mathematics

MDPI

Article

On the *Embed and Project* Algorithm for the Graph Bandwidth Problem

Janez Povh [1,2]

[1] Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva ulica 6, SI-1000 Ljubljana, Slovenia; janez.povh@fs.uni-lj.si

[2] Institute of Mathematics, Physics and Mechanics, Jadranska 19, SI-1000 Ljubljana, Slovenia

Abstract: The graph bandwidth problem, where one looks for a labeling of graph vertices that gives the minimum difference between the labels over all edges, is a classical NP-hard problem that has drawn a lot of attention in recent decades. In this paper, we focus on the so-called *Embed and Project Algorithm* (EPA) introduced by Blum et al. in 2000,which in the main part has to solve a semidefinite programming relaxation with exponentially many linear constraints. We present several theoretical properties of this special semidefinite programming problem (SDP) and a cutting-plane-like algorithm to solve it, which works very efficiently in combination with interior-point methods or with the bundle method. Extensive numerical results demonstrate that this algorithm, which has only been studied theoretically so far, in practice gives very good labeling for graphs with $n \leq 1000$.

Keywords: graph bandwidth problem; semidefinite programming; combinatorial optimization; embed and project algorithm; approximation algorithm

Citation: Povh, J. On the *Embed and Project* Algorithm for the Graph Bandwidth Problem. *Mathematics* **2021**, *9*, 2030. https://doi.org/10.3390/math9172030

Academic Editor: Takayuki Hibi

Received: 30 July 2021
Accepted: 20 August 2021
Published: 24 August 2021

1. Introduction

1.1. The Graph Bandwidth Problem

Motivation for the graph bandwidth problem dates back to the 1950s, when industrial mathematicians were challenged to perform the Gaussian elimination faster in order to solve large sparse systems of linear equations $Ax = b$, which are inevitable parts of almost all numerical methods for solving, e.g., partial differential equations or instances of linear or nonlinear programming. A natural idea was to permute rows and columns of A such that all non-zero entries of the permuted matrix lie within a very narrow band along the main diagonal. This application also gave the name to the problem: the matrix bandwidth problem (see, e.g., [3] and the references therein).

The graph bandwidth problem was introduced a decade later in [4]. It is actually a matrix bandwidth problem, applied to the adjacency matrix of a graph. More precisely, suppose we consider a simple (undirected, without loops and multi-edges) and connected graph $G = (V, E)$, where the vertex set is simply $V = \{1, 2, \dots, n\}$. The graph *bandwidth problem* (shortly, GBP) is the problem of finding the permutation of graph vertices such that the maximum difference of end point numbers, taken over all edges, is minimum:

$$OPT_{GBP} := \min \{\max_{ij \in E} |\varphi(i) - \varphi(j) \mid \varphi \text{ a permutation of } V\}. \quad (1)$$

The GBP is one of the hardest optimization problems on graphs. Papadimitriou proved in 1976 [5] that the (1) equation is NP-hard, and it remains NP-hard if graph G is very simple, such as a tree with a maximum degree of at most 3 [6] or a caterpillar with a hair length of at most 3 [7]. The problem of finding good approximate solutions for GBP is also very hard. Blache et al. [8] proved that if $P \neq NP$, then there is no polynomial time algorithm, which can approximate OPT_{GBP} with an approximation ratio smaller than 1.5.

1.2. Our Contribution

In this paper, we consider the so-called *Embed and Project Algorithm* (EPA), which follows the idea of distance and volume respecting embeddings. Feige [9] has introduced the notion of volume respecting embeddings and presented a polynomial randomized algorithm, which, for general graphs, gives with high probability labeling φ with bandwidth $\mathcal{O}(D\log^{3.5} n\sqrt{\log\log n})$, where D is the local density of a graph, defined in Equation (2). This is actually a very good result for general graphs according to the known $\Omega(\log n)$ gap on Cantor combs [10] .

Blum et al. [1] presented the algorithm that we will study in this paper. We call it the *Embed and Project Algorithm* (EPA). It uses semidefinite programming to find an embedding of a graph's vertices into a unit sphere in $\mathbb{R}^n$, which keeps distances between vertices, and after projecting to a random line, gives a bandwidth within the ratio $\mathcal{O}(\sqrt{n}\log n/\sqrt[4]{OPT_{SDP}})$, where $\sqrt{OPT_{SDP}}$ is the maximal distance between the embeddings of the adjacent vertices. Dunagan and Vempala [2] refined the algorithm from [1]. By using the embedding algorithm of Rao [11], they showed that the resulting algorithm gives labeling φ with bandwidth $\mathcal{O}(OPT_{GBP}\log^3 n\sqrt{\log\log n})$ with high probability.

The main step in both algorithms from the previous paragraph is solving a semidefinite programming problem (SDP), which exponentially has many linear constraints. Its polynomial time complexity has been proven by presenting a polynomial time separation oracle for the feasibility set, which enables the ellipsoidal method to solve the problem in polynomial time. This is of big theoretical importance, but due to the known practical inefficiency of the ellipsoidal method, it has very weak implementation relevance. This is probably the reason why (to the best of our knowledge) no implementation of this algorithm is known, besides our first approach in [12] .

The main contribution of this paper is showing that EPA, introduced mostly for theoretical reasons, has a strong practical impact. The computational bottleneck of this algorithm, i.e., solving the SDP with exponentially many constraints, can be resolved computationally and efficiently using a cutting-plane-like algorithm in combination with interior point methods [13,14] or with the bundle method [15,16], where in each step, we include only a small (linear in size) subset of the exponential set of constraints. Beside this, we

- provide several interesting theoretical properties about the optimum solution of the SDP problem, especially in relation to the optimum OPT_{GBP};
- demonstrate the performance of EPA with extensive numerical results for various test instances, which show that EPA in practice yields very good bandwidth approximations and could be a method of choice for this problem.

1.3. Assumptions and Notation

Throughout this paper, we consider only the graphs that are simple (undirected, without loops and multi-edges) and connected with a vertex set $V = \{1, \ldots, n\}$ and an edge set E. By I_n, we denote the $n \times n$ identity matrix. When the dimension of I_n is obvious, we omit n. If G is a sub-graph of graph H, we denote this by $G \subseteq H$.

2. Related Work

2.1. Approximation Results about the Bandwidth

A survey of algorithms up to 2015, applied to solve GBP, is available in [17]. For some families of graphs, we are able to compute the bandwidth exactly in a polynomial time. For caterpillars with a hair length of at most 2 [18], for interval graphs [19], for chain graphs [20] and for bipartite permutation graphs [21], there is a polynomial algorithm that computes the bandwidth exactly.

