Graph Layout Problems Parameterized by Vertex Cover

Michael R. Fellows[1], Daniel Lokshtanov[2], Neeldhara Misra[3], Frances A. Rosamond[1], and Saket Saurabh[2]

[1] University of Newcastle,
Newcastle, Australia
{michael.fellows,frances.rosamond}@newcastle.edu.au
[2] Department of Informatics, University of Bergen,
N-5020 Bergen, Norway
{daniello,saket.saurabh}@ii.uib.no
[3] The Institute of Mathematical Sciences,
Chennai, 600 017, India
neeldhara@imsc.res.in

Abstract. In the framework of parameterized complexity, one of the most commonly used structural parameters is the *treewidth* of the input graph. The reason for this is that most natural graph problems turn out to be fixed parameter tractable when parameterized by treewidth. However, *Graph Layout* problems are a notable exception. In particular, no fixed parameter tractable algorithms are known for the Cutwidth, Bandwidth, Imbalance and Distortion problems parameterized by treewidth. In fact, Bandwidth remains NP-complete even restricted to trees. A possible way to attack graph layout problems is to consider structural parameterizations that are stronger than treewidth. In this paper we study graph layout problems parameterized by the size of the minimum vertex cover of the input graph. We show that all the mentioned problems are fixed parameter tractable. Our basic ingredient is a classical algorithm for Integer Linear Programming when parameterized by dimension, designed by Lenstra and later improved by Kannan. We hope that our results will serve to re-emphasize the importance and utility of this algorithm.

1 Introduction

Parameterized complexity can be thought of as a "multivariate" approach to complexity analysis and algorithm design. In addition to the overall input size n, a secondary measurement k, the *parameter*, is also considered. In the parameterized complexity framework the central notion is *fixed parameter tractability* (FPT), defined to be solvability in time $f(k)n^c$, where f is some arbitrary function and c is a constant. For further details and an introduction to parameterized complexity we refer to [8,11,27].

In the framework of parameterized complexity, an important aspect is the *choice of parameter* for a problem. Exploring how one parameter affects the

S.-H. Hong, H. Nagamochi, and T. Fukunaga (Eds.): ISAAC 2008, LNCS 5369, pp. 294–305, 2008.

Table 1. Problem Definitions

Problem Name	Objective Function	Problem Definition
BANDWIDTH	$f_{bw}(\pi) = \max_{uv \in E} \lvert\pi(u) - \pi(v)\rvert$	$\mathbf{bw}(G) = \min_{\pi} f_{bw}(\pi)$
CUTWIDTH	$f_{cw}(\pi) = \max_{1 \le i \le n} \lvert\partial(V_i)\rvert$	$\mathbf{cw}(G) = \min_{\pi} f_{cw}(\pi)$
IMBALANCE	$f_{im}(\pi) = \sum_{i=1}^{n} \lvert L_\pi(v_i) - R_\pi(v_i)\rvert$	$\mathbf{im}(G) = \min_{\pi} f_{im}(\pi)$
DISTORTION[1]	$f_{di}(\pi) = \max_{uv \in E} \sum_{i=\pi(u)}^{\pi(v)-1} D(v_i, v_{i+1})$	$\mathbf{di}(G) = \min_{\pi} f_{di}(\pi)$

complexity of different parameterized or unparameterized versions of the problem, often leads to non trivial combinatorics and better understanding of the problem. In general there are two kinds of parameterizations. In the first kind the parameter reflects the value of the objective function in question. The second kind, *structural parameterizations*, measure the structural properties of the input. A well developed structural parameter is the treewidth of the input graph. A celebrated result in this direction is that every problem expressible in monadic second order logic can be solved in time $O(f(t) \cdot n)$ for graphs of treewidth at most t [7]. Even though many problems become tractable when the treewidth of the input graph is bounded, there are quite a few that do not. For an example BANDWIDTH remains NP-complete even for trees. In these cases it is interesting to consider parameterizations which enforce more structure on the input than the treewidth. In this direction Fellows and Rosamond investigated how different problems behave when parameterized by the *max leaf number* of the input graph [10].

