DumbAndDumber (Students)

Kristoffer Sandvang
Student at the University of Copenhagen, Denmark

Mateusz Filipowski
Student at the University of Copenhagen, Denmark

Abstract

This report addresses the one-sided crossing minimization problem (OCMP) in two-layered bipartite graphs in the PACE 2024 challenge. The OCMP has been proven to be NP-hard, even when one level of vertices is fixed. We have developed a heuristic hybrid algorithm comprised of the median and barycenter heuristic and its tie-breaking methods.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases PACE2024, Crossing minimization, two-layered bipartite graphs, One-Sided Crossing Minimization in Two-Layered Bipartite Graphs

Digital Object Identifier 10.5281/zenodo.11520511

Supplementary Material `https://github.com/KristofferSandvang/PACE2024` (Source code), `https://github.com/KristofferSandvang/PACE2024_Bachelor` (Bachelor thesis)

Acknowledgements We want to thank our Bachelor thesis supervisor Mikkel Vind Abrahamsen for his support throughout this process.

1 Introduction

This year's PACE 2024 challenge involves the one-sided crossing minimization problem (OCMP) [1]. The bipartite graph, $G = (V_a \cup V_b, E)$ is drawn such that all vertices in G are on two horizontal lines, where the vertices in V_a are on the upper level, and the vertices in V_b are on the lower level. V_a is kept fixed, thus we only have to compute the optimal order of V_b to minimize the total number of edge crossings in G. The goal of the challenge is to compute a permutation of the non-fixed layer that produces the fewest crossings. We have entered the heuristic track. The minimization of the number of edge crossings in two-layered bipartite graphs is one of the most widely studied problems in the field of graph drawing [4], where several heuristic algorithms for this problem already exist, such as the barycenter and median heuristic [3].

2 Barycenter Heuristic

A common heuristic to reduce the number of crossings in a bipartite graph is the barycenter heuristic. The method will choose the x-coordinate of each vertex in the V_b of the bipartite graph, as the average of the x-coordinates of its neighbors. The average x-coordinate is referred to as the barycenter value [3]. The barycenter heuyistic has a lower bound of $\Omega(\sqrt{n})$, which was proven in [2]. The barycenter heuristic can easily be implemented in $O(|E| + |V_B| log |V_B|)$ time.

3 Median Heuristic

Another common approach to the one-sided crossing minimization problem in a bipartite graph is the median heuristic. Similar to the barycenter heuristic, in the median heuristic an x-coordinate is chosen for each vertex $v \in V_B$ to be a median of the x-coordinates of its neighbors. Let N_v be the neighboring vertices of v and if $N_v = w_1, w_2, w_3, \ldots, w_i$ with $x_0(w_1) \leq x_0(w_2) \leq x_0(w_3) \leq \cdots \leq x_0(w_i)$, then we define $med(v) = x_0(w_{\lfloor i/2 \rfloor})$. If N_v is empty, then $med(v) = 0$. It separates two vertices with the same median, by the degree of the vertices. If one vertex has an odd degree and the other even, the odd degree vertex will have a lower x-value than the even degree vertex [3]. The median heuristic has been proved to have a 3-approximation [2]. The median heuristic can easily be implemented in $O(|E| + |V_B|log|V_B|)$ time.

4 Tie-breaking

In the PACE 2024 competition, we must decide the optimal order of the vertices. We have therefore tested multiple tie-breaking methods to handle the situation that arises when two vertices have the same barycenter or median value and which case handles them best. Various tie-breaking methods have already been developed and thoroughly tested [4]. We could have used these results, however, we have decided to implement and test the most promising tie-breaking methods on the provided public test graphs as these sparse graphs would not deviate from the graphs used in the competition.

The first tie-breaking method, `the permutations method`, handles these cases by iterating through all possible permutations of the vertices with the same barycenter or median values. The permutations method then compares the number of crossings each permutation would produce. However, this would mean a significant increase in the time complexity of both heuristics.

The second tie-breaking method is a combination of both algorithms `BarycenterMed` and `MedianBary`, where vertices that have the same barycenter value produced by the barycenter heuristic are evaluated and reordered by applying the median heuristic and vertices that have the same median value produced by the median heuristic are evaluated and reordered by applying the barycenter heuristic [4].

The third and last approach is a `Barycenter and Median heuristic`, `BarycenterRev` and `MedianRev`, where we reverse the order of the vertices that have the same barycenter or median values [4].

5 Our solver

To determine what heuristic to use when, we tested all our variants of the barycenter and median heuristic on the test graphs provided by [1]. The results of which are shown in table 1.

We have decided to have our solver utilize the time limit constraint of five minutes in the competition, and thus our solver uses a priority queue. Before we dequeue a heuristic from the queue we ensure that there is a minimum of 20 seconds remaining, to ensure it does not exceed the time limit. The priority of the heuristics is as shown below:

1. Median - Permutations
2. Barycenter - Permutations
3. Median - Barycenter
4. Barycenter

5. Barycenter median
6. Median

Heuristic	Victories
Barycenter variants	99
Barycenter	40
Permutations	40
BarycenterMed	13
BarycenterRev	6
Median variants	66
Permutations	35
MedianBary	23
Median	7
MedianRev	1

(a) Overall

Heuristic	Victories
Barycenter variants	50
Permutations	26
Barycenter	12
BarycenterMed	8
BarycenterRev	4
Median variants	48
Permutations	28
MedianBary	14
Median	5
MedianRev	1

(b) Public test set

Table 1 The combined table of victories achieved from different median and barycenter variants across all the test sets(a) and the public set(b). A victory is achieved when the heuristic produces a permutation of V_b that results in the lowest amount of edge crossings compared to the other heuristics.

However, this priority can change based on the graph's properties. We identified two limits for when we could use the permutation tie-breaking method based on our testing, and shown in a. Thus if the graph exceeds these limits, $|V_a| + |V_b| \geq 13500$ or $|E| \geq 13000$, we do not enqueue the permutation tie-breaking methods for both the median and barycenter heuristics. Furthermore, if the density of the graph, ρ, $0.003 < \rho < 0.0048$ then the barycenter heuristics have a higher priority than the median heuristics. Both of these thresholds are explained in depth in section 8 in our bachelor thesis. Once we dequeue a heuristic, it is then used to compute a permutation, π_{V_b} of V_b, and count the crossings produced by this permutation. The number of crossings and π_{V_b} is stored as a pair in an array. Ultimately, when the time limit is reached or the queue is empty, the permutation that produces the fewest crossings is chosen and π_{V_b} is returned.

References

1 PACE 2024. `https://pacechallenge.org/2024/`, 2023. Accessed on 13th of may 2024.

2 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1st edition, 1998.

3 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, April 1994. doi:10.1007/BF01187020.

4 Timo Poranen and Erkki Mäkinen. Tie-breaking heuristics for the barycenter and median algorithms. 04 2006.