For other families of graphs, there are only a few approximation algorithms with good approximation guarantees. A polynomial time approximation algorithm for caterpillars that computes the bandwidth, which is at most $\mathcal{O}(\log n/(\log\log n))$ times the local density, was given by Feige and Talwar [22]. Kloks et al. presented a two-approximation algorithm for asteroidal triple-free graphs [23]; a $\log d$ approximating algorithm on general height-balanced trees with depth d was presented by Haralambides [24]; a three-approximation algorithm on dense graphs was presented in [25]; and Gupta [26] gave an $\mathcal{O}(\log^{2,5} n)$-approximation algorithm on trees. Führer et al. [27] presented a two-approximation algorithm that runs in $\mathcal{O}(1.9797^n)$ and needs a polynomial size memory.

Several authors approached the graph bandwidth problem (or matrix bandwidth problem) using hybrid methods [28,29]. The ant colony approach in combination with local search improvements is described in [28,30]. In [31], the authors studied the cyclic bandwidth sum problem and proposed a heuristic algorithm, which first finds a set of paths that follow the structure of the graph and then merge the obtained paths based on a greedy approach. The edge-bandwidth of graph G was studied in [32], where asymptotically tight bounds on the edge bandwidth of two-dimensional grids and tori were presented.

GBP can be considered as a permutation problem. In the last decade, meta-heuristics based on the permutation representations were used to approximately solve such problems. However, to the best of our knowledge, GBP has not been considered so far. The linear ordering problem with cumulative costs and the flow shop scheduling problem were approached by this type of meta-heuristics in [33,34], respectively. Random Key Estimation of Distribution Algorithms (RK-EDA) has been proposed in [35] and applied to flow shop scheduling, linear ordering, quadratic assignment and traveling salesman problems. Algebraic Particle Swarm Optimization (APSO) for permutation problems was introduced in [36] and successfully applied to a well-known list of benchmark instances for four permutation problems. In [37], the bandwidth coloring problem was considered and solved approximately by a tabu search and GRASP.

Quantum algorithms for GBP and some other NP-hard problems were studied in [38]. Theoretical results of speedups were presented, albeit without any numerical results.

Tight lower bounds for OPT_{GBP} are very important, especially for evaluating heuristic approaches for computing bandwidth. In [3], the reader can find many different inequalities upon which the latter work was based. Among the most famous lower bounds is the *local density* D of a graph, defined as

$$D = \max\left\{\frac{|V(H)|-1}{\mathrm{diam}(H)} : H \text{ connected sub-graph of } G\right\}. \tag{2}$$

This lower bound is tight if the graph is a caterpillar with a hair length of at most 2 [18]. The exact strength of this lower bound is still an open question, but we know that there exist graphs (the so-called Cantor combs) where the gap is $\Omega(\log n)$, see [10] for definition and details. In [39], it is shown that the problem of determining D is APX-complete.

A very productive line of research on the graph bandwidth problem followed the idea of estimating the graph bandwidth using the recent semidefinite programming-based results for graph partitioning problems and the quadratic assignment problem, sometimes also further enhanced with symmetry reduction methods [12,40–45]. However, their results are applicable to graphs of small or medium range (up to few hundreds), and only the lower bounds are given, not the labeling. The reason for that is hidden in the fact that they embed the problem in the cone of positive semidefinite matrices of the order nk, where $k > 1$ and can be in some cases even equal to n. Additionally, it is not clear how to reconstruct a good labeling from the optimum solution of the semidefinite embedding.

Jiang et al. [46] analyzed the bandwidth of Kneser graphs and provided new lower and upper bounds for this family of graphs in terms of the main graph parameters n and r. No numerical demonstration of this bound is available.

Several authors tried to compute the graph bandwidth using a traditional combinatorial optimization approach, such as integer programming modeling and branch and bound [47,48]. Their approach is capable of approximating, and in some cases, even solving to optimality, the graph bandwidth problem on instances with several hundreds of vertices.

2.2. *Closed form Expressions for OPT_{GBP} for Some Families of Graphs*

In this subsection, we report results of graph bandwidth for families of graphs for which there are known closed form expressions for OPT_{GBP}. The results of their bandwidth are reported as Theorem 1.

We define the *path* P_n as a graph with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $E = \{ij \colon |i - j| = 1\}$ (this is path P^{n-1} from [49] (p. 6)). Similarly, the *cycle* C_n is a graph with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $E = \{ij \colon |i - j| \equiv 1 \pmod n\}$ (Diestel [49] (p. 7) used notation C^n). The *grid graph* $P_{m,n}$ is a Cartesian product of paths P_m and P_n. Similarly, the torus graph T_n [44] is the Cartesian product of cycle C_n with itself.

The *complete k-level t-ary tree* $T_{t,k}$ is a tree with root vertex v at level 1, where each vertex on levels $1, \ldots, k-1$ has exactly t successors on the next level, and each vertex on levels $2, \ldots k$ has one predecessor on the previous level (hence we have $|V(T_{t,k})| = (t^k - 1)/(t-1)$).

The *complete graph* on n vertices K_n is a graph, where all pairs of vertices are adjacent. The *complete bipartite graph* $K_{m,n}$ is a graph on $m + n$ vertices, where $V = V_1 \cup V_2$ with $|V_1| = m$, $|V_2| = n$, and (p, q) is an edge if and only if p and q are not from the same set V_i, $i = 1, 2$. The *complete k-partite graph* $K_{m_1, m_2, \ldots, m_k}$ is defined similarly.

The *n-cube* Q_n is a graph where $V = \{0, 1\}^n$, and two 0-1 sequences are adjacent if and only if they differ in exactly one position. Thus, Q_1 is the path P_2, and Q_2 is the cycle C_4.

In the following theorem, we cite some well-known solutions of the bandwidth problem on some particular graphs. They are either obvious or adopted from [50,51].

Theorem 1.

(i) $OPT_{GBP}(P_n) = 1.$
(ii) $OPT_{GBP}(K_n) = n - 1.$
(iii) $OPT_{GBP}(C_n) = 2.$
(iv) $OPT_{GBP}(P_{m,n}) = \min\{m, n\}.$
(v) $OPT_{GBP}(T_n) = 2n - 1.$
(vi) $OPT_{GBP}(T_{t,k}) = \lceil \frac{t(t^{k-1}-1)}{2(k-1)(t-1)} \rceil.$
(vii) $OPT_{GBP}(K_{m_1, m_2, \ldots, m_k}) = \sum_i m_i - \lceil \frac{1}{2}(m_1 + 1) \rceil$, *if* $m_1 = \max_i m_i$.
(viii) $OPT_{GBP}(Q_n) = \sum_{k=0}^{n-1} \binom{k}{\lfloor \frac{k}{2} \rfloor}.$

