Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-025-09985-7

RESEARCH

Encoding paths with binary arrays in a king's graph for error-free data transmission

Gökçe Çaylak Kayaturan[1] · Alexei Vernitski[2]

Accepted: 7 May 2025

Abstract
In this study, we have chosen the computer network with the shape of a king's graph. The king's graph G is defined as a set of edges, that is $E = \{((i, j), (p, q)) | i, p \in [0, M], j, q \in [0, N], M, N \in \mathbf{Z}, ((i, j), (p, q)) \; is \; an \; edge \iff i = p \;\; and \;\; j = q \pm 1 \;\; or \;\; i = p \pm 1 \;\; and \;\; j = q \;\; or \;\; i = p \pm 1 \;\; and \;\; j = q \pm 1\}$. We also set a delivery rule, in which the shortest paths in the graph are used for the message deliveries, to restrict the source consumption. Then, the paths are encoded in a way that we discover using binary arrays based on other well-known encoding methods. We prove that the path-coding method we present prevents errors denoted by false positives from the graph. Data transfer issues from computer science served as the motivation for this study.

Keywords King's graph · Shortest path · False positives · Computer network

Mathematics Subject Classification (2010) 05C90 · 68R10 · 68R05

1 Introduction

In this paper, we consider a graph denoted by king's graph $G = (V, E)$ that models a computer network [5]. We also consider a message forwarding scenario in graph. We suppose that each vertex v in the graph represents a computer and that communication between computers is supported through the shortest paths between two distinct vertices (see Fig. 1). The similar routing model has been introduced in [4]. The path is known before the message is sent, and once a message is forwarded through the path, it is not directed backward. This routing scenario in grid models was proposed by [6, 10, 15].

Some studies have been conducted to understand the structure of king's graphs in literature [8, 9]. A king's graph has been preferred as a model of tracking vehicles by [19]. Encoding models in king's graphs has been considered in the literature. The codes for vertices on a king's

✉ Gökçe Çaylak Kayaturan
gokcekayaturan@odu.edu.tr

Alexei Vernitski
asvern@essex.ac.uk

1 Ordu University, Cumhuriyet, Mustafa Kemal Blv. No:478, 52200 Altınordu/Ordu, Turkey

2 University of Essex, Colchester, UK

Published online: 15 May 2025

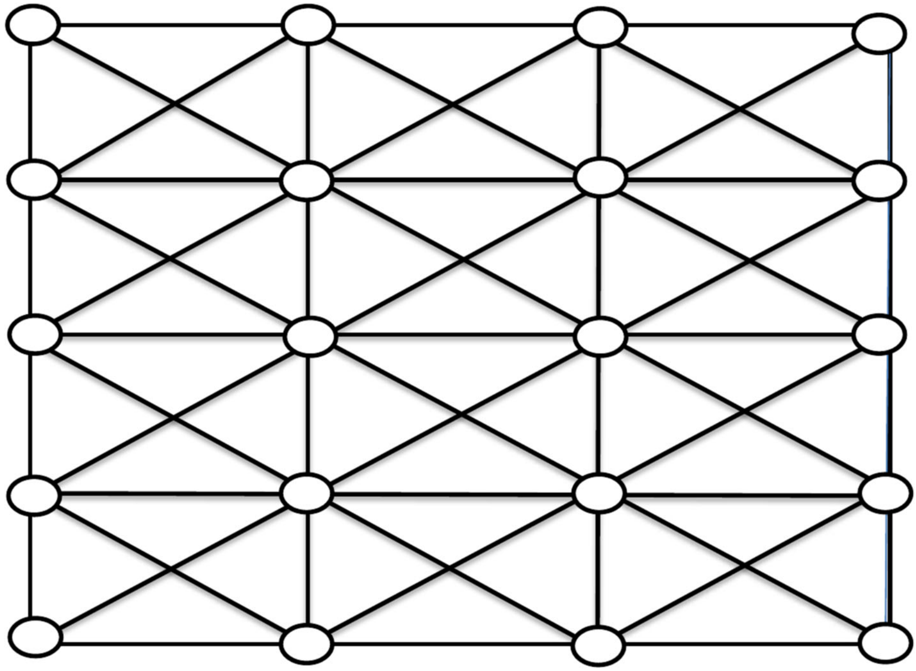

Fig. 1 A king's graph represents a computer network. [5]

graph have been identified in [7]. In this study, we have chosen the king's graph as a network model and we introduce an encoding method that encodes the links in the network having the shape of a king's graph. The labeling method introduced in this research is inspired by the study [2]. The method proposed by a Bloom filter is a technique to keep the data in compressed form. In practical applications, it is widely used to reduce network traffic, data mining, data transmission between parties in network models [3]. Recently Bloom filter techniques have found applications as effective alternatives for countering network attacks [18]. In recent research, the Bloom filter may still be preferable for building privacy-preserving techniques for some applications [1, 21]. Another study proposes a method to address path reconstruction and the identification of attack sources. This method is basically the use of the Bloom filter [16] As a labeling method, the Bloom filters save time and space. The Bloom filter is used to generate a solution for saving space in huge graphs presently [20].

We have invented path-labeling methods for the shortest paths in some graphs such as rectangular grids [10], hexagonal grids [12] and triangular grids [13]. We also consider another edge-labeling model in a king's graph [5]. Here, we have obtained more obtimised results than the results proposed in the study [5]. A path-labeling method has been considered in [11], unlike edge-labeling, vertex-labeling has been studied using Bloom filters in [11]. In this research, the paths' labels occupy less space than the labels obtained in [5]. This is an advantage for some applications that cares about the spaces, since the space can be saved by using Bloom filters [17].

The path encoding method presented in this study is based on a conventional approach, denoted by the Bloom filter, which is an approach of the encoding methods previously explored under a data transmission scenario between computers in a computer network represented by mathematical graphs. The mathematical theorems in the following sections demonstrate that the encoding approach we generalized does not result in errors. At the same time, it is demonstrated that this path-coding technique is space- and time-efficient.

The structure of this paper is that we introduce both standard Bloom filters and our labeling method, then we show how our labeling method does not produce any error, and finally we show why the method introduced in this paper is better than some other methods.

