Improved Approximation for Broadcasting in k-cycle Graphs

Jeffrey Bringolf[1][0009−0003−0928−9146], Anne-Laure Ehresmann[2][0000−0002−6116−3212], and Hovhannes A. Harutyunyan[1][0000−0001−7260−4186]

[1] Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada
jeffrey.bringolf@mail.concordia.ca
haruty@cs.concordia.ca
[2] YesWeHack, Singapore, Singapore
al@ehresmann.eu

Abstract. Broadcasting is an information dissemination primitive where a message originates at a node (called the originator) and is passed to all other nodes in the network. Broadcasting research is motivated by efficient network design and determining the broadcast times of standard network topologies. Verifying the broadcast time of a node v in an arbitrary network G is known to be NP-hard. Additionally, recent findings show that the broadcast time problem is also NP-complete in general cactus graphs and some highly restricted subfamilies of cactus graphs. These graph families are structurally similar to k-cycle graphs, in which the broadcast time problem is also believed to be NP-complete. In this paper, we present a simple $(1.5 - \epsilon)$-approximation algorithm for determining the broadcast time of networks modeled using k-cycle graphs, where $\epsilon > 0$ depends on the structure of the graph.

1 Introduction

Broadcasting is a crucial process for information dissemination in interconnected networks. As a result, a variety of network models have been studied at length [13, 15, 17, 19–22]. These models have varying rules describing the means by which information can be disseminated. They may have different numbers of originators, numbers of receivers at each time unit, distances of each call, numbers of destinations, and other characteristics of the network such as knowledge of the neighbourhood available to each node. In this paper, we will focus on the classical model of broadcasting, sometimes referred to as *telephone broadcasting*. The network is modeled as an undirected connected graph $G = (V, E)$, where $V(G)$ and $E(G)$ denote the vertex set and the edge set of G, respectively. The classical model is described by the following assumptions.

1. The broadcasting process is split into discrete time units.
2. The only vertex that has the message before the first time unit is called *originator*.

3. In each time unit, an informed vertex (*sender*) can *call* at most one of its uninformed neighbors (*receiver*).
4. During each time unit, all calls are performed in parallel.
5. The process halts as soon as all the vertices in the graph are informed.

Each call in the broadcasting process can be defined as $\Gamma = (u, v)$ where u is the sender and v is the receiver. The *broadcast scheme* is represented by an ordered list containing the calls made by each vertex throughout the broadcast. This list can be represented by the sequence $(\Gamma_1, ..., \Gamma_t)$, where Γ_i is the set of calls made at time i and t is the time unit at which the network is fully informed.

Definition 1. *The broadcast time of a vertex v in a given graph G is the minimum number of time units required to broadcast in G if v is the originator and is denoted by $b(v, G)$. The broadcast time of a given graph G, is the maximum broadcast time from any originator in G, formally $b(G) = max_{v \in V(G)}\{b(v, G)\}$.*

The general broadcast time decision problem is formally defined as follows. Given a graph $G = (V, E)$ with a specified set of vertices $V_0 \subseteq V$ and a positive integer k, is there a sequence $V_0, E_1, V_1, E_2, V_2, ..., E_k, V_k$ where $V_i \subseteq V, E_i \subseteq E(1 \leq i \leq k)$, for every $(u, v) \in E_i, u \in V_{i-1}, v \in V_i, v \notin V_{i-1}$ and $V_k = V$. Here, k is the total broadcast time, V_i is the set of informed vertices at round i, and E_i is the set of edges used at round i. Clearly, when $|V_0| = 1$, this problem becomes our broadcast problem of determining $b(v, G)$ for an arbitrary vertex v in an arbitrary graph G.

The broadcast time decision problem is NP-complete in an arbitrary graph [14, 27]. Furthermore, the problem was shown to be NP-complete in some restricted families of graphs, such as 3-regular planar graphs [23]. The study of the parameterized complexity of the broadcast time problem was initiated in [11] and has progressed in the following years [12, 28]. Moreover, there are few classes of graphs for which an exact polynomial-time algorithm is known. Some examples of graph families for which exact linear time algorithms exist for the broadcast time problem are the following: trees [24, 27], unicyclic connected graphs [18, 19], necklace graphs (chain of rings) [16], and Harary-like graphs [5, 6].

The intractability of the broadcast time problem has led to significant work on approximation algorithms. Ravi devised a polynomial-time $(O(\frac{\log^2 n)}{\log(\log n)}))$-approximation algorithm for the general broadcast time problem in [25]. This approximation ratio was improved to $O(\log n)$ in [2] by Bar-Noy et al. Subsequently, Elkin and Kortsarz again improved the approximation ratio to the current best known approximation in general graphs of $O(\frac{\log n}{\log(\log n)})$ [9, 10]. Additionally, the general broadcast time problem is known to be $\frac{3}{2}$-inapproximable from [26]. This was increased to a $(3 - \epsilon)$-inapproximability result in [9]. Due to the significant advancements in the approximation of the general broadcast time problem, attention has been drawn to more constrained classes of graphs.

Cactus graphs are a class of connected graphs in which any two simple cycles share at most one vertex. They have been a significant subject of interest in the field of graph algorithms as they are a simple class of graphs for which NP-hard problems have polynomial solutions. An exact algorithm to solve the broadcast time problem in cactus graphs was given in [7]. The complexity of the algorithm on a cactus graph $G = (V, E)$ is $O(kk!m + n)$ where $m = |E|$, $n = |V|$, and k is the maximum number of cycles that share a vertex. In a *k-restricted cactus graph*, k is held constant, and the algorithm is polynomial. Nonetheless, until recently, the tractability of the problem on general cactus graphs remained an open question. Moreover, the problem remains open even for very simple graph families, such as those formed by merging k cycles at a single vertex. These graphs, called *k-cycle graphs* or *flower graphs*, are the focus of this paper.