3. Embed and Project Algorithm (EPA)

The bandwidth problem may be interpreted as looking for such an embedding of vertex set V into the integer line that minimizes the maximum distance between two adjacent vertices. Blum et al. [1] noticed that it is equivalent to the problem where we consider embeddings of V into a set of distinct equispaced points along the quarter-circle of radius n in the positive quadrant of a two-dimensional space, i.e., into set $U := \left\{ \left(n\cos(\frac{j\pi}{2n}), n\sin(\frac{j\pi}{2n}) \right);\ j = 1, 2, \ldots, n \right\}$. The equivalence follows from the fact that the distance between two points from U is uniquely determined by the number of points from U that lie between them, and the same is true when we consider embeddings into an integer line. Relaxing the demand to be on a quarter circle to allow the embeddings into an $n-1$ dimensional sphere of radius n leads to the following SDP:

$$
(\mathrm{GBP}_{SDP}) \quad
\begin{array}{rcll}
OPT_{SDP} = \min & b & & \\
\text{s. t.} \quad Y & \in & \mathcal{S}_n^+, & \\
y_{ij} & \geq & 0, \quad \forall i, j, & (3) \\
y_{ii} & = & n^2, \quad \forall i, & (4) \\
2y_{ij} + b & \geq & 2n^2, \quad \forall ij \in E, & (5) \\
\dfrac{2}{|S|} \sum_{j \in S} y_{ij} & \leq & 2n^2 - \alpha_{|S|}, \quad \forall i,\ \forall S \subseteq \{1, \ldots, n\} \setminus \{i\} & (6)
\end{array}
$$

where

$$\alpha_k = \frac{1}{6}\left(\frac{k}{2}+1\right)(k+1). \tag{7}$$

Here, the matrix variable Y represents the scalar products of the vectors, which the graph vertices are mapped to. Solving this SDP is actually the "embed" part of EPA from [1], which is summarized in Algorithm 1.

Algorithm 1: Embed and Project Algorithm (EPA) for solving (1).

INPUT: graph $G = (V, E)$, maximum number of projections $M \in \mathbb{N}$.

1. Solve the SDP program GPB_{SDP} for the graph G to obtain Y.
2. Find the matrix W such that $W^T W = Y$ (e.g., $W = Y^{1/2}$).
3. Let vectors w_i be the columns of W, $1 \le i \le n$. Define $\varphi =$ identity and $OPT_{EPA} = n - 1$.
4. **For** $s = 1, 2, \ldots, M$
   - 4.1 Choose random unit vector $\ell \in \mathbb{R}^n$, according to a uniform distribution on the unit sphere.
   - 4.2 Define labeling $\varphi_s \colon V \to \{1, \ldots, n\}$ such that

$$\varphi_s(i) \le \varphi_s(j) \iff w_i^T \ell \le w_j^T \ell.$$

   - 4.3 Compute bandwidth of φ_s as $BW_{\varphi_s} := \max\{|\varphi_s(i) - \varphi_s(j)|; ij \in E\}$.
   - 4.4 If $BW_{\varphi_s} \le BW_{\varphi}$ then $\varphi = \varphi_s$ and $OPT_{EPA} = BW_{\varphi_s}$.

OUTPUT: φ, OPT_{EPA}.

We can see that computationally, the hardest part of EPA is solving SDP in Step 1, especially since this SDP has exponentially many linear constraints. A theoretically important but practically irrelevant answer to this question has already been provided by the authors in [1]. Constraint (6) includes $n(2^{n-1} - (n+1)/2)$, but we can check their feasibility in a polynomial time: for each row $i = 1, \ldots, n$, we sort the off-diagonal elements $y_{i1}, y_{i2}, \ldots, y_{i\,i-1}, y_{i\,i+1}, \ldots, y_{in}$ in decreasing order: $y_{ij_1} \ge y_{ij_2} \ge \cdots \ge y_{ij_{n-1}}$ and then check the feasibility only the first k elements in this order, for every $k = 1, \ldots, n-1$, for Equation (6). If these inequalities are satisfied, then for every $S \subset \{1, \ldots, n\} \setminus i$ with $|S| = k$, we have

$$\frac{2}{|S|}\sum_{j \in S} y_{ij} \le \frac{2}{|S|}\sum_{\ell=1}^{k} y_{ij_\ell} \le 2n^2 - \alpha_k.$$

4. Some Theoretical Guarantees for OPT_{GBP} and OPT_{SDP}

In this section, we present some new results of the solution for GBP$_{SDP}$ and relations between OPT_{SDP} and OPT_{GBP}.

4.1. Theoretical Guarantees for OPT_{SDP}

First, we consider the feasibility of GBP$_{SDP}$. We need the following lemma, which underlies the main results of [1] and is taken from [12]. We provide the proof since it reveals the main idea of constraint (6).

Lemma 1. *Let $A = \{1, 2, \ldots, n\}$ and $i \in A$. For arbitrary $S \subseteq A \setminus \{i\}$, the following is true:*

$$\frac{1}{|S|}\sum_{j \in S}(i-j)^2 \ge \alpha_{|S|}.$$

Proof. Suppose first that $|S| = 2k$. Then

$$\sum_{j \in S}(i-j)^2 \ge 2\sum_{l=1}^{k} l^2 = \frac{2k(k+1)(2k+1)}{6} = \frac{1}{6}|S|(\frac{|S|}{2}+1)(|S|+1)$$

with equality holding when S consists of first k left and right neighbors of i. If $|S| = 2k+1$, then

$$\begin{aligned}\sum_{j\in S}(i-j)^2 &\geq 2\sum_{l=1}^{k} l^2 + (k+1)^2 = \frac{(k+1)(4k^2+8k+6)}{6} \\ &\geq \frac{(k+1)(4k^2+8k+3)}{6} = \frac{1}{6}|S|(\frac{|S|}{2}+1)(|S|+1).\end{aligned}$$

□

The feasibility of GBP_{SDP} is an important question. It is resolved in the following trivial lemma.

Lemma 2. *The pair $b = 2n^2, Y = n^2 I$ is feasible for GBP_{SDP}.*

We will also use the following result from [12] (Lemma 4.9).

Lemma 3. *For arbitrary graph $G = (V, E)$ with maximum vertex degree Δ and local density D, we have*

$$OPT_{SDP} \geq \max\left\{\frac{(\Delta+1)(\Delta+2)}{12}, \frac{D^2}{12}\right\}. \quad (8)$$

For complete graph K_n, we can show that the bound from Lemma 3 is tight.

Lemma 4. *If G is a complete graph K_n on n vertices, then the optimum value of OPT_{SDP} is $\frac{n(n+1)}{12}$.*

Proof. Lemma 3 implies for K_n that $OPT_{SDP} \geq \frac{n(n+1)}{12}$. For the other direction, we need to construct a feasible solution $(\hat{b}, \hat{Y})$, such that $\hat{b} \leq \frac{n(n+1)}{12}$. We claim that $\hat{b} = \frac{n(n+1)}{12}$ and $\hat{Y} = n^2 I + (n^2 - \frac{\alpha_{n-1}}{2})(J - I)$ is such a pair. Indeed, by construction, it follows that $\hat{Y} \geq 0$ and $\hat{Y} \succeq 0$ since the smallest eigenvalue of $\hat{Y}$ is $\alpha_{n-1}/2$. The pair $\hat{b}, \hat{Y}$ satisfies constraints shown in Equations (4) and (5). While the former is trivial, the latter follows from the fact that for $i \neq j$, we have $\hat{y}_{ij} = n^2 - \alpha_{n-1}/2$, which implies that $\hat{b} = \frac{n(n+1)}{12} = 2n^2 - 2y_{ij} = \alpha_{n-1}$. It remains to show the feasibility for the last constraint (Equation (6)). This follows, since for every i and every $S \subset \{1, \ldots, n\} \setminus \{i\}$, we have

$$\frac{2}{|S|}\sum_{j\in S}\hat{y}_{ij} = 2n^2 - \alpha_{n-1} \leq 2n^2 - \alpha_{|S|},$$

since α_k increases with k. □

For sub-graphs, we can prove the following relations.

