Computer Vision Studies
Volume 2, Issue 1, No. 2
Received 3 March 2023; Accepted 22 March, 2023
Published online 11 April 2023

Algorithms for closeness, additional closeness and residual closeness

Chavdar Dangalchev
Institute of Mathematics and Informatics Bulgarian Academy of Sciences, Sofia, Bulgaria
Email: dangalchev@hotmail.com

Abstract. The residual and additional closeness are very important characteristics of graphs. They are measures of graphs' vulnerability and growth potentials. Calculating the closeness, the residual, and the additional closeness of graphs is a difficult computational problem. In this article we propose an algorithm for additional closeness and an approximate algorithm for closeness. Calculating the residual closeness of graphs is the most difficult of the three closenesses. We use Branch and Bound like algorithms to solve this problem. In order for the algorithms to be effective, we need good upper bounds of the residual closeness. In this article we have calculated upper bounds for the residual closeness of 1-connected graphs. We use these bounds in combination with the approximate algorithm to calculate the residual closeness of 1-connected graphs. We have done experiments with randomly generated graphs and have calculated the decrement in steps, delivered by the proposed algorithm.

Keywords. Closeness, residual closeness, additional closeness.

1. Introduction

Research of networks is a very important subject in different fields like mathematics, informatics, social science, chemistry etc. Networks' centrality measures identify the importance of the roles which every vertex plays in the network. These roles are different depending on the characteristic of a network we want to explore: vulnerability, access and spread of information, growth potentials etc. Most of the centrality measures (like degrees, betweenness, and eigenvector) can have very big differences between the values of two neighboring vertices. One of the advantages of the closeness is that it is changing more "smoothly" from vertex to vertex (see section 3).

In a work on network vulnerability [1], Dangalchev proposed one of the most sensitive characteristics - residual closeness. It measures the closeness 1 of a graph after removing a vertex or a link (edge). The definition for the closeness of vertex i in simple undirected graphs, used in [1], is:

$$\mathrm{C(i)} = \sum_{j\neq i} 2^{-d(i,j)} \quad (1)$$

In the above formula, $d(i,j)$ is the standard distance between vertices i and j. The graph G closeness is the sum of all the vertices' closenesses:

$$\mathrm{C}(G) = \sum_i \sum_{j\neq i} 2^{-d(i,j)} \quad (2)$$

The advantages of the above definition are that it can be used for not connected graphs and it is convenient for creating formulae for graph operations (see next section).

Let r and s be a pair of connected vertices in graph G and graph $G_{r,s}$ be the graph, constructed by removing link (r, s). Let $d_{r,s}(i,j)$ be the distance between vertices i and j in graph $G_{r,s}$. Using formula (2), we can calculate the closeness of graph $G_{r,s}$:

$$\mathrm{C}(G_{r,s}) = \sum_i \sum_{j\neq i} 2^{-d_{r,s}(i,j)} \quad (3)$$

The link residual closeness LR, a measure of graph G vulnerability, is defined in [1] as:

$$LR(G) = min_{r,s}\{C(G_{r,s})\} \quad (4)$$

In a similar way we define vertex residual closeness VR.

Let p and q be a pair of not connected vertices in graph G and graph $G_{p,q}$ be the graph, constructed by connecting p and q. The additional closeness is a measure of graph G growth potential and it is defined in Dangalchev [2] as:

$$A(G) = max_{p,q}\{C(G_{p,q})\} \quad (5)$$

Bounds for additional closeness are proven in [2] - for any graph G the additional closeness satisfies:

$$C(G) + \frac{1}{2} \leq A(G) \leq C(G) + (1 + C(k))^2 \quad (6)$$

where k is the vertex of G with the maximal closeness. The right side of the inequality is satisfied as equality for not connected graphs. It could be approached for some graphs (see [3]). In this article we will give upper bounds for residual closeness.

Unfortunately, finding the closeness of a graph is time and operations consuming. For example, Floyd–Warshall algorithm for finding the graph distances [4] has time complexity O (n^3). With some restriction on the type of graphs, the time could be decreased (see [5, 6]). The residual closeness is even more difficult to be calculated – O ($m.n^3$), where n is the number of vertices and m is the number of links of the graph.

Having the distances of a graph, the additional closeness can be easily calculated. In this article we present a way to decrease the operations for calculation of the additional closeness. We also propose an approximate algorithm for closeness with guaranteed precision. Here we present an algorithm for residual closeness of 1-connected graphs, based on the upper bounds and the approximate algorithm.

