

Improved Lower Bounds for the Cyclic Bandwidth Problem

Hugues Déprés¹, Guillaume Fertin²[0000–0002–8251–2012], and Eric Monfroy³[0000–0001–7970–1368]

¹ ENS de Lyon, Computer Science Department, Lyon, France
hugues.depres@ens-lyon.fr

² Université de Nantes, LS2N (UMR 6004), CNRS, Nantes, France
guillaume.fertin@univ-nantes.fr

³ LERIA, Université d'Angers, France
eric.monfroy@univ-angers.fr

Abstract. We study the classical CYCLIC BANDWIDTH problem, an optimization problem which takes as input an undirected graph $G = (V, E)$ with $|V| = n$, and asks for a labeling φ of V in which every vertex v takes a unique value $\varphi(v) \in [1; n]$, in such a way that $B_c(G, \varphi) = \max\{\min_{uv \in E(G)}\{|\varphi(u) - \varphi(v)|, n - |\varphi(u) - \varphi(v)|\}\}$, called the *cyclic bandwidth of G* , is minimized.

We provide three new and improved lower bounds for the CYCLIC BANDWIDTH problem, applicable to any graph G : two are based on the neighborhood vertex density of G , the other one on the length of a longest cycle in a cycle basis of G . We also show that our results improve the best known lower bounds for a large proportion of a set of instances taken from a frequently used benchmark, the Harwell-Boeing sparse matrix collection. Our third proof provides additional elements: first, an improved sufficient condition yielding $B_c(G) = B(G)$ (where $B(G) = \min_{\varphi} \{\max_{uv \in E(G)} \{|\varphi(u) - \varphi(v)|\}\}$ denotes the *bandwidth* of G); second, an algorithm that, under some conditions, computes a labeling reaching $B(G)$ from a labeling reaching $B_c(G)$.

Keywords: Cyclic Bandwidth problem, Graph labeling, NP-hard problem, Optimization, Lower bounds

1 Introduction

Let $G = (V, E)$ be an undirected graph without loops and multiple edges. Let $|V| = n$, and let φ be a bijection from V to $[n]$, where $[n]$ denotes the set $\{1, 2, 3 \dots n\}$. In the rest of the paper, φ will be called a *cb-labeling* of G (where “*cb*” stands for “cyclic bandwidth”). For any value $p \in \mathbb{Z}$, we let $|p|_n = \min\{|p|, n - |p|\}$. The Cyclic Bandwidth of G induced by φ , which we denote $B_c(G, \varphi)$, is defined as follows: $B_c(G, \varphi) = \max_{uv \in E} \{|\varphi(u) - \varphi(v)|_n\}$. The *Cyclic Bandwidth* of a graph G , denoted $B_c(G)$, is the minimum value $B_c(G, \varphi)$

among all possible labelings φ , and such a labeling will be called an *optimal cb-labeling*. We are now ready to define the CYCLIC BANDWIDTH problem, which is the following optimization problem: given a graph $G = (V, E)$ with n vertices, find $B_c(G)$. In the following, we shall denote by φ^* any optimal *cb-labeling* of G , i.e., any labeling of G satisfying $B_c(G, \varphi^*) = B_c(G)$.

The CYCLIC BANDWIDTH problem has been introduced in [6], in the context of designing a ring interconnection network between computers. It can be seen as a variant of the well-known BANDWIDTH MINIMIZATION problem, whose formal introduction is due to Harper [3] in 1964 – see also the survey from Chinn et al. [1] for a more detailed historical account. The BANDWIDTH MINIMIZATION problem also asks for a *bm-labeling* φ (where “*bm*” stands for “bandwidth minimization”), i.e., a bijection from V to $[n]$. The value computed from φ is in that case $B(G, \varphi) = \max_{uv \in E} \{|\varphi(u) - \varphi(v)|\}$. The BANDWIDTH MINIMIZATION problem asks for a labeling φ^* such that $B(G, \varphi^*)$ is minimized, and the latter value is denoted $B(G)$. Any optimal labeling will be called an *optimal bm-labeling* of G . Note that there is a strong connection between $B(G)$ and $B_c(G)$, starting with the trivial fact that for any graph G , $B_c(G) \leq B(G)$.

The CYCLIC BANDWIDTH has been extensively studied. It has been shown to be NP-hard, even in the case of trees with maximum degree 3 [8]. The value of $B_c(G)$ has also been determined when G belongs to a specific class of graphs, e.g., paths, cycles, Cartesian products of paths (resp. of cycles, of paths and cycles), full k -ary trees, complete graphs, complete bipartite graphs, and unit interval graphs [2, 7, 4, 8]. Some other works studied the relationship between $B_c(G)$ and $B(G)$, and notably aimed at determining sufficient conditions under which the equality $B(G) = B_c(G)$ holds [9, 5]. In particular, it has been shown that $B(T) = B_c(T)$ for any tree T [4]. Another set of results is concerned with determining bounds for $B_c(G)$, and notably lower bounds, for general graphs [15]. First, it is easy to see that for any graph G , $B_c(G) \geq \frac{\Delta(G)}{2}$, where $\Delta(G)$ denotes the maximum degree of G . Besides, for any graph G , we have $\frac{B(G)}{2} \leq B_c(G) \leq B(G)$, the leftmost bound being from [9]. Other lower bounds have been obtained, most of these results being based on the density or on a relevant cycle basis of the studied graph (see, e.g., [4, 15] which are in connection with our results).

Finally, more recent papers are concerned with designing efficient heuristics for the CYCLIC BANDWIDTH problem (see, e.g., [14, 13, 12]) where both execution time and upper bounds for $B_c(G)$ are examined – the latter being tested on a subset of the classical Harwell-Boeing sparse matrix collection (see, e.g., <https://math.nist.gov/MatrixMarket/collections/hb.html>).

Our goal in this paper is to provide lower bounds for $B_c(G)$ that apply to *any* graph G . We essentially prove three lower bounds: two of them rely on the graph density (Theorems 1 and 3), and both improve the best known results; the third one relies on a relevant cycle basis for G (Theorem 9), and also improves the best known related result. The latter result also presents improved sufficient conditions under which $B_c(G) = B(G)$, and provides a relabeling algorithm which, under these conditions, and given a *cb-labeling* φ of G , computes a *bm-*

labeling φ' of G such that $B_c(G, \varphi) = B(G, \varphi')$. We also successfully applied our results to a benchmark of 28 Harwell-Boeing graphs.

This paper is organized as follows: in Section 2 we discuss our graph density lower bounds, and in Section 3 our lower bound based on a relevant cycle basis of G . Section 4 describes our results on the abovementioned benchmark.