2 Standard bloom filter and its applications

The Bloom filter introduced in [2] is a procedure that represents a subset S of a universal set U, with a binary string of length $|U|$. The bits 1 in the string representing S correspond to an element x from the set U. Hence, the bit 1 in a certain place in the representative binary string of S means that $x \in S$. If we denote the label of element x by $\beta(x)$, then the label of the elements in S is shown as $\beta(S) = \cup_{x \in S} \beta(x)$. By this definition, it can be easily recognised that if $\beta(x) > (S)$, then we can say that $x \notin S$. However, if $\beta(S)$ does not represent S faithfully, then there may be a false positive in U adjacent to the subset S. In this case, it can be complicated to answer if $x \in S$ or $x \notin S$.

The probability of false positives is generally found with the calculation of $(1 - e^{\frac{-kn}{m}})^k$ [2] where $m = |U|$, $n = |S|$, and k is the number of bits 1 in Bloom filters. The choice of the parameter k affects the number of false positives. The optimum k can be found with $\lceil ln2 \times \frac{m}{n} \rceil$ which is received by computing the derivation of the formula for the probability of false positives [3].

Here, we handle the structural properties of the Bloom filters as the labels of the paths in the graphs. According to the routing scenario, the set $U = E$ represents the header of data transmitted from one computer to another in the graph. We assume that each edge e is encoded by a subset P of E. If the label of e is denoted by $\beta(e)$, then the labels that belong to the edges of a subset P of E are shown as $\beta(P) = \cup_{e \in P} \beta(e)$. The inquiry about the existence of the edges in the set P is extremely easy and fast by comparing the labels of the set and the edges. However, we can observe that $e \notin P$ where e is identified by $\beta(P)$. In this case, e is defined as a false positive of $\beta(P)$. According to the routing scenario, the edges linked to the chosen shortest path for data transmission are probably false positives of $\beta(P)$. If these kinds of edge, denoted by an adjacent edge, of the set P are not in the set P, one can say that the set is represented by $\beta(P)$ accurately. Accordingly, any computer on the chosen shortest path decides the next edge to which the message should be sent by comparing the header of the message and the labels of the edges.

3 Labels for the shortest paths in king's graphs

We assume that there are four types of imaginary lines crossing the edges in the direction of north ($\uparrow$), east ($\rightarrow$), north-west ($\nwarrow$) and north-east ($\nearrow$). (A similar approach has been discussed in research [5]). Each line is assigned by a positive integer. Lines oriented in the north-west direction are enumerated beginning from the bottom left corner (see Fig. 2) and lines directed north-east are enumerated beginning from the top left corner (see Fig. 2). Similarly, the lines directed to the way of north are enumerated beginning from the left-hand side to the right-hand side of the graph, and the lines directed to the way of east are counted from the bottom to the top of the graph.

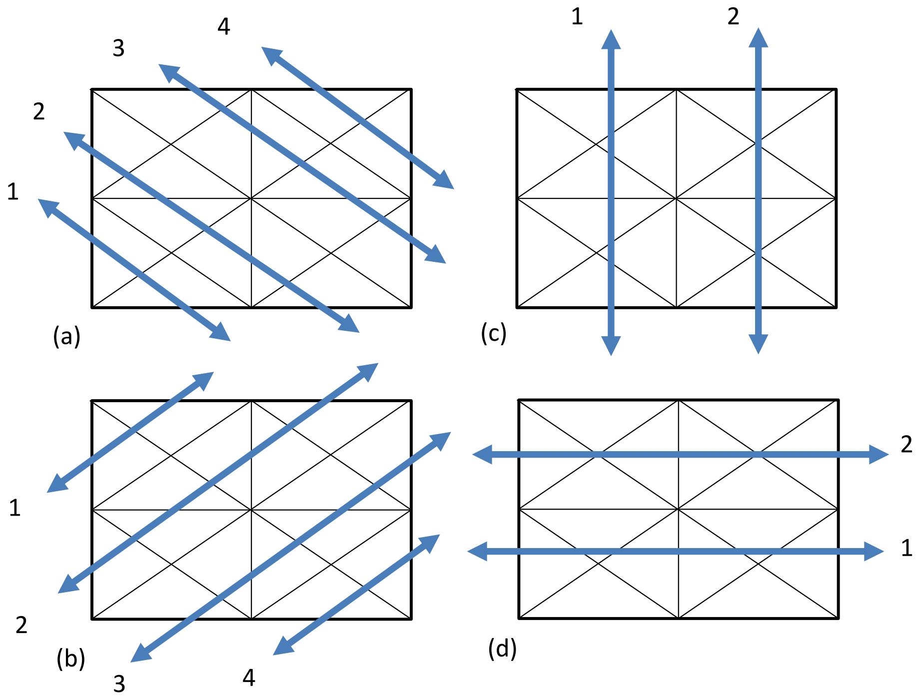

Fig. 2 The encoding lines of the edges in a king's graph [5]

We construct each Bloom filter of the edges with four main sections. Each section is divided into four- and two-bit subsections called after blocks. The first and second sections of the Bloom filters of the edges consist of four bits blocks and, the third and fourth sections consist of two bits blocks. In order to distinguish the orientations of the edges, we use the indicies v, h, nw, ne for the vertical, horizontal, and diagonal edges which lead to the north-west (or south-east) and north-east (or south-west), respectively. We aim to encode edges faithfully. Therefore, we use four bit blocks depending on the positions of the edges to represent them. The Bloom filters for the horizontal edge e_h, the vertical edge e_v, and the diagonal edge e_{nw}, e_{ne} contain blocks 1000, 0100, 0010 and 0001, respectively, in the first two sections. In addition, the Bloom filters of the diagonal edges e_{nw}, e_{ne} include blocks 10 and 01 respectively in the third and the fourth sections. Note that the encoding technique we introduce in this section differs from the method introduced in [5] with blocks indicating the edges and the length of these blocks in the third and fourth sections of the Bloom filters of the edges.

As an example, the Bloom filters of a diagonal edge e and a horizontal edge f in the king's graph of size 1×1 are shown in Fig. 3.

The lines oriented to the ways of north-west, north-east, north and east identify the places of the representative blocks of the edges in the first, second, third and fourth sections of Bloom filters of the edges, respectively. The numbers of the lines specify the block positions of 1000, 0100, 0010 and 0001; and 10 and 01 in the corresponding sections of the Bloom filters of the edges.

