

Article

Further Exploration of an Upper Bound for Kemeny's Constant

Robert E. Kooij ^{1,2,*}  and Johan L. A. Dubbeldam  ¹
¹ Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands; j.l.a.dubbeldam@tudelft.nl

² TNO (Unit ICT, Strategy & Policy, Netherlands Organisation for Applied Scientific Research), 2595 DA The Hague, The Netherlands

* Correspondence: r.e.kooij@tudelft.nl

Abstract: Even though Kemeny's constant was first discovered in Markov chains and expressed by Kemeny in terms of mean first passage times on a graph, it can also be expressed using the pseudo-inverse of the Laplacian matrix representing the graph, which facilitates the calculation of a sharp upper bound of Kemeny's constant. We show that for certain classes of graphs, a previously found bound is tight, which generalises previous results for bipartite and (generalised) windmill graphs. Moreover, we show numerically that for real-world networks, this bound can be used to find good numerical approximations for Kemeny's constant. For certain graphs consisting of up to 100 K nodes, we find a speedup of a factor 30, depending on the accuracy of the approximation that can be achieved. For networks consisting of over 500 K nodes, the approximation can be used to estimate values for the Kemeny constant, where exact calculation is no longer feasible within reasonable computation time.

Keywords: Kemeny's constant; effective graph resistance; random walks; spectral graph theory; pseudo-inverse Laplacian

MSC: 05C50; 05C75; 05C82



Academic Editors: Yi-Cheng Zhang and Shimin Cai

Received: 12 February 2025

Revised: 1 April 2025

Accepted: 2 April 2025

Published: 4 April 2025

Citation: Kooij, R.E.; Dubbeldam, J.L.A. Further Exploration of an Upper Bound for Kemeny's Constant. *Entropy* **2025**, *27*, 384. <https://doi.org/10.3390/e27040384>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Kemeny's constant, a graph metric first proposed in 1960 [1], links random walks, Markov chains, and spectral graph theory; see, for instance, [2–4]. An intuitive way to understand Kemeny's constant is by random walks on a graph, which was also how it was originally presented by Kemeny [1]. For an undirected connected graph with an adjacency matrix A , we can define a transition matrix $P_{ij} = A_{ij}/d_i$ for the transition from state i to j , where d_i is the degree of node i . This defines an irreducible finite-state Markov chain in discrete time with an $N \times N$ transition matrix P_{ij} [5]. If we also know the mean first-passage time matrix m_{ij} denoting the average time to go from a vertex i to a vertex j (we take $m_{ii} = 0$ by convention), the Kemeny constant is defined by

$$K(P) = \sum_{j=1}^N \pi_j m_{ij}, \quad (1)$$

where π_j is the j -th component of the stationary solution of the random walk. The fact that $K(P)$ does not depend on the index i , which can be interpreted as the starting state of the random walk and is therefore truly a constant, was discussed in a number of papers [6,7]. Hunter [8] and Kirkland [9] have analysed Relation (1) and established a connection with generalised matrix inverses.

The Kemeny constant also has an interpretation as a ‘mixing time’, which was originally proposed by Hunter in [7]. Here, we briefly repeat the demonstration that the Kemeny constant can be identified by a mixing time and show that this can be directly interpreted in terms of entropy. Let us define the ‘time to mixing’, T , of a Markov chain $\{X_n\}$ following [7], as the smallest index k at which $X_k = Y$, where Y is a random variable distributed according to the stationary distribution of the Markov chain $\{\pi_j\}$. We can now calculate the conditional expectation value of T , $E[T|Y, X(0) = i]$,

$$\begin{aligned} E[T|Y, X(0) = i] &= \sum_j E[T, Y = j|X(0) = i]P[Y = j] \\ &= \sum_j E[T_{ij}|X(0) = i]\pi_j = \sum_j m_{ij}\pi_j = K(P), \end{aligned} \quad (2)$$

where T_{ij} is the mean first-passage time for going from node i to node j .

Expression Equation (2) for the mixing time permits an interpretation in terms of *relative entropy* or Kullback–Leiber divergence $D(p||\pi)$, which measures the distance between the distributions p and π ; see also [10]. The relative entropy is defined as

$$D_n(p||\pi) = \sum_j p_j(n) \log \frac{p_j(n)}{\pi_j}.$$

Since $D_n(p||\pi) \geq 0$ with equality only when $p_j(n) = \pi_j$ for all $j = 1, \dots, N$, the time to mixing can be interpreted as the smallest value of n for which the relative entropy $D_n(p||\pi) = 0$.

Kemeny’s constant has recently also been suggested as a metric to identify bottleneck roads whose removal would greatly reduce the connectivity of the network [11] or as a metric to determine the ‘superspreader’ links that transmit disease between different communities [12].

It has already been established that there are several equivalent ways to express Kemeny’s constant: using effective graph resistance, random walks, spectral graph theory, and pseudo-inverse Laplacians; see [8].

The study of Kemeny’s constant is still an active and relevant research field, as was showcased by the mini-symposium “Kemeny’s constant on networks and its application”, which was organised as part of the 24th Conference of the International Linear Algebra Society, which took place in Galway, Ireland, 20–24 June 2022 [13] as well as recent papers addressing applications of Kemeny’s constant to different networks [14,15].

In 2017, Wang et al. [4] derived a closed-form formula for Kemeny’s constant, $K(P)$ for a random walk on a graph G with N nodes and L edges, where the transition matrix was given by $P = \Delta^{-1}A(G)$, where $A(G)$ is the adjacency matrix of G and Δ is a diagonal matrix containing the degrees of the nodes. In [4], it was shown that $K(P)$ can be expressed in terms of the Moore–Penrose pseudo-inverse Q^+ of the Laplacian matrix of G , as

$$K(P) = \zeta^T d - \frac{d^T Q^+ d}{2L}, \quad (3)$$

where the column vector $\zeta = (Q_{11}^+, Q_{22}^+, \dots, Q_{NN}^+)$ and $d(G) = (d_1, d_2, \dots, d_N)$ denotes the degree vector for the graph.

