Approximating Bandwidth

Martin Pei

April 22, 2005

1 Introduction

Given a graph $G = (V, E)$ on n vertices, a *vertex ordering* (or *layout*) is a bijection $f : V \to \{1, 2, \ldots, n\}$. In the *bandwidth minimization* problem, the goal is to find a vertex ordering f of G such that its bandwidth

$$B(G, f) = \max_{uv \in E} |f(u) - f(v)|$$

is minimized. We denote $B(G) = \min_f B(G, f)$ to be the bandwidth of G.

This is equivalent to the *matrix bandwidth minimization* problem, which is the form it was introduced in the 1960s. In the matrix version of the problem, a symmetric matrix M is given, and the goal is to find a permuation matrix P such that the nonzero entries of PMP^T are all within a "band of minimum width" about the diagonal. An application of this is to reduce the amount of space required to store the matrix, and simplify some matrix operations such as the LU factorization. In the graph version, if we imagine the vertex ordering as a layout on a line, then the problem is to make the longest edge as short as possible. This has many engineering applications such as VLSI, where if the edges are wires, the problem becomes finding a layout such that the length of the longest wire is minimized, which is a factor in transmission delays.

In terms of complexity, Papadimitriou [14] showed that the bandwidth minimization problem is NP-hard. Garey *et al.* [8] showed that the problem remains NP-hard even for trees with maximum degree 3. So this is one of a few graph theoretical problems that is hard even for trees. Subsequent investigations into this problem had been mainly focused on subclasses of graphs with specific structures, examples include caterpillars, chordal graphs, and even one on asteroidal triple-free claw-free graphs.

The class of (generalized) caterpillars seems to be of some interest in the literature. A caterpillar is a graph that consists of a path P called the *backbone*, and several paths (called *hairs* or *legs*) that are attached (on one end) to vertices of the backbone. Clearly, caterpillars are trees. Assman *et al.* [1] gave a polynomial time algorithms for finding minimum bandwidth of caterpillars of hair length at most 2. However, that is the best that we can do for exact algorithms, since Monien [13] showed that for caterpillars of hair length at most 3, the problem is NP-hard.

Now that you are convinced about the super hardness of this problem, we can probably be happy with approximations. In the next section, we will focus on a few techniques in designing approximations algorithms as seen in the literature.

2 Approximation Algorithms in the Literature

2.1 Lower Bounds

Before we talk about approximation algorithms for the bandwidth problem, it is useful to see some lower bounds for it. One lower bound that I came up with when I first pondered about this problem is the following:

Lemma 1. *Let $\Delta(G)$ be the maximum degree of G. Then $B(G) \geq \lceil \Delta(G)/2 \rceil$.*

One can easily see why this is true: Take a vertex v of maximum degree and consider the differences between the label of v with its neighbours. Each "difference value" can appear at most twice, so the maximum difference has to be at least $\Delta(G)/2$.

This lower bound is apparently too easy, and one quick glance at the literature (e.g. [3]) will give a generalization of this bound:

Lemma 2. *Let $N(v, d)$ be the number of vertices that are at most d away from v. Then*

$$B(G) \geq \max_{v \in V} \max_{d} \lceil N(v, d)/2d \rceil.$$

The value on the RHS of the inequality above is called the *local density* of the graph, and it is widely used to measure the performance of an approximation algorithm.

There are other measures for lower bounds. Harper [12] gave a lower bound of

$$B(G) \geq \max_{k} \min_{|S|=k} |\partial(S)|$$

where $\partial(S)$ is the set of vertices in S that is adjacent to some vertex in $V - S$. Later, Chvátalová [4] and others improved on this bound. However, these measures are more difficult to compute than the local density bound.

Dewdney [5] showed some other interesting lower bounds, but they are difficult to work with as well. He showed that $B(G) \geq \chi(G) - 1$ and $B(G) \geq \kappa(G)$, where $\chi(G)$ and $\kappa(G)$ are the chromatic number and the connectivity of G, respectively.

