which is an ordered $K$-tuple $\mathbf{x} = (x^{(0)}, x^{(1)}, \ldots, x^{(K-1)})$. For $0 \leq j < K$, $x^{(j)}$ is the value of the $j^{th}$ *attribute* (also $j^{th}$ *coordinate*) of $\mathbf{x}$. Each $x^{(j)}$ belongs to some totally ordered domain $D_j$, and thus $\mathbf{x}$ is an element of $D = D_0 \times D_1 \times \cdots \times D_{K-1}$. Therefore, each (multidimensional) key may be viewed as a point in a $K$-dimensional space. The analogy can be further exploited as we can, without loss of generality, assume that $D_j = [0, 1]$ for all $0 \leq j < K$ and hence that $D$ is the hypercube $[0, 1]^K$.

For $K \geq 2$, there are several types of queries we might be interested in (actually many of these queries make sense for $K = 1$ too). Among them, we have

- Exact match queries: given a key $\mathbf{x}$, determine whether a record with key $\mathbf{x}$ is present in the file $F$ or not;
- Range queries: given an hyperrectangle $[\ell_0, u_0] \times [\ell_1, u_1] \times \cdots \times [\ell_{K-1}, u_{K-1}]$, return the subset of records in $F$ which fall inside the given hyperrectangle, i.e. those keys $\mathbf{x}$ such that $\ell_j \leq x^{(j)} \leq u_j$ for all $0 \leq j < K$;
- Nearest neighbor queries: given a key $\mathbf{x}$, find the $m$ keys in the file closest to the given one, according to some "distance" measure (we should better say a similarity measure) defined over $D$;
- Partial match queries: given a query $q = (q_0, q_1, \ldots, q_{K-1})$ where each $q_j$ is either a value in $D_j$ (specified) or $q_j = *$ (unspecified), return the subset of records in $F$ whose attributes coincide with the specified attributes of $q$, i.e. return $\mathbf{x}$ if and only if, for any $j$, $q_j = *$ or $q_j = x^{(j)}$.

In this paper we will analyze the performance of partial match for several different multidimensional data structures. Before going on, and describing these data structures, we fix a bit of notation that will be consistently used through the paper. The number of specified attributes in a partial match query will be denoted $s$. We assume that $0 < s < K$. Otherwise, when $s = K$, we have exact match queries; and if $s = 0$, all keys in the file satisfy the query. As we will see, the performance of partial match actually depends on the ratio $\rho = s/K$ rather than on the actual values of $s$ and $K$. By hypothesis, $0 < \rho < 1$.

In the last twenty years, several data structures have been proposed for the dynamic maintenance of a file of multidimensional records (insertions and deletions) and a variety of associative queries. Clearly, it is desirable that any multidimensional data structure efficiently supports several associative queries, rather than being tailor-made for just one type of associative query.

$K$-dimensional binary search trees ($K$d-trees, for short) were introduced by Bentley [1] as a practical solution to this problem. Algorithms for $K$d-trees are simple, robust, flexible and usually exhibit good average performance. All the associative queries mentioned above are efficiently supported by $K$d-trees.

A $K$d-tree for a file $F$ can be incrementally built by successive insertions into an initially empty $K$d-tree as follows. The first key is put into a single node with two empty subtrees. The first attribute of the second key is compared then with the first attribute of the key at

$x_1 = (0.692, 0.703)$
$x_2 = (0.286, 0.495)$
$x_3 = (0.410, 0.895)$
$x_4 = (0.522, 0.953)$
$x_5 = (0.507, 0.394)$
$x_6 = (0.295, 0.300)$
$x_7 = (0.811, 0.605)$
$x_8 = (0.912, 0.807)$
$x_9 = (0.093, 0.210)$
$x_{10} = (0.188, 0.109)$

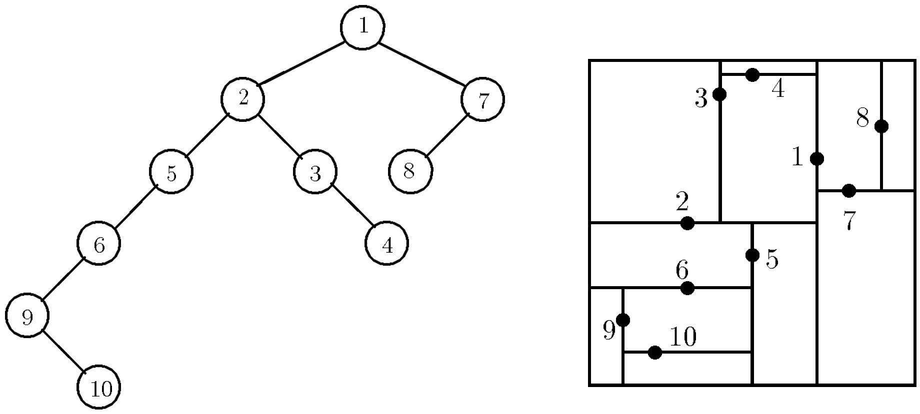

Figure 1: A 2d-tree and its corresponding partition of the search space

the root: if it is smaller or equal, the second key is recursively inserted in the (empty) left subtree; otherwise, it is recursively inserted in the (empty) right subtree. The first attribute of the third key is compared with the first attribute of the key at the root, and recursively inserted into the left or right subtree, as before. But if the subtree is not empty (contains the second key), we shall not compare the respective first attributes of the keys, but their second attributes. In general, when inserting a key $\mathbf{x}$, we compare the key to be inserted with the key $\mathbf{z}$ at the root of some subtree: if $\mathbf{z}$ is at level $j$, we compare $x^{(j \bmod K)}$ and $z^{(j \bmod K)}$, and recursively continue the insertion in the left or the right subtree of $\mathbf{z}$, until a leaf (empty subtree) is found, assuming that $\mathbf{x}$ is not already present in the tree. In Figure 1 we depict the 2d-tree that results from the insertion of $N = 10$ keys into an initially empty tree, in the same order as they have been listed in the left part of the figure. The figure also shows the partition of $[0,1]^2$ induced by the 2d-tree. It should be clear that if the same points were inserted in a different order they could yield a substantially different 2d-tree.

Exact match queries in $K$d-trees obviously operate by following a path down the tree, exactly the same way as if we were inserting that key: either we find it and report success or we reach a leaf, reporting failure.

The standard model for the probabilistic analysis of $K$d-trees is that a random $K$d-tree of size $N$ is built by inserting $N$ points drawn from some continuous probability distribution defined over $[0,1]^K$. This is equivalent to assume that the probability that an insertion fails in a given leaf of a random $K$d-tree of size $N$ is the same for any of its $N+1$ leaves. Under this assumption, the performance of updates and exact match queries coincides with the performance of updates and searches in binary search trees (notice that 1d-trees and binary search trees are the same); therefore, the expected cost of updates and exact match queries in random $K$d-trees is $\mathcal{O}(\log n)$. Furthermore, these costs are independent of $K$. For partial match queries, Flajolet and Puech [7] have shown that the average cost of a partial match query $q$ with $s$ out of $K$ specified attributes is

$$\mathbb{E}[C_n] = \beta_q n^{1-\rho+\vartheta(\rho)} + o(n^{1-\rho+\vartheta(\rho)}),$$

where $\vartheta(x)$ is the unique real solution of

$$(\vartheta(x) + 3 - x)^x (\vartheta(x) + 2 - x)^{1-x} - 2 = 0,$$

$$
\begin{aligned}
x_1 &= (0.10110\ldots, 0.10110\ldots)\\
x_2 &= (0.01001\ldots, 0.01111\ldots)\\
x_3 &= (0.01101\ldots, 0.11100\ldots)\\
x_4 &= (0.10000\ldots, 0.11110\ldots)\\
x_5 &= (0.10000\ldots, 0.01100\ldots)\\
x_6 &= (0.01001\ldots, 0.01001\ldots)\\
x_7 &= (0.11001\ldots, 0.10011\ldots)\\
x_8 &= (0.11101\ldots, 0.11001\ldots)\\
x_9 &= (0.00010\ldots, 0.00110\ldots)\\
x_{10} &= (0.00110\ldots, 0.00011\ldots)
\end{aligned}
$$

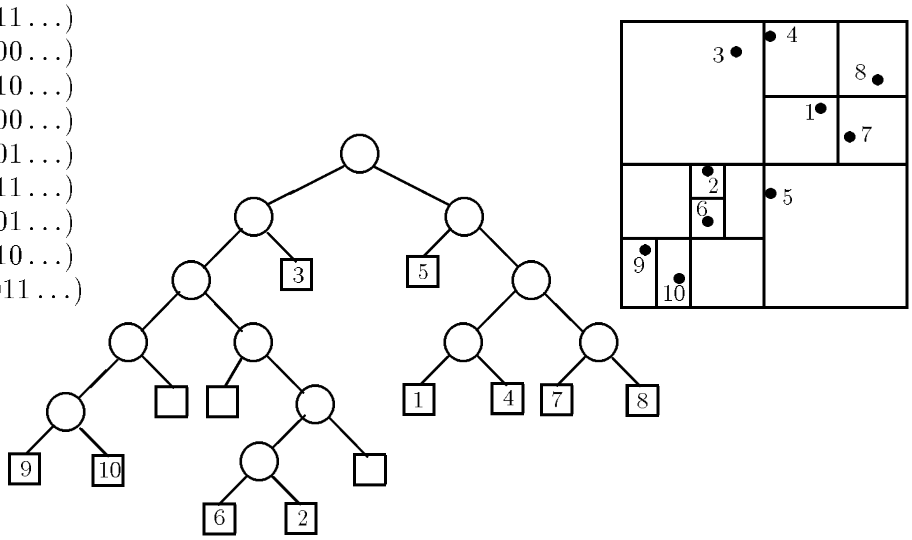

Figure 2: A 2d-trie and its corresponding partition of the search space

and $\beta_q$ is a constant which only depends on the pattern of alternance between specified and unspecified attributes in $q$. Furthermore, for $0 < x < 1$, $\vartheta(x) \leq 0.07$, but the result above disproved an old conjecture that $\mathbb{E}[C_n] = \Theta(n^{1-\rho})$.