In [4], not only Equation (3) was derived, but also a closely connected upper bound:

$$K(P) \leq \zeta^T d - \frac{H(G)}{D(G)\mu_1(G)} \equiv K_U(P), \quad (4)$$

where $D(G)$ is the average degree and $\mu_1(G)$ is the largest eigenvalue of the Laplacian matrix $\Delta(G) = A(G)$ corresponding to graph G . Here, $\Delta(G)$ denotes the diagonal matrix containing the degrees of the nodes. The heterogeneity index $H(G)$, measuring the variability in the degrees of the nodes (see [16]) is defined as

$$H(G) = \frac{1}{N} \sum_{i=1}^N (d_i - D(G))^2,$$

where d_i is the degree of the i -th node.

It was shown in [17] that the upper bound given in Equation (4) is tight, meaning that we have an equality in Equation (4), for two classes of graphs, namely complete bipartite graphs and (generalised) windmill graphs. A windmill graph $W(\eta, k)$ consists of η copies of the complete graph K_k , with each node connected to a common node. Two generalisations of windmill graphs were suggested by Kooij [18] in 2019. For both generalisations, we replace the central node, connecting all η copies of the complete graph K_k , with l central nodes. For the first generalisation, we assume that the l central nodes are all connected, i.e., they form a clique K_l . We call this a generalised windmill graph of Type I and denote it by $W'(\eta, k, l)$. For the second generalisation, we assume that the l central nodes have no connections among each other. We will refer to it as a Type II generalised windmill graph and denote it by $W''(\eta, k, l)$. Figure 1 shows examples of a windmill graph and its two generalisations,

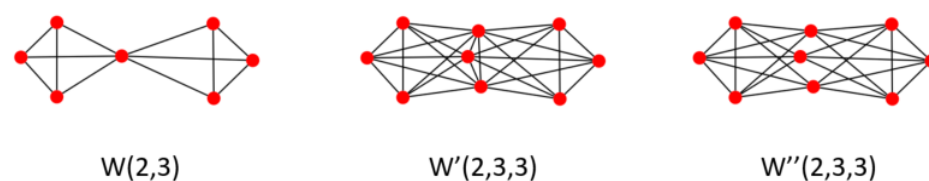


Figure 1. A windmill graph and generalised windmills of Types I and II.

The aim of this paper is four-fold. First, we will consider a broad family of graphs, which contain complete bipartite and (generalised) windmill graphs as special cases, and show analytically that for these graphs, the bound Equation (4) is tight. Graphs in this family have in common that they are bimodal and have a diameter of two. However, we will also show that these conditions are not sufficient to ensure that Equation (4) is tight. Next, we compare the complexity of the computation of the upper-bound Equation (4) with the exact expression for Kemeny's constant, given by Equation (3). In [17], we have already compared the exact value of $K(P)$ with the upper bound for some real-world networks. However, the considered networks were of rather moderate size ($N \leq 754$). Here, we will assess the performance of $K_U(P)$ on real-world networks of sizes up to around 365 K nodes and 1.72 M edges.

Finally, in addition to Equation (4), we also assess the performance of an upper bound suggested by de Vriendt [19] based on the so-called resistance radius of a graph:

$$K(P) \leq L\sigma^2 \equiv K^*, \quad (5)$$

where the resistance radius σ^2 is defined as

$$\sigma^2 = \frac{1}{2} (u^T \Omega^{-1} u)^{-1}, \quad (6)$$

with Ω denoting the resistance matrix and u the all-one vector. The upper-bound Equation (5) is tight for vertex-transitive graphs. Here, we remark that vertex-transitive graphs are rather exceptional and are typically highly symmetric; examples of vertex-

transitive graphs are Cayley graphs and the Petersen graph [20]. We will show in this paper that the bound K^* is not a good estimate for the Kemeny constant for the classes of graphs that are considered in this paper and that K_U is in general a much better estimate.

2. A Family of Biregular Graphs with Diameter 2

2.1. Construction

The aim is to construct a family of graphs that contains the complete bipartite and (generalised) windmill graphs as special cases and is commonly known as the combination of two regular graphs, denoted $G_1 \vee G_2$. We start the construction by considering a d_1 -regular graph G_1 on N_1 nodes, and a k_2 -regular graph G_2 on N_2 nodes. We assume $k_1 \geq 0$ and also $k_2 \geq 0$. Finally, we connect every node in G_1 to every node in G_2 to obtain the graph G . The nodes in G that are also in G_1 have degree $k_1 + N_2$, while the nodes in G_2 have degree $k_2 + N_1$. This construction yields a graph $G = G_1 \vee G_2$ that is a so-called biregular graph in which all nodes of G_1 have the same degree and the same holds for all nodes of G_2 ; see also [21]. Only if $k_1 + N_2 = k_2 + N_1$ is the graph G regular. By construction, G has diameter 2.

The choice of $k_1 = 0$ and $k_2 = 0$ leads to the complete bipartite graph K_{N_1, N_2} . If we take η isolated copies of the complete graph K_k as G_1 and an isolated node for G_2 , then G is the windmill graph $W(\eta, k)$. If instead, we let G_2 be a complete graph K_l , then G is a generalised windmill graph of Type I, $W'(\eta, k, l)$, whereas if we let G_2 consist of l isolated nodes, G is a generalised windmill graph of Type II, $W''(\eta, k, l)$.

Figure 2 shows an example of a graph that belongs to the suggested family of graphs. Here, G_1 , on the left side of the figure, is a random regular graph with $k_1 = 3$, on $N_1 = 10$ nodes, while G_2 is a graph on $N_2 = 8$ nodes, where each node has degree $k_2 = 5$. For the graph G , the nodes in G_1 have degree 11, while the nodes in G_2 have degree 15.