2. Previous Results

Let us consider some graph, received after operations between two graphs.

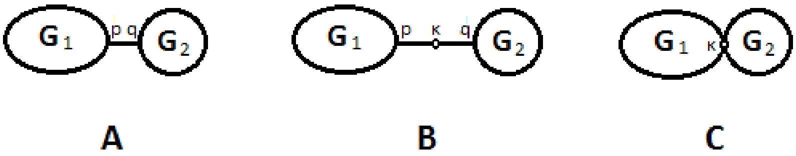

Figure 1. 1-connected graphs

All three graphs from Figure 1 are 1-connected: Graphs from Figure 1 cases A and B have vertex-connectivity and edge-connectivity equal to 1. Removing vertex p and its edges, or vertex q and its edges, or edge (p,q) will disconnect the graph from Figure 1 A. Removing one of vertices p, k or q or one of the edges(p,k) and (k,q) will disconnect the graph B. Graph G is 1-vertex-connected - removing vertex k and its edges will disconnect it.

Let vertex p (with vertex closeness $C(p)$) from graph G_1 and vertex q (with closeness $C(q)$) from graph G_2 be connected (Fig. 1 case A). The formula for closeness of graph A is given in [1]:

$$C(GA) = C(G1) + C(G2) + (1 + C(p))\,(1 + C(q)) \tag{7}$$

Let graphs G_1 and G_2 be connected by 2 links: vertex p (with vertex closeness $C(p)$) from graph G_1 is connected to a new vertex k, which is connected to vertex q (with closeness $C(q)$) from graph G_2 (Figure 1 case B). The formula for closeness of graph B, also given in [1], is:

$$C(GB) = C(G1) + C(G2) + 2 + C(p) + C(q) + 1\ 2\,(1 + C(p))\,(1 + C(q)) \tag{8}$$

Let vertices p (with vertex closeness $C(p)$) from graph G_1 and q (with vertex closeness $C(q)$) from graph G_2 coincide into one vertex k to create graph G_c (Figure 1 case C) . The formula for closeness of graph G_c is given in [2]:

$$C(G_c) = C(G_1) + C(G_2) + 2C(p)C(q) \tag{9}$$

After removing any vertex k from graph G, the smallest possible decrement in closeness is $2C(k)$. In [7] is proven a condition for it:

Theorem 1. If a vertex k does not belong to any unique geodesic linking any other 2 vertices in graph G then:

$$C(G\backslash k) = C(G) - 2C(k) \tag{10}$$

Some other results, related to closeness, additional closeness, and residual closeness can be found in [8-27].

3. Adjacent values of centrality measures

Let us investigate the changing of values between two adjacent vertices for the major centrality measures. It can easily be shown that degrees, betweenness and eigenvector centralities of two adjacent vertices can be very different.

Let us, for example, consider a star graph S_n with n vertices. The degree of the center is $n-1$ and the degree of any leaf is 1: the normalized degree centrality of a leaf is $1/(n-1)$. The normalized betweenness of the center is 1 while of a leaf is 0. If we normalize the eigenvector so its component of the center is 1 then the leaves' components of the eigenvector are $1/\sqrt{n-1}$.

In contrast, the closeness centrality is not changing that much. The closeness of the center is $(n-1)/2$ and the closeness of a leaf is $n/4$. The normalized value of the leaf's closeness is $1/2 + 1/(2n - 2)$.

We can see that the normalized values of a leaf's degree, betweenness and eigenvector are close to (approaching) 0, while the normalized closeness is greater than 0.5. The difference in the closeness of two adjacent vertices is smaller not only for the star graphs but for any graph.

Let there be link (p,q) between vertexes p and q of graph G.

Theorem 2. The closeness of vertex p satisfies:

$$\frac{1}{2}C(q) + \frac{1}{4} \leq C(p) \leq 2C(q) - \frac{1}{2} \tag{11}$$

Proof. From the triangle inequality, using the fact that the vertices p and q are linked, we have:

$$d(p,i) \leq d(p,q) + d(q,i) = 1 + d(q,i) \tag{12}$$

The closeness of vertex p can be restricted:

$$C(p) = \sum_{i\neq p} 2^{-d(p,i)} = 2^{-d(p,q)} + \sum_{i\neq p,q} 2^{-d(p,i)} \geq \frac{1}{2} + \sum_{i\neq p,q} 2^{-1-d(q,i)} = \frac{1}{2} + \frac{1}{2}\left(\sum_{i\neq p,q} 2^{-d(q,i)} + 2^{-d(p,q)}\right) - \frac{1}{4}$$
$$= \frac{1}{4} + \frac{1}{2}\sum_{i\neq q} 2^{-d(q,i)}$$

or

$$C(p) \geq \frac{1}{4} + \frac{1}{2}C(q) \tag{13}$$

Writing inequality (13) for vertex q we receive:

$$C(q) \geq \frac{1}{4} + \frac{1}{2}C(p) \tag{14}$$

or

$$C(p) \leq 2C(q) - \frac{1}{2} \tag{15}$$

Combining (13) and (15) we prove Theorem 2.