In this paper we consider parameterizing by the *vertex cover number* ($vc(G)$) of the graph. The vertex cover number of a graph G is the size of smallest set of vertices such that every edge has at least one end-point in this set. We study the graph layout problems CUTWIDTH, BANDWIDTH, IMBALANCE and DISTORTION parameterized by $vc(G)$. In a graph layout problem, we are given a graph $G = (V, E)$ as input and asked to find a permutation $\pi : V \to \{1, 2, \ldots, n\}$ that minimizes a certain problem specific objective function of π. In order to define the problems considered we need to introduce some notation. A permutation $\pi : V \to \{1, 2, \ldots, n\}$ orders the vertex set into $v_1 <_\pi v_2 <_\pi \ldots <_\pi v_n$. For every i, the set V_i is $\{v_1, \ldots, v_i\}$ and $\partial(V_i) = \{uv \mid uv \in E, u \in V_i, v \in V \setminus V_i\}$. We define $L_\pi(v)$ to be $\{u \mid u \in N(v), u <_\pi v\}$ and $R_\pi(v)$ is $\{u \mid u \in N(v), v <_\pi u\}$ where $N(v) = \{u\colon uv \in E\}$ is the neighborhood of v. For a pair of vertices u and v, the shortest path distance between u and v is denoted by $D(u, v)$. The precise definitions of the problems studied in the paper are given in Table 1.

Many problems in different domains can be formulated as graph layout problems. These include optimization of networks for parallel computer architectures, VLSI design, numerical analysis, computational biology, graph theory, scheduling

[1] The presented definition is equivalent to the original definition of distortion for embedding into line. Details are given in the section about DISTORTION.

and archaeology. In particular an algorithm for IMBALANCE is used as a starting point for many algorithms in graph drawing [19,20,28,30]. On the other hand BANDWIDTH is equivalent to the problem of minimizing bandwidth of a sparse symmetric square matrix which is useful for the storage and manipulations of these matrices, including Gaussian elimination [5,24]. CUTWIDTH was proposed as a model to minimize the number of channels in a circuit [1,25], and recently it has found applications in protein engineering [3], network reliability [21], automatic graph drawing [26], information retrieval [4], and as a subroutine in the cutting plane algorithm for TSP [17]. The problem of DISTORTION, or rather low distortion embeddings of a graph metric into simple metric spaces has proved to be a useful tool in designing algorithms in various fields. A long list of applications given in [14] includes approximation algorithms for graph and network problems, such as sparsest cut, minimum bandwidth, low-diameter decomposition and optimal group steiner trees, and online algorithms for metrical task systems and file migration problems.

Our Contributions

- We show that CUTWIDTH, BANDWIDTH, IMBALANCE and DISTORTION parameterized by the vertex cover number of the input graph are FPT. Notice that even though a graph G with $vc(G) \leq k$ has treewidth at most k, this can not be directly applied to obtain our results. The reason for this is that graph layout problems parameterized by treewidth have proven hard to cope with. In particular, the parameterized complexity of CUTWIDTH parameterized by treewidth is a non trivial problem left open in [29]. BANDWIDTH is NP-complete for trees and the parameterized complexity of IMBALANCE and DISTORTION with treewidth as parameter is unknown.
- A classical result in parameterized algorithms is that p-VARIABLE INTEGER LINEAR PROGRAMMING FEASIBLITY (p-ILP) is FPT. This powerful result, first proved by Lenstra in [23][2] and later improved by Kannan [18], is very rarely used in parameterized complexity. The only previously known examples of applications of this result in parameterized algorithms is in an FPT algorithm for the CLOSEST STRING problem [13] and in an EPTAS for MIN-MAKESPAN-SCHEDULING problem [2]. In fact, Niedermeier has explicitly asked for more applications of the result that p-ILP is FPT. In this context we quote Niedermeier [[27], Page Number:184]

  > "... it remains to investigate further examples besides CLOSEST STRING where the described ILP approach turns out to be applicable. More generally, it would be interesting to discover more connections between fixed-parameter algorithms and (integer) linear programming. ... "

  We extensively use this result in all our algorithms, thus giving more examples of its applicability.

[2] This paper received Fulkerson Prize in 1985 for an outstanding contribution in the area of discrete mathematics.

We would like to point out that an improved version of the Lenstra/Kannan algorithm for p-ILP designed by Frank and Tardos [12] uses space polynomial in p and input size. We apply this to give a polynomial space FPT algorithm for Bandwidth parameterized by $vc(G)$. This gives an interesting distinction between $vc(G)$ and treewidth parameterizations, because almost all algorithms for graphs of bounded treewidth apply dynamic programming and thus need exponential space.

In Section 2, we give a brief introduction to integer linear programming parameterized by the number of variables. Sections 3, 4, 5 and 6 contain FPT algorithms for Imbalance, Cutwidth, Bandwidth and Distortion respectively. The reader is encouraged to read the section on Imbalance before proceeding to the later sections because this section contains a description of general scheme used in all our algorithms. Finally we conclude with some remarks and open problems in Section 7.