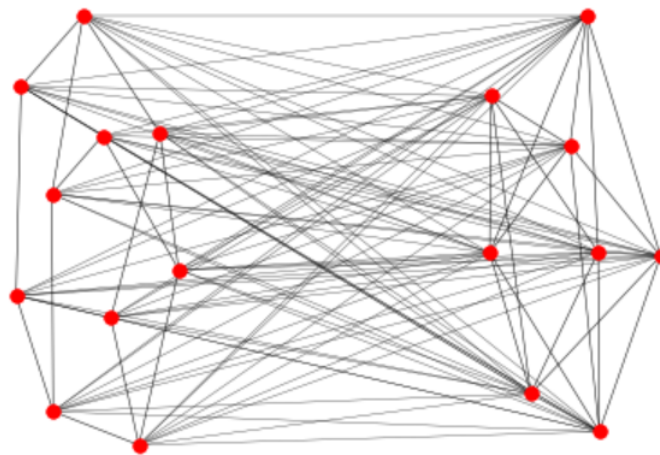


Figure 2. Graph G on 18 nodes, where G_1 is a random 3-regular graph on 10 nodes, and G_2 is a 5-regular graph on 8 nodes.

2.2. Tightness of the Upper Bound $K_U(G)$

We will now show for the family of graphs proposed in the previous subsection that the upper-bound Equation (4) for Kemeny's constant is tight.

Theorem 1. Consider two graphs G_1 and G_2 with all vertices in G_1 with degree d_1 and those in G_2 degree d_2 . If we connect each of the vertices in G_2 with all nodes of G_1 , then Kemeny's constant $K(P)$ for the resulting graph G is given by $K(P) = \zeta^T d - \frac{H(G)}{D\mu_1}$, that is, the upper-bound Equation (4) is tight.

Proof. First, we give expressions for the average degree D and the heterogeneity index H , which appear in the upper-bound Equation (4). Denoting the degrees of the nodes in G in G_1 and G_2 as D_1 and D_2 , respectively, we obtain

$$D_1 = D(G_1) + N_2 = d_1 + N_2 \quad (7)$$

and

$$D_2 = D(G_2) + N_1 = d_2 + N_1 \quad (8)$$

The average degree of G , $D(G)$, which we abbreviate for notational convenience to D , is defined by

$$D = \frac{D_1 N_1 + D_2 N_2}{N_1 + N_2}. \quad (9)$$

The heterogeneity index $H(G)$, a metric which quantifies the variability of the degree distribution (see [16]), is defined as follows:

$$H(G) = \frac{1}{N} \sum_{i=1}^N (D_i - D)^2, \quad (10)$$

where D_i denotes the degree of node i in graph G . Using the expressions for degrees D_1 and D_2 found in (7) and (8) and expression (9) for D , we obtain

$$\begin{aligned} H(G) &= \frac{1}{N_1 + N_2} \left(\sum_{i=1}^{N_1} (D_1 - D)^2 + \sum_{i=N_1+1}^{N_2} (D_2 - D)^2 \right) \\ &= \frac{1}{N_1 + N_2} (N_1 (D_1 - D)^2 + N_2 (D_2 - D)^2) = \frac{N_1 N_2 (D_1 - D_2)^2}{(N_1 + N_2)^2}. \end{aligned} \quad (11)$$

We will now prove the statement by first calculating the Laplacian matrix Q for the graph G , which has the following special structure:

$$Q = \begin{pmatrix} A_1 & -J_{N_1 \times N_2} \\ -J_{N_2 \times N_1} & A_2 \end{pmatrix}, \quad (12)$$

where $J_{N_2 \times N_1}$ is an all-one $N_2 \times N_1$ matrix, and the square matrices A_1 and A_2 are defined as

$$\begin{aligned} A_1 &= Q_{G_1} + N_2 I_{[N_1, N_1]} \\ A_2 &= Q_{G_2} + N_1 I_{[N_2, N_2]}, \end{aligned} \quad (13)$$

where $Q_{G_1(G_2)}$ is the Laplacian of graph G_1 (G_2), and $I_{[N_1, N_1]}, I_{[N_2, N_2]}$ denote the identity matrices of size $N_1 \times N_1$ and $N_2 \times N_2$, respectively. The decomposition of Q into 4 blocks can be understood by realising that the upper right-hand block, $-J_{N_1 \times N_2}$, represents the N_2 links that exist between each vertex of G_1 and all the vertices of G_2 . Since Q is a Laplacian matrix, we have to ensure that all rows sum up to zero, which can be achieved by adding N_2 to each of the diagonal entries of the $N_1 \times N_1$ block in the upper left-hand corner, that is, the block A_1 should be as defined above. Analogously, we find that the lower left-hand and right-hand blocks should be equal to $-J_{N_2 \times N_1}$ and A_2 , respectively.

Two eigenvectors, v_1 and v_N , can be found by inspection. $v_N = (1, \dots, 1)^T$, which corresponds to eigenvalue $\mu_N = 0$, and $v_1 = (N_2, \dots, N_2, -N_1, \dots, -N_1)^T$, which has N_1 entries equal to N_2 and N_2 entries equal to $-N_1$ and corresponds to $\mu_1 = N_1 + N_2$.