Lemma 5. *Suppose G and H are graphs on n vertices and $G \subseteq H$. If Y_G and Y_H are the corresponding optimum solutions for GBP_{SDP}, then $OPT_{SDP}(G) \leq OPT_{SDP}(H)$.*

Proof. Note that semidefinite programs GBP_{SDP}, which correspond to G and H, differ only in Constraint (5). From $G \subseteq H$, it follows that all inequalities (Equation(5)) in the SDP, which corresponds to G, are also included in the SDP that corresponds to H. Hence, if Y_H is feasible for GBP_{SDP} corresponding to H, then it is feasible also for GBP_{SDP} corresponding to G and consequently $OPT_{SDP}(G) \leq OPT_{SDP}(H)$. □

The above results imply the following corollary.

Corollary 1. *For any simple and connected graph G on n vertices, the following holds:*

$$OPT_{SDP}(P_n) \leq OPT_{SDP} \leq \frac{n(n+1)}{12},$$

where $OPT_{SDP}(P_n)$ is the optimum value of GBP_{SDP} for path P_n.

Proof. This follows from the fact that for every simple and connected G, we have $P_n \subseteq G \subseteq K_n$ and from Lemma 4. □

For the path on n vertices, we could not derive a closed formula solution for GBP_{SDP}, as we did for K_n. However, the following lemma provides a very tight upper bound.

Lemma 6. *If G is simple path P_n on n vertices, then for the optimum value of GBP_{SDP}, it holds that*

$$OPT_{SDP} \leq 2n^2(1 - \cos(\frac{\pi}{3n})).$$

Proof. Let us define Y by $y_{ij} = \langle w_i, w_j \rangle$, where $w_i = \Big(n\cos(\frac{i\pi}{3n}), n\sin(\frac{i\pi}{3n})\Big)$, for $i = 1, 2, \ldots, n$. We show that pair (b, Y) with $b = 2n^2(1 - \cos(\frac{\pi}{3n}))$ is feasible for GBP_{SDP}.

Constraint (4) is trivial. Constraint (3) is satisfied since $y_{ij} = n^2\cos((i-j)\pi/(3n)) \geq 0$. This follows from $|i-j|\pi/(3n) \leq \pi/3$. The feasibility for Constraint (5) follows since for every edge $ij \in E$, we have $|i-j| = 1$, hence $2y_{ij} + b = 2n^2\cos(\pi/(3n)) + 2n^2(1 - \cos(\pi/(3n))) = 2n^2$.

It remains to prove that Y is feasible for Constraint (6). We first show that $\|w_i - w_j\| \geq |i-j|$. Indeed, $\|w_i - w_j\|^2 = 4n^2\sin^2\frac{(i-j)\pi}{6n}$, hence we only need to show that

$$2n\sin(i\pi/(6n)) \geq i, \quad \text{for all } i = 1, \ldots, n-1.$$

This is equivalent to $\sin(i\pi/(6n)) \geq i/(2n)\ \forall i$, and follows from the fact that $\sin(x\pi/(6n)) - x/(2n)$ is concave for $x \in (0, n)$ and has zeros in 0 and n.

Therefore, $2y_{ij} = 2n^2 - \|w_i - w_j\|^2 \leq 2n^2 - (i-j)^2$. Hence, Constraint (6) follows since Lemma 1 implies that

$$\frac{2}{|S|}\sum_{j\in S} y_{ij} \leq \frac{2}{|S|}\sum_{j\in S}(n^2 - \frac{(i-j)^2}{2}) \ \leq\ 2n^2 - \alpha_{|S|}.$$

□

We have also derived a closed formula solution for OPT_{SDP}. It strengthens the result from Lemma 6 by showing that the angle $\pi/(3n)$ in the definition of w_i can be further decreased to some β_{opt}, and the resulting matrix Y remains feasible for GBP_{SDP}. Unfortunately, we have not been able to prove the optimality of the Y related to β_{opt} for general n. However, we computed these values of β_{opt} numerically, using MATLAB, for $n \leq 1024$ (the size of the largest graph we numerically studied for this paper), and then constructed Y in a similar way as above. In all these cases, Y was feasible for GBP_{SDP}, i.e., Constraint (6) was satisfied with a relative error below 10^{-4}, and for the case of simple paths, we obtained value OPT_{SDP} (up to numerical error), see Table 1. This is the reason why we formulate our result as a conjecture. We keep working on the proof, but our current results consist of a long and tedious trigonometric analysis, which is in any case out of the scope of the current paper.

Conjecture 1. *If G is a simple path P_n on n vertices, then the optimum value of GBP_{SDP} is*

$$OPT_{SDP} = 2n^2(1 - \cos(\beta_{opt})),$$

where β_{opt} is the smallest positive solution of equation

$$\sin((k+\frac{1}{2})\beta) = \sin(\frac{\beta}{2})\frac{23n^2+1}{24n}, \text{ if } n \text{ is an odd number and } k=(n-1)/2 \tag{9}$$
$$\sin((k+\frac{1}{2})\beta) = \sin(\frac{\beta}{2})\frac{23n^3-21n^2-2n}{24n^2}, \text{ if } n \text{ is an even number and } k=n/2-1. \tag{10}$$

4.2. Theoretical Guaranties for OPT_{GBP}

Blum et al. [1] have analyzed the expected quality of the labeling generated by EPA. Their results are summarized in the following theorem.

Theorem 2 ([1])**.** *Let OPT_{EPA} be the bandwidth of labeling computed by EPA from Algorithm 1, and let OPT_{SDP} be the optimal value of GBP_{SDP}. With high probability, we have*

$$OPT_{EPA} \leq \mathcal{O}(\sqrt{n}\log n/\sqrt[4]{OPT_{SDP}})\cdot OPT_{GBP}.$$

The proof of this theorem is quite complex and long. In the original paper [1], it is done in very dense form and might be demanding to understand. It was redone in a more understandable way in [12].

Note that the term "high probability" has a classical meaning: if the number of projections M in Step 4 of EPA is of polynomial size of n and large enough, then this probability is arbitrary close to 1.

We can use Lemma 3 to simplify the guaranty from Theorem 2.