2 Integer Linear Programming with Few Variables

Integer linear programming (ILP) is the framework in which we will eventually formulate all the problems studied. In this section we describe the required results in this direction.

> p-Variable Integer Linear Programming Feasiblity (p-ILP): Given matrices $A \in \mathbb{Z}^{m\times p}$ and $b \in \mathbb{Z}^{m\times 1}$, the question is whether there exists a vector $\bar{x} \in \mathbb{Z}^{p\times 1}$ satisfying the m inequalities, that is, $A \cdot \bar{x} \leq b$. The number of variables p is the parameter.

Lenstra [23] showed that p-ILP is FPT with running time doubly exponential in p. Later, Kannan [18] provided an algorithm for p-ILP running in time $p^{O(p)}$. The algorithm uses Minkowski's Convex Body theorem and other results from Geometry of Numbers. A bottleneck in this algorithm was that it required space exponential in p. Using the method of simultaneous Diophantine approximation, Frank and Tardos [12] describe preprocessing techniques, using which it is shown that Lenstra's and Kannan's algorithms can be made to run in polynomial space. They also slightly improve the running time of the algorithm. For our purposes, we use this algorithm.

Theorem 1 ([18],[23],[12]). p-Variable Integer Linear Programming Feasiblity *can be solved using* $O(p^{2.5p+o(p)} \cdot L)$ *arithmetic operations and space polynomial in* L*. Here* L *is the number of bits in the input.*

Later, a randomized algorithm for p-ILP was provided by Clarkson, we refer to [6] for further details. The result of Lenstra was extended by Khachiyan and Porkolab [22] to semidefinite integer programming. In their work, they show that if Y is a convex set in R^k defined by polynomial inequalities and equations of degree at most $d \geq 2$, with integer coefficients of binary length at most l, then for fixed k, the problem of computing an optimal integral solution y^* to the problem min $\{y_k \mid y(y_1, \ldots, y_l) \in Y \cup \mathrm{Z}^k\}$ admits an FPT algorithm. Their algorithm was further improved by Heinz [16] in the specific case of minimizing a polynomial $\hat{F}$ on the set of integer points described by an inequality system $F_i \leq 0$, $1 \leq i \leq s$ where the F_i's are quasiconvex polynomials in p variables with integer coefficients. This algorithm generalizes Lenstra's algorithm. In our algorithms we need the optimization version of p-ILP rather than the feasibility version. We proceed to define the minimization version of p-ILP.

> p-Variable Integer Linear Programming Optimization (p-Opt-ILP): Let matrices $A \in \mathbb{Z}^{m \times p}$, $b \in \mathbb{Z}^{m \times 1}$ and $c \in \mathbb{Z}^{1 \times p}$ be given. We want to find a vector $\bar{x} \in \mathbb{Z}^{p \times 1}$ that **minimizes** the objective function $c \cdot \bar{x}$ and satisfies the m inequalities, that is, $A \cdot \bar{x} \geq b$. The number of variables p is the parameter.

Now we are ready to state the theorem we will use in the later sections.

Theorem 2. $[\star]^3$ p-Opt-ILP *can be solved using* $O(p^{2.5p+o(p)} \cdot L \cdot \log(MN))$ *arithmetic operations and space polynomial in L. Here, L is the number of bits in the input, N is the maximum of the absolute values any variable can take, and M is an upper bound on the absolute value of the minimum taken by the objective function.*

3 Imbalance: The Inner Order Is Irrelevant

The solutions to all the problems considered in this paper follow the same basic scheme. The case of Imbalance is the simplest exhibition of this theme, and our algorithm for Imbalance will act as a template for the other algorithms to follow. We now proceed to give an FPT algorithm for the Imbalance problem parameterized by the size of the minimum vertex cover of the input graph. Our input consists of a graph $G = (V, E)$, and a vertex cover $C = \{c_1, \ldots, c_k\}$ of size k.

Fixing the order of appearce of vertices in C: We are looking for a permutation $\pi : V \rightarrow \{1, 2, \ldots, n\}$ for which $f_{im}(\pi)$ is minimized. In order to do this, we loop over all possible permutations of the vertex cover C and for each such permutation π_c, find the best permutation π of V that agrees with π_c. We say that π and π_c *agree* if for all $c_i, c_j \in C$ we have that $c_i <_\pi c_j$ if and only of $c_i <_{\pi_c} c_j$. In other words, the relative ordering π imposes on C is precisely π_c. Thus, at a cost of a factor of $k!$ in the running time we can assume that there exists an optimal permutation π such that $c_1 <_\pi c_2 <_\pi \ldots <_\pi c_k$.