Very recent results come quite close to determining the complexity of the broadcast time problem in k-cycle graphs. In particular, several new findings have been published regarding the broadcast time problem in cactus graphs, which are closely related to k-cycle graphs. The first finding is the proof that the broadcast time problem is NP-complete in so-called "snowflake graphs" by Aminian et al. [1]. Snowflake graphs are a subclass of cactus graphs that have a pathwidth of at most 2. Thus, by proving the intractability of the problem in this particular graph family, they also showed that the complexity of the broadcast time problem is NP-complete in both general cactus graphs and general graphs with pathwidth at most 2. See Figure 1 for an example of a snowflake graph.

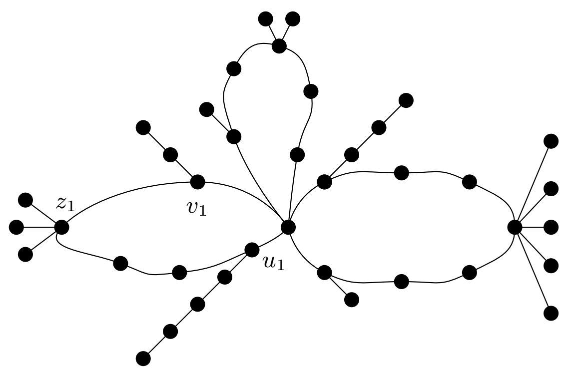

Fig. 1. An example of a snowflake graph. A snowflake graph consists of a k-cycle graph with paths originating from the vertices adjacent to the central vertex. One vertex on each cycle can also have any number of additional adjacent vertices. Snowflake graphs are cactus graphs with pathwidth at most 2 [1].

Recently, Egami et al. proved that the broadcast time problem is NP-complete in cactus graphs that are distance-1 to path forest [8] (see Figure 2). The reduction was performed from a restricted version of the the numerical matching with target sums problem, which simplified the reduction in [1]. Roughly speaking, given three sets of integers A, B, C, where $|A| = |B| = |C| = n$, introduce a cycle with $c + 3$ vertices for each $c \in C$. Then, for each $h \in \{1, 2, ..., max(C)\}$, add a path of length $max(C) - h + 1$ and connect it to the central vertex if $max(C) - h \notin A \cup B$. Figure 2 is an example of a graph resulting from this reduction where $A = \{1,3\}, B = \{2,4\}, C = \{3,7\}$. The class of cactus graphs that are distance-1 to path forest is a strict subclass of cactus graph. The class is also extremely similar to k-cycle graphs. More precisely, it is a k-cycle graph with some paths of arbitrary length originating from the central vertex. This finding indicates that the broadcast time problem is NP-hard even for very restricted families of cactus graphs. This makes the problem much more interesting in k-cycle graphs, which are a very closely related subclass of graphs for which the broadcast time problem remains open.

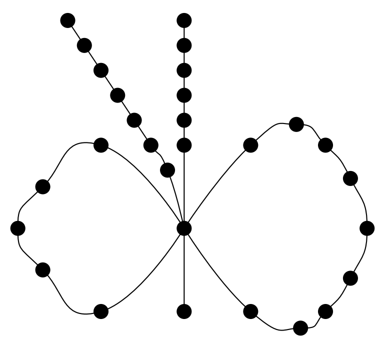

Fig. 2. An example of a graph created by the reduction in [8]. The graph consists of a k-cycle graph with paths connected to the central vertex.

The aforementioned findings regarding the NP-hardness of highly restricted families of cactus graphs raise the question of whether the broadcast time problem is also NP-hard for the next logical subfamily—k-cycle graphs. In fact, we are currently preparing a manuscript in which we prove that the broadcast time problem is NP-complete for k-cycle graphs and k-path graphs, and present a PTAS for each graph class. In light of these findings, approximation algorithms become much more important. While we do not close the problem, in this paper, we present a simple polynomial-time $(1.5 - \epsilon)$-approximation algorithm for the broadcast time problem in k-cycle graphs. By $(1.5 - \epsilon)$, we mean an approximation factor strictly less than 1.5. For any given graph, the factor is $1.5 - \epsilon$, where $\epsilon > 0$ depends on the structure of the graph. This improves the current best known approximation ratio of 2 [3].

The paper is organized as follows. In Section 2, we formally describe the family of k-cycle graphs. We also discuss previous related findings. In Section 3, we present our approximation algorithm. In Sections 4 and 5, we provide a proof that our proposed algorithm has a $(1.5 - \epsilon)$ approximation ratio. In Section 7, we analyze the algorithm in particular cases where the approximation is tight or the algorithm is optimal. Finally, we conclude the paper in Section 6.

2 k-Cycle Graphs

In this section, we discuss a subfamily of cactus graphs that are usually referred to as *k-cycle* graphs or *flower graphs.* A k-cycle graph $G = (C_1, C_2, ..., C_k)$ is obtained by connecting k cycles of arbitrary lengths at a single vertex v_c, called the central vertex.

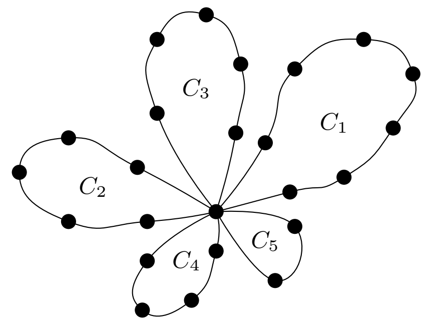

Fig. 3. An example of a k-cycle graph with $k = 5$ and $l_1 = 7, l_2 = 5, l_3 = 5, l_4 = 4, l_5 = 2$.