2.2 Approximation Techniques

The first group of algorithms that we are going to mention are heuristics from the late 1960s and the 1970s. A perhaps well-known algorithm was introduced by Cuthill and McKee [6], and this was later improved by Gibbs, Poole and Stockmeyer [9]. These algorithms are based on the idea of *level structures* of the graph. A level structure L is a partition of the vertices of the graph into *levels* $L_1, L_2, \ldots, L_k$ such that vertices adjacent to a vertex in level L_i ($2 \leq i \leq k-1$) must be in L_{i-1}, L_i, or L_{i+1}. Having generated such a structure, a vertex ordering f can be obtained by numbering level by level, i.e. the first $|L_1|$ numbers go to vertices in L_1, the next $|L_2|$ numbers go to vertices in L_2 etc. One can easily see that such an ordering satisfies

$$B(G) \leq B(G, f) \leq 2w(L) - 1,$$

where $w(L)$ is the size of the largest level in L (the *width* of L). The bad news is that this numbering will always satisfy

$$B(G, f) \geq w(L).$$

The algorithms mentioned above both tried to minimize the maximum width of the levels, and one can perhaps achieve this by increasing the number of levels (the *depth*) of the structure. They also number the vertices more carefully within each level. The result is that both algorithms (especially the one by Gibbs, Poole and Stockmeyer) work very well in practice, and they are quite efficient with an average-time complexity of $O(n^{1.2})$ for the algorithm by Gibbs *et al.* However, from a theoretical point of view, these are not so useful as there are no proofs that these algorithms will output an ordering whose bandwidth is within a certain factor from the optimal.

In a 1991 paper, Haralambides, Makedon and Monien [11] took this idea of level structures and applied it to obtain an approximation algorithm for caterpillars. Their idea is to take the backbone of the caterpillar and create a level for each vertex. Then for each leg of the caterpillar, "fold" it (in a fixed direction) so that starting from the vertex in the backbone it is attached to, the vertices in the leg are placed so that consecutive vertices in the leg are in consecutive levels of the structure. (Of course this is only a vague description of the algorithm. The actual algorithm is slightly more complicated.) The good news is that the authors gave an analysis that showed this algorithm will always produce a vertex ordering that is no more than $\log n$ times the minimum bandwidth. This is (as far as I know) the first paper that gives a concrete bound in the bandwidth minimization problem for any classes of graphs.

In 1998, Feige [7] introduced a new technique for approximating bandwidth in general graphs. His algorithm involves two steps. The first step is to find a *volume respecting embedding* of the graph in a normed space. This is an extension of ordinary embeddings in the sense that instead of considering the distance between two vertices, the volume between any subset of k vertices is considered. After finding such an embedding that has small distortions of volume, the next step is to project it onto a random line and use the ordering on the line as the ordering for the bandwidth problem. Feige proved that this is a $O(\log^{3.5} n \sqrt{\log \log n})$-approximation of the bandwidth problem for general graphs. Later, Gupta [10] gave a similar algorithm that improved this to a $O(\log^{2.5} n)$-approximation for trees.

2.3 Inapproximability

We end this section with a few words on the inapproximability of the bandwidth problem. Blache, Karpinski and Wirtgen [2] first proved that there is no PTAS for the bandwidth problem unless P=NP, not even for trees. In particular, they reduced the 3SAT problem to a tree in the bandwidth problem to show that no polynomial time approximation algorithm can give a approximation ratio better than 4/3 for trees, and 1.5 for general graphs. This is improved by Unger [15] who showed that there is no polynomial time algorithm that has an approximation ratio of k for any constant k. In fact, this result holds even for trees of maximum degree 3. So it looks like the $\log n$-approximation for caterpillars mentioned above is pretty good.

3 Integer Programming Formulation

The bulk of my work on the bandwidth problem is in creating an integer programming formulation for the problem. I have not seen any such work in the literature, so I thought I would try my hands on it. (Then again, this probably just means that people have tried it and were unsuccessful, so I was pretty pessimistic to begin with.) It is easy to characterize the "edge difference" property in an integer program. The main obstacle is to somehow characterize a vertex ordering in terms of actual numbers. For this part, I turned to L. Szegő's integer programming course this semester (of which I'm a TA, fortunately) and borrowed an assignment question from it. We will give the formulation first, and then explain why it works. First, let $\bar{0}$ be a new vertex, and let $V' = V \cup \{\bar{0}\}$. Let the variable b represent the bandwidth of the graph. Let w_u represent the label given to vertex u. Let x_{uv} be some 0,1-variable, to be explained later.