2 Graph Density Lower Bounds

In this section, we present, in Theorems 1 and 3, two new lower bounds on $B_c(G)$ that apply to any graph G . As discussed later, they can be seen as generalizations of previously existing lower bounds.

Theorem 1. *Let G be a graph, v any vertex of G , and $i \geq 1$ an integer. Let $N_i(v)$ denote the set of vertices within distance i from v in G . Then we have $B_c(G) \geq \left\lceil \frac{|N_i(v)|-1}{2^i} \right\rceil$.*

In order to prove Theorem 1, the following lemma is needed.

Lemma 2 *Let $d \geq 0$ be an integer, and let u_0, u_1, \dots, u_d be a path of length d in G . Let φ^* be an optimal cb -labeling of G . Then, $|\varphi^*(u_d) - \varphi^*(u_0)|_n \leq d \cdot B_c(G)$.*

Proof. First, note that $|x - y|_n$ satisfies the triangle inequality. Since φ^* is an optimal cb -labeling of G , by definition we have $|\varphi^*(u_{i+1}) - \varphi^*(u_i)|_n \leq B_c(G)$ for any $0 \leq i < d$. Consequently, summing the above expression for all $0 \leq i < d$ yields the required result. \square

Proof (of Theorem 1). Let $i \geq 1$ be an integer, and recall that $N_i(v)$ denotes the set of vertices lying in distance less than or equal to i from v in G . Let φ^* be an optimal cb -labeling of G , and by extension let $\varphi^*(N_i(v))$ be the set of labels used by φ^* to label the vertices of $N_i(v)$. From Lemma 2 we conclude that $\varphi^*(N_i(v)) \subseteq [\varphi^*(v) - i \cdot B_c(G); \varphi^*(v) + i \cdot B_c(G)] \bmod n$, since for any vertex $u \in N_i(v)$ there exists a path of length at most i between v and u .

The above interval $[\varphi^*(v) - i \cdot B_c(G); \varphi^*(v) + i \cdot B_c(G)] \bmod n$ contains $2i \cdot B_c(G) + 1$ distinct values. Since φ^* is injective, we have that $|N_i(v)| \leq 2i \cdot B_c(G) + 1$, which concludes the proof. \square

Note that Theorem 1 generalizes the trivial lower bound $B_c(G) \geq \left\lceil \frac{\Delta(G)}{2} \right\rceil$ (where $\Delta(G)$ denotes the maximum degree of G): take for this $i = 1$ and any vertex v of degree $\Delta(G)$. Our second result has the same flavor as Theorem 1 above. However, instead of relying on $N_i(v)$, it considers the number of vertices that lie at distance at most i from *either one of two vertices u or v* of G , where $uv \in E(G)$.

Theorem 3. *Let $G = (V, E)$ be a graph, u and v two distinct vertices of G , $i \geq 1$ an integer, and $N_i(u, v)$ the set of vertices within distance i from either u or v in G . Then, for every edge $uv \in E$, we have $B_c(G) \geq \left\lceil \frac{|N_i(u, v)|-1}{2^{i+1}} \right\rceil$.*

Proof. The arguments are somewhat similar to those used in proof of Theorem 1. Take any edge $uv \in E(G)$, and observe an optimal cb -labeling φ^* of G . First, note that $|\varphi^*(u) - \varphi^*(v)|_n \leq B_c(G)$ by definition. Moreover, by Lemma 2, we know that the set of labels used by φ^* to label the vertices of $N_i(u, v)$ is included in a closed interval of size $(2i + 1) \cdot B_c(G)$, since any two vertices of $N_i(u, v)$ are connected by a path of length at most $2i + 1$. Since φ^* is injective, we conclude from the above that $|N_i(u, v)| \leq (2i + 1) \cdot B_c(G) + 1$, which proves the result. \square

We note that, although Theorems 1 and 3 look similar, none of these two results strictly contains the other. Note also that Theorem 3 generalizes a result for lower bounds on trees proved in [8]. Due to lack of space, the above two claims are not proved here. We finally note that the lower bounds of Theorems 1 and 3 are tight for several well-known classes of graphs, such as paths, cycles, complete graphs, and caterpillars.

3 Cycle Basis Lower Bound

The whole current section is devoted to proving Theorem 9, which provides a lower bound for $B_c(G)$ based on both $B(G)$ and the length ℓ of the longest cycle in a cycle basis of G . Theorem 9 also gives an improved sufficient condition under which $B_c(G) = B(G)$, also based on ℓ . Theorem 9 can be considered as somewhat similar to Theorem 3.1 from [4], but is actually different as it provides a lower bound for $B_c(G)$.

Another interesting point is that, on the way to proving Theorem 9, we also prove the following result, of independent interest: there exists a labeling algorithm (namely, Algorithm 1) that, when $\ell < \frac{n}{B_c(G)}$, computes a bm -labeling φ' of G starting from a cb -labeling φ of G , such that $B(G, \varphi') \leq B_c(G, \varphi)$. In particular, if φ is optimal, so is φ' , and thus $B_c(G) = B(G)$. The interest of Algorithm 1 lies in the fact that, when $\ell < \frac{n}{B_c(G)}$, a labeling φ satisfying $B_c(G, \varphi) = k$ does not necessarily satisfy $B(G, \varphi) = k$ – and in that case another labeling φ' is needed in order to reach $B(G, \varphi') = k$.

Theorem 9 relies on Lemma 7, which we prove first. Before that, we start with some definitions.

Definition 4 Let G_1 and G_2 two graphs such that $V(G_1) = V(G_2) = V$. The pairwise symmetric difference of G_1 and G_2 , denoted $psd(G_1, G_2)$, is the graph G' whose set of vertices is V , and whose edges are those belonging to exactly one of the two graphs G_1 and G_2 . By extension, the symmetric difference $sd(\mathcal{G})$ of a set $\mathcal{G} = \{G_1, G_2, \dots, G_p\}$ of graphs, each built on the same set V of vertices, is $psd(G_1, psd(G_2(\dots psd(G_{p-1}, G_p))))$.

Definition 5 A cycle basis of a graph G is a set of cycles \mathcal{C} of G such that any cycle of G either belongs to \mathcal{C} or can be obtained by symmetric difference of a subset of \mathcal{C} .

Finally, we use a definition from [4], which will prove useful in the following.

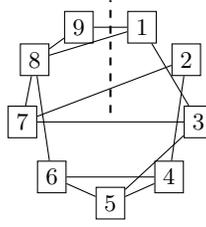


Fig. 1. Illustration of Def. 6 on a graph G with $n = 9$ with a cb -labeling φ such that $B_c(G, \varphi) = 4$. Each edge uv crossing the dashed vertical line has a non-zero $\alpha_\varphi(u, v)$.