Because the largest Laplacian eigenvalue is upper-bounded by N , the number of nodes in a graph (see [22]), we directly obtain that μ_1 is the largest eigenvalue of Q . Combining this with Equations (9)–(11), we obtain

$$\frac{H(G)}{D\mu_1} = \frac{N_1N_2(D_1 - D_2)^2}{(N_1 + N_2)^2(D_1N_1 + D_2N_2)}. \quad (14)$$

Since eigenvectors corresponding to different eigenvalues are all orthogonal and those corresponding to the same eigenvalues can be chosen to be orthogonal, due to the symmetry of Q , all eigenvectors $v = (x_1, x_2, \dots, x_{N_1+N_2})^T$ that are not equal to v_1 or v_N are subject to

$$\begin{aligned} x_1 + x_2 + \dots + x_{N_1+N_2} &= 0 \\ N_2(x_1 + \dots + x_{N_1}) - N_1(x_{N_1+1} + \dots + x_{N_1+N_2}) &= 0, \end{aligned}$$

which leads to

$$\begin{aligned} x_1 + x_2 + \dots + x_{N_1} &= 0 \\ x_{N_1+1} + \dots + x_{N_1+N_2} &= 0. \end{aligned} \quad (15)$$

We next turn to the expression $d^T Q^+ d$, where $Q^+ = \sum_{i=1}^{N_1+N_2-1} \frac{1}{\mu_i} \hat{v}_i \hat{v}_i^T$ where μ_i is the i -th eigenvalue of Q and $\hat{v}_i$ is the normalised eigenvector. The conditions for the eigenvectors (15) imply that all terms in the expression $d^T Q^+ d$ vanish except the term associated with v_1 . More precisely, we find that

$$d^T Q^+ d = \sum_{i=2}^{N_1+N_2-1} \frac{(d^T \hat{v}_i)^2}{\mu_i} + \frac{(\hat{v}_1^T d)^2}{\mu_1} = \frac{N_1N_2(D_1 - D_2)^2}{(N_1 + N_2)^2}, \quad (16)$$

where $d = (D_1, D_2)^T$, so the first N_1 components all have degree D_1 and the remaining components have degree D_2 , which implies $d^T v_i = 0$ by Equation (15). Finally, because from $D = \frac{2L}{N}$, we obtain

$$2L = D_1N_1 + D_2N_2 \quad (17)$$

it follows that $\frac{d^T Q^+ d}{2L}$ equals Equation (14), which proves the proposition. $\square$

2.3. Some Examples

As a first example, we consider the graph depicted in Figure 2, where $N_1 = 10$, $d_1 = 3$, $N_2 = 8$ and $d_2 = 5$. Using Python (<https://www.python.org/>) code, we have evaluated both K and K_U . For this network, we obtain $K = 16.33864$, which is equal to K_U to numerical precision, as should be according to Theorem 1. On the other hand, the upper bound K^* based upon the resistance radius gives $K^* = 19.29615$, which is reasonably close to the actual value.

Next, we consider a graph where $N_1 = 50$, $d_1 = 4$, $N_2 = 10$ and $d_2 = 6$; see Figure 3. Here, we get $K = 59.19805$, and again, K and K_U are numerically extremely close. On the other hand, for this graph, the bound Equation (5) is two orders larger than the actual value: $K^* = 533.03153$.

As a final example, we consider the case where $N_1 = 100$, $d_1 = 10$, $N_2 = 20$ and $d_2 = 8$; see Figure 4. Now, $K = 119.24078$ and again K and K_U are equal to numerical precision. Again, the bound based on the resistance radius is much higher: $K^* = 1652.63986$.

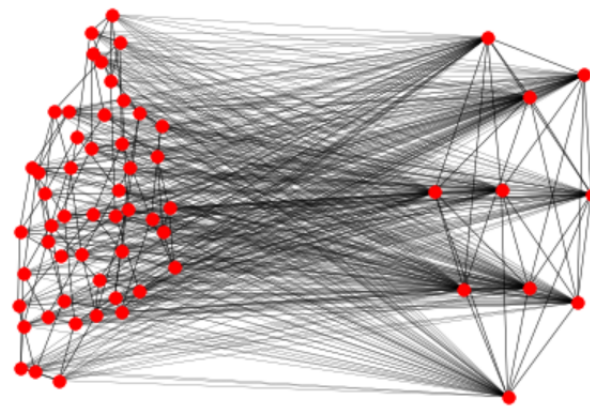


Figure 3. Graph G on 60 nodes, where G_1 is a random 4-regular graph on 50 nodes, and G_2 is a random 6-regular graph on 10 nodes.

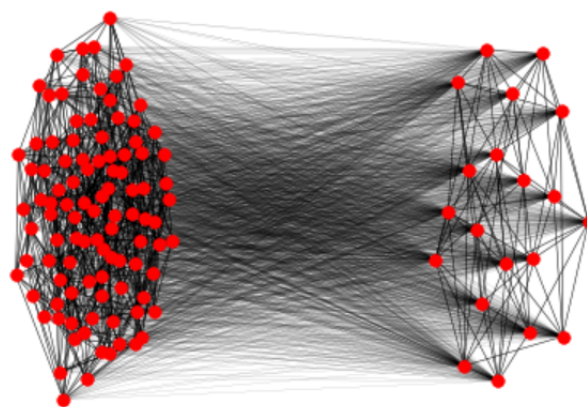


Figure 4. Graph G on 120 nodes, where G_1 is a random 10-regular graph on 100 nodes, and G_2 is a random 8-regular graph on 20 nodes.

We end this subsection by noting that the choice for the examples in this subsection was rather arbitrary. We also ran our Python script on several other graphs with sizes up to 1500 nodes. Each time, it yielded the same result: K and K_U have values that are numerically very close (see also [4,17] for more numerical comparisons), while the upper bound K^* exceeds Kemeny's constant by a few orders.

3. Graphs with Diameter 2 for Which $K_U(P)$ Is Not Tight

3.1. Bimodal Graphs with Diameter 2 for Which Equation (4) Is Not Tight

The numerical results of the examples on biregular graphs with diameter 2 from the previous section showed that in all these cases, the approximation of K by K_U is actually exact. In other words, the bound K_U is tight in these cases. Therefore, one might be tempted to believe that Equation (4) is tight for all biregular graphs with diameter 2. In this section, we prove that this is not the case by giving some counterexamples.