From Theorem 2 we can obtain:

Corollary 1. The ratio in closeness between two adjacent vertices is greater than 0.5.

The values of closeness from vertex to vertex are changing much smoothly than the ones of the other major central measures. As we have seen, for a star graph the ratio in degrees of two adjacent vertices could be in magnitude of $1\backslash n$, for the eigenvector the ratio could be in magnitude of $1\backslash\sqrt{(n)}$, for betweenness it could be 0. This "smoothness" of closeness is in the foundation of formulae (7), (8), and (9) and the bounds proven in the next sections.

4. Approximate algorithms for closeness

In this section we will describe an approximate algorithm for closeness with known (guaranteed) precision. The errors of this algorithm are very easy to be calculated. Let $N_k(p)$ be the k-neighborhood of vertex p in graph G (the vertices with distance to p equal to k, $k > 0$):

$$Nk(p) = \{v \in V : d(p,v) = k\} \qquad (16)$$

Let us define an approximation of the closeness $C_k(p)$ of vertex p as:

$$C_k(p) = \sum_{i=1}^{k} \sum_{v\in N_i(p)} 2^{-d(p,v)} = \sum_{i=1}^{k} 2^{-i}|N_i(p)| \qquad (17)$$

The error in closeness of vertex p after k steps is:

$$E_k(p) = C(p) - C_k(p) = \sum_{d(p,v)>k} 2^{-d(p,v)} \qquad (18)$$

Let us denote the number of vertices on distance to vertex p greater than k with m:

$$m = n - \sum_{0\le i\le k}|N_i(p)| \qquad (19)$$

The maximal possible error $E_{max}(p)$ of closeness is when all these m vertices are on distance $k + 1$:

$$E_k(p) \le E_{max}(p) = 2^{-k-1}(n - \sum_{i\le k}|N_i(p)|) = m \cdot 2^{-k-1} \qquad (20)$$

The minimal possible error $E_{min}(p)$ is zero when all m vertices are disconnected (in case of disconnected graph).

$$E_k(p) \ge E_{min}(p) \ge 0 \qquad (21)$$

In case of connected graph, the minimal error is when all remaining m vertices make a path, connected to p:

$$E_k(p) \ge Emin(p) = 2^{-k}(2^{-1} + 2^{-2} + \cdots + 2^{-m}) = 2^{-k}(1 - 2^{-m}) \qquad (22)$$

Combining the bounds for the error of the vertex p we receive:

$$1 - 2^{-m} \le 2^k E_k(p) \le \frac{m}{2} \qquad (23)$$

The total error in closeness E_k of graph G is:

$$\sum_p E_{min}(p) \le E_k = \sum_p E_k(p) \le \sum_p E_{max}(p) \qquad (24)$$

The approximate algorithm implements breadth-first traversal - it starts with calculation of the closeness of every vertex by including the neighbors on distance 1. Then we include the neighbors on distance 2, etc. At every step we calculate the total error (sum of all vertices' errors). When the total error (or its upper limit) is less than in advance given value, we stop the process.

We can see (formula 23) that the errors are decreasing exponentially with increasing of the maximal distance, hence the algorithm's performance is very suitable for graphs with bigger diameters (like path graphs).

5. Algorithm for additional closeness

Let us have a graph G with n vertices and m links. Let us add a new link (p, q) to receive graph G'. The number of possible new links are $n(n-1)/2 - m$. If we start to calculate the closeness of graph G' from scratch, we will have to make too many calculations.

Instead, we can calculate the closeness of graph G' using the closeness $C(G)$ of graph G. We have already calculated the matrix of distances $d(i, j)$ of graph G. We can calculate the distances d' between vertices i and j in graph G', after adding link (p, q):

$$d'(i,j) = min\{d(i,j), d(i,p) + 1 + d(q,j), d(i,q) + 1 + d(p,j)\} \qquad (25)$$

The closeness of graph G' is:

$$C(G') = C(G) + \sum_i \sum_{j\ne i}\left(2^{d'(i,j)} - 2^{d(i,j)}\right) \qquad (26)$$

Using the above formula we can decrease the number of calculations hundreds of times, depending on the size of the graph, compared to the calculation of distances from scratch.