Definition 1. *Let π_c be an ordering of C such that $c_1 <_{\pi_c} c_2 <_{\pi_c} \ldots <_{\pi_c} c_k$. We define C_i to be $\{c_1, c_2, \ldots, c_i\}$ for every i such that $1 \leq i \leq k$.*

Types of Vertices: Let I be the independent set $V \setminus C$. We associate a *type* with each vertex in I. A "type" is simply a subset of C.

Definition 2. *Let I be the independent set $V \setminus C$. The type of a vertex v in I is $N(v)$. For a type $S \subseteq C$ the set $I(S)$ is the set of all vertices in I of type S.*

Notice that two vertices of the same type are indistinguishable up to automorphisms of G, and that there are 2^k different types.

[3] Proofs of results marked with $[\star]$ will appear in the long version of the paper.

Inner Order: Observe that every vertex of I is either mapped between two vertices of C, to the left of c_1 or to the right of c_k by a permutation π. For a permutation π we say that a vertex v is at *location* 0 if $v <_\pi c_1$ and at location i if i is the largest integer such that $c_i <_\pi v$. The set of vertices that are at location i is denoted by L_i. We define the *inner order* of π at location i to be the permutation defined by π restricted to L_i.

The task of finding an optimal permutation can be divided into two parts. The first part is to partition the set I into $L_0, \ldots, L_k$, while the second part consists of finding an optimal inner order at all locations. One should notice that partitioning I into $L_0, \ldots, L_k$ amounts to deciding *how many* vertices of each type are at location i for each i. For most layout problems, figuring out the right partitioning turns out to be more difficult than determining the inner orders once the partitioning is known. For Imbalance, this turns out to be particularly true as the inner orders in fact are irrelevant. The reason for this is that permuting the inner order of π at location i does not change the imbalance of any single vertex where the imbalance of a vertex v is $|L_\pi(v) - R_\pi(v)|$. Finding the optimal ordering of the vertices thus reduces to finding the right partition of I into $L_0, \ldots, L_k$. We formalize this as an instance of p-Opt-ILP.

ILP Formulation: For a type S and location i we let x_S^i be a variable that encodes the number of vertices of type S that are at location i. Also, for every vertex c_i in C we have a variable y_i that represents the imbalance of c_i. In order to represent a feasible permutation, all the variables must be non-negative. Also the variables x_S^i must satisfy that for every type S, $\sum_{i=0}^{k} x_S^i = |I(S)|$. For every vertex c_i of the vertex cover let $e_i = \big||N(c_i) \cap C_{i-1}| - |N(c_i) \cap (C \setminus C_i)|\big|$ be a constant. Finally for every $c_i \in C$ we add the constarint $y_i = e_i + \big|\sum_{\{S \subseteq C | c_i \in S\}} \big(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j\big)\big|$.

One should notice that the last set of constraints is not a set of linear constraints. However, we can guess the sign of $y_i' = e_i + \sum_{\{S \subseteq C | c_i \in S\}} \big(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j\big)$ for every i in an optimal solution. This increases the running time by a factor of 2^k. For every i we let t_i take the value 1 if we have guessed that $y_i' \geq 0$ and we let t_i take the value -1 if we have guessed that $y_i' < 0$. We can now replace the non-linear constraints with the linear constraints $y_i = t_i y_i'$ for every i. Finally, for every type S and location i, let z_S^i be the constant $\big||S \cap C_i| - |S \cap (C \setminus C_i)|\big|$. We are now ready to formulate the integer linear program.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k} t_i \cdot y_i + \sum_{S \subseteq C} z_S^i \cdot x_S^i \\
\text{such that} \quad & \sum_{i} x_S^i = |I(S)| && \text{for all } i \in \{0, \ldots, k\}, S \subseteq C \\
& y_i = t_i e_i + \sum_{\{S \subseteq C | c_i \in S\}} \big(\textstyle\sum_{j=0}^{i-1} t_i x_S^j - \sum_{j=i}^{k} t_i x_S^j\big) && \text{for all } i \in \{1, \ldots, k\} \\
& x_S^i,\ y_i \geq 0 && \text{for all } i \in \{0, \ldots, k\}, S \subseteq C
\end{aligned}
$$

Since the value of $f_{im}(\pi)$ is bounded by n^2 and the value of any variable in the integer linear program is bounded by n, Theorem 2 implies that this integer linear program can be solved in FPT time, thus implying the following theorem.