The simplest counterexample we could find consists of the cycle graph C_5 with an additional link; see Figure 5.

For this graph, we get $K = 3.71212$, $K_U = 3.72380$ and $K^* = 4.21488$. There is a simple procedure to check whether or not a biregular graph G with diameter 2 belongs to the graph family constructed in the previous section. First, partition the nodes into two sets S_1 and S_2 where all the nodes in the set S_1 have degree D_1 , while all the nodes in the set S_2 have degree D_2 . Next, verify if the number of links between the 2 sets is $|S_1 \times S_2|$ and all nodes of S_1 are linked to all nodes of S_2 . If this is not the case, the graph $G \neq S_1 \vee S_2$. In the other case, remove all $|S_1 \times S_2|$ links between the sets S_1 and S_2 . If the remaining

two graphs are not both regular, then the original graph G does not belong to the family constructed in the previous section, that is, $G \neq S_1 \vee S_2$.

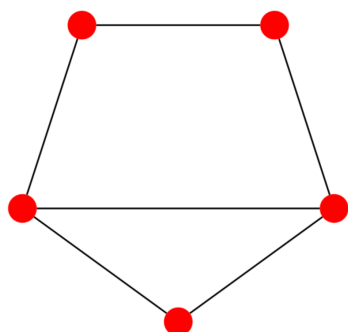


Figure 5. Smallest biregular graph with diameter 2 for which the upper-bound Equation (4) is not tight.

The second counterexample is constructed by adding a link to the Petersen graph; see Figure 6.

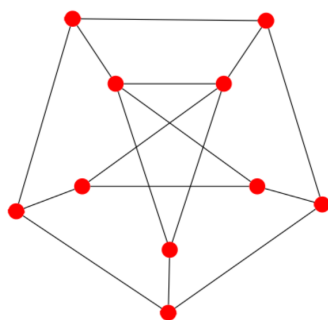


Figure 6. Petersen graph with one additional link.

For this graph, we get $K = 9.76597$, $K_U = 9.77389$ and $K^* = 10.26963$.

3.2. Non-Biregular Graphs with Diameter 2

We now give an example of a non-biregular graph with diameter 2, for which the upper-bound Equation (4) also does not equal Kemeny's constant. We construct the graph by first taking a complete graph K_N on N nodes. Next, we add one node and connect it to one node in K_N and therefore the resulting graph has diameter 2. The resulting graph has $N - 1$ nodes with degree $N - 1$, one node with degree N , and one node with degree 1. Figure 7 shows an example with $N = 10$.

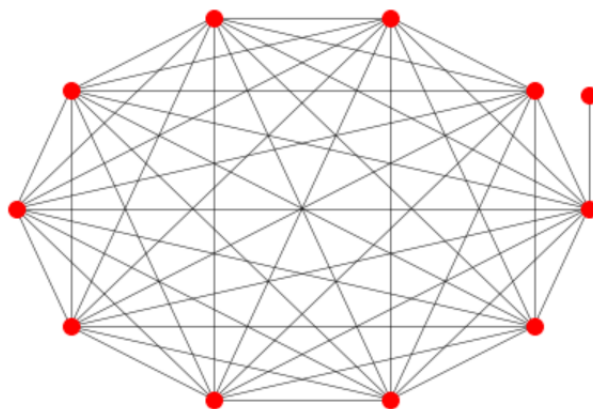


Figure 7. Non-biregular graph with diameter 2.

Applying Equation (3), we get $K = 9.26522$, while the upper bound of Equation (4) gives $K_U = 9.83439$, while $K^* = 31.28000$.

4. Regular Graphs

In this section, we consider regular graphs on N nodes with degree r . In this case, the relation between Kemeny's constant and the effective graph resistance was shown [23] to be

$$K(P) = \frac{r}{N} R_G, \quad (18)$$

where R_G denotes the effective graph resistance. Next, we show that for these graphs, the upper-bound Equation (4) is also tight. For this, we will use the following expression for the effective graph resistance (see [4]):

$$R_G = N \sum_{i=1}^N Q_{ii}^\dagger. \quad (19)$$

For r -regular graphs, $H = 0$, and therefore Equation (4) gives

$$K_U = \zeta^T d = r \zeta^T u = r \sum_{i=1}^N Q_{ii}^\dagger = \frac{r}{N} R_G, \quad (20)$$

hence $K_U = K$ according to Equation (18).

As an example, we consider a random 3-regular graph on 100 nodes (see Figure 8), which has a diameter 10. We get numerically $K = 195.30524$, which is indeed equal to K_U up to the numerical precision of 10^{-17} . Applying Equation (5) gives $K^* = 218.32805$. In this case, the upper-bound Equation (5) is not tight because the graph is not vertex-transitive.

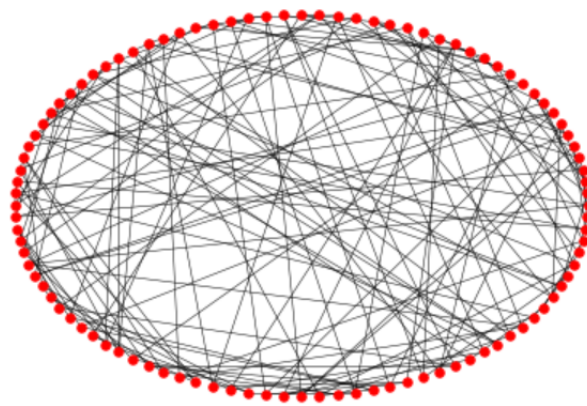


Figure 8. Random 3-regular graph on 100 nodes.