We can decrease the number of calculated distances even more. For every two vertices p and q of graph G, all vertices can be divided into 3 sets:

$$Np = \{v \in V : d(p,v) < d(q,v)\} \qquad (27)$$

$$Nq = \{v \in V : d(p,v) > d(q,v)\} \qquad (28)$$

$$N = \{v \in V : d(p,v) = d(q,v)\} \qquad (29)$$

The set Np contains vertex p and the set Nq includes vertex q.

The closeness of graph G can be calculated as:

$$C(G) = \sum_i \sum_{j\ne i} 2^{-d(i,j)} = \sum_{i\in N_p} \sum_{j\in N_p, j\ne i} 2^{-d(i,j)} + \sum_{i\in N_q} \sum_{j\in N_q, j\ne i} 2^{-d(i,j)}$$

$$+ 2\sum_{i\in N_p} \sum_{j\in N_q} 2^{-d(i,j)} + \sum_{i\in N} \sum_{j\in N, j\ne i} 2^{-d(i,j)} + 2\sum_{i\in N} \sum_{j\in N_p\cup N_q} 2^{-d(i,j)}$$

The upper formula is true for any pair of vertices p and q - connected or not connected.

Let us consider not connected vertices p and q. Let the distances in graph G', constructed from graph G by adding link

(p, q), be $d'(i, j)$. The shortest path $d'(i, j)$, when both i and j belong to Np, cannot include link (p, q) hence $d'(i, j) = d(i, j)$. Also, $d'(i, j) = d(i, j)$ when both i and j belong 7 to Nq, or N. The same is true when vertex i belongs to N and vertex j belongs to $N_p \cup N_q$. The closeness of graph G' can be calculated as:

$$C(G') = \sum_{i\in N_p}\sum_{j\in N_p, j\neq i} 2^{-d(i,j)} + \sum_{i\in N_q}\sum_{j\in N_q, j\neq i} 2^{-d(i,j)} + 2\sum_{i\in N_p}\sum_{j\in N_q} 2^{-d'(i,j)} + \sum_{i\in N}\sum_{j\in N, j\neq i} 2^{-d(i,j)} + 2\sum_{i\in N}\sum_{j\in N_p\cup N_q} 2^{-d(i,j)}$$

The difference between the two closenesses is:

$$C(G) - C(G') = 2\sum_{i\in N_p}\sum_{j\in N_p, j\neq i} 2^{-d(i,j)} - 2\sum_{i\in N_p}\sum_{j\in N_q} 2^{-d'(i,j)} \quad (30)$$

We can calculate the additional closeness by comparing only the distances between Np and Np:

$$C(G') = C(G) + 2\sum_{i\in N_p}\sum_{j\in N_q}(2^{d'(i,j)} - 2^{d(i,j)}) \quad (31)$$

and decrease the number of calculated distances.

6. Disconnecting a graph

Calculating the residual closeness is much more difficult than calculating the additional closeness. One can expect that removing a link that disconnects a graph will provide the residual closeness. In general, this is not true. Removing a link, which disconnects a graph, is not a sufficient condition for residual closeness. For example, a path graph can be disconnected by removing any of its links, but only the removing of the central link(s) provides the residual closeness.

Removing a link, which disconnects the graph, is not also a necessary condition for residual closeness. Below is an example for it.

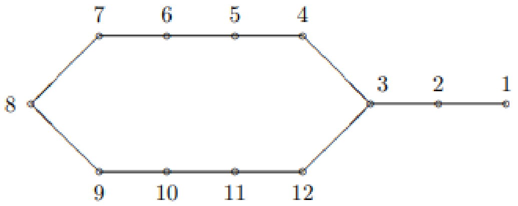

Figure 2. Not necessary condition for RC

The graph G from Figure 2 consists of a path graph P_2 (vertices 1 and 2), connected with link (2, 3) to a cycle graph C_{10} (vertices from 3 to 12). For their closenesses it is true:

$$C(P_2) = 1 \quad (32)$$

$$C(C_{10}) = 10\left(2\cdot\frac{1}{2} + 2\cdot\frac{1}{4} + 2\cdot\frac{1}{8} + 2\cdot\frac{1}{16} + \frac{1}{32}\right) = 10\left(1\frac{29}{32}\right) = 19\frac{1}{16} \quad (33)$$