Definition 6 Given a graph $G = (V, E)$, let $S \subset V \times V$ be such that $(u, v) \in S$ iff $uv \in E(G)$. Let φ be a cb -labeling of V , and let $\alpha_\varphi : S \rightarrow \{-1, 0, 1\}$ denote the following function:

$$\alpha_\varphi(u, v) = \begin{cases} 0 & \text{if } |\varphi(u) - \varphi(v)| \leq B_c(G, \varphi) \\ 1 & \text{if } \varphi(u) - \varphi(v) \geq n - B_c(G, \varphi) \\ -1 & \text{if } \varphi(u) - \varphi(v) \leq B_c(G, \varphi) - n \end{cases}$$

In other words, $\alpha_\varphi(u, v) = 0$ whenever u and v have “close enough labels”. Otherwise, $\alpha_\varphi(u, v) = \pm 1$, depending on which vertex among u and v has the greatest label. Hence, α_φ is not symmetric, since for any edge uv in $E(G)$, $\alpha_\varphi(u, v) = -\alpha_\varphi(v, u)$. In the following, we will say that an edge $uv \in E(G)$ has a *zero* α_φ when $\alpha_\varphi(u, v) = 0$, and a *non-zero* α_φ otherwise.

Figure 1 illustrates the above definition. This figure represents a graph G with $n = 9$ vertices, together with a cb -labeling φ satisfying $B_c(G, \varphi) = 4$ (note that φ is not optimal, since $B(G) = 2$ in this case). Vertices of G are displayed along a cycle, ordered (clockwise) by their labels in φ . For readability, we assume that labels and vertex identifiers are the same.

It can be seen that all edges having a non-zero α_φ are the ones that cross the dashed vertical line – because that line separates edges whose end vertices carry labels that are strictly more than $B_c(G, \varphi) = 4$ apart. The important thing to notice here is that, for any edge uv of G , the sign of the corresponding α_φ depends in which direction the dashed line is crossed, for example $\alpha_\varphi(7, 2) = \alpha_\varphi(8, 1) = \alpha_\varphi(9, 1) = 1$ (the dashed line is crossed from left to right), while $\alpha_\varphi(2, 7) = \alpha_\varphi(1, 8) = \alpha_\varphi(1, 9) = -1$ (the dashed line is crossed from right to left). In other words, when computing $\alpha_\varphi(u, v)$ for an edge $uv \in E(G)$, an orientation is given to that edge.

Let us define $s_\alpha(C) = \alpha_\varphi(x_p, x_0) + \sum_{i=0}^{p-1} \alpha_\varphi(x_i, x_{i+1})$ for any cycle $C = (x_0, x_1, \dots, x_p, x_0)$ of G , and let us call $s_\alpha(C)$ the α -sum of C .

It can then be seen that, if C_1 and C_2 are two cycles in G , and if $C = psd(C_1, C_2)$ is also a cycle, not all edges of C necessarily have the same orientation (induced by α_φ) as in C_1 or C_2 . Such a case is illustrated in Figure 2, where, e.g., the edge connecting vertices 2 and 5 in C_1 (left) is oriented from 2 to 5 in the α -sum of C_1 , while in $C = spd(C_1, C_2)$ (right), the same edge is

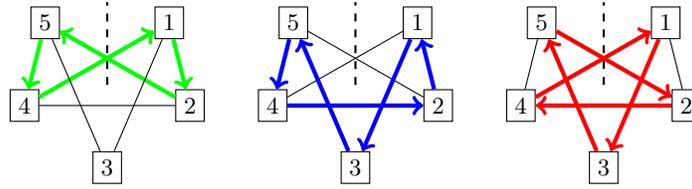


Fig. 2. A graph G with $n = 5$ vertices, such that $B_c(G) = 2$, in which the labels of an optimal cb -labeling φ are also the vertices identifiers. Edges in C_1 (left), C_2 (middle), and $C = spd(C_1, C_2)$ (right) are arbitrarily oriented to compute their respective α -sum. It can be seen that, in any case, $s_\alpha(C_1) = s_\alpha(C_2) = 0$, while $s_\alpha(C) = 2$.

oriented from 5 to 2 in the α -sum of C . As a consequence, in this case, we have $s_\alpha(C_1) = s_\alpha(C_2) = 0$ and $s_\alpha(C) = 2 \neq 0$. The example shown in Figure 2 shows that s_α is not necessarily conserved by symmetric difference. However, in the following lemma, which is a stronger version of Lemma 3.2 from [4], we are able to show that all cycles have a 0 α -sum under certain conditions.

Lemma 7 *Let G be a graph, and φ an optimal cb -labeling of G . If the set of cycles of G of length at most $\ell < \frac{n}{B_c(G)}$ contains a cycle basis, then for any cycle C of G , $s_\alpha(C) = 0$.*

Proof. Let φ be an optimal cb -labeling of G . Assume, by contradiction, that there is a cycle C of G such that $s_\alpha(C) \neq 0$. Remove from G all edges that have a non-zero α_φ , and let S_1, S_2, \dots, S_p denote the connected components of the resulting graph. For each connected component S_i , $1 \leq i \leq p$ and any subgraph $G' = (V', E')$ of G , we define the following function $\gamma_i(G')$: $\gamma_i(G') = \sum_{e \in E(G')} \gamma_i(e)$, where for any edge $uv \in E(G')$, $\gamma_i(uv) = 1$ if $\alpha(u, v) \neq 0$ and at least one vertex (say x) among u, v satisfies (i) $\varphi(x) > \frac{n}{2}$ and (ii) $x \in V(S_i)$; and $\gamma_i(uv) = 0$ otherwise.

One of the interests of γ_i is that, unlike α_φ , this function is symmetric: any edge uv will contribute the same way to γ_i , whether it is considered from u to v or from v to u . We will now prove Lemma 7 through a series of three claims.

Claim 1. Let $1 \leq i \leq p$, and let G_1 and G_2 be two subgraphs of G such that $\gamma_i(G_1)$ and $\gamma_i(G_2)$ are even. Then, $G' = psd(G_1, G_2)$ is such that $\gamma_i(G')$ is even.