5. Complexity for the Computation of $K_U(P)$

The time complexity of $K(P)$, computed via Equation (3), is dominated by the Laplacian pseudo-inverse, which is as expensive as performing a dense matrix multiplication and takes $\mathcal{O}(N^3)$ in practice with standard tools. On the other hand, the time complexity of $K_U(P)$ mainly depends on two operations: computing the largest Laplacian eigenvalue and performing the dot product of a degree vector and the diagonal element vector of the Laplacian pseudo-inverse. Interestingly, to compute $K_U(P)$, we can avoid the full pseudo-inversion as it only requires the diagonal elements of the Laplacian pseudo-inverse. Algorithms that approximate the diagonal (or the trace) of matrices often use iterative methods, sparse direct methods [24], Monte Carlo [25] or deterministic probing techniques [26]. Although faster than computing the full inversion, these approaches are still time-consuming in practice for large graphs [27]. For that reason, we employ a recently

proposed algorithm that approximates the diagonal entries of the Laplacian pseudo-inverse using combinatorial connections [27]. This algorithm exploits the relation between effective resistance and the pseudo-inverse Laplacian. In order to calculate the diagonal elements of $Q^\dagger$, it is sufficient to compute the electrical farness $f_{\text{el}}(u)$ of each node u in the set of all nodes V ; the farness is defined by

$$f_{\text{el}}(u) = \sum_{v \in V/\{u\}} R(u, v) = NQ_{uu}^\dagger + \text{Tr}(Q^\dagger)$$

Here, $R(u, v)$ is the effective resistance between node u and v , which is the potential difference between u and v when a unit current is injected in graph G at node u and extracted at node v [28]. Rather than calculate $R(u, v)$ for each pair of nodes, we sample a set of uniform spanning trees. This approach provides a probabilistic absolute approximation guarantee.

The algorithm's time complexity is summarised in the following proposition:

Proposition 1 ([27]). *Let $G = (V, E)$ be an undirected and weighted graph with N nodes and L edges. The sampling algorithm, briefly described above, gives an approximation of the diagonal elements of $Q^\dagger$ with absolute error $\pm\epsilon$ with probability $1 - \delta$ in an expected time $\mathcal{O}(L \cdot \text{ecc}^3(u) \cdot \epsilon^{-2} \cdot \log(L/\delta))$, where $\text{ecc}(u)$ is the length of the longest shortest path (eccentricity) starting in a selected node u . For small-world graphs and $\delta = 1/N$ (for high probability), this yields a time complexity of $\mathcal{O}(L \log^4 N \cdot \epsilon^{-2})$.*

For networks that have small-world characteristics, a common feature for many real-world networks [29], the above algorithm obtains a $\pm\epsilon$ -approximation with high probability, in a time that is linear in L up to polylogarithmic terms and quadratic in $1/\epsilon$. Furthermore, computing the largest Laplacian eigenvalue does not change the overall complexity bound. More precisely, this step often takes $\mathcal{O}(L)$ time for sparse matrices using standard iterative methods, such as the Lanczos algorithm [30]. In general, the actual running time for this step highly depends on the desired accuracy and the eigenvalue distribution of the involved matrix. Overall, the complexity bound for computing $K_U(P)$ for small-world graphs using the above techniques is linear in the number of links L (up to a polylogarithmic factor).

6. Analysis of Some Large Real-World Networks

In this section, we analyse the performance of our proposed bound, $K_U(P)$, compared to Kemeny's constant, $K(P)$, in terms of accuracy and running time results. For $K_U(P)$, our implementation uses the NetworKit [31] graph library to compute the diagonal elements of $Q^\dagger$ (via the algorithm of Angriman et al. [27]) and the SLEPC library (<https://slepc.upv.es/>) (accessed on 2 December 2024) to compute the largest Laplacian eigenvalue. $K(P)$, in turn, is computed via Equation (3) and our implementation uses the Eigen library (<http://eigen.tuxfamily.org>) (accessed on 2 December 2024) to compute the entire pseudo-inverse, $Q^\dagger$. We do not include any comparisons against K^* since, computationally, it is as expensive as the exact computation of Kemeny's constant. Our test machine is a shared-memory server with a 2x 18-Core Intel Xeon 6154 CPU and a total of 1.5 TB RAM. To ensure reproducibility, experiments are managed by SimexPal [32]. In Table 1, we list the real-world graphs that are used in our experiments, downloaded from SNAP [33] and NR [34] public repositories. In this context, we consider as medium graphs those whose vertex count is <57 K. The largest graph has around 365 K nodes and 1.72 M edges.

For the medium graphs of Table 1, we are able to compare our bound $K_U(P)$ relatively to Kemeny's constant $K(P)$, and the results are illustrated in Figure 9. $K_U(P)$ is computed with different error bounds (ϵ) for the approximation of the diagonal elements (via the

algorithm of Angriman et al. [27])—they correspond to the respective numbers next to the names in Figure 9. Regarding the accuracy, we observe that our approach for computing $K_U(P)$ is overall highly accurate for all values of ϵ and graphs. More precisely, on average (computed via geometric mean) over the medium-size graphs, our approach is 0.33% 0.27% 0.25% and 1.26% away from the exact Kemeny’s constant for $\epsilon = 0.1, 0.3, 0.5$ and 0.9 , respectively. Meanwhile, the running time is on average $2, 18, 48$ and $141 \times$ faster than the exact computation for each ϵ , respectively. Figure 9a shows that on individual graphs, a larger ϵ value ($\epsilon = 0.9$) may result in a slightly less accurate bound—up to 10% away from the exact value (arx). Moreover, in Figure 9b, we observe that for the inf graph, computing the exact Kemeny’s constant is much faster than computing $K_U(P)$ via Algorithm [27]. The primary reason for that is the small size (6K edges) for which an exact computation of the entire pseudo-inverse is still fast enough. A second reason for the slow performance of the algorithm of Angriman et al. could be due to the high diameter of the graph in question ($\gg \log N$).