We assume that cycles are indexed in non-increasing order of their lengths. Formally, $l_1 \geq l_2 \geq ... \geq l_k \geq 2$, where l_i is the number of vertices on cycle C_i (excluding v_c) for all $1 \leq i \leq k$.

As noted above, the broadcast time problem has been researched at length for cactus graphs and graphs with pathwidth at most 2 in [1]. The authors proved that the broadcast time problem is NP-complete in such graphs through a series of reductions originating at the known NP-hard problem 3, 4-SAT and transforming it into an instance of the broadcast time problem in a snowflake graph (Figure 1). The authors also present a polynomial-time 2-approximation algorithm. This extends the best-known approximation ratio of 2 in k-cycle graphs found in [3] to general cactus graphs.

In [3], the authors introduced a 2-approximation algorithm for the broadcast time problem in general k-cycle graphs. Additionally, for some infinite input classes of k-cycle graphs, the authors showed that their algorithm has an improved approximation or is optimal. The results introduced in [3] are presented in Table 1.

The main algorithm introduced in [3] (S_{cycle}) maintains 2 lists of cycles, sorted in non-increasing order of the number of uninformed vertices. One contains all cycles that have been informed once, and the other contains all cycles which have not yet been informed. In each time unit, it compares the number of uninformed vertices in the cycles in the first position of each list to determine which to inform. The same paper presents a second algorithm (A_{cycle}) which is optimal in certain cases. A_{cycle} informs the $\left\lfloor \frac{k}{2} \right\rfloor$ smallest cycles in decreasing order of length, then the $\left\lceil \frac{k}{2} \right\rceil$ largest cycles in decreasing order of length.

Then, each cycle is informed in decreasing order of the number of remaining vertices at time unit k. In this paper, we design a simple algorithm that achieves a better approximation considering only the lengths of the cycles. Through this technique, the order of calls made by v_c is entirely predetermined at time unit 1.

Case	Algorithm	Result
General k-cycle	S_{cycle}	$(2-\epsilon)$-approximation
$l_j \geq l_{j+1}+4$	S_{cycle}	optimal
$l_{j+1}+4 \geq l_j \geq l_{j+1}+3$	S_{cycle}	$\frac{7}{6}$-approximation for $k \geq 10$
$l_j = l_{j+1}+2$	S_{cycle}	optimal
$l_j = l_{j+1}$	S_{cycle}	optimal
$l_j \leq l_{j+1}+1$	S_{cycle}	$(1.5-\epsilon)$-approximation
$l_j = l_{j+1}+1$	A_{cycle}	optimal

Table 1. Summary of known results for k-cycle graphs [3, 4]

In the same paper, the authors also presented several lower bounds for the broadcast time problem in k-cycle graphs. Given a k-cycle graph G_k with a central vertex v_c, and an originator u, the authors proved the following.

Lemma 1. *([3]). Let G_k be a k-cycle graph where the originator is the central vertex u. Then*

1. $b(u) \geq k+1$
2. $b(u) \geq \left\lceil \frac{l_j+2j-1}{2} \right\rceil$ *for any* $j, 1 \leq j \leq k$.

Lemma 2. *([3]) Let G_k be a k-cycle graph where the originator is any vertex w on a cycle C_m and the length of the shortest path from w to vertex u is d. Then* $b(w) \geq d + \left\lceil \frac{l_j+2j-2}{2} \right\rceil$ *for any* j, $1 \leq j \leq k$, *where* $j \neq m$.

3 Simple-k-cycle Algorithm

In this section, we will describe Simple-k-cycle, our approximation algorithm for the broadcast time problem in k-cycle graphs. Unlike the 2-approximation algorithm described above, our algorithm is very simple. There are two distinct cases: the originator vertex (u) is the central vertex v_c, or the originator vertex is found on a cycle C_m and is $d \geq 1$ vertices away from v_c. When $u = v_c$, v_c informs each cycle C_i in time units i and $i+k$ for any i, $1 \leq i \leq k$. When $u \neq v_c$ and is found on C_m, d vertices away from v_c, the algorithm is as follows:

1. u informs its neighbour along the shortest path to v_c in time unit 1
2. In time unit 2, u informs its other neighbour
3. In time unit d, v_c is informed by its neighbour on C_m

4. Then, in time units $d+1, ..., d+m-1$, v_c informs cycles $C_1, ..., C_{m-1}$
5. In time unit $d+m$, v_c skips informing C_m
6. Then, in time units $d+m, ..., d+k-1$, v_c informs cycles $C_{m+1}, ..., C_k$
7. v_c informs each cycle C_i for a second time in time unit $d+k-1+i$ for all $i = 1, ..., k$

Algorithm 1 SIMPLE-K-CYCLE

Input A k-cycle graph $G = \{C_1, ..., C_k\}$ with central vertex v_c and an originator vertex u

Output A broadcast scheme with time $t_A(v_c, G)$

```
procedure SIMPLE-K-CYCLE(G, u)
    if u = v_c then
        for 1 ≤ i ≤ k do
            v_c informs C_i in time unit i
        end for
        for k + 1 ≤ i ≤ 2k do
            v_c informs C_{i−k} in time unit i
        end for
    else
        In time unit 1, u informs its neighbour along the shorter path to v_c
        In time unit 2, u informs its neighbour along the longer path to v_c
        v_c is informed by its neighbour on C_m in time unit d
        for 1 ≤ i ≤ m − 1 do
            v_c informs C_i in time unit d + i
        end for
        for m + 1 ≤ i ≤ k do
            v_c informs C_i in time unit d + i − 1
        end for
        for 1 ≤ i ≤ k do
            v_c informs C_i in time unit d + k − 1 + i
        end for
    end if

end procedure
```

In the following sections, we will show that SIMPLE-K-CYCLE is a $(1.5-\epsilon)$-approximation algorithm. We will show this by considering 2 cases. The 2 cases are $u = v_c$, and $u \neq v_c$. We then analyze the performance of the algorithm in specific cases. To conclude this section, we provide a complexity analysis of our algorithm.