The idea of $K$d-trees can easily be generalized to digital search trees: instead of partitioning the search space using the data, we can make a regular partition of the search space, based on the digits. The recursive partition of a region of the search space is not further continued when it does contain one or no data points. Thus in a search in a binary $K$d-trie we would use the first bit of the first attribute of the given key in the first level, going to the left if it were a zero, and to the right if it were a one. Then, the first bit of the second key in the second level, the first bit of the third key, and so on, down to the $(K-1)^{\text{th}}$ level. Then, in $K^{\text{th}}$ level, we should use the second bit of the first key, etc. In general, in the $j^{\text{th}}$ level we would be using the $\lfloor j / K \rfloor^{\text{th}}$ bit of the $(j \bmod K)^{\text{th}}$ attribute of the given key. Figure 2 depicts a 2d-trie for the same set of keys (now expressed in binary) as in Figure 1, and the induced partition of the search space. Notice that both the tree and the partition depend on the given set of keys but not on the order those were inserted.

Digital multidimensional data structures were first proposed around 1975 [2, 14] and further analyzed in [7, 11, 12, 17]. The performance of partial match in $K$d-tries and other multidimensional digital data structures (digital search trees, Patricia) is on the average $\Theta(n^{1-\rho})$, the only difference being the coefficient —which is not constant, but fluctuating between two constants— of the main order term. The usual probability model for the analysis of these digital data structures, the so-called Bernoulli model, is that each of the $K$ attributes of a key $\mathbf{x}$ is an independently generated infinite string of 0's and 1's, with each bit independently generated from the others, with identical probability 1/2 (symmetric Bernoulli model) or with fixed probabilities $p$ and $q = 1 - p$ (asymmetric Bernoulli model).

Quite recently, Duch *et al.* [4] have used a technique called *relaxation* to obtain a variant of $K$d-trees that facilitates the introduction of randomization, rotations, and other similar operations which tend to be complex and inefficient in the standard variant. The *relaxed* multidimensional trees differ from their standard counterparts because in the relaxed trees each node explicitly stores a value $j$ that will be used as the discriminator when a search path "passes" through that node. So, instead of using the preceptive attribute given by the cyclic rule of standard $K$d-trees, in relaxed $K$d-trees a field of the node is examined to know what attribute has to be used for the comparison at that point. This idea was already used by Friedman *et al.* [9] in their *optimized* $K$d-trees, where explicitly stored discriminators are used to build an optimal $K$d-tree, a fortiori, given all the keys in the file in advance, and under the assumption that no insertions nor deletions are made afterwards. In [4], however, relaxed $K$d-trees are used for the dynamic maintenance of a file, and the discriminator assigned to each node is given by an independent random variable uniformly distributed over $\{0, \ldots, K-1\}$.

Random relaxed $K$d-trees thus satisfy the following recursive definition: $T$ is a random relaxed $K$d-tree if and only if

1. $|T| = 0$ (i.e. $T$ is empty) or,
2. $|T| = n$, its left subtree $L$ is a random relaxed $K$d-tree of size $i$, $0 \le i < n$, its right subtree $R$ is a random relaxed $K$d-tree of size $n-1-i$, and for any $\mathbf{x} \in T$ and any discriminator $j$, $0 \le j < K$,

$$\mathbb{P}\left[\langle \mathbf{x}, j\rangle \text{ is the root of } T\right] = \frac{1}{nK}.$$

Random $K$d-trees satisfy a similar (at least in spirit) but more complex recursive characterization, as the discriminators in successive levels must follow the cyclic sequence: $0, 1, \ldots, K-1, 0, 1, \ldots$

As we shall see the recursive definition above makes the analysis of partial match in relaxed $K$d-trees much easier. We tackle this analysis in Sections 2, 3 and 4.

We prove that the expected cost of a partial match in a random relaxed $K$d-tree is $\beta \cdot n^{\epsilon} + \mathcal{O}(1)$ (Theorems 1 and 2), where $\epsilon$ and $\beta$ are constants only depending on $\rho$, and $\epsilon(\rho) \ge 1 - \rho + \vartheta(\rho)$ for all $\rho$. Nevertheless, the difference between $\epsilon$ and $1 - \rho + \vartheta(\rho)$ is always "small" (never exceeding 0.08). This means that relaxed $K$d-trees perform slightly worse than $K$d-trees, but the difference is almost insignificant. Also, the constant $\beta$ for relaxed $K$d-trees is independent of the query pattern, while for standard $K$d-trees there is a possibly distinct constant $\beta_q$ for each of the $2^K - 2$ relevant query patterns $q$. Thus relaxed $K$d-trees are not sensitive to the actual query, whereas $K$d-trees are, and the difference in the performance between two different partial match queries on $K$d-trees could be (for values of $n$ in practical ranges) of the same order of magnitude or even greater than the differences of performance between $K$d-trees and relaxed $K$d-trees.

Next, in Section 4, we compute the variance of the cost of partial match in $K$d-trees, which is $\beta' \cdot n^{2\epsilon} + \mathcal{O}(n^{\epsilon})$. In Section 5 we make a slight detour from the main theme of the paper, and briefly examine the analysis of the variance of the cost of partial match in quadtrees, a rather different multidimensional data structure, whose analysis has remarkable similarities with that of $K$d-trees (both standard and relaxed). We find the variance for 2-dimensional quadtrees. The variance for higher dimensional quadtrees still remains open.

The following sections (Sections 6, 7, 8 and 9) cover the analysis of partial match in relaxed multidimensional digital trees, which can be defined analogously to relaxed $K$d-trees. In particular, we analyze the expected cost of partial match in random relaxed $K$d-tries and its variance, and compare the expected costs for the relaxed variants of $K$d-tries, $K$d-digital search trees and $K$d-Patricia.

## 2 Partial match in relaxed $K$d-trees

Let $C_n$ be the cost of partial match in a random relaxed $K$d-tree of size $n$ when $s$ out of the $K$ attributes have been specified, with $0 < s < K$.

The analysis of partial match in standard $K$d-trees explicitly involves the query pattern and requires the solution of a system of $K$ equations [7]. On the contrary, the analysis of partial match in relaxed $K$d-trees is considerably less difficult than the analysis of partial match in standard $K$d-trees, because it is equivalent to the analysis of a "valuation" $\mathrm{pm}(\cdot)$ over random binary search trees. Let $\mathrm{pm}(t)$ denote the cost of a partial match query in a relaxed $K$d-tree with shape $t$. This random variable is defined by

$$\begin{aligned}\mathbb{P}\left[\mathrm{pm}(t) = k\right] = \rho \left( \frac{|t_1|+1}{|t|+1} \mathbb{P}\left[\mathrm{pm}(t_1) = k-1\right] + \frac{|t_2|+1}{|t|+1} \mathbb{P}\left[\mathrm{pm}(t_2) = k-1\right] \right) \\ + (1-\rho)\, \mathbb{P}\left[\mathrm{pm}(t_1) + \mathrm{pm}(t_2) = k-1\right],\end{aligned}$$

if $t = \circ(t_1, t_2)$ and $k > 0$, $\mathbb{P}\left[\mathrm{pm}(\square) = 0\right] = 1$ and $\mathbb{P}\left[\mathrm{pm}(t) = k\right] = 0$ in any other case.

The recurrence above is simply the formalization of the following intuitive argument: if the given tree is not empty then with probability $(1-\rho)$ the root of the tree is not specified and one needs to visit the root and then continue the search in both subtrees; on the other hand, with probability $\rho$ the root of the tree is specified and one needs to visit the root and then proceeds recursively on the left subtree with probability $\frac{\ell+1}{n+1}$ (where $\ell$ is the size of the left subtree and $n$ the size of the tree) and on the right subtree with complementary probability.

**Proposition 1** *Let $C(z,u)$ be the bivariate generating function*

$$C(z,u) = \sum_{n\geq 0} \sum_{k \geq 0} \mathbb{P}\left[C_n = k\right] z^n u^k.$$

*It satisfies*

$$\frac{\partial^2}{\partial z^2} zC(z,u) = \frac{2\rho}{1-z} u \frac{\partial}{\partial z} zC(z,u) + \frac{1-\rho}{z} u \frac{\partial}{\partial z} \left(zC(z,u)\right)^2$$

*with the initial conditions $C(0,u) = 1$ and $C_z(0,u) = u$. Here and in the sequel $C_x(z,u)$ denotes the partial derivative of $C(z,u)$ with respect to the variable $x$.*

**Proof.** To find the functional equation satisfied by $C(z,u)$, we first introduce the convenient and appealing notation

$$u^{\mathrm{pm}(t)} = \sum_{k\geq 0} \mathbb{P}\left[\mathrm{pm}(t) = k\right] u^k,$$

for the PGF associated to the random variable $\mathrm{pm}(t)$.

Then the recurrence for $\mathbb{P}[\mathrm{pm}(t)=k]$ translates to

$$u^{\mathrm{pm}(t)} = u \cdot \left( \rho \frac{|t_1|+1}{|t|+1} u^{\mathrm{pm}(t_1)} + \rho \frac{|t_2|+1}{|t|+1} u^{\mathrm{pm}(t_2)} + (1-\rho)\, u^{\mathrm{pm}(t_1)+\mathrm{pm}(t_2)} \right), \tag{1}$$

if $t$ is non-empty, and $u^{\mathrm{pm}(\square)} = 1$.