$$\min b$$

$$|w_u - w_v| \leq b \qquad \forall uv \in E \tag{1}$$

$$\sum_{v \in V'-u} x_{uv} = 1 \qquad \forall u \in V' \tag{2}$$

$$\sum_{v \in V'-u} x_{vu} = 1 \qquad \forall u \in V' \tag{3}$$

$$w_u - w_v + n x_{uv} \leq n - 1 \qquad \forall uv \in V \times V, u \neq v \tag{4}$$

$$x_{uv} \in \{0, 1\}$$

$$w_i \in \mathbb{Z}$$

Equation (1) ensures that the bandwidth b is at least as large as the edge differences. Equations (2) to (4) characterizes the feasible solutions to the asymmetric traveling salesman problem for the *complete graph* on the vertices V', where $x_{uv} = 1$ if uv is in the (directed) tour, 0 otherwise. In particular, (2) and (3) ensure that the solution is a collection of tours, and (4) eliminates any subtour that does not contain the vertex $\bar{0}$ (one can show this by summing this inequality over all edges in such a subtour to obtain a contradiction $nk \leq (n-1)k$ for some k). We want the w_u's to represent the labels of V in the tour (e.g. first vertex reached from $\bar{0}$ gets labelled 1, next vertex gets labelled 2 etc.). Notice that in (4), if $x_{uv} = 1$, then $w_u - w_v \leq -1$, which means that $w_v \geq w_u + 1$. This says that if uv is an arc in this tour, then the label of v must be at least one larger than the label of u, which is sort of what we want. In particular, the objective function ensures that equality holds in this case. Now the labels w_u may not be $\{1, 2, \ldots, n\}$, but we know for sure that they are some consecutive n integers, and that does not change the problem.

Now that you are (sort of) convinced of the correctness of this integer program, let's make some changes to it. First, (1) is really two inequalities,

$$\begin{aligned} w_u - w_v &\leq b \\ -w_u + w_v &\leq b. \end{aligned}$$

To make my life slightly simpler, we define $A = \{uv, vu : uv \in E\}$ to be the set of directed arcs of E (each edge is replaced with two arcs, one in each direction). Then these constraints reduce to

$$w_u - w_v \leq b \quad \forall uv \in A.$$

In order to take the dual, we turn the inequalities around and put the variables to the LHS. Now we relax the variables x and w to get the following linear program:

$$\begin{aligned} & \min b & & \\ w_u - w_v + b &\geq 0 & \forall uv &\in A & (5) \\ \sum_{v \in V'-u} x_{uv} &= 1 & \forall u &\in V' & (6) \\ \sum_{v \in V'-u} x_{vu} &= 1 & \forall u &\in V' & (7) \\ -w_u + w_v - nx_{uv} &\geq 1 - n & \forall uv &\in V \times V, u \neq v & (8) \\ x_{uv} &\geq 0 & & \end{aligned}$$

(Notice that $x_{uv} \leq 1$ is implied by (6) and (7).) Now we can take the dual. Let y_{uv} be the dual variables for (5), z_{u^+} for (6), z_{u^-} for (7), and α_{uv} for (8) (obviously running out of letters). Then if I didn't make any mistake, here is the dual:

$$\begin{aligned} \max \sum_{uv \in V \times V, u \neq v} (1-n)\alpha_{uv} + \sum_{u \in V'} (z_{u^+} + z_{u^-}) & & \\ \sum_{uv \in A} y_{uv} = 1 & & (9) \\ \sum_{uv \in A} y_{uv} - \sum_{vu \in A} y_{vu} + \sum_{v \in V-u} (-\alpha_{uv} + \alpha_{vu}) = 0 & \quad \forall u \in V & \\ z_{u^+} + z_{v^-} - n\alpha_{uv} \leq 0 & \quad \forall uv \in V \times V, u \neq v & \\ z_{u^+} + z_{v^-} \leq 0 & \quad \forall uv \in V' \times V', u = \bar{0} \text{ or } v = \bar{0} & \\ y_{uv} \geq 0, \quad \alpha_{uv} \geq 0 & & \end{aligned}$$

Ok, so that is quite horrific, and this is pretty much where I get stuck. Notice that there are only a polynomial number of constraints and variables, so one could presumably solve these linear programs in polynomial time. And once an optimal solution is obtained, one can presumably use (9) to obtain a probability distribution among the y_{uv}'s and perform some randomized rounding. One observation that I can make is that by complimentary slackness, if $y_{uv} > 0$, then it implies uv is a tight edge in the primal, which means uv realizes the bandwidth b. Not sure how that would help, though.