Table 1. Summary of graph instances, providing (in order) network name, type, abbreviation, vertex count, and edge count.

Graph	Type	ID	V	E
inf-power	infrastructure	inf	4 K	6 K
facebook-ego-combined	social	fac	4 K	8.8 K
p2p-Gnutella04	internet	p2p	10 K	39 K
ca-HepPh	collaboration	ca-	11 K	117 K
arxiv-astro-ph	collaboration	arx	17 K	196 K
eat	words	eat	23 K	297 K
arenas-pgp	infrastructure	are	24 K	10 K
as-caida20071105	internet	as-	26 K	53 K
ia-email-EU	communication	ia-	32 K	54.4 K
loc-brightkite	social	lob	57 K	213 K
soc-Slashdot0902	social	soc	82 K	504 K
flickr	images	fli	106 K	2.31 M
livemocha	social	liv	104 K	2.19 M
loc-gowalla-edges	social	log	196 K	950 K
web-NotreDame	web	web	325 K	1.09 M
citeseer	citation	cit	365 K	1.72 M

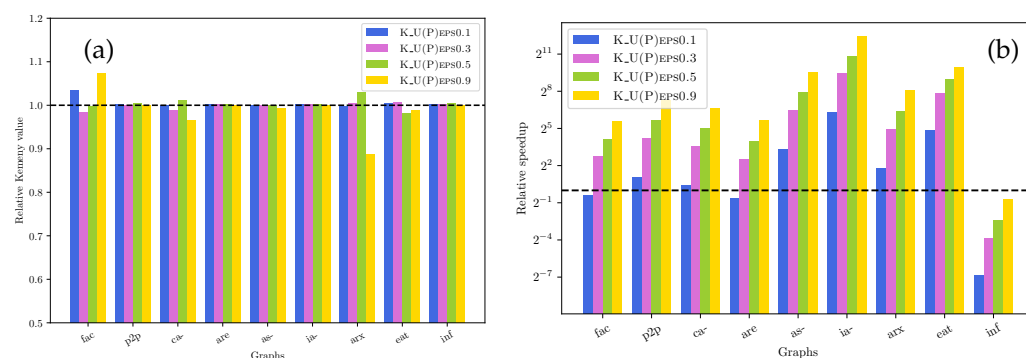


Figure 9. Relative quality (a) and speedup (b) results (per graph) for computing $K_U(P)$ for medium graphs ($n < 57$ K) of Table 1. Results are relative to exact computation of $K(P)$.

In Table 2, we illustrate our results for the largest graphs of Table 1. For this experiment, we set $\epsilon = 0.5$ for the approximation of the diagonal elements of Q^+ as this offers the best trade-off between accuracy and speed, according to the previous experiment. Unfortunately, we were not able to compute exact values for Kemeny’s constant for these graphs, as all

involved runs timed out at 18,000 s. This is due to the prohibitive time and space complexity of the pseudo-inversion operation required by $K(P)$.

Table 2. Absolute results for $K_U(P)$ on the largest graphs of Table 1. Comparison to $K(P)$ is prohibitive due to the (large) size of the graphs in question.

Graph	$K_U(P)$	Time (h:min:s)
lob	80,903	48.83 s
soc	96,102	50.87 s
fli	122,185	1 min:38.11 s
liv	120,525	37.07 s
log	271,577	5 min:10.77 s
web	1,009,760	1 h:11 min:19.36 s
cit	508,244	1 h:16 min:11.51 s

7. Conclusions

We have investigated Kemeny’s constant $K(P)$ for a number of networks using the exact expression from [4] and compared this expression with two upper bounds: one $K^*(P)$ that was derived in Ref. [19] and is known to be tight for vertex-transitive graphs, and the other bound $K_U(P)$ was derived in [4] and is written in terms of degrees of the nodes, the diagonal elements of the pseudo-inverse Laplacian, the largest eigenvalue of the Laplacian matrix and the heterogeneity of the degrees of the nodes.

We have numerically demonstrated that the bound $K_U(P)$ is generally a much better approximation for $K(P)$ than $K^*(P)$ for the networks that we have explored. Moreover, we have proved that for any graph G composed of two regular graphs G_1 and G_2 with all nodes of the graph G_1 connected to each node of G_2 , the bound $K_U(P)$ is tight. This generalises earlier findings that the bound $K_U(P)$ is tight for (generalised) windmill and complete bipartite graphs.

As an illustration of the advantages of using the expression $K_U(P)$ to estimate the Kemeny constant, we numerically calculated the Kemeny constant for a number of real-world large networks. We find that the calculation of $K_U(P)$ can be performed very efficiently, displaying efficiency gains in the order of a factor 100–1000, for networks up to 57 K nodes. The upper bound can still be obtained in a reasonable time for networks up to 365 K nodes.

Author Contributions: Conceptualisation, R.E.K. and J.L.A.D.; methodology, R.E.K. and J.L.A.D.; software, R.E.K.; formal analysis, R.E.K. and J.L.A.D.; investigation, R.E.K. and J.L.A.D.; writing—original draft preparation, R.E.K. and J.L.A.D.; writing—review and editing, R.E.K. and J.L.A.D.; visualisation, R.E.K. All authors have read and agreed to the published version of the manuscript.

Funding: J.L.A. Dubbeldam was partially supported by the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 955708 and the Dutch National Foundation projects OCENW.KLEIN.277.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kemeny, J.G.; Snell, J.L. *Finite Markov Chains*; D. Van Nostrand: Princeton, NJ, USA, 1960.
2. Lovász, L. Random Walks on Graphs: A Survey. In *Paul Erdős is Eighty*; Bolyai Society, Mathematical Studies: Keszthely, Hungary, 1993; Volume 2, pp. 1–46.