The number of lines in the north-west and north-east directions is $M + N$ if the graph is $M \times N$ in size, with M and N representing the number of horizontal and vertical edges on

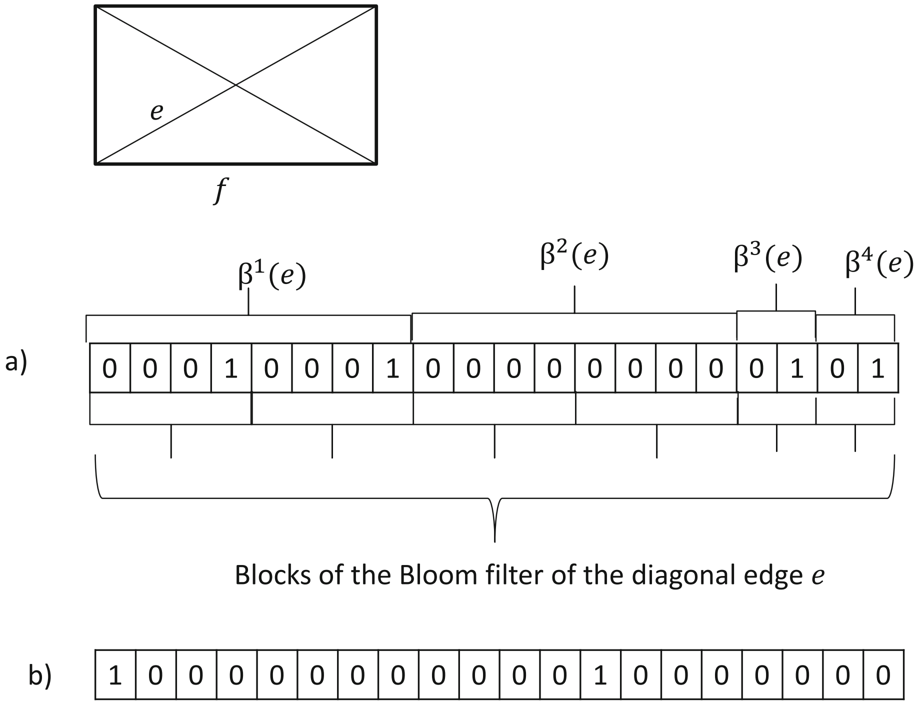

Fig. 3 The Bloom filters of the edges e and f from the 1×1 size king's graph. $\beta^i(e)$ where $i \in \{1, 2, 3, 4\}$ is the sections of the Bloom filter of the edges

one of the boundaries of the king's graph. In addition, the number of lines with north and east directions is M and N, respectively. Consequently, the number of bits in the Bloom filter of an edge is computed as $2 \times 4 \times (M + N) + 2M + 2N = 10 \times (M + N)$. This is denoted as the length of the Bloom filter.

According to the routing scenario, the shortest paths in the king's graph are used for the message distribution [5]. Hence, after all edges have their representative Bloom filters, the Bloom filter of a path is received after applying a bitwise OR operation to the Bloom filters of edges where the edges belong to the path.

4 Attributes of the encoded shortest paths in king's graphs

We take into consideration certain helpful aspects of the shortest pathways in a king's graph since the structural feature of a king's graph provides distinct Bloom filters for edges.

Lemma 4.1 *In a king's graph, the shortest path established between the vertices* $u = (r, s)$ *and* $v = (k, l)$ *where* $r \leq k - 1$ *and* $s \leq l - 1$ *does not include both vertical and horizontal edges* [5].

Proof Suppose a fragment P_F of a shortest path P in a king's graph G contains a diagonal edge e_d ending with vertices (r, s) and $(r + 1, s + 1)$ where $r, s \in \mathbb{Z}$. This edge can be denoted by $e_d = \{(r, s), (r + 1, s + 1)\}$. The diagonal edges can also have the form of $\{(r, s), (r - 1, s - 1)\}$ or $\{(r, s), (r + 1, s - 1)\}$ or $\{(r, s), (r - 1, s + 1)\}$. However, there is a path $P_F^{'}$ that contains one horizontal edge $e_h = \{(r, s), (r+1, s)\}$ and one vertical edge $e_v = \{(r+1, s), (r+1, s+1)\}$ between (r, s) and $(r+1, s+1)$. It is evident that the path $P_F^{'}$ has two edges that are longer than the single diagonal edge that connects these ends.

Suppose a path P with n edges between the vertices $u = (i, j)$ and $v = (k, l)$, where $r \leq k-1$ and $s \leq l-1$, contains vertical and horizontal edges.

We assume that the path P contains a horizontal edge e_h that is between the vertices $u = (r, s)$ and $(r+1, s)$ and a vertical edge e_v that lies between the vertices $(k, l-1)$ and $v = (k, l)$. Hence, the sequence of the vertices of the edges in the path P is $\{u = (r, s), (r+1, s), (r+2, s+1), (r+3, s+2), ..., (r+n-1, s+n-2), (r+n-1, s+n-1) = v = (k, l)\}$. The components of the vertices of an edge differ one unit from one vertex to another. If the edge is horizontal, its vertices on its first component of the pair of forms (x, y) diverge by one unit. Similar differences exist between the ordered pairs of vertices of a diagonal edge and a vertical edge: one unit separates them on the second component in the representative coordinates of a diagonal edge, while one unit separates them on the first.

Consider a shortest path P' between $u = (r, s)$ and $v = (r+n-1, s+n-1)$. The order of vertices of the edges in P' could be $\{u = (r, s), (r+1, s+1), (r+2, s+2), (r+3, s+3), ..., (r+n-2, s+n-2), (r+n-1, s+n-1) = v = (k, l)\}$ where all edges can be diagonal. There are $n-1$ edges in P'. This contradicts the number of edges in P located between u and v, since $|P'| < |P|$ where both paths lie between the same vertices.

Four compass orientations are present in the edges of a king's graph: north (or south), east (or west), north-east (or south-west), and north-west (or south-east). For example, the alignment of the vertical and horizontal edges is equally south or north and east or west, respectively. The diagonal edges are oriented to the north-east (south-west) and the north-west (south-east) in a king's graph.