Complexity Analysis The complexity of SIMPLE-K-CYCLE is $O(k)$. $k < n$, so the complexity of the algorithm is also $O(n)$. The algorithm iterates over each of the k cycles at most twice. Once the cycles have been iterated over, we know exactly in which time units each cycle is informed. In constant time, one can calculate the time it takes for all the vertices of a cycle to be informed given the time units at which it was called from v_c. So, each cycle is iterated over twice, and we perform a constant-time operation for each cycle. Thus, as stated above, the complexity of SIMPLE-K-CYCLE is $O(k)$ and consequently, is also $O(n)$.

4 Broadcasting from the Central Vertex

This section discusses the broadcast time generated by SIMPLE-K-CYCLE when the originator is v_c. Recall, the algorithm informs cycle C_i, $1 \leq i \leq k$ in time units i and $k+i$. See Figure 4.

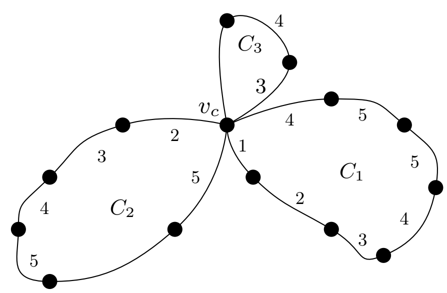

Fig. 4. An example of a broadcast scheme generated by SIMPLE-K-CYCLE on a k-cycle graph with $l_1 = 6, l_2 = 5, l_3 = 2$ and originator v_c. v_c calls cycle C_1 in time units 1 and 4, cycle C_2 in time units 2 and 5, and cycle C_3 in time unit 3. The broadcast is completed in 5 time units.

Theorem 1. *Algorithm 1 is a polynomial-time* $(1.5-\epsilon)$*-approximation algorithm for general* k*-cycle graphs when the originator is* v_c.

Proof. Let $t_A(v_c)$ be the broadcast time generated by Algorithm 1, and $t_{opt}(v_c)$ be the optimal broadcast time for G with originator v_c. To calculate $\frac{t_A(v_c)}{t_{opt}(v_c)}$, we will consider two cases: $t_A(v_c) \geq 2k$, and $t_A(v_c) < 2k$.

Case 1 $\boldsymbol{t_A(v_c) \geq 2k}$ If $t_A(v_c) \geq 2k$, then all cycles $C_1, ..., C_k$ get 2 calls from v_c. Since cycle C_i is informed by v_c at times i and $k+i$, k vertices in C_i are informed by time $k+i$. Thus, starting at time $k+i$, there are $l_i - k$ uninformed vertices in cycle C_i. At each time unit starting at time $k+i$, two additional vertices are informed in C_i. So, it takes $\lceil \frac{l_i-k}{2} \rceil$ time units to inform the remaining vertices in cycle C_i once it receives the second call from v_c at time unit $k+i$. So, for all $i, 1 \leq i \leq k$, cycle C_i is fully informed by time:

$$t_i = k + i - 1 + \left\lceil \frac{l_i - k}{2} \right\rceil = \left\lceil \frac{2i - 2 + k + l_i}{2} \right\rceil \tag{1}$$

Also, from lower bound (Lemma 1) we get

$$2t_{opt}(v_c) - 2i + 1 \geq l_i \tag{2}$$

Naturally, $t_A(v_c) = max_{1 \leq i \leq k}\{t_i\}$. Combining (1), and (2), we get the following.

$$t_i = \left\lceil \frac{2i-2+k+l_i}{2} \right\rceil \leq \left\lceil \frac{2i-2+k+2t_{opt}(v_c)-2i+1}{2} \right\rceil = t_{opt}(v_c) + \left\lceil \frac{k-1}{2} \right\rceil \tag{3}$$

Additionally from lower bound (Lemma 1), we have $t_{opt}(v_c) - 1 \geq k$. So, we know that $t_A(v_c) = max_{1 \leq i \leq k}\{t_i\} \leq t_{opt}(v_c) + \left\lceil \frac{t_{opt}(v_c)-2}{2} \right\rceil$. As a result,

$$\frac{t_A(v_c)}{t_{opt}(v_c)} \leq \frac{t_{opt}(v_c) + \left\lceil \frac{t_{opt}(v_c)-2}{2} \right\rceil}{t_{opt}(v_c)} \leq 1 + \frac{\left\lceil \frac{t_{opt}(v_c)-2}{2} \right\rceil}{t_{opt}(v_c)} = 1 + \frac{\left\lceil \frac{t_{opt}(v_c)}{2} \right\rceil}{t_{opt}(v_c)} - \frac{1}{t_{opt}(v_c)}$$
$$\leq 1 + \frac{0.5t_{opt}(v_c) + 0.5}{t_{opt}(v_c)} - \frac{1}{t_{opt}(v_c)} = 1.5 - \frac{0.5}{t_{opt}(v_c)} < 1.5$$

So, $\frac{t_A(v_c)}{t_{opt}(v_c)} < 1.5$ when $t_A(v_c) \geq 2k$

Case 2: $\boldsymbol{t_A(v_c) < 2k}$ We will prove this case by contradiction. If $t_A(v_c) = k+p$, for some $1 \leq p < k$, then cycles $C_1, ..., C_p$ are informed by v_c twice, while cycles $C_{p+1}, ...C_k$ are only informed once. Cycles $C_1, ..., C_p$ are informed exactly like in case 1, so they are fully informed before $1.5t_{opt}$. Consider cycles $C_{p+1}, ..., C_k$.