Proof (of Claim 1). Take any $1 \leq i \leq p$, and let G_1 and G_2 be two subgraphs of G such that both $\gamma_i(G_1)$ and $\gamma_i(G_2)$ are even. Then $\gamma_i(psd(G_1, G_2)) = \gamma_i(G_1) + \gamma_i(G_2) - 2 \cdot \sum_{e \in E(G_1) \cap E(G_2)} \gamma_i(e)$, since all edges e present in both G_1 and G_2 are removed from $psd(G_1, G_2)$. Since all numbers in the above equation are even, $\gamma_i(psd(G_1, G_2))$ is also even. \square

Claim 2. Any cycle C' of G whose length ℓ satisfies $\ell < \frac{n}{B_c(G)}$ is such that $\gamma_i(C')$ is even for any $1 \leq i \leq p$.

Proof (of Claim 2). Observe a cycle $C' = (x_0, x_1, \dots, x_\ell)$ (with $x_0 = x_\ell$) of G , of length $\ell < \frac{n}{B_c(G)}$, and consider the set \mathcal{L} of labels that are used to label $V(C')$.

We claim that \mathcal{L} belongs to a (cyclic) interval I of cyclic length strictly less than $\frac{n}{2}$. In order to show this, let u (resp. v) be the vertex of C' whose label is minimum (resp. maximum), and let $\varphi_{\min} = \varphi(u)$ (resp. $\varphi_{\max} = \varphi(v)$). Now let us consider the set \mathcal{L} of labels that are used to label $V(C')$. There are three possible cases for \mathcal{L} : first, suppose $\mathcal{L} \subseteq [\varphi_{\min}; \varphi_{\max}]$. Then, $I = [\varphi_{\min}; \varphi_{\max}]$, and I is of length strictly less than $\frac{n}{2}$ by definition of the cyclic bandwidth and since $B_c(G) < \frac{n}{\ell}$. Second, suppose $\mathcal{L} \subseteq [1; \varphi_{\min}] \cup [\varphi_{\max}; n]$. For the same reasons as in the previous case, we conclude that the interval $I = [1; \varphi_{\min}] \cup [\varphi_{\max}; n]$ is of cyclic length strictly less than $\frac{n}{2}$. Third, suppose we are not in one of the two above cases, and let us show that this case cannot happen: indeed, this means that at least one vertex $x \neq u, v$ (resp. $y \neq u, v$) of $V(C')$ satisfies $\varphi(x) \in [1; \varphi_{\min}[\cup]\varphi_{\max}; n]$ (resp. $\varphi(y) \in]\varphi_{\min}; \varphi_{\max}[$). We will say that x (resp. y) has a *small* (resp. *high*) label. Now let P_x denote the path from u to v , that contains x , and that follows the edges of C' . Let S_x be the sum of $|\varphi(a) - \varphi(b)|_n$ over all edges ab in P_x . Then, we know that $S_x \geq \varphi_{\max} - \varphi_{\min}$. Now consider the path P_y going from u to v that contains y , and that follows the edges of C' . Let S_y be the sum of $|\varphi(c) - \varphi(d)|_n$ over all edges cd in P_y . Then we know that $S_y \geq n - \varphi_{\max} + \varphi_{\min}$. Hence, $S_x + S_y \geq n$. We will now show that, since we supposed $\ell < \frac{n}{B_c(G)}$, this cannot happen. Indeed, we have $S_x + S_y = \sum_{i=0}^{\ell-1} |\varphi(x_{i+1}) - \varphi(x_i)|_n \leq \ell \cdot B_c(G)$, because for any two neighbors x_{i+1} and x_i in C' we have $|\varphi(x_{i+1}) - \varphi(x_i)|_n \leq B_c(G)$, by definition of $B_c(G)$. Moreover, we know by hypothesis that $\ell \cdot B_c(G) < n$, thus we conclude that $S_x + S_y < n$, a contradiction.

We just proved that \mathcal{L} belongs to a (cyclic) interval I of cyclic length strictly less than $\frac{n}{2}$. Let us now observe interval I . There are two cases: first, if either 1 or n is not in I , then all edges uv of C' satisfy $\alpha_\varphi(u, v) = 0$, by definition of α_φ and because $B_c(G) \leq \frac{n}{2}$ for any graph G . Thus, every edge of C' contributes to zero in $\gamma_i(C')$, and consequently $\gamma_i(C') = 0$ for any $1 \leq i \leq p$.

Second, if both 1 and n are present in I , take any integer $1 \leq i \leq p$ and partition the vertices of C' in two sets: the first set V_1 contains every vertex $u \in V(C')$ such that $\varphi(u) > \frac{n}{2}$ and $u \in V(S_i)$; the second set is $V_2 = V(C') - V_1$.

Let uv be an edge of C' . We will say that uv is a *crossing* edge if $u \in V_1$ and $v \in V_2$, or vice-versa – this term refers to the fact that, as we will prove it, a crossing edge uv necessarily satisfies $\alpha(u, v) \neq 0$, and thus, crosses the dashed vertical as in Figure 1. Take any edge uv in C' . By definition, $\gamma_i(uv) \in \{0, 1\}$. Our goal here is to show that $\gamma_i(uv) = 1$ iff uv is a crossing edge.

(\Leftarrow) Suppose uv is a crossing edge, and suppose wlog that $u \in V_1$ and $v \in V_2$. In that case, $\varphi(v) < \frac{n}{2}$. Otherwise, we would have $\alpha_\varphi(u, v) = 0$ (by definition of α_φ , because the labels of u and v are separated by less than $B_c(G)$), which implies that both vertices u and v belong to S_i , since $u \in V(S_i)$. But in that case v would belong to V_1 , since $v \in V(S_i)$ and $\varphi(v) > \frac{n}{2}$. This is a contradiction since v cannot simultaneously belong to V_1 and V_2 . Thus, every crossing edge uv with $u \in V_1$ and $v \in V_2$ is such that $\varphi(v) < \frac{n}{2}$, and consequently $\alpha_\varphi(u, v) \neq 0$. This implies that for any such edge uv , $\gamma_i(uv) = 1$.

(\Rightarrow) Now suppose $\gamma_i(uv) = 1$, which implies $\alpha_\varphi(u, v) \neq 0$. Thus, there are two cases: $\alpha_\varphi(u, v) > 0$ or $\alpha_\varphi(u, v) < 0$. First, if $\alpha_\varphi(u, v) > 0$, and since $\gamma_i(uv) = 1$, then $u \in V(S_i)$ by definition. Moreover, $\alpha_\varphi(u, v) > 0$ implies (by definition) $\varphi(u) \geq \varphi(v) + n - B_c(G, \varphi)$. Since $B_c(G, \varphi) \leq \frac{n}{2}$ for any graph G and any labeling φ , and since $\varphi(v) \geq 1$, we conclude that $\varphi(u) > \frac{n}{2}$. We then know that $u \in V_1$ by definition. Moreover, $v \notin V_1$, otherwise $uv \in E(S_i)$, a contradiction to the fact that all edges in $E(S_i)$ have a zero α_φ . Thus, we conclude that $v \in V_2$, and that uv is a crossing edge. Now if $\alpha_\varphi(u, v) < 0$, by a symmetrical argument as above, we conclude that $v \in V_1$ and $u \in V_2$, i.e., uv is a crossing edge as well.