Theorem 3. *The* Imbalance *problem parameterized by the vertex cover number of the input graph is fixed parameter tractable.*

4 Cutwidth: The Inner Order Is Known

In the Cutwidth problem, we are to find the permutation of the vertices of the input graph that minimizes $f_{cw}(\pi)$, the maximum cut in the permutation. We proceed to give an FPT algorithm for minimizing $f_{cw}(\pi)$ in graphs with small vertex covers. The input is a graph $G = (C \cup I, E)$ with C being a vertex cover of size k. We define the *rank* of a vertex v with respect to a vertex set S to be $rank(S, v) = |N(v) \setminus S| - |N(v) \cap S|$. Notice that $|\partial(S \cup v)| = |\partial(S)| + rank(S, v)$.

Just as for the Imbalance problem, we guess the order $c_1 <_{\pi_c} \ldots <_{\pi_c} c_k$ of the vertices in C in an optimal permutation π. We consider the inner order of L_i for some i between 0 and k. Suppose $\pi(c_i) = s$, then, for any t with $s < t \leq s + |L_i|$ we have that $|\partial(V_t)| = |\partial(V_s)| + \sum_{j=s+1}^{t} rank(V_{j-1}, v_j)$. Since the set of vertices in the locations form an independent set, $rank(V_{j-1}, v_j) = rank(C_i, v_j)$ for every j between $s+1$ and t. This gives the equation $|\partial(V_t)| = |\partial(V_s)| + \sum_{j=s+1}^{t} rank(C_i, v_j)$.

Hence if we start with an optimal permutation π and reorganize the inner order at each location i to sort the vertices by rank with respect to C_i in non-decreasing order, we get another optimal ordering with a fixed inner order for each location. In such orderings the largest values of $|\partial(V_i)|$ occur either at $i = \pi(c_j) - 1$ or at $i = \pi(c_j)$ for some j between 1 and k. Since the rank of a vertex $v \in I$ with respect to C_i only depends on i and the type of v, we can use this together with the fact that $|\partial(V_t)| = |\partial(V_s)| + \sum_{j=s+1}^{t} rank(C_i, v_j)$ in order to give an integer linear programming formulation for the Cutwidth problem.

For every type S and location i we introduce a variable x_S^i that tells us the number of vertices of type S that are at location i. For every i between 1 and k we add a variable y_i which encodes $rank(V_{\pi(c_i)-1}, c_i)$ and the constant $e_i = |N(c_i) \cap (C \setminus C_i)| - |N(c_i) \cap C_{i-1}|$. For every type S and location i we also compute the constant e_S^i that indicates the rank of a vertex of type S with respect to C_i. Finally we need a variable c that represents the cutwidth of G. For the constraints, as for the Imbalance problem, we need to make sure the variables x_S^i represent a valid partitioning of I into $L_0, \ldots, L_k$. Finally we need constraints to encode the rank of the vertex cover vertices and the connection between the partitioning of I and the cutwidth c. This yields the following integer linear program:

$$
\begin{array}{lll}
\min & c & \\
\text{such that} & \sum_i x_S^i = |I(S)| & \text{for all } S \subseteq C \\
& y_i = e_i + \sum_{\{S \subseteq C | c_i \in S\}} \left(\sum_{j=i}^{k} x_S^j - \sum_{j=0}^{i-1} x_S^j \right) & \text{for all } i \in \{0, \dots, k\} \\
& c \geq \sum_{j=0}^{i} y_j + \sum_{j=0}^{i-1} \sum_{S \subseteq C} e_S^j \cdot x_S^j & \text{for all } i \in \{1, \dots, k\} \\
& c \geq \sum_{j=0}^{i-1} y_j + \sum_{j=0}^{i-1} \sum_{S \subseteq C} e_S^j \cdot x_S^j & \text{for all } i \in \{1, \dots, k\} \\
& x_S^i \geq 0 & \text{for all } i \in \{0, \dots, k\}, S \subseteq C
\end{array}
$$

Since the value of $f_{cw}(\pi)$ is bounded by n^2 and the value of any variable in the integer linear program is bounded by n^2, Theorem 2 implies that this integer linear program can be solved in FPT time, yielding the following theorem.

Theorem 4. *The* CUTWIDTH *problem parameterized by the minimum vertex cover of the input graph is fixed parameter tractable.*

5 Bandwidth: The Inner Order Is Structured I

In the BANDWIDTH problem the aim is to minimize the function $f_{bw}(\pi) = \max_{uv \in E} |\pi(u) - \pi(v)|$. As for the previous cases we guess the ordering $c_1 <_{\pi_c} \dots <_{\pi_c} c_k$ of the vertices in C in an optimal permutation π. Since we now are looking for the optimal permutation π that agrees with this ordering of the vertices in C, we observe that for a vertex $v \in I$ the only relevant neighbours in C are the leftmost and rightmost neighbour. We can thus delete the edges from v to all other neighbours of v. After this reduction every vertex in I has degree at most 2, and thus the number of different types is bounded by k^2 rather than 2^k.