Consider the shortest path between $u = (r, s)$ and $v = (k, l)$. If $|k-l| \geq 0$, then the path includes horizontal and diagonal edges or all horizontal edges, since the first component of representative points of a vertex changes from one vertex to the other vertex of a horizontal edge. Similarly, if $|l-k| \geq 0$, then the second component of representative points of a vertex shifts from one vertex of a vertical edge to the other vertex of a vertical edge, and the path is made up of vertical and diagonal edges or only vertical edges.

Consequently, the orientations of edges in a shortest path, where the path has both horizontal and diagonal edges, are east, south-east and north-east, or equivalently, west, north-west and south-west. Similarly, the orientations of edges in a shortest path that has both vertical and diagonal edges are north, north-east and north-west, or equally south, south-west and south-east.

Lemma 4.2 *In a king's graph, the number of edges in a shortest path between two distinct vertices $u = (i, j)$ and $v = (i+n, i+m)$ where $i, j, m, n \in \mathbb{Z}$ is $max(m, n)$.*

Proof Consider a shortest path P with vertices $u = (i, j)$ and $v = (i+n, i+m)$ where $i, j, m, n \in \mathbb{Z}$. If $n = 1$ and $m = 1$, then P includes one diagonal edge by Lemma 4.1. Hence, the number of edges on the path P is $max(m, n) = 1$.

Suppose that the number of edges on a shortest path in the king's graph is $max(m, n) = m$ where $m, n > 1$ between $u = (i, j)$ and $v = (i+n, i+m)$. In this case, we can take the inequality $m > n$.

Suppose $m > n$. If $m > n$, then $m-1 \geq n$. Hence, $max(m-1, n) = m-1$. We assume that the number of edges on the shortest path between $u = (i, j)$ and $v = (i+n, i+m)$ is m. The edges link the vertex $v = (i+n, i+m)$ in the shortest path lying between u and v can be $e_h = \{(i+m-1, j+n), (i+m, j+n)\}$ or $e_v = \{(i+m, j+n-1), (i+m, j+n)\}$

or $e_d = \{(i+m-1, j+n-1), (i+m, j+n)\}$. Therefore, the path from the vertex u to any of these vertices $\{(i+m-1, j+n)\}$ or $\{(i+m, j+n-1)\}$ or $\{(i+m-1, j+n-1)\}$ has $m-1$ edges. Therefore, the proof holds for $max(m-1, n) = m-1$ and the proof of the induction step is completed.

Lemma 4.3 *A diagonal line crosses at most one edge belonging to the shortest path in a king's graph* [5].

Proof Let $I = \{0, 1, 2, \ldots, n\}$ be the set of indices and e_{h_k} be the horizontal edge where $k \in I$. The horizontal edges crossed with the same line have an order of $e_{h_0} = \{(i, j), (i-1, j)\}$, $e_{h_1} = \{(i-1, j-1), (i, j-1)\}$, $e_{h_2} = \{(i-2, j-2), (i-1, j-2)\}, \ldots, e_{h_n} = \{(i-n, j-n), (i-n-1, j-n)\}$ where $i, j \in \mathbb{Z}$. Suppose a diagonal line crosses two of these horizontal edges $e_{h_k} = \{(i-k, j-k), (i-k+1, j-k)\}$ and $e_{h_l} = \{(i-l, j-l), (i-l+1, j-l)\}$ where both lie on a shortest path P. The path, which is the sub-path of P, between $(i-k, j-k)$ and $(i-l, j-l)$ includes the edges e_{h_k} and e_{h_l} and diagonal edges by Lemma 4.1. However, there can be another path between vertices $(i-k, j-k)$ and $(i-l, j-l)$ that is 2 edges shorter than the sub-path of P including two horizontal edges e_{h_k} and e_{h_l}. This contradicts the fact that the chosen path P including the edges e_{h_k} and e_{h_l} ,which are crossed by a diagonal line, is the shortest path.

Likewise, we may assume that a line intersects two vertical edges lying on a shortest path P' together. These edges are denoted by e_{v_m} and e_{v_n}, where the set of indices is $I = \{0, 1, 2, \ldots, t\}$ and $m, n \in I$. The vertical edges on one line must have the form $e_{v_0} = \{(p, q), (p, q-1)\}, e_{v_1} = \{(p-1, q-1), (p-1, q-2)\}, e_{v_2} = \{(p-2, p-2), (p-2, q-3)\}, \ldots, e_{v_n} = \{(p-n, q-n), (p-n, q-n-1)\}$ where $p, q \in \mathbb{Z}$. Let the edge e_{v_m} lie between the vertices $(p-m, q-m)$ and $(p-m, q-m-1)$ and the edge e_{v_n} link the vertices $(p-n, q-n)$ and $(p-n, q-n-1)$. The path, which is the sub-path of P', between the vertices $(p-m, q-m)$ and $(p-n, q-n)$ includes the edges e_{v_m} and e_{v_n} and diagonal edges by Lemma 4.1. Yet, there is another path between the vertices $(p-m, q-m)$ and $(p-n, q-n)$ that consists of only diagonal edges and this path is two edges shorter than the sub-path of P' including two vertical edges e_{v_m} and e_{v_n}. This contradicts the fact that the chosen path P' including the edges e_{v_m} and e_{v_n} ,which are crossed by a diagonal line, is the shortest path.

We assume that a line crosses two diagonal edges that are part of the same shortest path P''. The diagonal edges on the same line have the form $e_{d_0} = \{(r, s+1), (r+1, s)\}$, $e_{d_1} = \{(r-1, s), (r, s-1)\}, e_{d_2} = \{(r-2, s-1), (r-1, s-2)\}, \ldots$ where $r, s \in \mathbb{Z}$. If two of these diagonal edges with vertices $\{(r, s+1), (r+1, s)\}$ and $\{(r-2, s-1), (r-1, s-2)\}$ are on the sub-path of chosen path P'', then it is easy to see that there is another path between the vertices $(r, s+1)$ and $(r-2, s-1)$ which is shorter than the sub-path of chosen path P'' including the diagonal edges can be found.