We thus know that each edge $uv \in E(C')$ contributes to 1 to γ_i iff uv is a crossing edge. This allows us to prove the claim: if we follow the edges of C' to compute $\gamma_i(C')$, the number of crossing edges encountered is necessarily even, since C' starts and ends in the same vertex. Since only a crossing edge increases γ_i (by exactly 1), we conclude that $\gamma_i(C')$ is even. \square

Claim 3. There exists an integer $1 \leq j \leq p$ and a cycle C' of G such that $\gamma_j(C') = 1$.

Proof (of Claim 3). Recall that we suppose there exists a cycle C in G such that $s_\alpha(C) \neq 0$. For each $1 \leq i \leq p$, let Σ_i be the sum of the $\alpha_\varphi(u, v)$ for each edge uv of C that has an extremity in S_i , counting uv twice in Σ_i if both $u, v \in V(S_i)$. The sum of the Σ_i s over every $1 \leq i \leq p$ is thus equal to $2s_\alpha(C)$.

Since we supposed $s_\alpha(C) \neq 0$, we know there exists at least one j , $1 \leq j \leq p$, such that $\Sigma_j \neq 0$. Thus, there exists at least one edge $uv \in E(C)$ such that (i) at least one of $u, v \in V(S_j)$, and (ii) $\alpha_\varphi(u, v) \neq 0$.

Now let us look at every edge along C (following the natural order of the cycle), and let us observe each pair (uv, wx) of edges (with possibly $uv = wx$) satisfying the following: (i) $\alpha_\varphi(u, v) \neq 0$ and $u \in V(S_j)$ and (ii) $\alpha_\varphi(w, x) \neq 0$ and $x \in V(S_j)$. Because $\Sigma_j \neq 0$, such a pair (again with possibly $uv = wx$) must exist. Moreover, among such pairs, and since $\Sigma_j \neq 0$, there necessarily exists a pair, say (ab, cd) , such that $\alpha_\varphi(a, b) = \alpha_\varphi(c, d)$.

There are now two cases to consider. First, if $ab = cd$, then we construct the cycle C_1 containing edge ab and edges of a path $P_{a,b}$ joining a to b in S_j (we know there exists such a path, since $a, b \in V(S_j)$, and S_j is connected by definition). By definition also, $P_{a,b}$ only contains zero α_φ edges. Thus $\gamma_j(C_1) = 1$, since only edge ab contributes to 1 to $\gamma_j(C_1)$ – all other edges of C_1 contribute to 0.

Second, if $ab \neq cd$, take the path $P = ab \dots cd$ in C that connects a to d through ab and cd . Path P has no internal vertex in S_j by construction. Consider a second path $P' = a \dots d$ connecting a to d within S_j (again, P' exists since $a, d \in V(S_j)$ by definition, and since S_j is connected). Note that P and P' only intersect in a and d . Then, we can construct cycle C_2 , which is the concatenation of P and P' . What is important to observe is that only one of the two edges ab or cd will contribute to 1 in the computation of $\gamma_j(C_2)$ – the other will contribute to 0. Moreover, all other edges from P contribute to 0 to $\gamma_j(C_2)$ (because none contains a vertex of S_j), and the same conclusion is reached with any edge of P' : by definition, each such edge belongs to S_j , and thus is a zero α_φ edge. \square

Algorithm 1: Transformation of a *cb*-labeling φ into a *bm*-labeling solution φ' such that $B(G, \varphi') \leq B_c(G, \varphi)$, under the conditions of Lemma 8

Data: A graph $G = (V, E)$ and a *cb*-labeling φ of G
Result: A *bm*-labeling φ' of G

```

1 for each connected component  $S$  of  $G$  do
2   | Choose a vertex  $s$  in  $S$  and set  $\varphi^*(s) = \varphi(s)$  and  $\beta(s) = 0$ 
3   | Apply a Depth-First Search from  $s$ 
4   | for each newly visited vertex  $y$  discovered by a vertex  $x$  do
5   |   |  $\beta(y) = \beta(x) + \alpha_\varphi(x, y)$ 
6   |   |  $\varphi^*(y) = \varphi(y) + n \cdot \beta(y)$ 
7   |   end
8 end
9 for  $i$  from 1 to  $n$  do
10  | Set  $\varphi'(i)$  to be the rank of  $\varphi^*(i)$ 
11 end
12 Return  $\varphi'$ 

```

We are now ready to prove Lemma 7. Suppose the set of cycles of G of length at most $\ell < \frac{n}{B_c(G)}$ contains a cycle basis \mathcal{C} . Observe an optimal *cb*-labeling φ of G , and suppose by contradiction that there exists a cycle C in G such that $\alpha_\varphi(G) \neq 0$. By Claim 3, we know that there exists a cycle C' and an integer j such that $\gamma_j(C') = 1$. Besides, we know by definition that C' can be obtained by symmetric difference of elements of \mathcal{C} .

However, Claims 2 and 1 respectively yield that any cycle in \mathcal{C} has even γ_i and thus that any symmetric difference of elements of \mathcal{C} also has even γ_i . We thus get the desired contradiction, which in turn proves Lemma 7. \square

Note that Lemma 7 remains correct if φ is not an optimal labeling (but in that case, $B_c(G)$ should be replaced by $B_c(G, \varphi)$ in the lemma's statement).

The next lemma provides a sufficient condition yielding $B(G) = B_c(G)$, improved from [4]. It is the last step towards proving Theorem 9, and needs Lemma 7 to be proved.

Lemma 8 *If the set of cycles of G of length at most $\ell < \frac{n}{B_c(G)}$ contains a cycle basis, then $B_c(G) = B(G)$.*

Proof. Suppose G satisfies the conditions of the lemma, and let φ be a *cb*-labeling of G . We will show in the following that Algorithm 1 below computes a *bm*-labeling φ' of G satisfying $B(G, \varphi') \leq B_c(G, \varphi)$. If φ is optimal, then $B_c(G, \varphi) = B_c(G)$, and consequently $B(G, \varphi') = B_c(G) = B(G)$ (i.e., φ' is an optimal *bm*-labeling) since $B(G) \geq B_c(G)$ for any graph G .