Let i be the smallest value $p+1 \leq i \leq k$ such that cycle C_i is only fully informed at time unit $t_A(v_c) = p+k$. For the sake of contradiction, assume $t_A(v_c) = k+p \geq 1.5t_{opt}(v_c)$. In this case, C_i is fully informed at exactly $t_A(v_c)$, so $t_i = t_A(v_c)$. Additionally, since cycle C_i is informed only once at time i, we have $t_A(v_c) = i - 1 + l_i$. From this and (2),

$$t_A(v_c) = p + k = i - 1 + l_i$$
$$t_A(v_c) = p + k \leq i - 1 + 2t_{opt}(v_c) - 2i + 1$$
$$t_A(v_c) = p + k \leq 2t_{opt}(v_c) - i$$

From our assumption that $k + p \geq 1.5t_{opt}(v_c)$ and above,

$$1.5t_{opt}(v_c) \leq p + k \leq 2t_{opt}(v_c) - i$$

Then, $1.5t_{opt}(v_c) \leq 2t_{opt}(v_c) - i$ which leads to $i \leq \frac{t_{opt}(v_c)}{2}$. However, from our selection of i, we know that $p+1 \leq i$. Transiently, this means that $p+1 \leq \frac{t_{opt}(v_c)}{2}$, so $p \leq \frac{t_{opt}(v_c)}{2} - 1$. Also, since $t_{opt}(v_c) \geq k+1$, we have that $k \leq t_{opt}(v_c) - 1$. Thus, substituting p and k from above,

$$t_A(v_c) = p + k \leq \frac{t_{opt}(v_c)}{2} - 1 + t_{opt}(v_c) - 1 = 1.5t_{opt}(v_c) - 2$$

This is a contradiction with the assumption that $t_A(v_c) \geq 1.5t_{opt}(v_c)$, so $\frac{t_A(v_c)}{t_{opt}(v_c)} < 1.5$ when $t_A(v_c) < 2k$. Therefore, Algorithm 1 is a $(1.5 - \epsilon)$-approximation algorithm when $u = v_c$.

5 Broadcasting from a Cycle Vertex

This section discusses the broadcast time generated by SIMPLE-K-CYCLE when the originator is not v_c. Instead, in this section, we assume that the originator u is found d vertices away from v_c on some cycle C_m. Recall, the algorithm informs along the shortest path to v_c. Once v_c is informed, it proceeds as in the previous section, but skips the first call to C_m since it is already informed. See Figure 5.

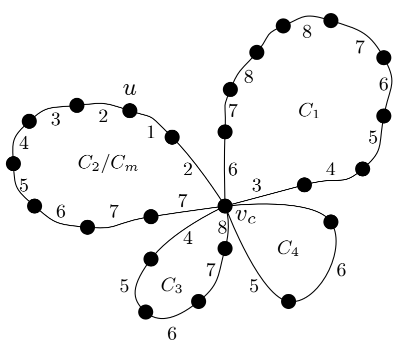

Fig. 5. An example of a broadcast scheme generated by SIMPLE-K-CYCLE on a k-cycle graph with $l_1 = 9, l_2 = 8, l_3 = 4, l_4 = 2$ and originator u with $dist(u, v_c) = d = 2$. u informs towards v_c in time unit 1, then informs its other neighbour in time unit 2. v_c is informed in time unit 2. Then, v_c calls cycle C_1 in time units 3 and 6, cycle C_2 in time unit 7, cycle C_3 in time units 4 and 8, and cycle C_4 in time unit 5. The broadcast is completed in 8 time units.

Theorem 2. *Algorithm 1 is a polynomial-time* $(1.5 - \epsilon)$*-approximation algorithm for general k-cycle graphs when the originator u is not* v_c.

Proof. Since $u \neq v_c$, we have that $dist(u, v_c) = d \geq 1$. It is clear that all the vertices on cycle C_m are informed no later than when the originator is v_c. Additionally, we know that $t_{opt}(u) \geq t_{opt}(v_c)$. So, by Theorem 1, cycle C_m is fully informed in less than $1.5t_{opt}(v_c) \leq 1.5t_{opt}(u)$ time units. We will therefore only consider the case where the broadcast time of our algorithm is not bounded by cycle C_m. This will become useful when we assume that some cycle C_i, with $i \neq m$, is fully informed in exactly $t_A(u)$ time units. Again, we will consider two cases: $t_A(u) \geq 2k + d - 1$ and $t_A(u) < 2k + d - 1$.

Case 1: $\boldsymbol{t_A(u) \geq 2k + d - 1}$ By lower bound (Lemma 2), we have $t_{opt}(u) \geq d + \lceil \frac{l_i+2i-2}{2} \rceil \geq d + \frac{l_i+2i-2}{2}$ where $1 \leq i \leq k$, $i \neq m$. This gives us

$$2t_{opt}(u) - 2d - 2i + 2 \geq l_i \tag{4}$$

In our approximation algorithm, all cycles C_i, $i \neq m$, $1 \leq i \leq k$ are informed by v_c by time units $d+i$ and $d+i+k-1$, irrespective of their size relative to cycle C_m.