We now may symbolically write for $C(z,u)$

$$C(z,u) = \sum_t \lambda(t) z^{|t|} u^{\mathrm{pm}(t)},$$

where the sum extends to all binary trees $t$, and $\lambda(t)$ is the probability that a random permutation produces the binary search tree $t$ [13, 19]

$$\lambda(t) = \begin{cases} 1 & \text{if } t \text{ is empty,} \\ \frac{\lambda(t_1)\cdot\lambda(t_2)}{|t|} & \text{if } t = \circ(t_1, t_2). \end{cases}$$

Using the recursions for $\lambda(t)$ and $u^{\mathrm{pm}(t)}$ we get, by routine algebraic manipulations, the statement of the proposition. $\square$

Proposition 1 is the basis for our study, as successive differentiation of $C(z,u)$ with respect to $u$ and setting $u=1$ will yield the corresponding differential equations satisfied by the different GFs for factorial moments, including the GF of expectations (the factorial moment of order 1).

## 3 The average cost

To compute the expectation of $C_n$ we first introduce the generating function

$$C_1(z) = \sum_{n\geq 0} \mathbb{E}[C_n]\, z^n. \tag{2}$$

**Proposition 2** *The generating function $C_1(z)$ of the first moments of $C_n$ satisfies the following second-order linear differential equation*

$$zC_1''(z) - 2\frac{(2z-1)}{1-z} C_1'(z) - 2\frac{2-\rho-z}{(1-z)^2} C_1(z) - 2\frac{1}{(1-z)^3} = 0$$

*with the initial conditions $C_1(0) = 0$ and $C_1'(0) = 1$.*

**Proof.** The statement of the proposition easily follows by differentiating w. r. t. $u$ and evaluating at $u=1$ both sides of Proposition 1, as $C_1(z) = C_u(z,1)$. The initial conditions can be found from simple reasoning about the combinatorics of the problem: $C_0 = 0$ and $C_1 = 1$. $\square$

**Proposition 3** *The generating function $C_1(z)$ is*

$$C_1(z) = \frac{1}{1-\rho}\left(\frac{{}_2F_1\left(\left.{a,b \atop 2}\right| z\right)}{(1-z)^\alpha} - \frac{1}{1-z}\right),$$

*where*

$${}_2F_1\left(\left.{a,b \atop c}\right| z\right) = {}_2F_1\left(a,b;c;z\right) = \sum_{k\geq 0} \frac{(a+k-1)^{\underline{k}}(b+k-1)^{\underline{k}}}{(c+k-1)^{\underline{k}}}\frac{z^k}{k!},$$

*denotes the classical hypergeometric function [10, 20], and in this particular instance, $a = 2-\alpha$, $b = 1-\alpha$, and $\alpha = \frac{1}{2}+\frac{1}{2}\sqrt{9-8\rho}$.*

**Proof.** To begin with, it is not difficult to check that

$$-\frac{1}{(1-\rho)(1-z)} \tag{3}$$

is a particular solution of the differential equation given in Proposition 2. From this starting point, we shall proceed by considering the general solution for the homogeneous equation. The following is mostly inspired by the analysis of quadtrees by Flajolet *et al.* [5].

The only possible singularities of the solution of the homogeneous equation

$$zg''(z) - 2\frac{(2z-1)}{(1-z)}g'(z) - 2\frac{(2-\rho-z)}{(1-z)^2}g(z) = 0 \tag{4}$$

are the singularities of the coefficients, and the zeros of the leading coefficient, i.e. $z=1$, $z=0$ and $z=\infty$. As the combinatorial origin of the problem entails a polynomial growth of the coefficients of $C_1(z)$, we study the local behavior of $g(z)$ around $z=1$. It is natural to set then

$$g(z) = y(z)/(1-z)^\alpha,$$

where $\alpha$ is still to be determined. Then (4) transforms to

$$\begin{aligned} &z(1-z)^2y''(z) - 2(2z-\alpha z-1)(1-z)y'(z) + \\ &\qquad ((\alpha-1)(\alpha-2)z + 2(\alpha+\rho-2))y(z) = 0 \end{aligned} \tag{5}$$

Now, we set $\alpha$ so that the coefficient of $y(z)$ is of the form $B_0\cdot(1-z)$, with an arbitrary constant $B_0$ and therefore we impose

$$-\alpha+\alpha^2-2(1-\rho) = 0,$$

whose solutions are $(1+\Delta)/2$ and $(1-\Delta)/2$, with

$$\Delta = \Delta(\rho) = \sqrt{9-8\rho}.$$

We select the largest root, namely $\alpha = (1+\Delta)/2$, and then substituting in (5) we get the hypergeometric differential equation

$$z(1-z)y''(z) + (2-3z+z\Delta)y'(z) + (2\rho-3+\Delta)y(z) = 0 \tag{6}$$

whose solution is

$$y(z) = B_1 \cdot {}_2F_1 \left( \left. {a, b \atop 2} \right| z \right), \tag{7}$$

where ${}_2F_1 \left( \left. {a,b \atop 2} \right| z \right)$ is the classical hypergeometric function, with $a = 2 - \alpha$ and $b = 1 - \alpha$.

Actually, there is a second solution of the hypergeometric equation (6), but it has a logarithmic singularity at $z = 0$, and hence the combinatorial nature of the problem excludes its contribution to $C_1(z)$ (i.e. the constant $B_2$ multiplying such a second term of the general solution must be 0 in this case).

Adjusting the initial conditions, and undoing the manipulations that we have already done, we have finally the closed form for $C_1(z)$ given in the proposition. □

**Theorem 1** *Let $C_n$ denote the cost of a partial match query with $s$ specified attributes out of $K$ in a random relaxed $K$d-tree of size $n$. Then*

$$\mathbb{E}\left[C_n\right] = -\frac{1}{1-\rho} + \frac{1}{1-\rho} \sum_{k=0}^{n} \binom{n-k+\alpha-1}{n-k} \binom{\alpha-1}{k} \binom{\alpha-2}{k},$$

*where $\alpha = \frac{1}{2} + \frac{1}{2}\sqrt{9 - 8\rho}$.*

**Proof.** The formula for the coefficients $\mathbb{E}\left[C_n\right]$ is just the convolution that follows from Proposition 3 and the binomial and hypergeometric expansions. □

**Theorem 2** *Let $C_n$ denote the cost of a partial match query with $s$ specified attributes out of $K$ in a random relaxed $K$d-tree of size $n$. Then*

$$\mathbb{E}\left[C_n\right] = \beta n^{\alpha-1} + \mathcal{O}(1), \tag{8}$$

*where $\alpha$ is as in Theorem 1, $a = 1 - \alpha$, $b = 2 - \alpha$ and*

$$\beta = \frac{{}_2F_1 \left( \left. {a,b \atop 2} \right| 1 \right)}{(1-\rho)\Gamma(\alpha)} = \frac{\Gamma(2\alpha - 1)}{(1-\rho)\alpha\Gamma^3(\alpha)}. \tag{9}$$

**Proof.** To get the asymptotic behavior of $\mathbb{E}\left[C_n\right]$ it suffices to study the asymptotic behavior of $C_1(z)$ when $z \to 1$. The second term makes a contribution which can be neglected, and the hypergeometric function is analytic at $z = 1$. Thus,

$$C_1(z) \sim \beta'(1-z)^{-\alpha} + \mathcal{O}((1-z)^{-1}),$$

with $\beta' = (1-\rho)^{-1} \cdot {}_2F_1\left(a, b; 2; 1\right)$ and from standard singularity analysis [6]

$$[z^n]C_1(z) = \frac{\beta'}{\Gamma(\alpha)} n^{\alpha-1} + \mathcal{O}(1).$$

The hypergeometric function is analytic at $z = 1$ and its value at this point can be found [10] because $\Re(c) = 2 > \Re(a) + \Re(b) = 3 - 2\alpha$. Indeed, for any $\rho$, $0 < \rho < 1$, we have $1 < \alpha < 2$, and then $2 > 3 - 2\alpha$. To evaluate ${}_2F_1\left(a, b; 2; 1\right)$ we apply the well known identity

$${}_2F_1 \left( \left. {a, b \atop c} \right| 1 \right) = \frac{\Gamma(c-a-b)\Gamma(c)}{\Gamma(c-a)\Gamma(c-b)}.$$

□

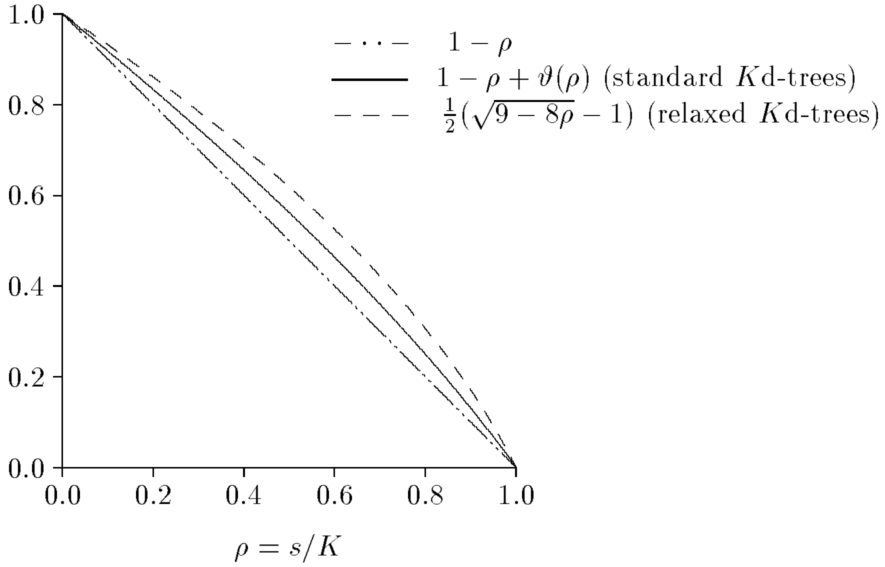


Figure 3: The value of the exponent in the average cost of partial match in standard and relaxed $K$d-trees.