For BANDWIDTH, we are not able to determine the inner orders a priori, contrary to the situation we had for CUTWIDTH. Instead we will show that there is an optimal permutation where the inner orderings have a specific structure. We say that an interval $[a, b]$ on the integer line is *uniform* if all vertices π maps to $[a, b]$ have the same type. A *zone* is an inclusion maximal uniform interval, and for a layout π of the vertices of G, the *zonal dimension* of π at location i, $\zeta_i(\pi)$, is the number of zones inside $[\pi(c_i) + 1, \pi(c_{i+1}) - 1]$. The zonal dimension of π is $\zeta(\pi) = \max_{i=0}^{k} \zeta_i(\pi)$. Our approach consists of two parts. First we show that there is an ordering π minimizing bandwidth such that $\zeta(\pi) \leq k^2(2k + 1) + 2k$. We then use this to show that BANDWIDTH parameterized by the size of the minimum vertex cover of the input graph is fixed parameter tractable.

Lemma 1. [$\star$] *For a graph $G = (C \cup I, E)$, there is an optimal bandwidth ordering π with $\zeta(\pi) \leq k^2(2k + 1) + 2k$.*

So, how can one use Lemma 1 to give an integer linear program for the BANDWIDTH problem? The trick is to guess the correct values of $\zeta_i(\pi)$ for every i and guess which type of vertices appears in each zone. We can do this at a cost of a factor $(3k^3)^{k+1}(k^2)^{3k^3} = k^{O(k^3)}$ in the running time. Note that the zones are ordered from left to right. We can now set up an integer linear program where the variables encode how many vertices there are in each zone. Let x_i be a variable that encodes the number of vertices in zone number i from the left. For each type $S \subseteq C$ such that $I(S)$ is nonempty, we let $Z(S)$ be the set of integers such that for each $i \in Z(S)$ we have guessed that the vertices in the zone i have type S. Let l_S and r_S be the smallest and largest numbers in $Z(S)$ respectively. Now, for an integer $1 \leq i \leq k$ we let e_i be the number of zones guessed to be to the left of c_i. Finally, for an integer i between 1 and k and a type S we define the constant $t_1(i, S)$ to be the number of vertices from C to the right of zone number l_S and to the left of c_i. Similarly, let $t_2(i, S)$ be the number of vertices from C to the left of zone number r_S and to the right of c_i. Having made the discussed guesses, we can formulate the BANDWIDTH problem as an integer linear program as follows:

$$
\begin{aligned}
&\min \quad b \\
&\text{such that} \sum_{i \in Z(S)} x_i = |I(S)| && \text{for all } S \subseteq C : I(S) \neq \emptyset \\
&\qquad b \geq j - i - 1 + \sum_{q=e_i+1}^{e_j} x_q && \text{for all } c_i c_j \in E \\
&\qquad b \geq t_1(i, S) + \sum_{j=l_S}^{e_i} x_j && \text{for all } i \in \{1, \ldots, k\}, S \subseteq C\colon I(S) \neq \emptyset, c_i \in S \\
&\qquad b \geq t_2(i, S) + \sum_{j=e_{i+1}}^{r_S} x_j && \text{for all } i \in \{1, \ldots, k\}, S \subseteq C\colon I(S) \neq \emptyset, c_i \in S \\
&\qquad x_i \geq 0 && \text{for all } i \in \{0, \ldots, k\}
\end{aligned}
\tag{1}
$$

Because the value of $f_{bw}(\pi)$ is bounded from above by n and the value of any variable in the integer linear program is bounded by n, Theorem 2 implies that this integer linear program can be solved in FPT time, yielding the following theorem.