3. Palacios, J.L.; Renom, J.M. Bounds for the Kirchhoff index of regular graphs via the spectra of their random walks. *Int. J. Quantum Chem.* **2010**, *110*, 1637–1641. [CrossRef]
4. Wang, X.; Dubbeldam, J.L.A.; Van Mieghem, P. Kemeny's constant and the effective graph resistance. *Linear Algebra Its Appl.* **2017**, *535*, 231–244.
5. Noh, J.D.; Rieger, H. Random Walks on Complex Networks. *Phys. Rev. Lett.* **2004**, *92*, 118701. [CrossRef] [PubMed]
6. Levene, M.; Loizou, G. Kemeny's Constant and the Random Surfer. *Am. Math. Mon.* **2002**, *109*, 741–745. [CrossRef]
7. Hunter, J. Mixing times with applications to perturbed Markov chains. *Linear Algebra Its Appl.* **2006**, *417*, 108–123. [CrossRef]
8. Hunter, J.J. The role of Kemeny's constant in properties of Markov chains. *Commun. Stat.-Theory Methods* **2014**, *43*, 1309–1321.
9. Kirkland, S.; Zeng, Z. Kemeny's Constant and an Analogue of Braess' Paradox for Trees. *Electron. J. Linear Algebra* **2016**, *31*, 444–464. [CrossRef]
10. Thomas, M.; Cover, J.A.T. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
11. Altafini, D.; Bini, D.A.; Cutini, V.; Meini, B.; Poloni, F. An edge centrality measure based on the Kemeny constant. *arXiv* **2022**, arXiv:2203.06459. [CrossRef]
12. Yilmaz, S.; Dudkina, E.; Bin, M.; Crisostomi, E.; Ferraro, P.; Murray-Smith, R.; Parisini, T.; Stone, L.; Shorten, R. Kemeny-based testing for COVID-19. *PLoS ONE* **2020**, *15*, e0242401. [CrossRef]
13. Available online: <https://www.niallmadden.ie/ILAS-2022.pdf> (accessed on 2 December 2024).
14. Breen, J.; Crisostomi, E.; Kim, S. Kemeny's constant for a graph with bridges. *Discret. Appl. Math.* **2022**, *322*, 20–35. [CrossRef]
15. Bini, D.A.; Durastante, F.; Kim, S.; Meini, B. On Kemeny's constant and stochastic complement. *Linear Algebra Its Appl.* **2024**, *703*, 137–162. [CrossRef]
16. Bell, F. A note on the irregularity of graphs. *Linear Algebra Its Appl.* **1992**, *161*, 45–54. [CrossRef]
17. Kooij, R.E.; Dubbeldam, J.L. Kemeny's constant for several families of graphs and real-world networks. *Discret. Appl. Math.* **2020**, *285*, 96–107. [CrossRef]
18. Kooij, R.E. On generalized windmill graphs. *Linear Algebra Its Appl.* **2019**, *565*, 25–46. [CrossRef]
19. Devriendt, K.; Martin-Gutierrez, S.; Lambiotte, R. Variance and Covariance of Distributions on Graphs. *SIAM Rev.* **2022**, *64*, 343–359. [CrossRef]
20. Godsil, C.; Royle, G.F. *Algebraic Graph Theory*; Springer: Berlin/Heidelberg, Germany, 2013.
21. Scheinermann, E.; Ullman, D. *Fractional Graph Theory: A Rational Approach*; Wiley and Sons: Hoboken, NJ, USA, 2008.
22. Van Mieghem, P. *Graph Spectra for Complex Networks*; Cambridge University Press: Cambridge, UK, 2011.
23. Palacios, J.L.; Renom, J. Broder and Karlin's formula for hitting times and the Kirchhoff Index. *Int. J. Quantum Chem.* **2011**, *111*, 35–39. [CrossRef]
24. Jacquelin, M.; Lin, L.; Yang, C. PSELInv—A distributed memory parallel algorithm for selected inversion: The non-symmetric case. *Parallel Comput.* **2018**, *74*, 84–98. [CrossRef]
25. Hutchinson, M.F. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *J. Commun. Statist. Simula.* **1990**, *19*, 433–450. [CrossRef]
26. Bekas, C.; Kokiopoulou, E.; Saad, Y. An Estimator for the Diagonal of a Matrix. *Appl. Numer. Math.* **2007**, *57*, 1214–1229. [CrossRef]
27. Angriman, E.; Predari, M.; van der Grinten, A.; Meyerhenke, H. Approximation of the Diagonal of a Laplacian's Pseudoinverse for Complex Network Analysis. In Proceedings of the ESA. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Pisa, Italy, 7–9 September 2020; LIPIcs; Volume 173, pp. 6:1–6:24.
28. Bollobás, B. *Modern Graph Theory*; Springer: Berlin/Heidelberg, Germany, 1998.
29. Newman, M. *Networks*, 2nd ed.; Oxford University Press: Oxford, UK, 2018.
30. Paige, C. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra Its Appl.* **1980**, *34*, 235–258. [CrossRef]
31. Staudt, C.L.; Sazonovs, A.; Meyerhenke, H. NetworKit: A tool suite for large-scale complex network analysis. *Netw. Sci.* **2016**, *4*, 508–530. [CrossRef]
32. Angriman, E.; van der Grinten, A.; von Looz, M.; Meyerhenke, H.; Nöllenburg, M.; Predari, M.; Tzovas, C. Guidelines for experimental algorithmics: A case study in network analysis. *Algorithms* **2019**, *12*, 127. [CrossRef]
33. Leskovec, J. Stanford Network Analysis Package (SNAP). Available online: <https://snap.stanford.edu/> (accessed on 2 December 2024).
34. Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Proceedings of the AAAI, Austin, TX, USA, 25–30 January 2015.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.