4 Acyclic Orientations and Other Failed Attempts

In this section, I will record some random musings on other approaches to this problem that have raced through my brain in the past. I couldn't do much with any of them, unfortunately. The first idea is to relate the bandwidth problem to acyclic orientations. Given a vertex ordering, one can obtain an acyclic orientation by directing each edge towards the endpoint that has a larger label. On the other hand, given any acyclic orientation (which is also a partially ordered set), one can obtain an ordering by taking any linear extension of the poset. So perhaps some results in the theory of poset can be applied to bandwidth, but no luck so far for me. Also, there is a theorem saying that a graph has chromatic number k if and only if there exists an acyclic orientation whose longest path has length $k-1$. So maybe there really is a serious connection with graph colourings as well.

I have also pondered the possibility of a local search. My idea for a local search would start with an arbitrary ordering, look at the longest edge, and find possible candidates that could switch labels with endpoints of that edge such that the bandwidth can be reduced. The problem is, however, that exchanging labels could create an unpredictable mess. It's hard to imagine a consistent criteria that would efficiently switch labels without creating long edges elsewhere.

5 Final Remarks

It is a bit unfortunate that the survey portion of this paper is longer than the creative portion. The reason being, bandwidth minimization is indeed a difficult problem, and it has a really long history. A lot of people have worked on it, and recent techniques are quite difficult to understand. Hence my inability to do much "playing" with it. Also, procrastination doesn't help...

References

[1] Assman, S.F., Peck, G.W., Syslo, M.M. and Zak, J., The bandwidth of caterpillars with hairs of length 1 and 2, *SIAM J. Algebraic Discrete Mathods* (1981), 387–391.

[2] Blache, G., Karpinski, M. and Wirtgen, J., On approximation intractability of the bandwidth problem, *Electronic Colloquium on Computational Complexity*, Report No. 14 (1998).

[3] Chinn, P.Z., Chvátalová, J., Dewdney, A.K. and Gibbs, N.E., The bandwidth problem for graphs and matrices–a survey, *Journal of Graph Theory* 6(1982), 223–254.

[4] Chvátalová, J., On the bandwidth problem for graphs, Ph.D. dissertation, Department of Combinatorics and Optimization, University of Waterloo (1980).

[5] Chvátalová, J., Dewdney, A.K., Gibbs, N.E. and Korphage, R.R., The bandwidth problem for graphs: a collection of recent results, Research Report #24, Department of Computer Science, UWO, London, Ontario (1975).

[6] Cuthill, E. and McKee, J.M., Reducing the bandwidth of sparse symmetric matrices, *Proc. 24th Nat Conf, Assocn Comp Mach, ACM Pub. P69* 1969, 157–172.

[7] Feige, U., Approximating the bandwidth via volume respecting embeddings, *Journal of Computer and System Sciences* 60(2000), 510–539.

[8] Garey, M.R., Graham, R.L., Johnson, D.S. and Knuth, D.E., Complexity results for bandwidth minimization, *SIAM Journal of Applied Mathematics* 34(1978), 477–495.

[9] Gibbs, N.E., Poole Jr., W.G. and Stockmeyer, P.K., An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM J. Numerical Analysis* 13(1976), 236–250.

[10] Gupta, A., Improved bandwidth approximation for trees and chordal graphs, *Journal of Algorithms* 40(2001), 24–36.

[11] Haralambides, J., Makedon, F. and Monien, B., Bandwidth minimization: an approximation algorithm for caterpillars, *Mathematical Systems Theory* 24(1991), 169-177.

[12] Harper, L.H., Optimal numberings and isoperimetric problems on graphs, *Journal of Combinatorial Theory* 1(1966), 385-393.

[13] Monien, B., The bandwidth-minimization problem for caterpillars with hairlength 3 is NP complete, *SIAM J. Algebraic Discrete Methods* 7(1986), 505–512.

[14] Papadimitriou, C.H., The NP-completeness of the bandwidth minimization problem, *Journal of Computing* 16(1976), 263–270.

[15] Unger, W., The complexity of the approximation of the bandwidth problem, *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science* (1998), 82–91.