It is interesting to point out that, although Theorem 2 is valid only if $0 < \rho < 1$, it provides meaningful information for the limiting cases to some extent. If $\rho = 0$, that is, no attribute is specified, then $\mathbb{E}[C_n] = n$. Indeed, $\alpha - 1 \to 1$ and $\beta \to 1$ as $\rho \to 0$. On the other hand, in an exact match all attributes are specified and $\rho = 1$. In this case, we know that $\mathbb{E}[C_n] = \mathcal{O}(\log n)$. And we have that $\alpha - 1 \to 0$ and $\beta \to \infty$ if $\rho \to 1$, which is an approximate way to express by a formula of the type $\beta \cdot n^{\alpha-1}$ that $\mathbb{E}[C_n]$ grows with $n$, but slower than any function of the type $n^\epsilon$, for positive $\epsilon$.

In Figure 3 we plot $1-\rho+\vartheta(\rho)$ against $\alpha(\rho)-1$, the exponents in the main order term of the average cost of partial match in standard and relaxed $K$d-trees respectively. The plot shows that standard $K$d-trees perform slightly better than relaxed $K$d-trees for all values of $\rho$.

An intuitive explanation for this difference is that in standard $K$d-trees a partial match will follow, for every $K$ steps, one branch $s$ times and both branches $K-s$ times, whereas these figures hold only on the average in relaxed $K$d-trees. Since there is a non-null probability that the event "both subtrees are explored" occurs more than $K-s$ times in $K$ steps, the expected cost of a partial match in relaxed $K$d-trees is larger. Relaxed $K$d-trees exhibit other advantages, though: their average performance does not depend on the actual pattern of the query, and they are less sensitive to deviations from the hypothesis of "anisotropy", i.e. no coordinate axis is privileged over the others.

# 4 The variance

The steps we shall follow to find $\mathbb{V}[C_n]$ are basically the same as in the previous section. Alas, it will be considerably more difficult to find a closed form for $C_2(z)$, the generating function for the second factorial moments of $C_n$

$$C_2(z) = \sum_{n\geq 0} \mathbb{E}\left[C_n^{\underline{2}}\right] z^n.$$

For the rest of this section, we define $\alpha = (1+\Delta)/2$, $\Delta = \sqrt{9-8\rho}$, $a = 2-\alpha$ and $b = 1-\alpha$, thus bearing the same meanings as in the previous section.

**Proposition 4** *The generating function $C_2(z)$ of the second factorial moments of $C_n$ satisfies the second-order linear differential equation*

$$C_2''(z) - \frac{2(2z-1)}{z(1-z)} C_2'(z) - \frac{2(2-\rho-z)}{z(1-z)^2} C_2(z) = s(z)$$

*where the inhomogeneous part is*

$$\begin{aligned} s(z) = {} & \frac{2(2-z+z\Delta)\cdot {}_2F_1\left(\left.{a,b \atop 2}\right| z\right)}{(1-\rho)z(1-z)^{2\alpha+1}} - \frac{4\rho\cdot {}_2F_1\left(\left.{a,b \atop 2}\right| z\right)}{(1-\rho)z(1-z)^{\alpha+2}} \\ & - \frac{2(\Delta+2\rho-3)\cdot {}_2F_1\left(\left.{a,b \atop 2}\right| z\right)\cdot {}_2F_1\left(\left.{a+1,b+1 \atop 3}\right| z\right)}{(1-\rho)(1-z)^{2\alpha}} - \frac{4}{z(1-z)^3} \end{aligned}$$

*and the initial conditions are $C_2(0) = 0$ and $C_2'(0) = 0$.*

**Proof.** Differentiating the equation given in Proposition 1 twice w. r. t. $u$ and evaluating at $u = 1$ leads to the given equation for $C_2(z) = C_{uu}(z,1)$. Also, we use the result of Proposition 3 in the derivation of this equation. $\square$

**Proposition 5** *The generating function $C_2(z)$ is*

$$C_2(z) = \frac{1}{(1-z)^\alpha}\cdot {}_2F_1\left(\left.{a,b \atop 2}\right| z\right) \int_{x=0}^{z} \left(\int_{t=0}^{x} \frac{t^2\, {}_2F_1\left(\left.{a,b \atop 2}\right| t\right) s(t)dt}{(1-t)^{\alpha-2}}\right) \frac{(1-x)^{2\alpha-2}dx}{x^2\, {}_2F_1^2\left(\left.{a,b \atop 2}\right| x\right)}.$$

**Proof.** Notice that the homogeneous equation corresponding to the differential equation above is the same as (4), but we do not have a particular solution for the inhomogeneous differential equation nor we can disregard the logarithmic term in the general solution of (4).

Therefore we write $g(z) = B_1\, g_1(z) + B_2\, g_2(z)$ for the general solution $g(z)$ of the homogeneous equation, with the nonlogarithmic term given by $g_1(z) = (1-z)^{-\alpha}\, {}_2F_1(a,b;2;z)$ (see previous section).

In the following we will solve this differential equation and obtain an integral representation for $g(z)$.

With the substitutions $g(z) = g_1(z)\widetilde{g}(z)$ and $\widetilde{g}'(z) = \widehat{g}(z)$ the *variation of constants method* [20] yields the following first order differential equation for $\widehat{g}(z)$:

$$g_1(z)\,\widehat{g}'(z) + \left(2g_1'(z) - \frac{2(2z-1)}{1-z} g_1(z)\right)\widehat{g}(z) = 0\,. \tag{10}$$

One solution for $\widehat{g}(z)$ is given by

$$\widehat{g}(z) = \frac{1}{g_1^2(z)\, z^2 (1-z)^2} = \frac{(1-z)^{2\alpha-2}}{z^2 \ {}_2F_1{}^2\left(\begin{smallmatrix} a,b \\ 2 \end{smallmatrix} \middle| z\right)}, \tag{11}$$

so that we get the second solution $g_2(z)$ by

$$g_2(z) = \frac{1}{(1-z)^\alpha} \, {}_2F_1\left(\begin{matrix} a,b \\ 2 \end{matrix} \middle| z\right) \int_{t=z_0}^{z} \frac{(1-t)^{2\alpha-2}}{t^2 \ {}_2F_1{}^2\left(\begin{smallmatrix} a,b \\ 2 \end{smallmatrix} \middle| t\right)} \, dt. \tag{12}$$

The integration path from the starting point $z_0$ to $z$ must not encounter the logarithmic singularity at $z = 0$.

The general solution of the homogeneous differential equation is hence

$$g(z) = \frac{B_1}{(1-z)^\alpha} \, {}_2F_1\left(\begin{matrix} a,b \\ 2 \end{matrix} \middle| z\right) + \frac{B_2}{(1-z)^\alpha} \, {}_2F_1\left(\begin{matrix} a,b \\ 2 \end{matrix} \middle| z\right) \int_{t=z_0}^{z} \frac{(1-t)^{2\alpha-2} dt}{t^2 \ {}_2F_1{}^2\left(\begin{smallmatrix} a,b \\ 2 \end{smallmatrix} \middle| t\right)} \tag{13}$$

and we can now apply again the variation of constants method to get a closed form for $C_2(z)$.

That means we write one particular solution for the equation given in the proposition as

$$g_p(z) = B_{p,1}(z) g_1(z) + B_{p,2}(z) g_2(z)\,. \tag{14}$$

This leads to the following system of linear equations for the derivatives $B'_{p,1}(z)$ and $B'_{p,2}(z)$:

$$\begin{aligned} B'_{p,1}(z) g_1(z) + B'_{p,2}(z) g_2(z) &= 0 \quad \text{and} \\ B'_{p,1}(z) g'_1(z) + B'_{p,2}(z) g'_2(z) &= s(z). \end{aligned}$$

With the solutions

$$\begin{aligned} B'_{p,2}(z) &= \frac{s(z)}{g_1(z)\widehat{g}(z)} \quad \text{and} \\ B'_{p,1}(z) &= -B'_{p,2}(z) \int_{t=z_0}^{z} \widehat{g}(t) dt \end{aligned}$$

we get one particular solution of the differential equation by integration and resubstitution, namely,

$$\begin{aligned} g_p(z) &= g_1(z) \int_{x=z_0}^{z} \left( \int_{t=x_0}^{x} \frac{s(t)}{g_1(t)\widehat{g}(t)} dt \right) \widehat{g}(x) dx \\ &= \frac{1}{(1-z)^\alpha} \, {}_2F_1\left(\begin{matrix} a,b \\ 2 \end{matrix} \middle| z\right) \int_{x=z_0}^{z} \left( \int_{t=x_0}^{x} \frac{t^2 \, {}_2F_1\left(\begin{smallmatrix} a,b \\ 2 \end{smallmatrix} \middle| t\right) s(t) dt}{(1-t)^{\alpha-2}} \right) \frac{(1-x)^{2\alpha-2} dx}{x^2 \ {}_2F_1{}^2\left(\begin{smallmatrix} a,b \\ 2 \end{smallmatrix} \middle| x\right)}. \end{aligned} \tag{15}$$

Adapting the initial conditions now can be done by choosing the starting points of the integration paths $x_0$ and $z_0$, thus getting the claimed integral representation of $C_2(z)$. □

**Theorem 3** *Let $C_n$ denote the cost of a partial match query with $s$ specified attributes out of $K$ in a random relaxed $K$d-tree of size $n$. Then*

$$\mathbb{V}[C_n] \sim \left(\frac{8\Gamma(2\alpha)}{\alpha^2(\alpha-1)^2(2\alpha-1)(3\alpha-2)\Gamma^4(\alpha)} - \frac{4\Gamma^2(2\alpha)}{\alpha^4(\alpha-1)^2(2\alpha-1)^2\Gamma^6(\alpha)}\right) n^{2\alpha-2}.$$

**Proof.** The coefficients $\mathbb{E}\left[C_n^2\right] = [z^n]C_2(z)$ can now be found asymptotically by means of singularity analysis, from its integral representation (Proposition 5).