By the second call from v_c, C_i has $k-1$ informed vertices. Starting at time unit $d+i+k-1$, C_i has two extra vertices informed at every time unit. So, all cycles C_i are fully informed at time unit $t_i = d+k+i-2+\left\lceil \frac{l_i-(k-1)}{2} \right\rceil = \left\lceil \frac{2i+k+2d-3+l_i}{2} \right\rceil$. Thus, substituting (4), we get

$$t_i \leq \left\lceil \frac{2i+k+2d-3+2t_{opt}(u)-2d-2i+2}{2} \right\rceil = \left\lceil \frac{k+2t_{opt}(u)-1}{2} \right\rceil = \left\lceil \frac{k-1}{2} \right\rceil + t_{opt}(u)$$

Since $t_A(u) = max_{1\leq i \leq k}(t_i)$, we have that $t_A(u) \leq \left\lceil \frac{k-1}{2} \right\rceil + t_{opt}(u)$. So,

$$\begin{aligned} \frac{t_A(u)}{t_{opt}(u)} &\leq \frac{t_{opt}(u) + \left\lceil \frac{k-1}{2} \right\rceil}{t_{opt}(u)} \leq 1 + \frac{\left\lceil \frac{t_{opt}(v_c)-2}{2} \right\rceil}{t_{opt}(u)} \leq 1 + \frac{\frac{t_{opt}(v_c)-1}{2}}{t_{opt}(u)} \\ &\leq 1 + \frac{0.5t_{opt}(u) - 0.5}{t_{opt}(u)} = 1.5 - \frac{0.5}{t_{opt}(u)} < 1.5 \end{aligned}$$

So, Simple-K-Cycle is a $(1.5-\epsilon)$-approximation algorithm when $t_A \geq 2k+d-1$.

Case 2: $\boldsymbol{t_A(u) < 2k+d-1}$ We will prove this case by contradiction. If $t_A(u) = k+p+d-1$, for some $1 \leq p < k$, then cycles $C_1, ..., C_p$ are informed by v_c twice, while cycles $C_{p+1}, ...C_k$ are only informed once. Cycles $C_1, ..., C_p$ are informed exactly like in case 1, so they must finish before $1.5t_{opt}(u)$. So, consider cases where the broadcast time is bounded some cycle $C_{p+1}, ..., C_k$.

Let i be the smallest value $p+1 \leq i \leq k$ such that cycle C_i is only fully informed at time unit $t_A(u) = p+k+d-1$. For the sake of contradiction, assume $t_A(u) = k+p+d-1 \geq 1.5t_{opt}(u)$. In this case, cycle C_i is fully informed at exactly $t_A(u)$, so $t_i = t_A(u)$. Note that all cycles C_j, $p+1 \leq j \leq k$ receive a call from v_c by time unit $d+j$ independent of whether $j < m$ or $j > m$. So, we can assume that cycle C_i receives a call from v_c at time $d+i$. Note that proving this case implicitly proves the case where C_i is informed earlier than time unit $d+i$. So, $t_A(u) = d+i-1+l_i$. Using (4) and the above, we get

$$\begin{aligned} t_A(u) &= p+k+d-1 = d+i-1+l_i \\ t_A(u) &= p+k+d-1 \leq d+i-1+2t_{opt}(u)-2i-2d+2 \\ t_A(u) &= p+k+d-1 \leq 2t_{opt}(u)-i-d+1 \end{aligned}$$

From our assumption that $k+p+d-1 \geq 1.5t_{opt}(u)$ and above,

$$1.5t_{opt}(u) \leq p+k+d-1 \leq 2t_{opt}(u)-i-d+1$$

Then, $1.5t_{opt}(u) \leq 2t_{opt}(u)-i-d+1$ which leads to $i \leq \frac{t_{opt}(u)}{2} - d+1$. However, from our selection of i we know that $p+1 \leq i$. Transiently, this means that $p+1 \leq \frac{t_{opt}(u)}{2} - d+1$. Thus, $p \leq \frac{t_{opt}(u)}{2} - d$. Also, since $t_{opt}(u) \geq k+1$, we have that $k \leq t_{opt}(u)-1$. Thus, substituting p and k from above,

$$t_A(u) = p+k+d-1 \leq \frac{t_{opt}(u)}{2} - d + t_{opt}(u) - 1 + d - 1 = 1.5t_{opt}(u) - 2$$

This is a contradiction with the assumption that $t_A(u) \geq 1.5t_{opt}(u)$, so $\frac{t_A(u)}{t_{opt}(u)} < 1.5$ when $t_A(u) < 2k + d - 1$. Therefore, Simple-K-Cycle is a $(1.5 - \epsilon)$-approximation algorithm when $u \neq v_c$.

6 Conclusion

In this paper, we designed a simple linear-time algorithm for the broadcast time problem in general k-cycle graphs. Further, we showed that the given algorithm is a $(1.5-\epsilon)$-approximation algorithm. This improves the previously best known approximation ratio of 2, found in [3] and generalized for cactus graphs in [1]. However, the complexity of the broadcast time problem in k-cycle graphs is still open. There is also no known inapproximability lower bound for this problem. This leaves several potential avenues for further research. It could be possible to devise an exact polynomial-time algorithm for this problem. Another potential direction for future research is to build upon the techniques used in [8] to prove the NP-hardness of the broadcast time problem in k-cycle graphs. The lack of any known inapproximability lower bound also leaves an opportunity to improve the known approximation ratio or to introduce such a lower bound. Another future direction is, of course, to extend this $(1.5 - \epsilon)$ approximation to general cactus graphs.

References

1. Aminian, A., Kamali, S., Seyed-Javadi, S.M., Sumedha: On the complexity of telephone broadcasting: From cacti to bounded pathwidth graphs (2025), `https://arxiv.org/abs/2501.12316`, to appear in Proceedings of ICALP 2025
2. Bar-Noy, A., Guha, S., Naor, J.S., Schieber, B.: Message multicasting in heterogeneous networks. SIAM Journal on Computing **30**(2), 347–358 (2000). https://doi.org/10.1137/S0097539798347906
3. Bhabak, P., Harutyunyan, H.A.: Constant approximation for broadcasting in k-cycle graph. In: Ganguly, S., Krishnamurti, R. (eds.) Algorithms and Discrete Applied Mathematics. pp. 21–32. Springer International Publishing, Cham (2015)
4. Bhabak, P., Harutyunyan, H.A., Concordia University (Montréal, Q., Concordia University (Montréal, Q.D.o.C.S., Engineering, S.: Approximation Algorithms for Broadcasting in Simple Graphs with Intersecting Cycles. Ph.D. thesis, [Concordia University], [Montréal, Québec] (2015), `http://spectrum.library.concordia.ca/979568/`
5. Bhabak, P., Harutyunyan, H.A., Kropf, P.: Efficient broadcasting algorithm in Harary-like networks. In: 2017 46th International Conference on Parallel Processing Workshops (ICPPW). pp. 162–170 (2017). https://doi.org/10.1109/ICPPW.2017.34
6. Bhabak, P., Harutyunyan, H.A., Tanna, S.: Broadcasting in Harary-like graphs. In: 2014 IEEE 17th International Conference on Computational Science and Engineering. pp. 1269–1276 (2014). https://doi.org/10.1109/CSE.2014.244
7. Čevnik, M., Žerovnik, J.: Broadcasting on cactus graphs. Journal of Combinatorial Optimization **33**(1), 292–316 (2017). https://doi.org/10.1007/s10878-015-9957-8