We need Claims 4 and 5 to show that $B(G, \varphi') \leq B_c(G, \varphi)$.

Claim 4. For any adjacent vertices u and v , $\beta(v) = \beta(u) + \alpha_\varphi(u, v)$.

Proof (of Claim 4). First suppose v has been discovered by u during the Depth-First Search (DFS) in Algorithm 1: then Line 6 of Algorithm 1 yields $\beta(v) = \beta(u) + \alpha_\varphi(u, v)$. Symmetrically, the result is proved if u has been discovered by v during the DFS, since $\alpha_\varphi(u, v) = -\alpha_\varphi(v, u)$ for any labeling φ and any edge $uv \in E(G)$. Now suppose none of u and v have discovered the other during the DFS. Since u and v are in the same connected component, by property of DFS, there exists a vertex s such that $\beta(s) = 0$ and two paths $P_u = u_0u_1u_2\dots u_p$ (with $u_0 = s$ and $u_p = u$) and $P_v = v_0v_1v_2\dots v_q$ (with $v_0 = s$ and $v_q = v$), such that $\beta(u_{i+1}) = \beta(u_i) + \alpha_\varphi(u_i, u_{i+1})$ for any $0 \leq i \leq p-1$ and $\beta(v_{i+1}) = \beta(v_i) + \alpha_\varphi(v_i, v_{i+1})$ for any $0 \leq i \leq q-1$. Therefore, we have $\beta(u) = \sum_{i=0}^{p-1} \alpha_\varphi(u_i, u_{i+1})$ and $\beta(v) = \sum_{i=0}^{q-1} \alpha_\varphi(v_i, v_{i+1})$.

Suppose first that P_u and P_v only share vertex s . In that case, we obtain a cycle C_{uv} by concatenating paths P_u, P_v and edge uv . Since we are in the conditions of Lemma 7, we conclude that $s_\alpha(C_{uv}) = 0$. Observe that $s_\alpha(C_{uv}) = S_{P_u} + \alpha_\varphi(u, v) + S_{P_v}$, where $S_{P_u} = \sum_{i=0}^{p-1} \alpha_\varphi(u_i, u_{i+1}) = \beta(u)$ and $S_{P_v} = \sum_{i=0}^{q-1} \alpha_\varphi(v_{i+1}, v_i) = -\sum_{i=0}^{q-1} \alpha_\varphi(v_i, v_{i+1}) = -\beta(v)$. Hence we conclude that $\beta(u) + \alpha_\varphi(u, v) - \beta(v) = 0$, which is the desired result.

If P_u and P_v share more than vertex s , consider the common vertex between both paths having the largest index $r > 0$ in $P_u = u_0u_1\dots u_p$, and let t be such that $u_r = v_t$. Thus, we can write $P_u = P_r \cdot P$ (where $P_r = u_0u_1\dots u_r$) and $P_v = P_t \cdot P'$ (where $P_t = v_0v_1\dots v_t$), and in that case $S_{P_u} = S_{P_r} + S$ and $S_{P_v} = S_{P_t} + S'$, where $S_{P_r} = \sum_{i=0}^{r-1} \alpha_\varphi(u_i, u_{i+1})$, $S = \sum_{i=r}^{p-1} \alpha_\varphi(u_i, u_{i+1})$, $S_{P_t} = \sum_{i=0}^{t-1} \alpha_\varphi(v_i, v_{i+1})$ and $S' = \sum_{i=t}^{q-1} \alpha_\varphi(v_i, v_{i+1})$. Again, if we concatenate P, P' and uv , we get a cycle C_{rt} and by Lemma 7 we have $s_\alpha(C_{rt}) = 0$. Thus, it remains to show that $S_{P_r} - S_{P_t} = 0$ to conclude. This is done by induction on r . When $r = 0$, we are in the case where P_u and P_v only share vertex s , which has already been discussed. When $r = 1$, $P_r = u_0u_1 = su_1$ is limited to one edge. If $t = 1$, then $v_1 = u_1$ and $P_t = P_r = su_1$ and by antisymmetry of α_φ we have $S_{P_r} - S_{P_t} = \alpha_\varphi(s, u_1) + \alpha_\varphi(u_1, s) = 0$. If $t > 1$, then concatenating edge P_r and path P_t yields a cycle C' , for which we know by Lemma 7 again that $s_\alpha(C') = S_{P_r} - S_{P_t} = 0$. Now if $r > 1$, apply induction hypothesis to r' , where r' is the largest index (with $r' < r$) of a vertex of P_u also belonging to P_v . Let t' be the corresponding index in P_v . Then, $P_u = P_{r'} \cdot P_{r'r} \cdot P$ and $P_v = P_{t'} \cdot P_{t't} \cdot P'$, where $P_{r'r}$ (resp. $P_{t't}$) is a path from r' to r (resp. from t' to t). By induction hypothesis, $S_{P_{r'}} - S_{P_{t'}} = 0$. Moreover, concatenating $P_{r'r}$ and $P_{t't}$ yields a cycle C'' , which satisfies, by Lemma 7, $s_\alpha(C'') = 0$. Since $s_\alpha(C'') = S_{P_{r'r}} - S_{P_{t't}}$, we obtain altogether that $(S_{P_{r'}} + S_{P_{r'r}}) - (S_{P_{t'}} + S_{P_{t't}}) = 0$. But $S_{P_{r'}} + S_{P_{r'r}} = S_{P_r}$ and $S_{P_{t'}} + S_{P_{t't}} = S_{P_t}$, hence we conclude $S_{P_r} - S_{P_t} = 0$, and the proof by induction holds. Altogether, we have $S_{P_r} - S_{P_t} = 0$ in all cases. This allows us to conclude that $\beta(v) = \beta(u) + \alpha_\varphi(u, v)$. \square

Claim 5. For any two adjacent vertices x and y in G , $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x)|_n$.