First we get a local expansion for $s(z)$ around $z=1$, namely

$$\begin{aligned} s(z) &= \frac{4\alpha\, {}_2F_1{}^2\left(\left.{a,b \atop 2}\right| 1\right)}{(1-\rho)(1-z)^{2\alpha+1}} + \mathcal{O}\left(\frac{1}{(1-z)^{\alpha+2}}\right) \\ &= \frac{4\alpha\Gamma^2(2\alpha-1)}{(1-\rho)\Gamma^2(\alpha)\Gamma^2(\alpha+1)(1-z)^{2\alpha+1}} + \mathcal{O}\left(\frac{1}{(1-z)^{\alpha+2}}\right). \end{aligned}$$

Notice that $\alpha+2 < 2\alpha+1$, so the asymptotic expansion is valid. Furthermore, this last inequality implies that that the initial conditions and the arising integration constants do not match, so we get

$$\begin{aligned} C_2(z) &= \frac{1}{(1-z)^\alpha}\int_{x=0}^{z}\left(\int_{t=0}^{x}\frac{B_3\,dt}{(1-t)^{3\alpha-1}}\right)(1-x)^{2\alpha-2}dx + \mathcal{O}\left(\max\left\{\frac{1}{(1-z)^\alpha}, \frac{1}{(1-z)^{2\alpha-2}}\right\}\right) \\ &= \frac{B_3}{(3\alpha-2)(\alpha-1)}\frac{1}{(1-z)^{2\alpha-1}} + \mathcal{O}\left(\max\left\{\frac{1}{(1-z)^\alpha}, \frac{1}{(1-z)^{2\alpha-2}}\right\}\right), \end{aligned}$$

where $B_3$ is the constant factor of the first term in the asymptotic expansion of $s(z)$.

This leads to the following asymptotic estimate for the coefficients

$$\begin{aligned} \mathbb{E}\left[C_n^2\right] = [z^n]C_2(z) &= \frac{B_3}{(3\alpha-2)(\alpha-1)}\frac{n^{2\alpha-2}}{\Gamma(2\alpha-1)} + \mathcal{O}\left(\max\left\{n^{\alpha-1}, n^{2\alpha-3}\right\}\right) \\ &= \frac{8\Gamma(2\alpha)n^{2\alpha-2}}{\alpha^2(\alpha-1)^2(2\alpha-1)(3\alpha-2)\Gamma^4(\alpha)} + \mathcal{O}\left(n^{\alpha-1}\right). \end{aligned}$$

The asymptotic equivalent for the variance given in the statement of the theorem is easily computed from the known asymptotic estimates for $\mathbb{E}[C_n]$ (Theorem 2), from the ones we just have obtained for $\mathbb{E}\left[C_n^2\right]$, and

$$\mathbb{V}[C_n] = \mathbb{E}\left[C_n^2\right] + \mathbb{E}[C_n] - \left(\mathbb{E}[C_n]\right)^2.$$

□

The order of magnitude of the variance for relaxed $K$d-trees is the square of that of its corresponding expected value; this phenomenon also occurs in standard $K$d-trees, where the variance is $\Theta(n^{2(1-\rho+\vartheta(\rho))})$ [3].

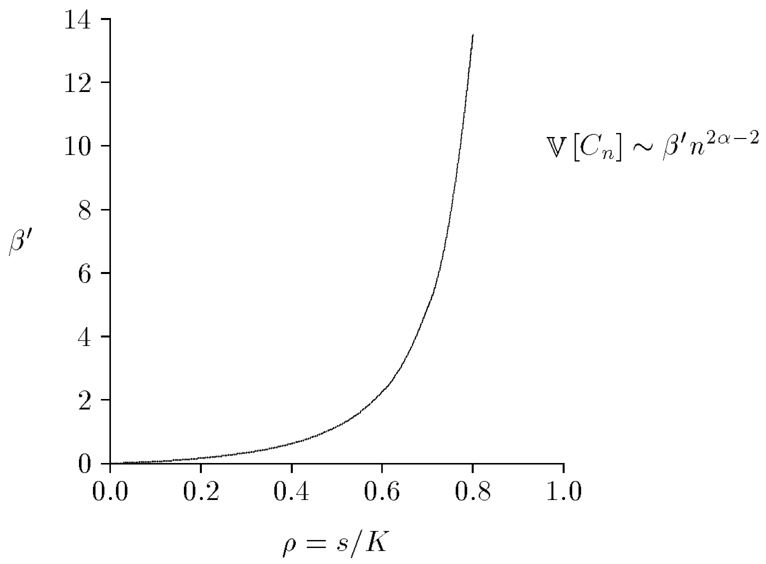


Figure 4: The value of the constant $\beta'$ in the variance of partial match in relaxed $K$d-trees.

## 5 A note on 2-dimensional quadtrees

We cannot refrain from making a few comment about quadtrees, as the analysis of partial match in quadtrees so closely resembles the analysis performed in the two previous subsections. In [5] the expected cost of partial match in quadtrees of arbitrary dimension was explicitly computed by means of the techniques we have sketched in previous sections. However, the variance of partial match in quadtrees, even for the particular case $K = 2$, was unknown. We consider here this particular case and compute this variance $\mathbb{V}[Q_n]$ using the arguments of the previous subsection almost verbatim. Notice that a partial match in dimension $K = 2$ means that exactly one coordinate is specified while the other is left unspecified, that is, $\rho = 1/2$.

**Theorem 4** *Let $Q_n$ be the cost of a partial match in a random 2-quadtree of size $n$. Then*

$$\mathbb{V}[Q_n] \sim \beta_Q n^{\epsilon}_Q = \left( \frac{(2\alpha-1)\Gamma(2\alpha)}{3\alpha(\alpha-1)\Gamma^4(\alpha)} - \frac{\Gamma^2(2\alpha)}{4\Gamma^6(\alpha)} \right) n^{2\alpha-2}$$

*where*

$$\alpha = \frac{\sqrt{17}-1}{2} \approx 1.561552813\ldots,$$
$$\beta_Q \approx 0.3493015\ldots, \qquad \textit{and}$$
$$\epsilon_Q = 2\alpha - 2 \approx 1.12310526\ldots.$$

**Proof.** We are not going to show here a full proof of the theorem, but rather sketch it. Let $Q_2(z)$ be the GF for the second factorial moment of $Q_n$. The recurrence for these costs translates to

$$Q_2''(z) + \frac{2}{z}Q_2'(z) - \frac{4}{z(1-z)^2}Q_2(z) = s(z) \tag{16}$$

where the initial conditions are $Q_2(0) = 0$ and $Q_2'(0) = 0$. From there, and in a manner similar to that shown in the proof of Theorem 3, we find

$$Q_2(z) = \frac{1}{(1-z)^\alpha}\,{}_2F_1\left(\left.\begin{matrix}1-\alpha,-\alpha\\2\end{matrix}\right| z\right)\int_{x=0}^{z}\left(\int_{t=0}^{x}\frac{t^2\,{}_2F_1\left(\left.\begin{smallmatrix}1-\alpha,-\alpha\\2\end{smallmatrix}\right| t\right)s(t)dt}{(1-t)^\alpha}\right)\frac{(1-x)^{2\alpha}dx}{x^2\,{}_2F_1{}^2\left(\left.\begin{smallmatrix}1-\alpha,-\alpha\\2\end{smallmatrix}\right| x\right)} \tag{17}$$

where now $\alpha$ is the solution of the indicial equation, given by $\alpha = \frac{\sqrt{17}-1}{2}$ and

$$s(z) = \frac{4}{z(1-z)^{2\alpha+1}}\,{}_2F_1{}^2\left(\left.\begin{matrix}1-\alpha,-\alpha\\2\end{matrix}\right| z\right) - \frac{4}{z(1-z)^3}.$$

The third and final step is to work out asymptotic expansions for $\mathbb{E}\left[Q_n^2\right]$ as in Section 4 and then for $\mathbb{V}\left[Q_n\right]$. $\square$

## 6 Partial match in relaxed $K$d-tries

As we did for relaxed $K$d-trees in the previous section, it is convenient to start defining a bivariate generating function $C(z,u)$ associated to the costs $C_n$ of a partial match in a random relaxed $K$d-trie of size $n$, where $s$ out of the $K$ attributes have been specified. Again, $\rho = s/K$ and $0 < \rho < 1$. However, the analysis of $K$d-tries makes it advisable to define $C(z,u)$ as the exponential GF of the PGFs $C_n(u) = n![z^n]C(u,z) = \sum_{k\geq 0}\mathbb{P}\left[C_n = k\right]u^k$ rather than the ordinary GF of the $C_n(u)$'s.

**Proposition 6** *Let $C(z,u)$ be the bivariate generating function*

$$C(z,u) = \sum_{n\geq 0}\sum_{k\geq 0}\mathbb{P}\left[C_n = k\right]u^k\frac{z^n}{n!}.$$

*It satisfies*

$$C(z,u) = \rho\,u\,e^{z/2}C(z/2,u) + (1-\rho)\,u\,C^2(z/2,u) + (1-u)(1+z).$$

**Proof.** It is more or less immediate from the recursion of $\mathrm{pm}(\cdot)$ and the Bernoulli model: the bits of the multidimensional keys are independently generated and with symmetric probability, i.e.,

$$\mathbb{P}\left[\text{the } i\text{-th bit of } x^{(j)} = 0\right] = \mathbb{P}\left[\text{the } i\text{-th bit of } x^{(j)}\right] = 1/2,$$

for any $i$, any key $\mathbf{x}$ and any of its attributes $j$, $0 \leq j < K$. Indeed, let

$$C(z,u) = \sum_t \lambda(t)\,u^{\mathrm{pm}(t)}\frac{z^{|t|}}{|t|!},$$

where the sum extends to all binary trees $t$ and $\lambda(t)$ denotes now the probability that $|t|$ random bitstrings produce a trie whose shape is $t$. For the Bernoulli model we have

$$\lambda(t) = \begin{cases}1 & \text{if } |t| \leq 1,\\ 2^{-|t|}\binom{|t|}{|t_1|}\cdot\lambda(t_1)\cdot\lambda(t_2) & \text{if } t = \circ(t_1,t_2).\end{cases}$$

The rest of the proof is routine algebraic manipulation. $\square$

The analysis of the expected value of $C_n$ and its variance follow from differentiation of $C(z,u)$ w. r. t. $u$ and setting $u=1$; but because of the difference-differential equation satisfied by $C(z,u)$, we will rather work with its Poisson transform $\hat{C}(z,u)$ of $C(z,u)$.