The vertical and diagonal edges crossed by the same line have a form of $e_{v_0} = \{(r, s), (r, s-1)\}, e_{d_0} = \{(r-1, s), (r, s-1)\}, e_{v_1} = \{(r-1, s), (r-1, s-1)\}, e_{d_1} = \{(r-2, s-1), (r-1, s-2)\} \ldots$ and the horizontal and diagonal edges crossed by the same line have a form of $e_{h_0} = \{(r, s), (r+1, s)\}, e_{d_0} = \{(r-1, s), (r, s-1)\}, e_{h_1} = \{(r-2, s-1), (r-1, s-1)\}, e_{d_1} = \{(r-2, s-1), (r-1, s-2)\} \ldots$ where $r, s \in \mathbb{Z}$. When there are horizontal and diagonal edges or vertical and diagonal edges that are crossed by the same diagonal line in chosen path which is supposed to be the shortest path, a similar approach can be used. Consequently, the number of edges on the path between the goal vertices contradicts that the path is the shortest.

In contrast to this lemma, more than one diagonal edge of the shortest path may be intersected by horizontal or vertical lines.

Lemma 4.4 *A horizontal line does not intersect more than one edge in a shortest path that consists of vertical and diagonal edges. Additionally, a vertical line in a shortest path made up of diagonal and horizontal edges does not cross more than one edge.*

Proof A horizontal line r_h lies between two lines from the y-coordinate, which we say $y_k = k$ and $y_{k+1} = k + 1$.

Let P be the shortest path between vertices (i, k) and $(i + m, k + 1)$. Note that there might be another path between the lines $y_k = k$ and $y_{k+1} = k + 1$ that the vertices are (i, k) and $(i + m, k)$. If $i < i + m$, then the edges on the shortest path are diagonal or horizontal, or horizontal and diagonal. Since the first components of representative points of vertices, where the vertices belong to horizontal edges, change 1, and both components of representative points of vertices,where the vertices belong to diagonal edges, differ 1.

The vertices of the last edge on the shortest path P are $\{(i + m - 1, k), (i + m, k + 1)\}$ if the edge is diagonal or $\{(i + m - 1, k + 1), (i + m, k + 1)\}$ if the edge is horizontal. Note that horizontal edges cannot be intersected by horizontal lines since they are parallel. On the other hand, the edges between the lines bounded by the points of the y-coordinate are diagonal and vertical. If there is another vertical edge on the path P intersected by the line r_h with the vertices $\{(i + m, k), (i + m, k + 1)\}$, this contradicts the Lemma 4.1.

Similarly, a vertical line r_v lies between two points from x-coordinate that we say $x_l = l$ and $x_{l+1} = l + 1$. Suppose that a shortest path lies between these lines. The vertices of this path are (l, j) and $(l, j + n)$ or (l, j) and $(l + 1, j + n)$. When $j < j + n$, the shortest path contains diagonal or vertical edges, or both. A vertical line intersects horizontal and diagonal edges. However, a shortest path cannot have a horizontal edge if it only has diagonal and vertical edges according to the Lemma 4.1.

5 The bloom filters for the shortest paths in king's graphs

The edge labeling introduced in this research generates a shorter length of Bloom filters than the method proposed in [5]. In addition, the encoding method proposed in this paper differs in some properties from the method introduced in [5] (see Section 3 in this paper). Therefore, the Bloom filters of the shortest paths have differences and the topological properties of king's graph affect the structure of our Bloom filters of the edges. The following lemmas show this special appearance.

Lemma 5.1 *The first and second sections of the Bloom filter of the shortest path in a king's graph exhibit the form of a successive sequence for the blocks containing one bit* 1.

Proof By definition, a path consists of connected edges. The lines that we use for encoding edges are assigned by consecutive numbers and intersect edges belonging to the shortest paths. The blocks presenting the edges in the Bloom filters are placed in some block positions specified by the number of lines. By Lemma 4.3, a diagonal line does not cross more than one edge in a shortest path; hence, the blocks representing the edges in the shortest path are not placed in the same block positions in the Bloom filter of the shortest path. Hence, blocks that include one bit 1 have the form of a consecutive sequence in the relevant sections of the Bloom filters of shortest paths in a king's graph.

Lemma 5.2 *The Bloom filter of any shortest path may include the block* 11 *in its either the third or fourth sections.*

Proof According to the labeling technique we introduced, the vertical and horizontal lines designate the places for the blocks where the blocks represent the diagonal edges in the third and fourth sections of Bloom filters of the edges, respectively.

Suppose a vertical line r_v between the coordinates $x = i$ and $x = i+i$ intersects more than one diagonal edge on a shortest path. The sequence of vertices of diagonal edges intersected by the line r_v is (i, j), $(i + 1, j + 1)$, $(i, j + 2)$, $(i + 1, j + 3)$, ..., $(i, j + n)$. The number of diagonal edges between the vertices (i, j) and $(i, j+n)$ is n according to the Lemma 4.2. That is, the number of vertical edges between the vertices (i, j) and $(i, j + n)$. Therefore, all these diagonal edges may belong to a shortest path, and the vertical line r_v intersects all these diagonal edges whose number is more than one. Note that this path contains both types of diagonal edges oriented north-east and north-west. Consequently, representative blocks 01 and 10 of these diagonal edges are set in the same block position, which is numbered by the line r_v in the third section of the Bloom filter of the shortest path. By Lemma 4.4, a horizontal line does not intersect more than one edge from a shortest path containing vertical and diagonal edges. Hence, the fourth section of the Bloom filter of the shortest path, including vertical and diagonal edges, does not contain the block 11.

6 Removing false positives from the shortest paths in king's graphs

The header, which is sent to the recipient along with the message, specifies the message's route based on the routing scenario we are considering. When the message reaches a computer v on the way, it takes into account each edge connected to v and contrasts the edges incidental to v in all bits with the Bloom filters of the shortest path. As a result, the edges that are connected to the shortest pathways could be false positives in this network architecture. The encoding technique presented in [5] does not generate a false positive. Also, in this section, we show that the Bloom filter that we improve in this research for the shortest paths in a king's graph does not generate false positives.

Theorem 6.1 *In a king's graph, the Bloom filter applied to the shortest path does not generate a false positive.*

Proof A shortest path P and an edge e in P are given in a king's graph G. Suppose the edges f, g, h, i, j, k, l are directly connected to the edge e (see the Fig. 4). The edge e can be any of vertical, horizontal or diagonal edge, we follow the same approach to any type of edge chosen from the shortest path. We examine that e is horizontal in P (see the Fig. 4). Note that, routing scenarios state that the message is not forwarded back, when a computer lying on the shortest path gets the message, therefore if the horizontal edge e receives the message from the computer placed on its left hand-side, then the edge e carries the message to the computer placed on its right hand-side.