The closeness of graph G, using formula (7), is:

$$C(G) = C(P_2) + C(C_{10}) + \left(1 + \frac{1}{2}\right)\left(1 + 1\frac{29}{32}\right) = 24\frac{27}{64} \quad (34)$$

Removing link (2,3) will reduce the closeness to:

$$C(G\backslash(2,3)) = C(P2) + C(C_{10}) = 20\frac{1}{16} \quad (35)$$

Removing link (3,4) will create a path graph P_{12}. The formula for closeness of path graphs, given in [1], is:

$$C(P_k) = 2k - 4 + 2^{2-k} \quad (36)$$

Replacing in the above formula k with 12 we receive for the closeness:

$$C(G\backslash(3,4)) = C(P_{12}) = 2\cdot 12 - 4 + 2^{-10} = 20\frac{1}{2^{10}} \quad (37)$$

Removing link (2,3) reduces the closeness of G from 24.421875 to 20.0625, while removing link (3,4) reduces it to 20.0009765625 - removing a link, which does not disconnect the graph, produces the residual closeness.

We have given examples that disconnecting a graph is not necessary nor sufficient condition for the residual closeness. In spite of these examples, the closeness of a disconnected graph is a very good upper limit for the residual closeness (compare 20.0625 to 20.0009765625).

7. An upper bound for residual closeness (Figure 1A)

In the next sections, we will consider the reversed situation of formulae (7), (9), and (9): instead of increasing the number of links or vertices we will decrease them. The obvious bound for residual closeness from below is zero, which is satisfied as equality for path P_2. The obvious upper bound is $LR(G) \leq C(G) - 0.5$, satisfied for complete graphs.

The graph, shown in Figure 1 A, is 1-connected - it will be disconnected after removing link (p, q) or one of the vertices p or q.

Theorem 3. *Let us have a connected graph G with closeness C(G). Let after removing the link between vertices p (with vertex closeness C(p)) and q (with vertex closeness C(q)) the resulting graph G' be disconnected (figure 1 A). Then the closeness of graph G' is*:

$$C(G') = C(G) - \frac{4}{9}5C(p)C(q) - 2C^2(p) - 2C^2(q) + C(p) + C(q) + 1 \quad (38)$$

Proof. Let us create a graph G by connecting graphs G_1 and G_2 - linking vertex p of graph G_1 (with vertex closeness $C_1(p)$ within graph G_1) with vertex q of graph G_2 (with closeness $C_2(q)$) - case A from Figure 1.

The closeness of vertex p within graph G is:

$$C(p)=\sum_{i\neq p}2^{-d(p,i)}=\sum_{i\in G_1,i\neq p}2^{-d(p,i)}+\sum_{i\in G_2}2^{-d(p,i)}=C_1(p)+2^{-d(p,q)}+\sum_{i\in G_2,i\neq q}2^{-1-d(q,i)}=C_1(p)+\frac{1}{2}+\frac{1}{2}C_2(q)$$

The corresponding result for $C(q)$ is:

$C(q)=\ C_2(q)+\frac{1}{2}+\frac{1}{2}C_1(p)$ (39)

We can solve a linear system with 2 variables ($C_1(p)$ and $C_2(q)$) and the two equations, given above. Eliminating $C_2(q)$ from the upper two equations we obtain:

$C(q)=2\ C(p)-2C_1(p)-1+\frac{1}{2}+\frac{1}{2}C_1(p)$ (40)

$C_1(p)=\frac{4C(p)-2C(q)-1}{3}$ (41)

The corresponding presentation for $C_2(q)$ is:

$C_2(q)=\frac{4C(q)-2C(p)-1}{3}$ (42)

The formula (7) for graph G is:

$C(G)=\ C(G_1)+C(G_2)+(1+C_1(p))(1+C_2(q))$ (43)

Now let us remove the link between vertices p and q:

$$C(G')=\ C(G_1)+C(G_2)=C(G)-\big(1+C_1(p)\big)\big(1+C_2(q)\big)=C(G)-\frac{1}{9}(4C(q)-2C(p)+2)(4C(p)-2C(q)+2)$$

$$=C(G)-\frac{4}{9}(5C(p)C(q)-2C^2(p)-2C^2\ (q)+C(p)+C(q)+1)$$

This finishes the proof.

The formula for closeness of Theorem 3 is a necessary condition for 1- connectivity. A hypothesis is that this formula is also a sufficient condition (like Theorem 1, which is a sufficient and a necessary condition).