Corollary 2. *If G is a graph with maximum degree Δ and diameter d, then EPA from Algorithm 1 with high probability returns labeling φ with*

$$OPT_{EPA} \leq \mathcal{O}(C\log n)\cdot OPT_{GBP},$$

where

$$C = \min\left\{\frac{\sqrt{n}}{\sqrt{\Delta+1}},\ \sqrt{d}\right\}.$$

Note that if Δ is close to n (that happens if, e.g., there exists a vertex which is adjacent to almost all vertices), then the constant C from Corollary 2 is close to 1. In graphs where $d=\mathcal{O}(\log^t n)$, for some $t\geq 0$ (this happens, e.g., in almost complete k-ary trees), we get $C=\mathcal{O}(\log^{t/2} n)$.

The results above imply the following new lower bound for OPT_{GBP}, which is an improvement of the lower bounds developed in Blum et al. [1,12]. The result follows from the observation that the vectors $w_i = \left(n\cos(\frac{i\pi}{2n}), n\sin(\frac{i\pi}{2n})\right)$ imply a feasible matrix Y for GBP_{SDP}. By using the result of Lemma 6 and Conjecture 1, we improve this bound.

Lemma 7. *For arbitrary graph $G=(V,E)$, we have*

$$OPT_{GBP} \geq \left\lceil\frac{\sqrt{OPT_{SDP}}}{n\beta}\right\rceil \geq \left\lceil\frac{3\sqrt{OPT_{SDP}}}{\pi}\right\rceil, \tag{11}$$

where $\beta\leq \pi/(3n)$ is the smallest angle, such that vectors

$$w_i := \Big(n\cos(\varphi(i)\beta), n\sin(\varphi(i)\beta), 0,\ldots,0\Big)\in\mathbb{R}^n,\ i=1,2,\ldots,n, \tag{12}$$

are feasible for Constraint (6).

Proof. Let φ be the optimal labeling of $V(G)$ (i.e., $OPT_{GBP} = \max_{ij \in E} |\varphi(i) - \varphi(j)|$). Lemma 6 implies that the vectors w_i yield feasible solution Y for GBP_{SDP}. Obviously, $OPT_{SDP} \leq b := \max_{ij \in E} \|w_i - w_j\|^2$. On the other hand, for every $ij \in E$, we have

$$\begin{aligned} \|w_i - w_j\|^2 = \; & 4n^2 \sin^2((\varphi(i) - \varphi(j))\beta/2) \leq 4n^2 \sin^2(OPT_{GBP} \cdot \beta/2) \leq \\ \leq \; & 4n^2 (OPT_{GBP} \cdot \beta/2)^2 \leq 4n^2 (OPT_{GBP} \cdot \pi/(6n))^2 = (\pi \cdot OPT_{GBP}/3)^2, \end{aligned}$$

hence

$$OPT_{SDP} \leq \max_{ij \in E} \|w_i - w_j\|^2 \leq 4n^2 (OPT_{GBP} \cdot \beta/2)^2 \leq (\pi \cdot OPT_{GBP}/3)^2,$$

which is equivalent to

$$OPT_{GBP} \geq \frac{\sqrt{OPT_{SDP}}}{n\beta} \geq \frac{3\sqrt{OPT_{SDP}}}{\pi}.$$

Since OPT_{GBP} is always a positive integer number, we can apply rounding up on the chain of inequalities from above. □

Remark 1. *As mentioned in the paragraph before Conjecture 1, we numerically checked for* $n \leq 1024$ *that values* β_{opt} *obtained by solving Equations (9) and (10) yield vectors* w_i*, which imply the feasible solution* Y*. Hence, these* β *values can be used to compute an enhanced lower bound from Lemma 7. Indeed, we report these bounds in all tables that we provide.*

5. Computational Results

5.1. Computational Issues with Solving GBP_{SDP}

Note that the GBP_{SDP} is an SDP in matrices of order n. Recently developed bounds for OPT_{GBP}, which are based on semidefinite programming relaxations of the quadratic assignment problem or graph partitioning problem [40,44,45], involve semidefinite programs in matrices of order kn, which is much worse compared to our SDP. However, Constraints (3), (5) and (6) are very expensive. To begin with, they include inequalities, and if we want to solve GBP_{SDP} by interior-point methods, we have to introduce one new non-negative slack variable for each inequality. Secondly, the number of inequalities in Constraint (6) is exponential in n. For any matrix $X \in \mathcal{S}_n^+$, we can decide in polynomial time whether it is feasible for Constraint (6) or not, as was mentioned in Section 3. This is a theoretically very strong result since we may apply the ellipsoid method, which needs only a polynomial separation oracle to solve the problem in polynomial time (see, e.g., [52]). It is well known that the ellipsoid method has very poor practical efficiency; therefore, we are interested in applying other, more efficient methods, such as interior-point methods [13,14,53] or the bundle method [15,16].

In Algorithm 2, we present a cutting-plane-like algorithm, which enables us to solve GBP_{SDP} to optimality in a reasonable time by interior-point methods if $|V(G)| \leq 200$ or if $|V| \leq 500$ and the graph is sparse or by the bundle method for graphs with $|V(G)| \leq 1000$.

Below, we explain in detail the steps from the cutting-plane algorithm from Algorithm 2.

Step 1. Here, we may take an arbitrary subset. Numerical experiments show that it makes sense to take only a few (default setting is 2) inequalities for each $1 \leq i \leq n$. We take those with $|S| = n - 1$.

Step 2. We solve the SDP from Step 1 to optimality by using interior-point methods (SDPT3 [13], SEDUMI [54], and MOSEK [55]), if $n \leq 200$ or if $n \leq 500$ and the graphs are sparse. Otherwise, we use the bundle method [15,16].

Step 3.1. The new subset is carefully selected. All the inequalities from the previous two iterations that are still important (have nonzero dual variable) are kept. Additionally, for each $1 \leq i \leq n$, we add some of the most violated inequalities. We detect them by sorting the ith row of $\hat{Y}$ from the previous iteration in decreasing order and then take inequalities with the largest numbers of variables (only the first few of them). If at some iteration, $\hat{Y}$ violates an inequality that was already involved but deleted, we take this inequality back and keep it forever.

Step 3.2. The same as Step 2.

Algorithm 2: Cutting-plane algorithm for solving the GBP_{SDP}

INPUT: graph $G = (V, E)$.

1. Select some subset of inequalities $\mathcal{A}_0(Y) \leq a_0$ from Constraint (6) with size linear in n.
2. Solve GBP_{SDP}, where Constraint (6) is replaced by $\mathcal{A}_0(X) \leq a_0$. Let $\hat{Y}$ be the optimum solution.
3. **While** $\hat{Y}$ is not feasible for Constraint (6) OR the maximum number of iterations is not reached **do**
   - 3.1 Select new small subset of inequalities $\mathcal{A}(Y) \leq a$ from Constraint (6), which contains all important inequalities from previous iterations.
   - 3.2 Solve GBP_{SDP} with Constraint (6) replaced by $\mathcal{A}(Y) \leq a$ using interior-point methods or the bundle method to obtain new optimum solution $\hat{Y}$.