Since e is horizontal, then the following vertical edges k, g cannot be in the same shortest path by the Lemma 4.1. Then one might think these edges can be false positives in the Bloom filter of P denoted by $\beta(P)$. The diagonal encoding lines of the edge e cross the edges k and g. Hence, if the edges e and k or the edges e and g on the shortest path, then the representative blocks of the vertical and horizontal edges appear on the same block position $\beta(P)$. However, this contradicts the Lemma 5.1. It is found that $\beta(k), \beta(g) \nleq \beta(P)$.

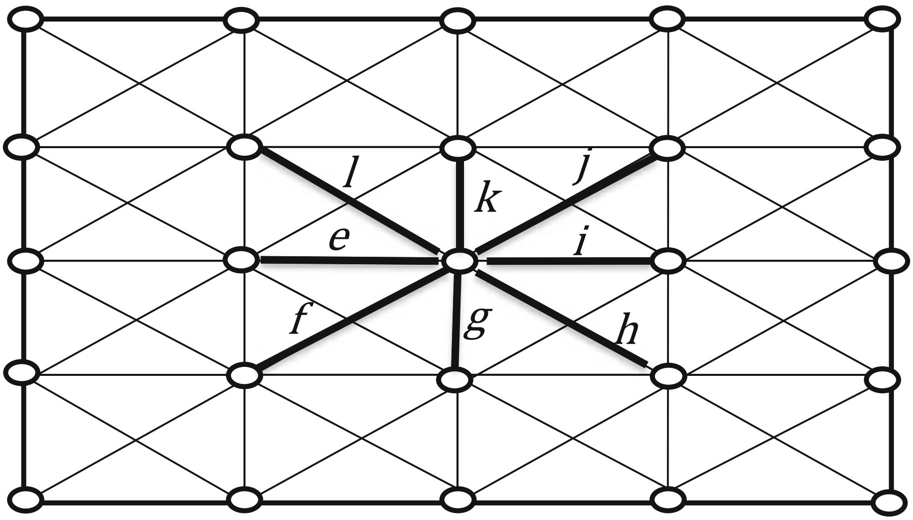

Fig. 4 The adjacent edges k, g, f, l, j, h, i to the edge e from a shortest path [5]

The edges l and f are crossed by the same vertical line. By Lemma 4.4, these edges can be false positives for $\beta(P)$. Also, these edges are crossed by diagonal lines with e. When the edge e and the edge f or the edge e and the edge l lie on P, then the descriptive blocks of these edges appear on the same block's place in the first or second sections of $\beta(P)$, respectively. Yet, this contradicts the Lemma 5.1. Therefore, any of the edges l, f do not exist in P and they are definitely not false positives.

Now, we examine if one of the next three edges h, i and j is on P with e. These three edges are crossed by the same vertical line and by Lemma 4.4 a vertical line does not cross more than one edge in the shortest path made up of the horizontal and diagonal edges. Hence the edges h, i and j cannot be on the shortest path all together.

Suppose the edge j following the edge e is on P. A diagonal line crosses the both edges j and i and a vertical line intersects both edges j and h (see Fig. 4), by Lemma 4.3 and Lemma 4.4 these edges cannot be on one shortest path. In this case the edges h and i can be false positives of $\beta(P)$. Since, the edge i exists on the same diagonal encoding line with the edge j, the representative 4-bit blocks of these edges lie on one block's place $\beta(P)$. Yet, this contradicts the Lemma 5.1. If $j \in P$ after the edge e, then it is obtained that $\beta(i) \nleq \beta(P)$. Also, the edges j and h are crossed by the same vertical line. If these edges exist in the same shortest path, then the representative 2-bit blocks of these edges would be seen on the same block's place in the third section of $\beta(P)$. Yet, the block 11 is found in the fourth section of the Bloom filter of the shortest path if it is composed of both horizontal and diagonal edges. By Lemma 5.2, both the third and fourth sections of the Bloom filter of the shortest path do not include the block 11 at the same time. Hence this is a contradiction and $\beta(h) \nleq \beta(P)$. Therefore, j is on P with e, then i and h are not false positives. We may assume i or j are on P with e, the similar process which proves that j is on the path can be applicable.

The approach which is considered for a horizontal edge from a shortest path as above is applicable to the other types of edges that are either vertical or diagonal. The same argument as above can be followed for the edge on the shortest path which is supposed to be diagonal or horizontal. We can make an assumption of the orientation of the edge on the shortest path, then we examine the following edges that are either on the shortest path or not. The adjacent edges are unquestionably not on the path when an edge is on the path. The Bloom filters of any of its neighboring edges are $\leq$ the Bloom filter of the shortest path in every bit position simultaneously.

7 Arguments of the encoding methods

7.1 One-bit per edge labeling

We assume that G is a king's graph with a set of edges E. In this way of path labeling, each edge $e \in E$ belonging to a path P, where $P \subset E$, is represented by one bit in $\beta(E)$. This approach also does not generate false positives, since in this encoding method the length of the Bloom filter is the number of edges in the graph and each edge is represented in a precise bit position in the Bloom filter.

In this path labeling method, the number of bits in the Bloom filter is $|E|$. Obviously, k denotes the number of edges in the path P, then $|E| = 4MN + M + N$ where the graph is in size of $M \times N$. On the other hand, according to our encoding method $|\beta(P)| = 10(M \times N)$. It is obtained that $10 \times (M + N) < 4MN + M + N$ for $M \leq 5$ and $N \leq 5$. Consequently, the parameters used for labeling edges limit the use of space in our path labeling approach for king's graphs where the graph size is larger than 5×5.

7.2 Standard bloom filter for edges in a king's graph

Here, we examine how parameters, which we use to build Bloom filters, work for standard Bloom filters. In the research that preceded ours, the k number of bits 1 of $\beta(e)$ of the edge e are placed in the string randomly. Hence, if we had labeled the edges with standard Bloom filters, we would probably obtain false positives. The formula $(1 - e^{\frac{-kn}{m}})^k$, where m is the length of the Bloom filter, $n = |S|$, and k is the number of bits 1 in the Bloom filters of the edges, is typically used to approximate the probability of false positives [2].