**Proposition 7** *The Poisson transform $\hat{C}(z,u) = e^{-z}C(z,u)$ satisfies*

$$\hat{C}(z,u) = \rho\, u\, \hat{C}(z/2,u) + (1-\rho)\, u\, \hat{C}^2(z/2,u) + (1-u)(1+z)e^{-z}.$$

## 7 The average cost

Let $C_1(z)$ be the exponential generating function of the expected values of $C_n$ and let $\hat{C}_1(z) = e^{-z}C_1(z)$ be its Poisson transform.
Hence, if $\hat{C}_n = n![z^n]\hat{C}_1(z)$ then

$$\mathbb{E}\,[C_n] = n![z^n]C_1(z) = \sum_{k\geq 0}\binom{n}{k}\hat{C}_k. \tag{18}$$

Our plan to get $\mathbb{E}\,[C_n]$ is therefore more or less standard: 1) from the functional equation satisfied by $\hat{C}_1(z)$ it is almost straightforward to obtain its coefficients $\hat{C}_n$; 2) and an asymptotic estimate for the sought expected values $\mathbb{E}\,[C_n]$ then follows from the application of Rice's method [8] to (18).

**Proposition 8** *The generating function $\hat{C}_1(z)$ satisfies*

$$\hat{C}_1(z) = 1 - (1+z)e^{-z} + (2-\rho)\hat{C}_1(z/2).$$

*Furthermore, its coefficients are, for $n \geq 1$,*

$$\hat{C}_n = \frac{(-1)^n(n-1)}{1-(2-\rho)2^{-n}}$$

*and $\hat{C}_0 = 0$.*

**Proof.** The functional equation satisfied by $\hat{C}_1(z) = e^{-z}C_1(z)$ trivially follows by differentiating the functional equation given in Proposition 7 w. r. t. $u$ and setting $u=1$, since

$$\hat{C}_1(z) = e^{-z}C_1(z) = e^{-z}C_u(z,1) = \hat{C}_u(z,1).$$

From the equation for $\hat{C}_1(z)$, we get its coefficients $\hat{C}_n$ almost effortlessly

$$\hat{C}_n = (-1)^n(n-1) + (2-\rho)2^{-n}\hat{C}_n.$$

$\square$

The proposition above completes the first part of the analysis, so we can now produce an exact solution for the expected value of $C_n$, as announced:

$$\mathbb{E}\left[C_n\right] = \sum_{k=1}^{n} \binom{n}{k} \frac{(-1)^k (k-1)}{1-(2-\rho)2^{-k}}. \tag{19}$$

However the solution above gives us little information about these expected values. Alternating sums such as the one above are also difficult to evaluate numerically because of the cancellations. Hence, the next step is to produce an asymptotic expansion for $\mathbb{E}\left[C_n\right]$, a task which is becoming almost mechanical thanks to increasingly better understood "technologies" such as Rice's method and others akin to the Mellin transform [8]. Briefly stated, we can convert the alternating sum into an integral in the complex plane by a judicious choice of the integration path and then apply the residue theorem:

$$\sum_{k=n_0}^{n} \binom{n}{k} (-1)^k f_k = -\frac{1}{2\pi \mathrm{i}} \int_{\mathcal{C}} f(z) \frac{\Gamma(n+1)\Gamma(-z)}{\Gamma(n+1-z)} dz =$$
$$\sum_{\sigma} \operatorname{Res}\left[f(z)\frac{\Gamma(n+1)\Gamma(-z)}{\Gamma(n+1-z)}; z=\sigma\right],$$

where the last sum extends to all poles $\sigma$ not in $[n_0, +\infty)$.

**Theorem 5** *Let $C_n$ be the cost of a partial matching query with $s$ out of $K$ specified attributes in a random $K$d-trie of size $n$. Then*

$$\mathbb{E}\left[C_n\right] \sim \frac{n^L}{\log 2}\big((L-1)\Gamma(-L) + \delta_L(\log_2 n)\big),$$

*where $L = \log_2(2-\rho)$ and*

$$\delta_L(x) = \sum_{k \neq 0} \left(L - 1 + \frac{2\pi i}{\log 2}k\right) \Gamma\Big(-L + \frac{2\pi i}{\log 2}k\Big) e^{2\pi i k x},$$

*is a periodic function of period 1, mean 0 and small amplitude.*

**Proof.** Set $f_k = (k-1)/(1-(2-\rho)2^{-k})$, so (19) reads

$$\mathbb{E}\left[C_n\right] = \sum_{k=1}^{n} \binom{n}{k} (-1)^k f_k.$$

A suitable analytic continuation for $f_k$ is obviously

$$f(z) = \frac{z-1}{1-(2-\rho)2^{-z}}$$

whose singularities are located at

$$z = \chi_k = \log_2(2-\rho) + \frac{2\pi \mathrm{i}}{\log 2}k, \quad k = 0, \pm 1, \pm 2, \ldots$$

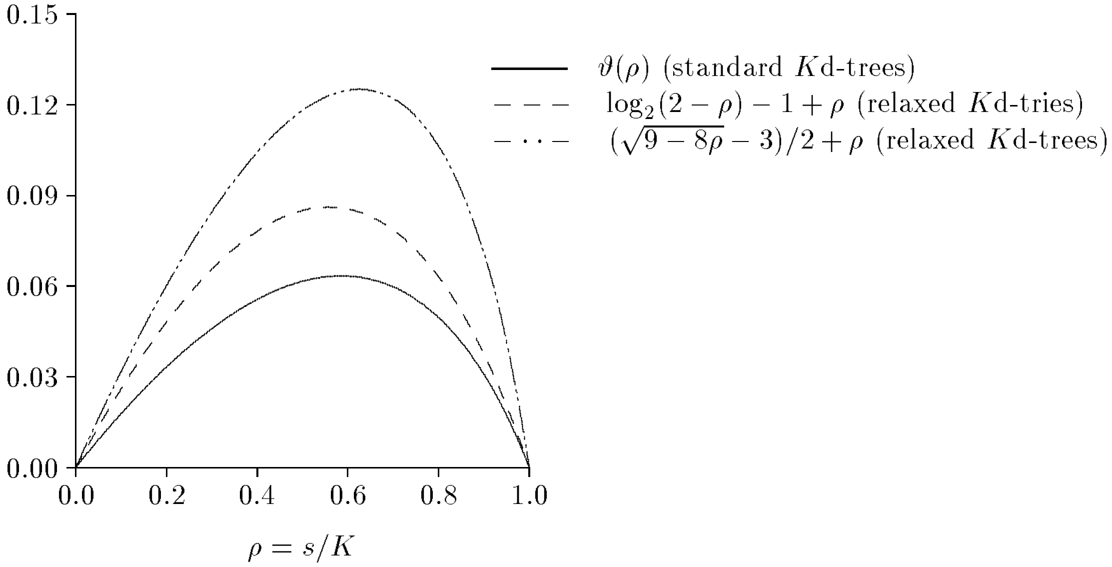


Figure 5: Excess (w. r. t. to $1-\rho$) of the exponent in the average cost of partial match

Each one of these is a simple pole, and hence,

$$\sum_{k=1}^{n} \binom{n}{k} (-1)^k f_k = \sum_k n! \operatorname{Res}\left[ f(z) \frac{\Gamma(-z)}{\Gamma(n+1-z)}; z = \chi_k \right].$$

Let $L = \log_2(2-\rho)$. The residue at $z = \chi_k$ is given by

$$n! \operatorname{Res}\left[ f(z) \frac{\Gamma(-z)}{\Gamma(n+1-z)}; z = \chi_k \right] = \frac{L-1}{\log 2} \left( 1 + \frac{2\pi \mathrm{i}}{(\log 2)(L-1)} k \right) \cdot n^L \cdot e^{2\pi \mathrm{i} k \log_2 n} \cdot \Gamma\Big( -L - \frac{2\pi \mathrm{i}}{\log 2} k \Big).$$

The statement of the theorem follows after a few simple manipulations and simplifications. □

In Figure 5 we plot the difference between $1-\rho$ and the exponent of $n$ in the cost of partial matches for standard $K$d-trees ($\vartheta(\rho)$), for relaxed $K$d-tries and for relaxed $K$d-trees. The horizontal axis would correspond to standard $K$d-tries as the exponent in their expected cost is $1-\rho$. The exponent in relaxed (standard) $K$d-trees is bigger than the exponent in relaxed (standard) $K$d-tries as expected: tries tend to be more well balanced than search trees. It is a bit more surprising that standard $K$d-trees perform slightly better than relaxed $K$d-tries (the difference never exceeds 0.025, but is not 0). The variance due to the relaxation of discriminators is the cause.

# 8 The variance

Let $C_2(z)$ be the exponential generating function of the second factorial moments of the number of comparisons and $\hat{C}_2(z)$ its Poisson transform.