OUTPUT: $\hat{Y}$, SDP_{OPT}.

If we keep all the important inequalities from the previous iteration and keep adding new inequalities, then, in the worst case scenario, we eventually (after many iterations) add all the inequalities from Constraint (6); hence, the optimum $\hat{Y}$ returned by the cutting-plane algorithm is the optimum solution for GBP_{SDP}. However, in practice, we limit the number of iterations (in numerical experiments we use only 12 iterations for the interior-point method and 7 iterations for the bundle method) and still get very close to an optimum point.

5.2. Results

In this subsection, we report numerical results, obtained by EPA on the test instances for which the problem GBP_{SDP} is solvable by our implementation of the cutting-plane algorithm. For graphs with fewer than 200 vertices or sparse graphs with fewer than 500 vertices, we used interior-point methods. In particular, we used solvers SDPT3, available at [13], SEDUMI [54], and MOSEK [55]. We also solved larger instances, where GBP_{SDP} is too big for interior-point methods. In these cases, we used the bundle method (for details about this method, see [15,16]).

We first demonstrate the behavior of EPA on complete graphs and simple paths since for the former, we have a closed formula solution (Lemma 4), while for the latter, we have upper bounds from Lemma 6 and the conjectured formula for the optimum values (Conjecture 1). The results are reported in Tables 1–2. Table 1 contains size n in the first column, the optimum value of (1) in the second column, the optimum value computed by EPA in the third, the optimum value of GBP_{SDP} in the fourth column, the upper bounds from Lemma 6, the values from Conjecture 1 in the fifth column, and angles $\pi/(3n)$ used in Lemma 6 and the β_{opt} angles from Conjecture 1 are in the last two columns.

We can see that for simple paths P_n, EPA always detects the optimum solution of (1). Additionally, the bound from Lemma 6 is good but does not approach OPT_{SDP} with increasing n. On the other hand, the sixth column contains the values conjectured by Conjecture 1, which are obviously equal to the optimum values of OPT_{SDP}. This demonstrates that this conjecture is valid (with a precision of 10^{-4}) at least for values n from Table 1. Actually, as explained in the text before Conjecture 1, we numerically validated this conjecture for $n \leq 1024$ and used the computed values of β_{opt} to compute the enhanced lower bounds for OPT_{GBP} based on Lemma 7.

The second group of numerical results pertains to complete graphs K_n, for which we know the optimum values of (1) and GBP_{SDP} (Theorem 1 and Lemma 4). In the first column, we put the size of the graph, i.e., the number of vertices n in graph K_n, the second column contains the (well-known) bandwidth of the graph, the third column contains the bandwidth of labeling obtained by EPA (we take $M = 10.000$ projections), the fourth column contains the optimal values of GBP_{SDP}, and the last two columns contain the lower bounds for OPT_{GBP} based on Lemma 7. We can observe that numerical results in Table 2 are well aligned with the results from Lemma 4. Additionally, the computed values of bandwidth are always equal to the optimum value $n-1$, which is not surprising since this is the only value that any projection from Step 4.2 of EPA can attain. Unsurprisingly, the lower bounds from the last two columns are rather weak since they are based on finding the longest path in a graph. The more the graph is similar to a path, the better this lower bound is.

Table 1. Numerical results obtained by the EPA algorithm on simple paths P_n.

n	OPT_{GBP}	OPT_{EPA}	OPT_{SDP}	**Bound Lemma 6**	**Bound Conj 1**	$\pi/(3n)$	β_{opt}
10	1	1	1.0091	1.0956	1.0091	0.1047	0.1005
15	1	1	1.0122	1.0962	1.0122	0.0698	0.0671
20	1	1	1.0112	1.0964	1.0112	0.0524	0.0503
25	1	1	1.0126	1.0965	1.0126	0.0419	0.0403
30	1	1	1.0118	1.0965	1.0118	0.0349	0.0335
35	1	1	1.0126	1.0965	1.0126	0.0299	0.0288
40	1	1	1.0120	1.0966	1.0120	0.0262	0.0252
45	1	1	1.0127	1.0966	1.0127	0.0233	0.0224
50	1	1	1.0122	1.0966	1.0122	0.0209	0.0201

Table 2. Results of the EPA algorithm on complete graphs K_n

n	OPT_{GBP}	OPT_{EPA}	OPT_{SDP}	$\lceil 3/\pi \cdot \sqrt{OPT_{SDP}} \rceil$	$\lceil \sqrt{OPT_{SDP}}/(n\beta) \rceil$
25	24	24	54.1667	8.0000	8.0000
40	39	39	136.6667	12.0000	12.0000
55	54	54	256.6667	16.0000	16.0000
70	69	69	414.1667	20.0000	21.0000
85	84	84	609.1667	24.0000	25.0000
100	99	99	841.6667	28.0000	29.0000

The third part of the numerical results consists of Table 3. These are the results obtained by EPA on the rest of the graphs, for which we know the optimum bandwidth (see Theorem 1, items (*iii*)–(*viii*). We can observe that the labelings obtained by EPA are, in our opinion, very close to the optimum. On the other hand, the lower bounds are tight only for cycles, while for the other graphs, there is a non-negligible gap between them and the optimum value of OPT_{GBP}.

Lastly, we illustrate the behavior of EPA on caterpillars with a hair length of 1. A caterpillar is a simple graph that consists of a backbone, which is, in fact, a simple path P, and with an arbitrary number of simple paths (called hairs), starting from vertices of P. A subcaterpillar is a sub-graph that is also a caterpillar. For caterpillars with a hair length of at most 2, there exists a polynomial time algorithm to compute OPT_{GBP}—it is equal to the local density D, defined in Equation (2). For more details, see [12,18].

Table 3. Results on graphs with a known bandwidth.

Instance	n	OPT_GBP	OPT_EPA	OPT_SDP	$\lceil 3/\pi \cdot \sqrt{OPT_{SDP}} \rceil$	$\lceil \sqrt{OPT_{SDP}/(n\beta)} \rceil$
C_{100}	100	2	2	1.6444	2	2
C_{150}	150	2	2	1.6446	2	2
C_{200}	200	2	2	1.6660	2	2
C_{250}	250	2	2	1.6607	2	2
C_{300}	300	2	3	1.7214	2	2
$P_{5,20}$	100	5	5	22.9788	5	5
$P_{5,25}$	125	5	5	23.7658	5	5
$P_{10,15}$	150	10	11	62.3827	8	8
$P_{10,20}$	200	10	11	73.8030	9	9
T_7	49	13	14	37.6510	6	7
T_8	64	15	16	49.9751	7	8
T_9	81	17	18	63.9479	8	8
T_{10}	100	19	20	79.5694	9	9
T_{15}	225	29	30	182.3621	13	14
T_{20}	400	39	41	326.2956	18	18
$T_{2,5}$	31	4	5	7.6207	3	3
$T_{4,5}$	341	43	55	970.5753	30	31
$T_{3,6}$	364	37	55	692.6553	26	27
$T_{2,6}$	63	7	8	19.4953	5	5
$T_{2,7}$	127	11	14	51.7816	7	8
$T_{2,8}$	255	19	30	178.4072	13	14
$K_{5,10,15,20}$	50	39	43	208.2501	14	15
$K_{10,20,30,40}$	100	79	79	833.2500	28	29
$K_{10,20,30,40,50}$	150	124	143	1874.9168	42	44
$K_{20,30,40,50,60}$	200	169	193	3333.2591	56	58
Q_5	32	13	13	34.1000	6	6
Q_6	64	23	23	113.7500	11	11
Q_7	128	43	43	390.0714	19	20
Q_8	256	78	83	1365.3128	36	37
Q_9	512	148	163	4854.5317	67	70
Q_{10}	1024	274	309	17476.6041	127	132