Using the above formula and the fact that the closeness $C(G')$ is an upper bound of the link residual closeness we receive:

Corollary 2. *The link residual closeness (LR) of graph G (from Figure 1 A) satisfies*:

$LR(G)\ \le\ C(G)-\frac{4}{9}(5C(p)C(q)-2C^2(p)-2C^2\ (q)+C(p)+C(q)+1)$ (44)

If instead of link (p, q) we remove vertex p with its links then the closeness will decrease even more - instead of $C(G1)$ we will have $C(G_1\backslash p)$. The decrement will be small if vertex p is pendant or does not belong to any unique geodesic linking any other two vertices in graph G_1. The smallest decrement, according to Theorem 1, will be $2C_1$(p). Subtracting $2C_1(p)$ (formula 41) from formula (38) of Theorem 3, we can receive:

Corollary 3. *The vertex residual closeness (VR) of graph G (from Figure 1 A) satisfies*:

$VR(G)\le\ C(G)-\frac{20C(p)C(q)-8C^2(p)-8C^2\ (q)+28C(p)-8C(q)-2}{9}$ (45)

A similar bound can be obtained if we remove vertex q.

8. An upper bound for residual closeness (Figure 1B)

The link residual closeness in case B from fig. 1 could be restricted using case A and Corollary 2. For the vertex residual closeness, we need:

Theorem 4. *Let us have a connected graph G with closeness C(G). Let vertex k have only 2 links: to vertices p and q with closeness C(p) and C(q) correspondingly. Let after removing vertex k and its links the graph be disconnected (case B from fig. 1). Then the closeness C(G′) of the resulting, disconnected graph G′ is*:

$C(G')=\ C(G)-\frac{136C(p)C(q)-32\left(C^2(p)+C^2\ (q)\right)+216\left(C(p)+C(q)\right)+198}{225}$ (46)

Proof. The proof is similar to the one of Theorem 3 (using formula 8, instead of formula 7). Let vertex p of graph G_1 has closeness $C_1(p)$ and vertex q of graph G_2 has closeness $C_2(q)$ (see case B from Figure 1).

The closeness of vertex p within graph G is:

$C(p)=\sum_{i\in G_1,i\neq p}2^{-d(p,i)}+2^{-d(p,k)}+2^{-d(p,q)}+\sum_{\in G_2}2^{-2-d(p,i)}=C_1(p)+2^{-1}+2^{-2}+\frac{1}{4}C_2(q)=C_1(p)+\frac{1}{4}(3+C_2(q))$

The corresponding result for $C(q)$ is:

$C(q)=C_2(q)+\frac{1}{4}(3+C_1(p))$ (47)

Solving a linear system with 2 variables ($C_1(p)$ and $C_2(q)$) we receive the presentations:

$C_1(p)=\frac{16C(p)-4C(q)-9}{15}$ (48)

$C_2(q)=\frac{16C(q)-4C(p)-9}{15}$ (49)

Replacing in formula (8) the above values we receive:

$$C(G')=\ C(G_1)+C(G_2)=\ C(G)-2C_1(p)-C_2(q)-\frac{1}{2}\big(1+C_1(p)\big)\big(1+C_2(q)\big)=$$

$$C(G)-\frac{136C(p)C(q)-32\left(C^2(p)+C^2\ (q)\right)+216\left(C(p)+C(q)\right)+198}{225},$$

which finishes the proof.

From Theorem 4 we receive:

Corollary 4. *The vertex residual closeness of graph G (from Figure 1 B) satisfies*:

$$VR(G) \leq C(G) - \frac{136C(p)C(q) - 32\left(C^2(p)+C^2(q)\right) + 216\left(C(p)+C(q)\right) + 198}{225} \quad (50)$$

9. An upper bound for residual closeness (Figure 1C)

The vertex residual closeness in case C from figure 1 can be restricted using:

Theorem 5. *Let connected graph G be created by collapsing vertex p of graph G_1 and vertex q of graph G_2 into vertex k. The closeness C(G′) of graph G′, constructed by removing vertex k, satisfies*:

$$C(G') \leq C(G) - 3C(k) + 0.5 \quad (51)$$

Proof. Using formula (9) we have:

$$C(G_1) + C(G_2) = C(G) - 2C(p)C(q) \quad (52)$$

The closeness of graph G_1 is decreasing at least with two times the closeness of vertex p (see Theorem 1). The equality is when p is not on the shortest path of any pair of vertices of G_1. Similar is situation with graph G_2. Using formula (9) and the fact that $C(p) + C(q) = C(k)$ we receive:

$$C(G') \leq C(G_1) - 2C(p) + C(G_2) - 2C(q) = C(G) - - 2C(p)C(q) - 2\,C(k) \quad (53)$$