In our model $m = |\beta(e)|$, n is the number of edges in a shortest path and k is the number of bits 1 in the Bloom filters of the edges. The number of edges in a shortest path that connect a king's graph's two opposed corners, with maximum number of edges, is $max(M, N)$ by Lemma 4.2 where the king's graph is in the size of $M \times N$. In order to find the optimal k, we examine the edges with their maximum number on a shortest path and the length of the Bloom filters of the edges, since the aim is to get the probability of false positives as minimal as possible.

The optimum k is calculated with the formula $\lceil ln2 \times \frac{m}{n} \rceil$ to find the minimum probability of false positives, [17]. We find $k = \lceil ln2 \times \frac{10(M+M)}{M} \rceil = 14$ with the parameters of our model. However, in the labeling method we present k is 2 in the Bloom filter of the vertical and horizontal edges and 4 in the Bloom filter of the diagonal edges in our method.

If the edges of a king's graph are labeled with the standard Bloom filter using the parameters introduced in our encoding method, then the probability of false positives would be computed as $(1 - e^{\frac{-kn}{m}})^k = (1 - e^{\frac{-14M}{10(M+M)}})^{14} \approx 0,00006$. This is another advantage of the encoding method presented in this research.

There are $6 \times max(M, N)$ adjacent edges that are the possible false positives to the shortest path with the highest number of edges. Therefore, when the edges would have been encoded with the standard Bloom filter, the probability of nonexistent false positives would be $(0.99994)^{6M}$ when $M = N$.

7.3 Other encoding method for a king's graph

The encoding methods introduced in [5] and in this research for the shortest paths in king's graph do not generate false positives.

Furthermore, the labeling method we discovered in this study generates shorter Bloom filters than the method proposed in [5].The length of the Bloom filter introduced in [5] is $12 \times (M \times N)$, where M and N are the number of horizontal and vertical edges on the horizontal and vertical sides of a king's graph, respectively. The length of the Bloom filters in this paper, that is, $10 \times (M \times N)$, is approximately 16, 6% shorter than the other method in [5]. The method proposed in this study uses less space.

If the bits 1 in the Bloom filters of the edges are arranged in random places, then the labeling method may produce false positives. As an example, when the size of the graph is 11×10, the length of the Bloom filter is 210 where $m = 10(M \times N)$ and 252 where $m = 12(M \times N)$. Edges that have connections to the shortest pathways could be false positives. Figure 5 shows the false positive probability for the Bloom filter with $m = 252$ and $m = 210$ and $k = 4$, when the number of edges n on the shortest path changes between

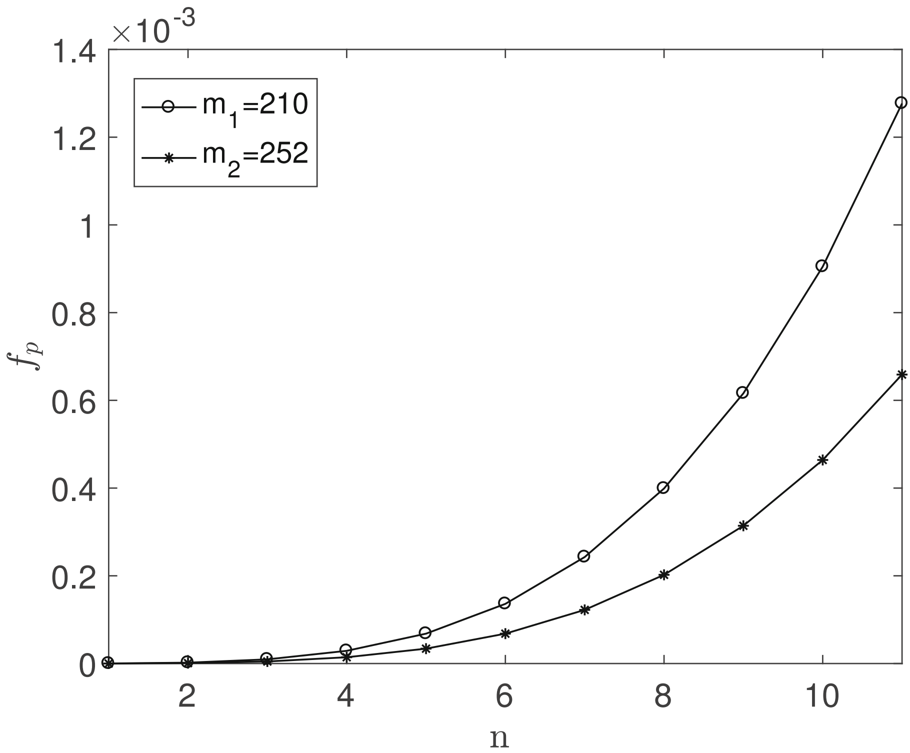

Fig. 5 The false positive probability for the Bloom filter with $m = 252$ and $m = 210$ and $k = 4$, when the number of edges in the paths takes different varieties

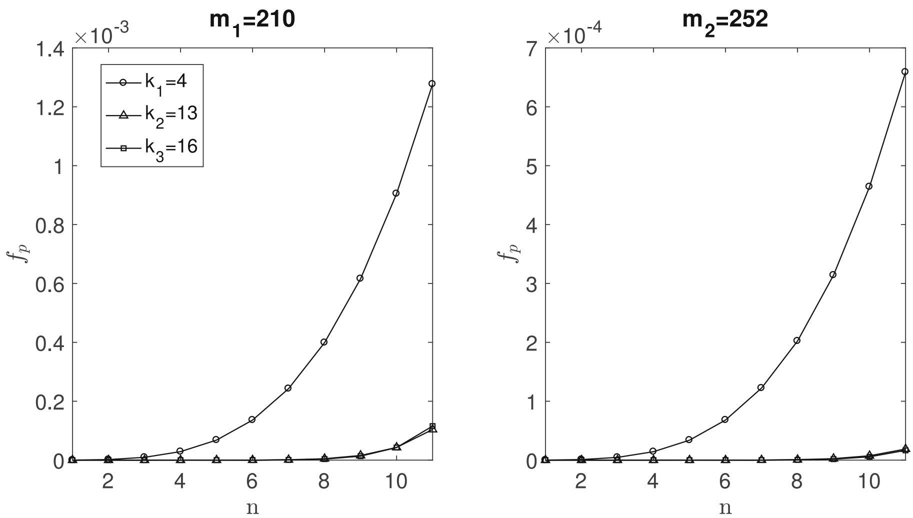

Fig. 6 The false positive probability for the Bloom filter of both $m = 210$ and $m = 252$ with $k = 4$, $k = 13$ and $k = 16$, when the number of edges in the paths takes different varieties

1 and 11 and the edges on the graph are labeled by using the standard Bloom filter. Note that the shortest path in a king's graph of size 11×10 contains at most 11 edges.