Proof (of Claim 5). If $\alpha_\varphi(x, y) = 0$, then $\beta(y) = \beta(x)$ by Claim 4. Thus, we have $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x)|$. However, by definition of α_φ , we know that

$|\varphi(y) - \varphi(x)| \leq B_c(G, \varphi)$ and thus $|\varphi(y) - \varphi(x)|_n = |\varphi(y) - \varphi(x)| = |\varphi^*(y) - \varphi^*(x)|$. Suppose now $\alpha_\varphi(x, y) = 1$. Then, by Claim 4 again, $\beta(y) = \beta(x) + 1$, and consequently $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x) + n| = n - (\varphi(x) - \varphi(y))$ (because $\alpha_\varphi(x, y) = 1$ implies $\varphi(x) > \varphi(y)$). We then conclude that $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x)|_n$. Finally, if $\alpha_\varphi(x, y) = -1$, by a symmetrical argument as above, we get that $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x) - n| = n - (\varphi(y) - \varphi(x))$ (because $\alpha_\varphi(x, y) = -1$ implies $\varphi(x) < \varphi(y)$). Finally, $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x)|_n$. \square

We are now ready to finish proof of Lemma 8. First, note that φ^* is injective, since for any vertex v , $\varphi^*(v)$ is congruent to $\varphi(v)$ modulo n . Moreover, since φ^* is injective, φ' is also injective. As $|V| = n$, φ' is a bijection from V to $[n]$. Moreover, $|\varphi'(y) - \varphi'(x)|$ is equal to the number of vertices v for which $\varphi^*(v)$ lies between $\varphi^*(y)$ (included) and $\varphi^*(x)$ (excluded). Since φ^* is injective, this number is less than or equal to $|\varphi^*(y) - \varphi^*(x)|$, so we have $|\varphi'(y) - \varphi'(x)| \leq |\varphi^*(y) - \varphi^*(x)|$. Since we know by Claim 5 that $|\varphi^*(y) - \varphi^*(x)| = |\varphi(y) - \varphi(x)|_n$, we conclude that φ' is a bm -labeling satisfying $B(G, \varphi') \leq B_c(G, \varphi)$. In particular, if φ is an optimal cb -labeling, then $B_c(G, \varphi) = B_c(G)$ and thus $B(G, \varphi') \leq B_c(G)$. But since for any graph G , $B(G) \geq B_c(G)$, we conclude that $B(G, \varphi') = B_c(G)$. Consequently φ' is an optimal bm -labeling, and $B(G) = B_c(G)$. \square

Our main result of the section is now stated.

Theorem 9. *Let G be a graph and let ℓ be the length of the longest cycle of a cycle basis of G . Then, we have $B_c(G) \geq \min\{B(G), \lceil \frac{n}{\ell} \rceil\}$. In particular, if $B(G) \leq \lceil \frac{n}{\ell} \rceil$, then $B(G) = B_c(G)$.*

Proof. Suppose first $B_c(G) \geq \lceil \frac{n}{\ell} \rceil$. Since $B(G) \geq B_c(G)$ for any graph, we have $B(G) \geq \lceil \frac{n}{\ell} \rceil$, which proves the result. Now suppose $B_c(G) < \lceil \frac{n}{\ell} \rceil$. In that case we have $B_c(G) < \frac{n}{\ell}$, since $B_c(G)$ is an integer, and by Lemma 8 we conclude that $B(G) = B_c(G)$.

4 Applying our results on 28 Harwell-Boeing instances

We now illustrate how the results we obtained in Sections 2 and 3 improve previous knowledge, by applying them on instances taken from the Harwell-Boeing sparse matrix collection. This benchmark contains a large number of matrices (that we interpret as graphs here) derived from several industrial applications. This collection has often been used as a benchmark for both the BANDWIDTH MINIMIZATION and CYCLIC BANDWIDTH problems, see e.g. [14, 12, 13, 11, 10]. Before going further, we note that all lower bound results provided in previous sections can be computed in polynomial time. Hence, computational performances are not provided here, as they do not play a major role.

We took a set of 28 Harwell-Boeing instances, that have recently been specifically used for the CYCLIC BANDWIDTH problem [14, 12, 13]. Our results are summarized in Table 1. The three leftmost columns describe the instance: respectively its usual name, its number n of vertices, and m of edges. The next

Graph G	n	m	Best Known Upper Bound	Lower Bound Density	ℓ	$B(G) =$ $B_c(G) ?$	Lower Bound Cycle Basis	Lower Bound from our results
jgl009	9	50	4*	4	3	no	3	4*
rgg010	10	76	5*	5	3	no	4	5*
jgl011	11	76	5*	5	3	no	4	5*
can_24	24	92	5*	5	3	yes	5	5*
pores_1	30	103	7*	6	4	yes	7	7*
ibm32	32	90	9*	8	5	no	7	8
bcsprw01	39	46	4*	4	13	no	3	4*
bcsstk01	48	176	12	9	4	no	12	12*
bcsprw02	49	59	7	6	8	yes	7	7*
curtis54	54	124	8*	8	9	no	6	8*
will57	57	127	6	6	4	yes	6	6*
impcol_b	59	281	17	14	6	no	10	14
ash85	85	219	9	8	3	yes	9	9*
nos4	100	247	10	9	4	yes	10	10*
dwt_234	117	162	11	10	8	yes	11	11*
bcsprw03	118	179	9*	9	10	yes	9	9*
bcsstk06	420	3720	45	31	4	yes	37	37
bcsstk07	420	3720	45	31	4	yes	37	37
impcol_d	425	1267	35	24	21	no	21	24
can_445	445	1682	46	36	10	?	45	45
494_bus	494	586	28	24	19	?	25	25
dwt_503	503	2762	41	29	8	yes	29	29
sherman4	546	1341	27	21	4	yes	21	21
dwt_592	592	2256	29	22	3	yes	22	22
662_bus	662	906	38	36	23	?	29	36
nos6	675	1290	16	15	4	yes	15	15
685_bus	685	1282	32	26	22	yes	30	30
can_715	715	2975	60	52	16	?	45	52

Table 1. Summary of results applied to a set of 28 instances from the Harwell-Boeing repository, sorted by increasing $n = |V(G)|$. The rightmost column presents the best lower bound obtained by our results (i.e., the maximum value obtained by Theorems 1, 3, and 9). In the leftmost and rightmost column, asterisks indicate optimality, while bold values in the rightmost column indicate a strict improvement compared to previous knowledge. Note that the “Best Known Upper Bound” for **bcsprw03** we report here is the only result not imported from [13] (where the reported bound is 10). Indeed, we know by [11] that $B(G) \leq 9$ for that instance; since $B_c(G) \leq B(G)$ for any graph G , we get a(n optimal) upper bound of 9 for **bcsprw03**.

column (“Best Know Upper Bounds”) provides the best upper bounds results concerning B_c , which are taken from the recent paper by Ren et al. [13]. Optimal values for $B_c(G)$ are marked with an asterisk.