In Table 4, we report the numerical results obtained on seven caterpillars with a hair length of 1. The first column contains the names of the instances. Caterpillar $C_{m_1,m_2,\dots,m_k}$ has k nodes on the main path (the backbone), each of them having attached m_i hairs of length 1. The last caterpillar $C^{15}_{1,2,3,2,1}$ has backbone of length 75, where the first 15 vertices on the backbone have 1 hair of length 1, the next 15 vertices have 2 hairs of length 1, the next 15 have 3 hairs of length 1, and in this manner symmetrically till the end. Note that in the last two lines, the lower bounds are tight. This conforms to intuition: the more the graph is similar to the path, the better the lower bounds from Lemma 11 are.

Table 4. Results on caterpillars with a hair length of 1.

Instance	n	OPT_GBP	OPT_EPA	OPT_SDP	$\lceil 3/\pi \cdot \sqrt{OPT_{SDP}} \rceil$	$\lceil \sqrt{OPT_{SDP}/(n\beta)} \rceil$
$C_{5,10,15,20}$	54	13	14	74.5537	9	9
$C_{10,20,30,40,50}$	155	31	33	635.3134	25	26
$C_{5,10,15,20,25,30,35,40}$	188	27	33	423.7338	20	21
$C_{15,15,15,15,15,15,15,15,15,15}$	160	15	17	160.9720	13	13
$C_{4,12,20,6,10,25,15,7,35}$	143	18	23	164.1064	13	13
$C_{5,5,5,\ldots,5}$	140	4	5	12.7798	4	4
$C^{15}_{1,2,3,2,1}$	150	6	9	34.0927	6	6

6. Conclusions and Future Work

In this paper, we studied the *Embed and Project Algorithm* (EPA) to approximately solve the bandwidth problem proposed in [1]. It consists of several important steps. The central step consists of solving a semidefinite program with exponentially many linear constraints. In the original paper, the ellipsoid method was proposed to solve this SDP since we can check the feasibility of each candidate solution in a polynomial time.

While the original result was mostly of theoretical importance, the results in this paper showed that we can devise a cutting-plane-like algorithm in combination with interior-point methods or the bundle method to solve the underlying SDP. This algorithm includes only a few (linearly many) of the most important constraints from this exponential set of constraints, which implies all the other constraints. We have also established new theoretical insights into EPA and into the underlying SDP, which help to understand EPA and were used to develop new lower bounds for OPT_{GBP}.

The extensive numerical results are very promising and confirm that EPA in practice yields very good bandwidth approximations and has strong potential for further research.

The main open question for future research is the development of new methods to solve GBP_{SDP}. We have already observed that the existing SDP solvers, which rely on interior-point methods, do not scale well to a large number of constraints. The bundle method, which we used when the number of constraints was too large, has a slow convergence, so there is a need for new methods to solve GBP_{SDP}. Based on our experiences related to other combinatorial optimization problems [56], we believe that the ADMM method, in combination with the augmented Lagrangian method, has strong potential, and we will test this in the future.

Another interesting question is how to extend EPA to other, similar layout problems, such as the topological bandwidth problem, the cutwidth problem, the edge-bandwidth problem, etc.

Funding: The work was supported by the Slovenian Research Agency through the projects N1-0071, J1-2453, J2-2512, J1-1691, and program P2-0162.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data about instances used in the numerical computations is available upon request.

Conflicts of Interest: No conflict of interest.

References

1. Blum, A.; Konjevod, G.; Ravi, R.; Vempala, S. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theor. Comput. Sci.* **2000**, *235*, 25–42.
2. Dunagan, J.; Vempala, S. On Euclidean embeddings and bandwidth minimization. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 229–240.