In order to minimize the number of false positives, the Bloom filter should have an optimal number of bits 1, that is, $\lceil ln2 \times \frac{m}{n} \rceil$, [17]. The optimal numbers of bits 1 with the parameters $m = 210$ and $m = 252$ are $k = \lceil ln2 \times \frac{10(M+N)}{max(M,N)} \rceil = 13$ and $k = \lceil ln2 \times \frac{12(M+M)}{max(M,N)} \rceil = 16$ in a 11×10 sized king's graph, respectively. As seen in Fig. 6, the false positive probability rate has a low level with optimum k for standard Bloom filters.

8 Conclusion

The motivation for this study came from data transfer problems between computers in computer networks. This study employs a rigorous theoretical approach and covers a novel method to encode the shortest paths in king's graphs using Bloom filters.

In theory, we proved that a path-coding method that we introduced based on the Bloom filter in a computer network that has the shape of a king's graph does not generate false positives regardless of the size of the graph. In this respect, it offers experts the convenience of choosing the size of the network. In applications, note that all parameters should maintain their optimal values to reduce network costs. If one is interested in networks having another structure, instead of king's graphs, it may be useful to refer to our earlier studies [10, 12–14].

In comparison with [5] this research of the Bloom filters for shortest paths in king's graph uses Bloom filters of a shorter length. This has advantages for users, as it uses less space than in previous research. Both methods are optimal in the sense that they do not produce false positives.

Some studies show that using standard Bloom filters is a reasonably good approach if they may produce false positives with a small amount. This is possible if $|U|$ is reasonably small [10]. We demonstrate that a model with fewer space requirements and no false positives can be obtained by building a Bloom filter under certain assumptions.

Author Contributions GCK was a major contributor in writing the manuscript. AV supervised the research. All authors read and approved the final manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Ahmed, N., Michelin, R.A., Xue, W., Putra, G.D., Ruj, S., Kanhere, S.S., Jha, S.: Dimy: Enabling privacy-preserving contact tracing. J. Netw. Comput. Appl. **202**, 103356 (2022)
2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**(7), 422–426 (1970)
3. Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. Internet Math. **1**(4), 485–509 (2004)
4. Vernitski, A., Reed, M., Carrea, L.: Optimized hash for network path encoding with minimized false positives. Comput. Netw. **58**, 180–191 (2014)
5. Kayaturan, G.C.: Representing Shortest Paths in Graphs Using Bloom Filters without False Positives and Applications to Routing in Computer Networks. PhD dissertation, University of Essex, UK, (2018)
6. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: High performance distributed computing, 2001. Proceedings. 10th IEEE International Symposium on, pp. 181–194. IEEE, (2001)
7. Dantas, R., Havet, F., Sampaio, R.M.: Minimum density of identifying codes of king grids. Discrete Math. **341**(10), 2708–2719 (2018)
8. Favaron, O., Fricke, G.H., Pritikin, D., Puech, J.: Irredundance and domination in kings graphs. Discrete Math. **262**(1), 131–147 (2003)
9. Ionascu, E.J., Pritikin, D., Wright, S.E.: k-dependence and domination in kings graphs. Am. Math. Mon. **115**(9), 820–836 (2008)
10. Kayaturan, G.C., Vernitski, A.: A way of eliminating errors when using bloom filters for routing in computer networks. In: Networks, ICN 2016. the fifteenth international conference on, pp. 52–57. IARIA, (2016)
11. Kayaturan, G.C.: Error elimination from bloom filters in computer networks represented by graphs. Fundamental J. Math. Appl. **5**(4), 240–244 (2022)
12. Kayaturan, G.Ç., Vernitski, A.: Routing in hexagonal computer networks: How to present paths by bloom filters without false positives. In: Computer science and electronic engineering (CEEC), 2016 8th, pp 95–100. IEEE, (2016)
13. Kayaturan, G.C., Vernitski A.: Encoding shortest paths in triangular grids for delivery without errors. In: Proceedings of the international conference on future networks and distributed systems, p. 7. ACM, (2017)
14. Kayaturan, G.C., Vernitski A.: Encoding shortest paths in graphs assuming the code is queried using bit-wise comparison. arXiv:1806.09442, (2018)

15. Li, X., Peng, L. and Zhang, C.: Application of bloom filter in grid information service. In: Multimedia information networking and security (MINES), 2010 international conference on, pp. 866–870. IEEE, (2010)
16. Ma, J., Su, W., Li, Y., Yao, F.: A low-overhead and high-precision attack traceback scheme with combination bloom filters. Secur. Commun. Netw. **2022**(1), 1639203 (2022)
17. Mitzenmacher, M.: Compressed bloom filters. IEEE/ACM Trans. Netw. (TON) **10**(5), 604–612 (2002)
18. Patgiri, R., Nayak, S., Muppalaneni, N.B.: Is bloom filter a bad choice for security and privacy? In: 2021 International conference on information networking (ICOIN), pp. 648–653. IEEE, (2021)
19. Punithan, M.X., Seo, S.W.: King's graph-based neighbor-vehicle mapping framework. IEEE Trans. Intell. Trans. Syst. **14**(3), 1313–1330 (2013)
20. Saha, A., Sengupta, N., Ramanath, M.: Reachability in large graphs using bloom filters. In: Proceedings - 2019 IEEE 35th international conference on data engineering workshops, ICDEW 2019, pp. 217–224. ICDEW, (2019)
21. Yin, W., Yuan, L., Ren, Y., Meng, W., Wang, D. and Yin, Q.W.W.: Differential cryptanalysis of bloom filters for privacy-preserving record linkage. IEEE Transactions on Information Forensics and Security, (2024)