8. Egami, Y., Gima, T., Hanaka, T., Kobayashi, Y., Lampis, M., Mitsou, V., Nemery, E., Otachi, Y., Vasilakis, M., Vaz, D.: Broadcasting under structural restrictions (2025), https://arxiv.org/abs/2504.13669, to appear in Proceedings of MFCS 2025
9. Elkin, M., Kortsarz, G.: Combinatorial logarithmic approximation algorithm for directed telephone broadcast problem. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing. p. 438–447. STOC '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/509907.509972
10. Elkin, M., Kortsarz, G.: Sublogarithmic approximation for telephone multicast. Journal of Computer and System Sciences **72**(4), 648–659 (2006). https://doi.org/https://doi.org/10.1016/j.jcss.2005.12.002
11. Fomin, F.V., Fraigniaud, P., Golovach, P.A.: Parameterized complexity of broadcasting in graphs. In: Paulusma, D., Ries, B. (eds.) Graph-Theoretic Concepts in Computer Science. pp. 334–347. Springer Nature Switzerland, Cham (2023)
12. Fomin, F.V., Fraigniaud, P., Golovach, P.A.: Parameterized complexity of broadcasting in graphs. Theoretical Computer Science **997**, 114508 (2024). https://doi.org/https://doi.org/10.1016/j.tcs.2024.114508
13. Fraigniaud, P., Lazard, E.: Methods and problems of communication in usual networks. Discrete Applied Mathematics **53**(1), 79–133 (1994). https://doi.org/https://doi.org/10.1016/0166-218X(94)90180-5
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., USA (1979)
15. Gholami, S., Harutyunyan, H.A.: Broadcast graphs with nodes of limited memory. In: Pacheco, D., Teixeira, A.S., Barbosa, H., Menezes, R., Mangioni, G. (eds.) Complex Networks XIII. pp. 29–42. Springer International Publishing, Cham (2022)
16. Harutyunyan, H.A., Hovhannisyan, N., Maraachlian, E.: Broadcasting in chains of rings. In: 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN). pp. 506–511 (2023). https://doi.org/10.1109/ICUFN57995.2023.10201234
17. Harutyunyan, H.A., Liestman, A.L., , Peters, J.G., Richards, D.: Handbook of Graph Theory, chap. 12, pp. 1477–1494. Chapman Hall (2013)
18. Harutyunyan, H.A., Maraachlian, E.: Linear algorithm for broadcasting in unicyclic graphs. In: Lin, G. (ed.) Computing and Combinatorics. pp. 372–382. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
19. Harutyunyan, H.A., Maraachlian, E.: On broadcasting in unicyclic graphs. Journal of Combinatorial Optimization **16**(3), 307–322 (2008). https://doi.org/10.1007/s10878-008-9160-2
20. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks **18**(4), 319–349 (1988). https://doi.org/https://doi.org/10.1002/net.3230180406
21. Hromkovič, J., Klasing, R., Monien, B., Peine, R.: Dissemination of Information in Interconnection Networks (Broadcasting & Gossiping), pp. 125–212. Springer US, Boston, MA (1996). https://doi.org/10.1007/978-1-4757-2491-2_5, https://doi.org/10.1007/978-1-4757-2491-2_5
22. Kortsarz, G., Peleg, D.: Approximation algorithms for minimum-time broadcast. SIAM Journal on Discrete Mathematics **8**(3), 401–427 (1995). https://doi.org/10.1137/S0895480193245923
23. Middendorf, M.: Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2. Inf. Process. Lett. **46**(6), 281–287 (Jul 1993). https://doi.org/10.1016/0020-0190(93)90066-I

24. Proskurowski: Minimum broadcast trees. IEEE Transactions on Computers **C-30**(5), 363–366 (1981). https://doi.org/10.1109/TC.1981.1675796
25. Ravi, R.: Rapid rumor ramification: approximating the minimum broadcast time. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 202–213 (1994). https://doi.org/10.1109/SFCS.1994.365693
26. Schindelhauer, C.: On the inapproximability of broadcasting time. In: Jansen, K., Khuller, S. (eds.) Approximation Algorithms for Combinatorial Optimization. pp. 226–237. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
27. Slater, P.J., Cockayne, E.J., Hedetniemi, S.T.: Information dissemination in trees. SIAM Journal on Computing **10**(4), 692–701 (1981). https://doi.org/10.1137/0210052
28. Tale, P.: Telephone broadcast on graphs of treewidth two. Theoretical Computer Science **1045**, 115282 (2025). https://doi.org/https://doi.org/10.1016/j.tcs.2025.115282

APPENDIX

7 Performance of Simple-K-Cycle in Specific Cases

In this section, we present two illustrative examples. The first demonstrates that our approximation upper bound is tight. Specifically, for k-cycle graphs with relatively large k and small cycles of equal length, our algorithm produces a broadcast time that asymptotically approaches $1.5t_{opt}$. The other example shows that the algorithm can, in some cases, produce an optimal solution.