3. Chinn, P.Z.; Chvátalová, J.; Dewdney, A.K.; Gibbs, N.E. The bandwidth problem for graphs and matrices—A survey. *J. Graph Theory* **1982**, *6*, 223–254.
4. Harary, F. Sparse matrices in graph theory. In *Large Sparse Sets of Linear Equations*; Academic Press: Cambridge, MA, USA, 1970; pp. 139–150.
5. Papadimitriou, C.H. The NP-completeness of the bandwidth minimization problem. *Computing* **1976**, *16*, 263–270.
6. Garey, M.R.; Graham, R.L.; Johnson, D.S.; Knuth, D.E. Complexity results for bandwidth minimization. *SIAM J. Appl. Math.* **1978**, *34*, 477–495.
7. Monien, B. The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM J. Algebr. Discret. Methods* **1986**, *7*, 505–512.
8. Karpinski, G.B.M.; Wirthgen, J. On Approximation Intractability of the Bandwidth Problem. *Electron. Colloq. Comput. Complex. Rep.* **1998**, *14*, 97–041.
9. Feige, U. Approximating the bandwidth via volume respecting embeddings. *J. Comput. Syst. Sci.* **2000**, *60*, 510–539.
10. Chung, F.R.; Seymour, P.D. Graphs with small bandwidth and cutwidth. *Discret. Math.* **1989**, *75*, 113–119.
11. Rao, S. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In Proceedings of the Fifteenth Annual Symposium on Computational Geometry, Miami Beach, FL, USA, 13–16 June 1999; pp. 300–306.
12. Povh, J. *Towards the Optimum by Semidefinite and Copositive Programming: New Approach to Approximate Hard Optimization Problems*; VDM Publishing: Saarbrucken, Germany, 2009.
13. Tütüncü, R.H.; Toh, K.C.; Todd, M.J. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.* **2003**, *95*, 189–217.
14. Terlaky, T. *Interior Point Methods of Mathematical Programming*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 5.
15. Anjos, M.F.; Ghaddar, B.; Hupp, L.; Liers, F.; Wiegele, A. Solving k-way graph partitioning problems to optimality: The impact of semidefinite relaxations and the bundle method. In *Facets of Combinatorial Optimization*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 355–386.
16. Fischer, I.; Gruber, G.; Rendl, F.; Sotirov, R. Computational experience with a bundle approach for semidefinite cutting plane relaxations of Max-Cut and equipartition. *Math. Program.* **2006**, *105*, 451–469.
17. Mafteiu-Scai, L.O. The bandwidths of a matrix. A survey of algorithms. *Ann. West Univ. Timis.-Math. Comput. Sci.* **2014**, *52*, 183–223.
18. Assmann, S.; Peck, G.; Sysło, M.; Zak, J. The bandwidth of caterpillars with hairs of length 1 and 2. *SIAM J. Algebr. Discret. Methods* **1981**, *2*, 387–393.
19. Kratsch, D. Finding the minimum bandwidth of an interval graph. *Inf. Comput.* **1987**, *74*, 140–158.
20. Kloks, T.; Kratsch, D.; Müller, H. Bandwidth of chain graphs. *Inf. Process. Lett.* **1998**, *68*, 313–315.
21. Heggernes, P.; Kratsch, D.; Meister, D. Bandwidth of bipartite permutation graphs in polynomial time. *J. Discret. Algorithms* **2009**, *7*, 533–544.
22. Feige, U.; Talwar, K. Approximating the bandwidth of caterpillars. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 62–73.
23. Kloks, T.; Kratsch, D.; Müller, H. Approximating the bandwidth for asteroidal triple-free graphs. *J. Algorithms* **1999**, *32*, 41–57.
24. Haralambides, J.; Makedon, F. Approximation algorithms for the bandwidth minimization problem for a large class of trees. *Theory Comput. Syst.* **1997**, *30*, 67–90.
25. Karpinski, M.; Wirtgen, J.; Zelikovsky, A. An approximation algorithm for the bandwidth problem on dense graphs. In Proceedings of the RALCOM'97, Santorini Island, Greece, 6–11 October 1997; pp. 1–14.
26. Gupta, A. Improved bandwidth approximation for trees and chordal graphs. *J. Algorithms* **2001**, *40*, 24–36.
27. Fürer, M.; Gaspers, S.; Kasiviswanathan, S.P. An exponential time 2-approximation algorithm for bandwidth. In *International Workshop on Parameterized and Exact Computation*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 173–184.
28. Crisan, G.C.; Pintea, C.M. A hybrid technique for a matrix bandwidth problem. *Sci. Stud. Res.* **2011**, *21*, 113–120.
29. Lim, A.; Lin, J.; Rodrigues, B.; Xiao, F. Ant colony optimization with hill climbing for the bandwidth minimization problem. *Appl. Soft Comput.* **2006**, *6*, 180–188.
30. Guan, J.; Lin, G.; Feng, H.B. Ant colony optimisation with local search for the bandwidth minimisation problem on graphs. *Int. J. Intell. Inf. Database Syst.* **2019**, *12*, 65–78.
31. Hamon, R.; Borgnat, P.; Flandrin, P.; Robardet, C. Relabelling vertices according to the network structure by minimizing the cyclic bandwidth sum. *J. Complex Netw.* **2016**, *4*, 534–560.
32. Balogh, J.; Mubayi, D.; Pluhár, A. On the edge-bandwidth of graph products. *Theor. Comput. Sci.* **2006**, *359*, 43–57.
33. Baioletti, M.; Milani, A.; Santucci, V. Variable neighborhood algebraic differential evolution: An application to the linear ordering problem with cumulative costs. *Inf. Sci.* **2020**, *507*, 37–52.
34. Ceberio, J.; Irurozki, E.; Mendiburu, A.; Lozano, J.A. A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. *IEEE Trans. Evol. Comput.* **2013**, *18*, 286–300.
35. Ayodele, M.; McCall, J.; Regnier-Coudert, O. RK-EDA: A novel random key based estimation of distribution algorithm. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Scotland, UK, 17–21 September 2016; pp. 849–858.

36. Santucci, V.; Baioletti, M.; Milani, A. Tackling permutation-based optimization problems with an algebraic particle swarm optimization algorithm. *Fundam. Inform.* **2019**, *167*, 133–158.
37. Marti, R.; Gortazar, F.; Duarte, A. Heuristics for the bandwidth colouring problem. *Int. J. Metaheuristics* **2010**, *1*, 11–29.
38. Ambainis, A.; Balodis, K.; Iraids, J.; Kokainis, M.; Prūsis, K.; Vihrovs, J. Quantum speedups for exponential-time dynamic programming algorithms. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, San Diego, CA, USA, 6–9 January 2019; pp. 1783–1793.
39. Marinček, J.; Mohar, B. On approximating the maximum diameter ratio of graphs. *Discret. Math.* **2002**, *244*, 323–330.
40. De Klerk, E.; E.-Nagy, M.; Sotirov, R. On semidefinite programming bounds for graph bandwidth. *Optim. Methods Softw.* **2013**, *28*, 485–500.
41. Povh, J.; Rendl, F. A copositive programming approach to graph partitioning. *SIAM J. Optim.* **2007**, *18*, 223–241.
42. Povh, J.; Rendl, F. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discret. Optim.* **2009**, *6*, 231–241.
43. Povh, J. Contribution of copositive formulations to the graph partitioning problem. *Optimization* **2013**, *62*, 71–83.
44. Rendl, F.; Sotirov, R.; Truden, C. Lower bounds for the bandwidth problem. *Comput. Oper. Res.* **2021**, *135*, 105422.
45. van Dam, E.R.; Sotirov, R. On bounding the bandwidth of graphs with symmetry. *INFORMS J. Comput.* **2015**, *27*, 75–88.
46. Jiang, T.; Miller, Z.; Yager, D. On the bandwidth of the Kneser graph. *Discret. Appl. Math.* **2017**, *227*, 84–94.
47. Caprara, A.; Salazar-González, J.J. Laying out sparse graphs with provably minimum bandwidth. *INFORMS J. Comput.* **2005**, *17*, 356–373.
48. Martí, R.; Campos, V.; Piñana, E. A branch and bound algorithm for the matrix bandwidth minimization. *Eur. J. Oper. Res.* **2008**, *186*, 513–528.
49. Diestel, R. *Graph Theory*; Graduate Texts in Mathematics; Springer: Berlin/Heidelberg, Germany, 2018.
50. Chung, F.R. Labelings of graphs. *Sel. Top. Graph Theory* **1988**, *3*, 151–168.
51. Lai, Y.L.; Williams, K. A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs. *J. Graph Theory* **1999**, *31*, 75–94.
52. Grötschel, M.; Lovász, L.; Schrijver, A. *Geometric Algorithms and Combinatorial Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 2.
53. De Klerk, E. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 65.
54. Sturm, J.F. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* **1999**, *11*, 625–653.
55. MOSEK ApS. *The MOSEK Optimization Toolbox for MATLAB Manual*, Version 9.0; MOSEK ApS: Copenhagen, Denmark, 2019.
56. Hrga, T.; Povh, J. MADAM: A parallel exact solver for Max-Cut based on semidefinite programming and ADMM. *arXiv* **2020**, arXiv:2010.07839.