The minimum of $C(p)C(q)$, with fixed sum ($C(p) + C(q) = C(k)$), is when one of the closeness is the minimal possible (0.5 for connected graphs):

$$2C(p)C(q) \geq 2\frac{1}{2}\left(C(k) - \frac{1}{2}\right) = C(k) - 0.5 \quad (54)$$

Combining both inequalities, we receive:

$$C(G') \leq C(G) - 2C(p)C(q) - 2\,C(k) \leq C(G) - 3C(k) + 0.5 \quad (55)$$

The equality is when one of the graphs is path with 2 vertices (for example G_1 is P_2) and the other graph (G_2) is linked by a vertex (q), which is not on the shortest path of any pair of vertices of this graph.

Corollary 5. *The vertex residual closeness of graph G (from Figure 1 C) satisfies*:

$$VR(G) \leq C(G) - 3C(k) + 0.5 \quad (56)$$

10. An algorithm for residual closeness of 1-connected graphs

The algorithm uses bounds, like B&B algorithms, but it is evaluating every possible solution, like brute-force algorithms. It uses the bounds from sections 7, 8, and 9 and the approximate algorithm to calculate the residual closeness of 1-connected graphs.

The solution set for residual closeness consists of all graphs with removed one link (or vertex). We remove a link and start calculating the closeness of the constructed graph using the approximate algorithm. At every step of the approximate algorithm, we calculate the errors (using formula (23)). If the lower limit of closeness of step k ($C_k + E_{min}$) is greater than the best-found solution (the one with lowest residual closeness) we stop calculating the closeness. If we have calculated the real closeness, because the approximate algorithm cannot stop, we compare it with the current best solution and possible update it.

The algorithm will work efficiently if we have a good starting solution (a solution with low residual closeness). A good candidate for an upper bound of the residual closeness of 1-connected graph is the closeness of a graph, which is disconnected.

We have done experiments with randomly generated connected graphs. We have generated graphs with 20, 30, and 40 vertices, divided into two subgraphs (10, 15 and 20 vertices each) with only one link connecting them. The two subgraphs have randomly generated links and are at least 2-connected. The percentage of links in a subgraph is set to be from 20 to 95 percentages of all possible links (45, 105 or 190 links correspondingly).

The graphs are divided into 3 groups, separated by ranges of percentage of the links of all possible links of the subgraphs: less than 45%, between 45% and 65%, and greater than 65%. The number of generated graphs in every range is bigger than 100. The best solution from the start is the closeness of disconnected two subgraphs, calculated using formula (38).

The results, shown in Table 1, contain the number of vertices, range of links in % from all possible links in the subgraphs, average percentage of all links, average diameter, average number of steps used by the algorithm, average percentage of steps related to the diameter.

Table 1. Decreasing the algorithm steps

Vertices	Range	Links %	Diam	Steps	Steps %
20	<45%	37.05	6.64	2.56	38.94
20	45-65	54.81	5.31	2.00	38
20	>65%	77.71	4.74	2.00	42.88
30	<45%	32.18	6.69	2.66	40.11
30	45-65	54.77	5.17	2.00	38.9
30	>65%	77.43	4.86	2.00	41.51
40	<45%	28.96	6.72	2.70	40.53

We can see that, after having a good starting solution for residual closeness (by removing the link which disconnects the two subgraphs), the steps for the algorithm and corresponding times are decreased by around 60 %.

11. Conclusions

In this article we have given an algorithm for additional closeness and an approximate algorithm for closeness of graphs.

Calculating the residual closeness of larger graphs is a difficult computational problem (O (mn^3)). We propose a solution for this problem - to use Branch and Bound like algorithms. For the algorithm to be effective, we need a good upper bound of the residual closeness. In this article, we have calculated bounds for 1-connected graphs and use them for calculating their residual closeness. We use the approximate algorithm to calculate the closeness of every solution, which give us around 60% decrement in the number of calculations and time.

The future efforts should be focus on developing effective algorithms for the closeness and the residual closeness of any graphs, graphs without special structure.

Conflicts of Interest: The author declares no conflict of interest.

References

[1] Dangalchev Ch. (2006) Residual closeness in networks. Phisica A. 365:556-564. doi:10.1016/j.physa.2005.12.020.