Approximation Ratio Approaching 1.5

Here, we will provide an example of an infinite input class of graphs for which the approximation ratio of Simple-K-Cycle approaches 1.5. Let G be a k-cycle graph with $l_1 = l_2 = ... = l_k$. Then, let $t_A(v_c)$ be the broadcast time of v_c in G under the broadcast scheme generated by Simple-K-Cycle. Also, let $t_{opt}(v_c)$ be the optimal broadcast time of v_c in G. Since all cycles have the same length, $t_A(v_c)$ is bound by the last cycle to be informed, C_k. Assuming $t_A(v_c) \geq 2k$, C_k is informed for a second time in time unit $2k$. By this time unit, k vertices have already been informed. So,

$$t_A(v_c) = (2k-1) + \left\lceil \frac{l_k - k}{2} \right\rceil = \left\lceil \frac{l_1 + 3k - 2}{2} \right\rceil \leq \frac{l_1 + 3k - 1}{2}$$

Thus, we have $\frac{l_1+3k-2}{2} \leq t_A(v_c) \leq \frac{l_1+3k-1}{2}$. The optimal broadcast time when all cycles are of equal length is achieved by informing each cycle C_i in time units i and $2k+1-i$. In other words, inform cycles in order, then in reverse order. This guarantees that each cycle is fully informed in the same time unit ($t_{opt}(v_c)$). So, in an optimal broadcast scheme C_k is informed in time units k and $k+1$. Also, C_k is fully informed in exactly $t_{opt}(v_c)$ time units. Thus, we have that

$$t_{opt}(v_c) = k + \left\lceil \frac{l_k - 1}{2} \right\rceil = \left\lceil \frac{2k + l_1 - 1}{2} \right\rceil \geq \frac{2k + l_1 - 1}{2}$$

Therefore, $\frac{2k+l_1-1}{2} \leq t_{opt}(v_c) \leq \frac{2k+l_1}{2}$. This gives us the following approximation ratio which depends on k and l_i.

$$\frac{l_1 + 3k - 2}{l_1 + 2k} \leq \frac{t_A(v_c)}{t_{opt}(v_c)} \leq \frac{l_1 + 3k - 1}{l_1 + 2k - 1}$$

This shows that when $l_1 = l_2 = ... = l_k$, the approximation ratio of Simple-K-Cycle asymptotically approaches 1.5 when k is very large in relation to l_1. More formally, for any $\epsilon > 0$, there exists a pair l_1, k such that $\frac{l_1+3k-2}{l_1+2k} < 1.5 - \epsilon$ and $\frac{l_1+3k-1}{l_1+2k-1} < 1.5 - \epsilon$, so $\frac{t_A(v_c)}{t_{opt}(v_c)} < 1.5 - \epsilon$. Furthermore, we see that $\frac{t_A(v_c)}{t_{opt}(v_c)} < 1.5$ since $l_1 > 1$.

Simple-k-cycle is Optimal

Here, we will provide several cases where Simple-K-Cycle is optimal. We continue to use the notation used in the previous example.

Any broadcast scheme in which all cycles are fully informed in exactly the last time unit is optimal by the pigeonhole principle. Let $t_A = p + k$, where $1 \leq p$. In this case, all cycles C_i, $1 \leq i \leq p$, $i \leq k$ are informed twice, while all cycles C_j, $p < j \leq k$, are informed only once. All cycles C_i are informed in time units i and $k+i$. So, they each have k informed vertices by time unit $k+i$ and have two additional vertices informed in each time unit between $k+i$ and $p+k$. So, they have $k + 2(p + k - (k + i - 1)) = k + 2p - 2i + 2$ informed vertices by time unit $k+p$. By the same logic, every cycle C_j is informed in only time unit j. So, they have $p + k - (j - 1) = p + k - j + 1$ informed vertices by time unit $p+k$.

For all cycles to be fully informed in exactly time unit $p+k$, using the logic above, the following must be true. $l_i = k + 2p - 2i + 2$, $1 \leq i \leq p$, $i \leq k$, and $l_j = p + k - j + 1$, $p < j \leq k$.

From this, we can see that $l_i - l_{i+1} = 2$, so the lengths of all cycles that are informed by v_c twice must differ by 2. Also, $l_j - l_{j+1} = 1$, so the lengths of all cycles that are informed only once must differ by 1. If $p < k$, then $l_p - l_{p+1} = k + 2p - 2p + 2 - (p + k - (p + 1) + 1) = 2$, so the last cycle to be informed twice is 2 vertices larger than the first cycle to be informed once. This has a few implications.

$\boldsymbol{t_A(v_c) \geq 2k}$ *and* $\boldsymbol{l_i = l_{i+1} + 2}$ When $t_A(v_c) \geq 2k$, all cycles are informed by v_c twice. By the logic above, this means that Simple-K-Cycle generates an optimal broadcast scheme for any graph that has $l_i = l_{i+1} + 2$, $1 \leq i < k$. Since the logic used to show that $l_i = l_{i+1} + 2$ is independent of k or p, this statement holds for any number of cycles and any $t_A(v_c) \geq 2k$.

$\boldsymbol{t_A(v_c) < 2k}$ When $t_A(v_c) < 2k$, there are some cycles that are only informed once. As seen above, the lengths of these cycles must differ from each other by exactly 1. Additionally, the first cycle to be informed only once C_{p+1} must be exactly 2 vertices smaller than the last cycle to be informed twice C_p. Thus, Simple-K-Cycle generates an optimal broadcast scheme for any graph with $l_1 = l_2 + 2 = ... = l_p + 2(p-1) = l_{p+1} + 2p = l_{p+2} + 2p + 1 = ... = l_k + 2p + k - 1$, where $l_1 = k + 2p$ and $t_A(v_c) < 2k$.