**Proposition 9** *The generating function $\hat{C}_2(z)$ satisfies*

$$\hat{C}_2(z) = 2(2-\rho)\hat{C}_1(\tfrac{z}{2}) + (2-\rho)\hat{C}_2(\tfrac{z}{2}) + 2(1-\rho)\Big(\hat{C}_1(\tfrac{z}{2})\Big)^2.$$

*Furthermore, its coefficients are, for $n \geq 2$,*

$$\hat{C}_n^{(2)} = \frac{(-1)^n(n-1)(2-\rho)2^{1-n}}{\Big(1-(2-\rho)2^{-n}\Big)^2} + \frac{(-1)^n(n-1)\Big(2+2^{n-2}(n-4)\Big)(1-\rho)2^{1-n}}{\Big(1-(2-\rho)^2 2^{-n}\Big)\Big(1-(2-\rho)2^{-n}\Big)}$$
$$+ \frac{(-1)^n(1-\rho)(2-\rho)2^{2-n}}{\Big(1-(2-\rho)^2 2^{-n}\Big)\Big(1-(2-\rho)2^{-n}\Big)} \sum_{\ell=1}^{n-1}\binom{n}{\ell}(n-\ell-1)2^{-\ell}\frac{\ell-1}{1-(2-\rho)2^{-\ell}},$$

*and $\hat{C}_0^{(2)} = \hat{C}_1^{(2)} = 0$.*

**Proof.** From Proposition 7 we get, upon differentiating twice w. r. t. $u$ and setting $u=1$ the functional equation for $\hat{C}_2(z)$.

Reading off the coefficients we find

$$\begin{aligned}\hat{C}_n^{(2)} &= \frac{n![z^n]\Big[2(2-\rho)\hat{C}_1(\tfrac{z}{2}) + 2(1-\rho)\Big(\hat{C}_1(\tfrac{z}{2})\Big)^2\Big]}{1-(2-\rho)2^{-n}} \\ &= \frac{(2-\rho)2^{1-n}(-1)^n(n-1) + (1-\rho)2^{1-n}n![z^n]\hat{C}_1^2(z)}{1-(2-\rho)2^{-n}}.\end{aligned}$$

We thus have to find the coefficients $\hat{A}_n = n![z^n]\hat{C}_1^2(z)$. From Proposition 8, squaring the closed form for $\hat{C}_1(z)$ we get

$$\hat{C}_1^2(z) = \Big(1-(1+z)e^{-z}\Big)^2 + 2(2-\rho)\Big(1-(1+z)e^{-z}\Big)\hat{C}_1(\tfrac{z}{2}) + (2-\rho)^2\hat{C}_1^2(\tfrac{z}{2}).$$

Therefore, we have

$$\hat{A}_n = \frac{n![z^n]\Big[\Big(1-(1+z)e^{-z}\Big)^2 + 2(2-\rho)\Big(1-(1+z)e^{-z}\Big)\hat{C}_1(\tfrac{z}{2})\Big]}{1-(2-\rho)^2 2^{-n}}.$$

Finally, we obtain the sought formula for the $\hat{A}_n$ from

$$n![z^n]\Big(1-(1+z)e^{-z}\Big)^2 = (-1)^n(n-1)\Big[2+2^{n-2}(n-4)\Big] \qquad (n \geq 1)$$

and

$$n![z^n]\Big(1-(1+z)e^{-z}\Big)\hat{C}_1(\tfrac{z}{2}) = (-1)^n\sum_{k=1}^{n-1}\binom{n}{k}(n-k-1)2^{-k}\frac{k-1}{1-(2-\rho)2^{-k}}$$

for $n \geq 2$. Putting everything together yields the form given in the proposition. $\square$

**Theorem 6** *Let $C_n$ denote the cost of a partial match query with $s$ specified attributes out of $K$ in a random relaxed $K$d-trie of size $n$. Then*

$$\mathbb{V}[C_n] \sim \frac{n^L}{\log 2}\Big(r_0(L)\cdot\Gamma(-2L) - \frac{(L-1)^2\Gamma^2(-L)}{\log 2} + \delta_L^{(2)}(\log_2 n)\Big),$$

*where $L = \log_2(2-\rho)$, and*

$$r_0(L) = \frac{4(2L-1)}{1-\rho} + (2L-1)(L-2)(2-\rho) + 4\sum_{\ell\geq 1}\binom{2L}{\ell}(2L-1-\ell)2^{-\ell}\frac{\ell-1}{1-(2-\rho)2^{-\ell}}.$$

*If $\rho = 2-\sqrt{2}$, i.e., $2L=1$, then $r_0(L)\Gamma(-2L)$ must be interpreted as the corresponding limit when $\rho \to 2-\sqrt{2}$.*

*Furthermore, the fluctuating term $\delta_L^{(2)}(x)$ has period 1 and small amplitude, and is given by*

$$\delta_L^{(2)}(x) = \sum_{k\neq 0} r_k(L)\cdot\Gamma\Big(-2L+\frac{2\pi i}{\log 2}k\Big)e^{2\pi i k x} - \delta_L^2(x),$$

$$\begin{aligned} r_k(L) = {} & \frac{4(2L-1+\frac{2\pi i}{\log 2}k)}{1-\rho} + (2L-1+\frac{2\pi i}{\log 2}k)(L-2+\frac{\pi i}{\log 2}k)(2-\rho) \\ & + 4\sum_{\ell\geq 1}\binom{2L+\frac{2\pi i}{\log 2}k}{\ell}(2L-1+\frac{2\pi i}{\log 2}k-\ell)2^{-\ell}\frac{\ell-1}{1-(2-\rho)2^{-\ell}}, \qquad k=\pm 1, \pm 2, \ldots \end{aligned}$$

*and $\delta_L(x)$ is as in Theorem 5.*

**Proof.** From the closed formula for the second factorial moments $\mathbb{E}\left[C_n^{\underline{2}}\right]$ in Proposition 9, it is relatively easy to obtain the suitable continuation (i.e., $f(k) = \mathbb{E}\left[C_k^{\underline{2}}\right]$ for any positive integer $k$)

$$\begin{aligned} f(z) = {} & \frac{(2-\rho)2^{1-z}(z-1)}{\Big(1-(2-\rho)2^{-z}\Big)^2} + \frac{(z-1)\Big(2+2^{z-2}(z-4)\Big)(1-\rho)2^{1-z}}{\Big(1-(2-\rho)^2 2^{-z}\Big)\Big(1-(2-\rho)2^{-z}\Big)} \\ & + \frac{(1-\rho)2^{2-z}(2-\rho)}{\Big(1-(2-\rho)^2 2^{-z}\Big)\Big(1-(2-\rho)2^{-z}\Big)}\sum_{\ell\geq 1}\binom{z}{\ell}(z-\ell-1)2^{-\ell}\frac{\ell-1}{1-(2-\rho)2^{-\ell}} \\ & + \frac{(1-\rho)(2-\rho)4^{1-z}(z-1)}{\Big(1-(2-\rho)^2 2^{-z}\Big)\Big(1-(2-\rho)2^{-z}\Big)^2}. \end{aligned}$$

In order to apply Rice's method, we must compute the residues of

$$\frac{\Gamma(n+1)\Gamma(-z)}{\Gamma(n+1-z)}f(z)$$

with real part $< 2$. First observe that there is no pole at $z=1$, since $f(1)=0$, unless $\rho = 2-\sqrt{2}$ (equivalently, $2L=1$). In any case, careful examination shows that the dominant singularities are located at $z = \chi_k = 2L + \frac{2\pi\mathrm{i}}{\log 2}k, k = 0, \pm 1, \pm 2, \ldots$ and are always simple poles, even if $2L = 1$.

The residue of $f(z)$ at $\chi_k$ is

$$r'_k(L) = \frac{4(2L - 1 + \frac{2\pi\mathrm{i}}{\log 2}k)}{(1-\rho)\log 2} + \frac{(2L - 1 + \frac{2\pi\mathrm{i}}{\log 2}k)(L - 2 + \frac{\pi\mathrm{i}}{\log 2}k)(2-\rho)}{\log 2}$$
$$+ \frac{4}{\log 2}\sum_{\ell \geq 1}\binom{2L + \frac{2\pi\mathrm{i}}{\log 2}k}{\ell}(2L + \frac{2\pi\mathrm{i}}{\log 2}k - \ell - 1)2^{-\ell}\frac{\ell - 1}{1 - (2-\rho)2^{-\ell}}.$$

In particular, for $k = 0$,

$$r'_0(L) = \frac{4(2L-1)}{(1-\rho)\log 2} + \frac{(2L-1)(L-2)(2-\rho)}{\log 2} + \frac{4}{\log 2}\sum_{\ell \geq 1}\binom{2L}{\ell}(2L - \ell - 1)2^{-\ell}\frac{\ell - 1}{1 - (2-\rho)2^{-\ell}}.$$

Furthermore,

$$\frac{\Gamma(n+1)\Gamma(-\chi_k)}{\Gamma(n+1-\chi_k)} \sim \Gamma(-\chi_k)n^{2L}e^{2\pi\mathrm{i}k\log_2 n}.$$

Factoring out $1/\log 2$ from the $r_k$'s and collecting together the contributions of the poles with imaginary part as the first term in the fluctuation $\delta_L^{(2)}(x)$, we get an asymptotic estimate of $\mathbb{E}\left[C_n^2\right]$ which, after subtracting the square of $\mathbb{E}[C_n]$ (and adding $\mathbb{E}[C_n]$, but this contribution is asymptotically negligible), yields the asymptotic behavior of $\mathbb{V}[C_n]$. $\square$

Thus the standard deviation of the cost of partial match in relaxed $K$d-tries is of the same order of magnitude as its expected value. This time such a behavior is rather surprising as the standard deviation of the cost of partial match in all variants of standard multidimensional digital trees is far below the expected value [12, 17].