[2] Dangalchev Ch. (2020) Additional Closeness and Networks Growth. Fundamenta Informaticae. 176(1):1-15. doi:10.3233/FI-2020-1960

[3] Dangalchev Ch. (2022) Additional Closeness of Cycle Graphs. IJFCS. 33(8): 1033-1052. doi:10.1142/s0129054122500149.

[4] Floyd, R.(1962) Algorithm 97: Shortest path. Communications of the ACM, 5(6):345. doi:10.1145/367766.368168

[5] Williams, R. (2014). Faster all-pairs shortest paths via circuit complexity. Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14). New York: ACM. 664–673. doi:10.1145/2591796.2591811.

[6] Thorup, M. (1999). Undirected single-source shortest paths with positive integer weights in linear time. Journal of the ACM. 46(3):362–394. doi:10.1145/316542.316548.

[7] Dangalchev Ch. (2011) Residual closeness and generalized closeness. IJFCS. 22(8):1939-1947. doi:10.1142/s0129054111009136.

[8] Aytac A, Odabas Z.N. (2011) Residual closeness of wheels and related networks. IJFCS. 22(5):1229-1240. doi:10.1142/s0129054111008660.

[9] Odabas Z.N, Aytac A. (2013) Residual closeness in cycles and related networks. Fundamenta Informaticae. 124 (3): 297-307. doi:10.3233/FI-2013-835

[10] Turaci T, Okten M. (2015) Vulnerability of Mycielski graphs via residual closeness, Ars Combinatoria. 118: 419-427.

[11] Aytac A, Berberler Z.N.O. (2017) Robustness of Regular Caterpillars. IJFCS. 28(7): 835-841.

[12] Aytac A, Odabas Z.N. (2017) Network robustness and residual closeness. RAIRO-Operations Research. 52(3), 839-847. doi:10.1051/ro/2016071

[13] Turaci T, Aytac V. (2017) Residual Closeness of Splitting Graphs. Ars Combinatoria. 130: 17-27.

[14] Aytac A, Berberler, Z.N.O. (2017) Residual Closeness for Helm and sunflower graphs. TWMS Journal of applied and Engineering Mathematics. 7(2), 209-220.

[15] Dangalchev Ch. (2018) Residual Closeness of Generalized Thorn Graphs. Fundamenta Informaticae. 162(1), 2018, p 1-15. doi:10.3233/FI-2018-1710

[16] Aytac V, Turaci T. (2018) Closeness centrality in some splitting networks. Computer Science Journal of Moldova. 26(3):251-269.

[17] Berberler ZN, Yigit E. (2018) Link Vulnerability in Networks. IJFCS. 29(03):447-456. doi:10.1142/S0129054118500144

[18] Rupnik D, Zerovnik J. (2019) Networks with Extremal Closeness. Fun- ˇ damenta Informaticae. 167(3):219-234. doi:10.3233/fi-2019-1815.

[19] Yigit E., Berberler ZN. (2019) A Note on the Link Residual Closeness of Graphs Under Join Operation. IJFCS. 30(03):417-424.

[20] Yigit E., Berberler ZN. (2019) Link failure in wheel type networks. IJMPC. 30(09):1-12.

[21] Dangalchev Ch. (2020) Closeness of Splitting Graphs. C.R. Acad. Bulg. Sci. 73(4): 461-466.

[22] B. Zhou, Z. Li, H. Guo. (2021) Extremal results on vertex and link residual closeness. IJFCS. 32:1-21. doi:10.1142/S0129054121500295

[23] Cheng M.Q., Zhou B. (2022) Residual closeness of graphs with given parameters. Journal of the Operations Research Society of China.:1-18. doi:10.1007/s40305-022-00405-9

[24] Wang Y., Zhou B. (2022) Residual Closeness, Matching Number and Chromatic Number. The Computer Journal.. doi:10.1093/comjnl/bxac004

[25] Zheng L., Zhou B. (2022) On the spectral closeness and residual spectral closeness of graphs. RAIRO-Operations Research. 56(4):2651-2668. doi:10.1051/ro/2022125.

[26] G¨olpek H.T. (2022) Vulnerability of Banana trees via closeness and residual closeness parameters. Maltepe Journal of Mathematics. 4(2):1-5. doi:10.47087/mjm.1156370.

[27] Zheng L, Zhou B. (2022) Spectra of closeness Laplacian and closeness signless Laplacian of graphs. RAIRO-Operations Research. 56(5):3525–3543. doi:10.1051/ro/2022161.