Theorem 5. [$\star$] *The* BANDWIDTH *problem parameterized by the size k of the minimum vertex cover of the input graph can be solved in time $k^{O(k^3)}n$ and polynomial space.*

6 Distortion: The Inner Order Is Structured II

In this section we consider the parametrized complexity of embedding graph metrics into the real line, parameterized by the size of the minimum vertex cover of the input graph. Given an undirected graph $G = (V, E)$, a natural metric associated with G is $M(G) = (V, D)$ where the distance function D is the shortest path distance between u and v for each pair of vertices $u, v \in V$. Given a graph metric M and another metric space M' (like real line) with distance functions D and D', a mapping $f : M \to M'$ is called an *embedding* of M into M'. The mapping f has *contraction* c_f and *expansion* e_f if for every pair of points p, q in M, $D(p, q) \leq D'(f(p), f(q)) \cdot c_f$ and $D(p, q) \cdot e_f \geq D'(f(p), f(q))$ respectively. A mapping f has *distortion* d if $(e_f \cdot c_f)$ is at most d. We say that f is *non-contracting* if c_f is at most 1. A non-contracting mapping f has *distortion* d if e_f is at most d. As observed by several authors before [9,15], the problem of finding a minimum distortion embedding of a graph metric into the line can be expressed as a problem of finding the permutation $\pi : V \to \{1, 2, \ldots, n\}$ that minimizes $f_{di}(\pi) = \max_{uv \in E} \sum_{i=\pi(u)}^{\pi(v)-1} D(v_i, v_{i+1})$.

Lemma 2 ([9]). *A graph $G = (V, E)$ has a distortion d embedding f into the real line if and only if there is a permutaion $\pi : V \to \{1, 2, \ldots, n\}$ such that $f_{di}(\pi) \leq d$.*

For a permutation π and two vertices u and v such that $u <_\pi v$ we define $D_\pi(u, v) = \sum_{i=\pi(u)}^{\pi(v)-1} D(v_i, v_{i+1})$. If $v <_\pi u$ then $D_\pi(u, v)$ is defined to be $D_\pi(v, u)$. We give a fixed parameter tractable algorithm for the DISTORTION problem parameterized by the size of the minimum vertex cover of the input graph. Our approach is similar to, albeit more involved than, the algorithm presented for the BANDWIDTH problem. As for the previous problems, we iterate over all $k!$ ways to order the vertices of C into $c_1 <_{\pi_c} \ldots <_{\pi_c} c_k$. We proceed to show that there is an optimal permutation π such that $\zeta(\pi) \leq (4k + 1)2^{2^k}$.

Lemma 3. [$\star$] *For a graph $G = (C \cup I, E)$, there is an optimal distortion ordering π with $\zeta(\pi) \leq (4k + 1)2^{2^k}$.*

Using Lemma 3 we can give an algorithm for the DISTORTION problem similar to the algorithm for BANDWIDTH. The algorithm proceeds excactly as for BANDWIDTH with the only differences being that the zonal dimension is much larger, and that one has to be careful to introduce constants that encode the distance between two consecutive vertices in the ILP. Notice that since the zonal dimension is not polynomial in k for the DISTORTION problem, we do not obtain a polynomial space algorithm.

Theorem 6. *The* DISTORTION *problem parameterized by the minimum vertex cover of the input graph is fixed parameter tractable.*

7 Conclusion and Discussions

In this paper we considered parameterization by vertex cover number of the graph, a structural parameter stronger than the treewidth. This enabled us to show that graph layout problems CUTWIDTH, BANDWIDTH, IMBALANCE and DISTORTION are FPT parameterized by vertex cover number of the graph. This is in contrast to the parameterization by treewidth for which the paramterized complexity of these problems is open. The structural parameterization of vertex cover number also brought forward the technique of bounded variable integer linear programming to importance. We believe that this (underused) powerful result will become one of the basic tools in classifying whether a problem is FPT, as well as in designing practical algorithms, because p-ILP is well solved for p up to 1000.

One may wonder whether there exists a problem which is not FPT for graphs of bounded vertex cover number. This in indeed the case, as List Coloring remains $W[1]$-hard even for graphs of bounded vertex cover number. An important graph layout problem is Optimal Linear Arrangement where the objective is to minimize the sum of $|\partial V_i|$. We can show that this problem is in XP by giving an algorithm of time complexity $n^{f(k)}$ when parameterized by the vertex cover number of the input graph. The main difficulty we face in encoding this problem as ILP is that the objective function is not linear, but quadratic. Hence in this direction the following questions still remain unanswered.

– Is Optimal Linear Arrangement FPT parameterized by the vertex cover number of the input graph?
– Is Cutwidth FPT parameterized by the treewidth of the input graph?