## 9 Other digital multidimensional trees

In [11] the cost of partial match for standard multidimensional digital search trees (DSTs) and Patricia trees (see [13, 18] for a description of these alternative digital data structures) was analyzed showing that in all cases its expected value is $\Theta(n^{1-\rho})$. Also, Kirschenhofer and Prodinger show that, with regard to the constant factor in the main order term of the expected cost, $K$d-Patricia perform better than $K$d-DSTs and, in turn, these perform better than $K$d-tries.

Relaxed variants of $K$d-DSTs and $K$d-Patricia trees are also easy to define. We will not give here the tedious details, as the techniques that we have to use to analyze partial match in these digital data structures are basically the same that we have used for $K$d-tries. For relaxed $K$d-Patricia trees we have that

$$\mathbb{E}\left[C_n^{[\mathrm{Pat}]}\right] \sim \frac{n^L}{\log 2}\Big(\Gamma(-L)\left(\frac{\rho}{2}L - \rho\right) + \delta_L^{[\mathrm{Pat}]}(\log_2 n)\Big),$$

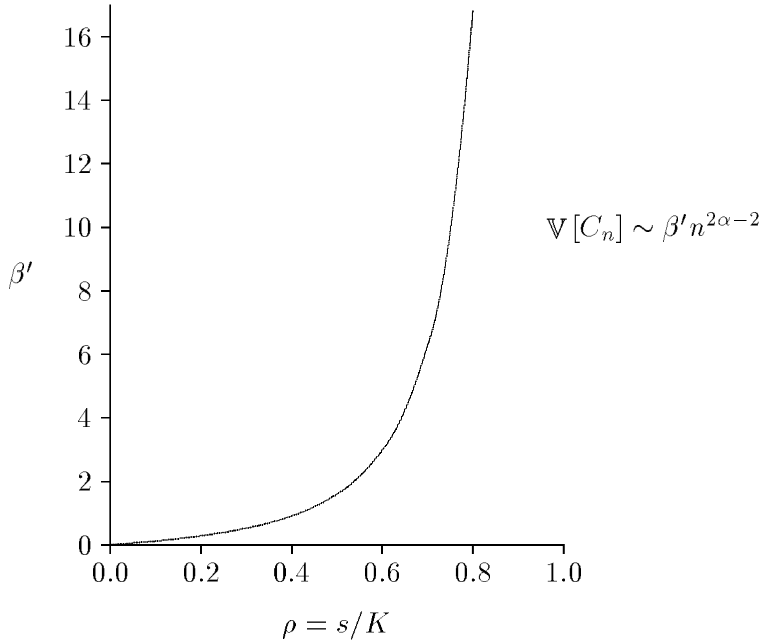


Figure 6: The value of the constant $\beta'$ in the variance of partial match in relaxed $K$d-tries

where $L$ is as in Theorem 5 and $\delta_L^{[\mathrm{Pat}]}(x)$ has similar characteristics to $\delta_L^{[\mathrm{Trie}]}(x) \equiv \delta_L(x)$. And for relaxed $K$d-DSTs we have

$$\mathbb{E}\left[C_n^{[\mathrm{DST}]}\right] \sim \frac{n^L}{\log 2}\left(\left(-\frac{Q(2-\rho)}{Q_\infty}\right)\Gamma(-L) + \delta_L^{[\mathrm{DST}]}(\log_2 n)\right),$$

where

$$Q(t) = \prod_{n\geq 1}\left(1 - \frac{t}{2^n}\right), \quad Q_\infty \equiv Q(1),$$

$L$ is as in Theorem 5 and $\delta_L^{[\mathrm{DST}]}(x)$ is yet another fluctuating function that behaves much like $\delta_L(x)$.

**Theorem 7** *Let $C_n^{[\mathrm{Pat}]}$, $C_n^{[\mathrm{DST}]}$ and $C_n^{[\mathrm{Trie}]}$ denote the cost of a partial match in a relaxed $K$-dimensional Patricia tree, a relaxed $K$-dimensional digital search tree and a $K$-dimensional trie, respectively. Then, for any sufficiently large value of $n$,*

$$\mathbb{E}\left[C_n^{[\mathrm{Pat}]}\right] \lesssim \mathbb{E}\left[C_n^{[\mathrm{DST}]}\right] \lesssim \mathbb{E}\left[C_n^{[\mathrm{Trie}]}\right].$$

**Proof.** Comparing Theorem 5 and the expressions given above for the expected costs of partial match in relaxed Patricia and relaxed DSTs, and disregarding the contributions of the fluctuating factors, the inequalities follow from (observe that $\Gamma(-L)$ is negative for all $0 < \rho < 1$)

$$\rho - \frac{\rho}{2}\log_2(2-\rho) > \frac{Q(2-\rho)}{Q_\infty} > 1 - \log_2(2-\rho), \tag{20}$$

which holds for any $\rho$, $0 < \rho < 1$. This can be proved in a direct manner; however, and surprisingly, it is sufficient to set $y = 1 - \log_2(2-\rho)$; then (20) reads

$$(1 - 2^{-y})(1+y) > \frac{Q(2^{1-y})}{Q_\infty} > y,$$

and those are exactly the inequalities from [11]. □

# 10 Final remarks

The analysis of partial match in relaxed multidimensional trees (either $K$d-trees or $K$d-tries) can be done using the same type of techniques that where used to analyze various relevant parameters in other multidimensional data structures, namely standard $K$d-trees, standard $K$d-tries and $K$-dimensional quadtrees. However, the analysis of relaxed multidimensional trees is usually simpler due to the nice recursive probability models that apply for relaxed trees. The analysis shown in this paper also illustrates that a commonly encountered mistake leads to wrong conclusions: in general it is not valid to assume that the expected value of $f(X)$ is about the same as applying $f$ to the expected value of $X$. In the terms of this paper, even though each possible coordinate is found $t/K$ times —on the average— while following any search path of length $t$ in a relaxed $K$d-tree, the average performance of partial match in $K$d-trees is not proportional to that in standard $K$d-trees (this also holds for relaxed $K$d-tries vs. standard $K$d-tries).

Nevertheless, the results of this paper also show that there are contexts where using relaxed variants of the multidimensional trees might be useful, as the differences in performace with other multidimensional data structures are not extremely huge (for practical values of $n$) and the relaxed trees offer several advantages for the designer of algorithms.

Last but not least, we would like to mention that relaxed $K$d-trees and relaxed $K$d-tries can be readily analyzed, at least in a first approximation, with the so-called *continuous master theorem* [15]. This theorem provides almost effortlessly the right order of magnitude ($\alpha(\rho)-1$, $\log_2(2-\rho)$) of the expected cost of partial matches in both relaxed $K$d-trees and relaxed $K$d-tries. Unfortunately, it cannot provide information about the constants (constant + fluctuation, in the case of $K$d-tries) nor about the order of magnitude of the lower order terms. Another interesting point is that, using precise asymptotic estimates for the expected values obtained by some other mean, the continuous master theorem can be applied to get good estimates of the variances, that is, it produces answers of the form $\beta n^\epsilon + \Theta(n^{\epsilon'})$ for the variance of relaxed $K$d-trees, 2-dimensional quadtrees and relaxed $K$d-tries, with closed forms for $\beta$, $\epsilon$ and $\epsilon'$ in each case, which obviously coincide with those given in Theorems 3, 4 and 6.

# References


[1] J. L. Bentley. Multidimensional binary search trees used for associative retrieval. *Communications of the ACM*, 18(9):509–517, 1975.

[2] W. A. Burkhard. Hashing and trie algorithms for partial match retrieval. *ACM Transactions on Database Systems*, 1(2):175–187, 1976.

[3] W. Cunto, G. Lau, and Ph. Flajolet. Analysis of $kdt$-trees: $k$d-trees improved by local reorganisations. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Algorithms and Data Structures (WADS'89)*, volume 382 of *LNCS*, pages 24–38. Springer-Verlag, 1989.

[4] A. Duch, V. Estivill-Castro, and C. Martínez. Randomized $k$-dimensional binary search trees. Technical Report LSI-98-??-R, LSI-UPC, 1998.

[5] Ph. Flajolet, G. Gonnet, C. Puech, and J. M. Robson. Analytic variations on quadtrees. *Algorithmica*, 10:473–500, 1993.

[6] Ph. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM Journal of Discrete Mathematics*, 3(1):216–240, 1990.

[7] Ph. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[8] Ph. Flajolet and R. Sedgewick. Mellin transforms and asymptotics: Finite differences and Rice's integrals. *Theoretical Computer Science*, 144(1–2):101–124, 1995.

[9] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition, 1994.

[11] P. Kirschenhofer and H. Prodinger. Multidimensional digital searching–alternative data structures. *Random Structures & Algorithms*, 5(1):123–134, 1994.

[12] P. Kirschenhofer, H. Prodinger, and W. Szpankowski. Multidimensional digital searching and some new parameters in tries. *Int. J. Foundations of Computer Science*, 4:69–84, 1993.

[13] D. E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, 1973.

[14] R. L. Rivest. Partial-match retrieval algorithms. *SIAM J. Computing*, 5(1):19–50, 1976.

[15] S. Roura. An improved master theorem for divide and conquer recurrences. In R. Gorrieri P. Degano and A. Marchetti-Spaccamela, editors, *Automata, Programming and Languages (ICALP' 97)*, volume 1256 of *LNCS*. Springer-Verlag, 1997.

[16] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison Wesley, 1990.

[17] W. Schachinger. The variance of a partial match retrieval in a multidimensional symmetric trie. *Random Structures & Algorithms*, 7(1):81–95, 1995.

[18] R. Sedgewick. *Algorithms in C*, volume 1. Addison-Wesley, 3rd edition, 1997.

[19] R. Sedgewick and Ph. Flajolet. *An Introduction to the Analysis of Algorithms*. Addison-Wesley, 1996.

[20] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis*. Cambridge Univ. Press, 1952.