The five remaining columns are computations and informations derived from the present work: in a nutshell, column “Lower Bound Density” contains the best result obtained by Theorems 1 and 3 (each theorem being applied for all possible values of i and any vertex v in the case of Theorem 1, and for all possible values of i and any edge uv in the case of Theorem 3), while column “Lower Bound Cycle Basis” contains the best results obtained by Theorem 9. Note that for computing the values in this column, we need to determine the length ℓ of the longest

cycle in a cycle basis for G (column “ ℓ ”); then, based on ℓ and on known upper bounds on $B(G)$ [11], we can (most of the time, see column “ $B(G) = B_c(G)$?”) determine whether $B(G) = B_c(G)$. If $B(G) = B_c(G)$, then we can invoke the best known lower bounds from $B(G)$ taken from [10]. If not, we apply Theorem 9 and take the minimum between $\frac{n}{\ell}$ and (the best lower bound on) $B(G)$. Finally, the rightmost column, “Lower Bound from our results”, simply takes the maximum value between the one in column “Lower Bound Density” (Theorems 1 and 3) and the one in column “Lower Bound Cycle Basis” (Theorem 9).

In the rightmost column of Table 1, which summarizes our results, values in bold show a strict improvement from previous lower bound results (based on the lower bounds for $B_c(G)$ taken from [13]), while an asterisk shows that the lower bound we obtain is actually optimal.

We first note that, among the 28 instances, our density lower bound is strictly better than our cycle basis lower bound 10 times, while the opposite happens 11 times (and thus it is a draw for the 7 remaining instances). This shows that both types of lower bounds are useful for improvement.

We also observe that Ren et al. [13] have determined the optimal value for $B_c(G)$ for 9 instances (the 7 first together with `curtis54` and `bscpwr03`), hence our results cannot strictly improve them. It should be noted however, that for 8 out of these 9 instances, our lower bounds results are actually optimal (as shown by the asterisks in the rightmost column). Our results also show that, among the 19 remaining instances, we have been able to determine the optimal values for $B_c(G)$ for 6 of them, since our new lower bounds appear to be matching the upper bounds for B_c provided by [13]. This clearly shows the interest of our work in aiming at increasing the lower bounds.

Concerning the 13 remaining instances, our results have drastically reduced the gap between lower and upper bounds for B_c , which now ranges from 1 (instances `can445` and `nos6`) to 12 (instance `dwt_503`), with an average gap of 5.54. Previously, based on the results presented in [13], the gap between lower and upper bounds ranged between 8 and 40, with an average of 24.31. Over the 28 studied instances, there are only two cases for which our results have neither reached the optimal value for B_c nor strictly improved previous lower bounds – namely instances `ibm32` and `can_715`.

5 Conclusion and Open Questions

In this paper, we have studied the CYCLIC BANDWIDTH problem, and have particularly focused on determining new lower bounds for $B_c(G)$ for any graph G . We have, on the way, provided a relabeling algorithm (Algorithm 1) of a graph G that, under specific conditions, computes a labeling φ' for the BANDWIDTH MINIMIZATION problem, starting from a labeling φ for CYCLIC BANDWIDTH such that $B(G, \varphi') \leq B_c(G, \varphi)$. We have applied our results to a benchmark of 28 Harwell-Boeing instances already used in [14, 12, 13], and showed how our results greatly improve the best current knowledge.

The results we provide here may open the door to further improvements. In particular, it would be worth investigating whether they could help determining $B_c(G)$ for new families of graphs. They could also, in a more general setting, help better characterize properties of graphs satisfying $B_c(G) = B(G)$.

Concerning the benchmark that we used, note that among the 28 Harwell-Boeing instances from Table 1, we have not been able to determine whether $B_c(G) = B(G)$ for 4 instances, and investigating that question could be of interest. Finally, we observe that there exists graphs for which Algorithm 1, starting from a suboptimal labeling φ for CYCLIC BANDWIDTH, is able to compute an optimal labeling φ' for both BANDWIDTH MINIMIZATION and CYCLIC BANDWIDTH. Hence, it would be interesting to apply Algorithm 1 on the instances from Table 1 for which conditions of Lemma 8 are fulfilled, in order to determine whether this technique helps improving the knowledge on B and/or B_c .

References

1. Chinn, P.Z., Chvátalová, J., Dewdney, A.K., Gibbs, N.E.: The bandwidth problem for graphs and matrices - a survey. *Journal of Graph Theory* **6**(3), 223–254 (1982)
2. Chung, F.R.: Labelings of graphs. *Selected topics in graph theory* **3**, 151–168 (1988)
3. Harper, L.H.: Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics* **12**(1), 131–135 (1964)
4. Hromkovič, J., Müller, V., Sýkora, O., Vrt' o, I.: On embedding interconnection networks into rings of processors. In: *PARLE'92. Lecture Notes in Computer Science*, vol. 605, pp. 53–62. Springer (1992)
5. Lam, P.C.B., Shiu, W.C., Chan, W.H.: Characterization of graphs with equal bandwidth and cyclic bandwidth. *Discrete Mathematics* **242**(1-3), 283–289 (2002)
6. Leung, J.Y., Vornberger, O., Witthoff, J.D.: On some variants of the bandwidth minimization problem. *SIAM Journal on Computing* **13**(3), 650–667 (1984)
7. Lin, Y.: A level structure approach on the bandwidth problem for special graphs. *Annals of the New York Academy of Sciences* **576**(1), 344–357 (1989)
8. Lin, Y.: The cyclic bandwidth problem. *Systems Science and Mathematical Sciences* **7** (01 1994)
9. Lin, Y.: Minimum bandwidth problem for embedding graphs in cycles. *Networks* **29**(3), 135–140 (1997)
10. Martí, R., Campos, V., Piñana, E.: A branch and bound algorithm for the matrix bandwidth minimization. *European Journal of Operational Research* **186**(2), 513–528 (2008)
11. Pop, P., Matei, O., Comes, C.A.: Reducing the bandwidth of a sparse matrix with a genetic algorithm. *Optimization* **63**(12), 1851–1876 (2014)
12. Ren, J., Hao, J., Rodriguez-Tello, E.: An iterated three-phase search approach for solving the cyclic bandwidth problem. *IEEE Access* **7**, 98436–98452 (2019)
13. Ren, J., Hao, J.K., Rodriguez-Tello, E., Li, L., He, K.: A new iterated local search algorithm for the cyclic bandwidth problem. *Knowledge-Based Systems* **203**, 106136 (2020)
14. Rodriguez-Tello, E., Romero-Monsivais, H., Ramírez-Torres, G., Lardeux, F.: Tabu search for the cyclic bandwidth problem. *Computers & Operations Research* **57**, 17–32 (2015)
15. Zhou, S.: Bounding the bandwidths for graphs. *Theoretical Computer Science* **249**(2), 357–368 (2000)