References

1. Adolphson, D., Hu, T.C.: Optimal linear ordering. SIAM J. Appl. Math. 25, 403–423 (1973)
2. Alon, N., Azar, Y., Woeginger, G.J., Yadid, T.: Approximation schemes for scheduling on parallel machines. J. of Scheduling 1, 55–66 (1998)
3. Blin, G., Fertin, G., Hermelin, D., Vialette, S.: Fixed-parameter algorithms for protein similarity search under RNA structure constraints. In: Kratsch, D. (ed.) WG 2005. LNCS, vol. 3787, pp. 271–282. Springer, Heidelberg (2005)
4. Botafogo, R.A.: Cluster analysis for hypertext systems. In: Proceedings of SIGIR 1993, pp. 116–125. ACM Press, New York (1993)
5. Chinn, P., Chvatalova, J., Dewdney, A., Gibbs, N.: The bandwidth problem for graphs and matrices – a survey. Journal of Graph Theory 6, 223–254 (1982)
6. Clarkson, K.L.: Las Vegas Algorithms for Linear and Integer Programming When the Dimension is Small. Journal of the Association for Computing Machinery 42(2), 488–499 (1995)
7. Courcelle, B.: The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs Inf. Comput. 85(1), 12–75 (1990)
8. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1999)
9. Fellows, M.R., Fomin, F.V., Lokshtanov, D., Losievskaja, E., Rosamond, F.A., Saurabh, S.: Parameterized Low-distortion Embeddings - Graph metrics into lines and trees, CoRR abs/0804.3028 (2008)
10. Fellows, M.R., Rosamond, F.A.: The Complexity Ecology of Parameters: An Illustration Using Bounded Max Leaf Number. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, pp. 268–277. Springer, Heidelberg (2007)
11. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)

12. Frank, A., Tardos, E.: An Application of Simultaneous Diophantine Approximation in Combinatorial Optimization. Combinatorica 7, 49–65 (1987)
13. Gramm, J., Niedermeier, R., Rossmanith, P.: Fixed-Parameter Algorithms for CLOSEST STRING and Related Problems. Algorithmica 37(1), 25–42 (2003)
14. Gupta, A., Newman, I., Rabinovich, Y., Sinclair, A.: Cuts, trees and l_1-embeddings of graphs. Combinatorica 24(2), 233–269 (2004)
15. Heggernes, P., Meister, D., Proskurowski, A.: Minimum distortion embeddings into a path of bipartite permutation and threshold graphs. In: Gudmundsson, J. (ed.) SWAT 2008. LNCS, vol. 5124, pp. 331–342. Springer, Heidelberg (2008)
16. Heinz, S.: Complexity of integer quasiconvex polynomial optimization. Journal of Complexity 21, 543–556 (2005)
17. Junguer, M., Reinelt, G., Rinaldi, G.: The traveling salesman problem. In: Handbook on Operations Research and Management Sciences, vol. 7, pp. 225–330. North-Holland, Amsterdam (1995)
18. Kannan, R.: Minkowski's Convex Body Theorem and Integer Programming. Mathematics of Operations Research 12, 415–440 (1987)
19. Kant, G.: Drawing planar graphs using the canonical ordering. Algorithmica 16(1), 4–32 (1996)
20. Kant, G., He, X.: Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. Theoret. Comput. Sci. 172(1-2), 175–193 (1997)
21. Karger, D.R.: A randomized fully polynomial approximation scheme for all terminal network reliability problem. In: Proceedings of STOC, pp. 11–17. ACM Press, New York (1996)
22. Khachiyan, L., Porkolab, L.: Integer Optimization on Convex Semialgebraic Sets. Discrte Computational Geometry 23, 207–224 (2000)
23. Lenstra, H.W.: Integer Programming with a Fixed Number of Variables. Mathematics of Operations Research 8, 538–548 (1983)
24. Linial, N., London, E., Rabinovich, Y.: The Geometry of Graphs and Some of its Algorithmic Applications. Combinatorica 15(2), 215–245 (1995)
25. Makedon, F., Sudborough, I.H.: Minimizing width in linear layouts. In: ICALP 1983. LNCS, vol. 154, pp. 478–490. Springer, Heidelberg (1983)
26. Mutzel, P.: A polyhedral approach to planar augmentation and related problems. In: Spirakis, P.G. (ed.) ESA 1995. LNCS, vol. 979, pp. 497–507. Springer, Heidelberg (1995)
27. Niedermeier, R.: Invitation to fixed-parameter algorithms. Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)
28. Papakostas, A., Tollis, I.G.: Algorithms for area-efficient orthogonal drawings. Computational Geometry 9, 83–110 (1998)
29. Thilikos, D.M., Serna, M.J., Bodlaender, H.L.: Cutwidth II: Algorithms for partial w-trees of bounded degree. J. Algorithms 56(1), 25–49 (2005)
30. Wood, D.R.: Optimal three-dimensional orthogonal graph drawing in the general position model. Theoret. Comput. Sci. 299(1-3), 151–178 (2003)