A List Heuristic for Vertex Cover

David Avis
School of Computer Science, McGill University *

Tomokazu Imamura
Graduate School of Informatics, Kyoto University

Abstract

We analyze a list heuristic for the vertex cover problem that handles the vertices in a given static order based on the degree sequence. We prove an approximation ratio of at most $\sqrt{\Delta}/2 + 3/2$ for a non-increasing degree sequence, and show that no ordering can achieve an approximation ratio of less than $\sqrt{\Delta}/2$.

Keywords: vertex cover, approximation algorithm, list heuristic

1 Introduction

Let $G = (V, E)$ be an undirected graph. The *minimum vertex cover problem* for an undirected graph G is to find the smallest cardinality subset of vertices such that every edge is incident to at least one of the chosen vertices. This is a well known NP-hard optimization problem, and thus no polynomial time algorithm is known. There are a number of approximation algorithms that have been proposed for this problem with various performance guarantees. These are described in various textbooks, see for example Cormen et al. [1] and Vazirani [2]. The best known performance bound on the approximation ratio is two, which is obtained by the matching heuristic (choose all vertices in any maximal matching) and by rounding the linear programming relaxation. Another well known approximation algorithm for vertex cover is the *greedy* algorithm. It repeatedly selects a vertex adjacent to the largest number of uncovered edges, and achieves an approximation ratio of $H(\Delta) = 1 + 1/2 + ... + 1/\Delta = O(\log \Delta)$, where Δ is the maximum degree in

*Corresponding author: 3480 University, Montreal, Quebec, Canada H3A 2A7, avis@cs.mcgill.ca

G. Note that the degrees of the unselected vertices are dynamically updated at each step.

In this paper we consider approximation algorithms based on a static ordering of the vertices determined by their degrees. We call such approximation algorithms *list heuristics*, in analogy with similar heuristics for machine scheduling. In particular, the list decreasing heuristic chooses vertices in order of decreasing degree, selecting a vertex if it is adjacent to an uncovered edge. Note the vertex degrees are *not* updated.

One would expect that the list-decreasing heuristic performs worse than the greedy algorithm, since less information is available at each step, and in fact this is the case. In this paper we will give an analysis for the list-decreasing heuristic and show that its approximation ratio is at most $\frac{\sqrt{\Delta}}{2} + \frac{3}{2}$. We will also give examples that achieve this bound, up to the final constant. These examples also apply to list heuristics based on other orderings of the degree sequence, showing that within this class of heuristics, the best performance guarantee is obtained by selecting the vertices in decreasing order by degree.

2 Preliminaries

Let $G = (V, E)$ be a simple undirected graph. A set of vertices $C \subseteq V$ is called a *vertex cover* if for any edge in E at least one of its endpoints is contained in C. Let n represent the number of vertices, m the number of edges, and Δ the maximum degree of G. The degree of vertex v is denoted by d_v, and the set of edges incident to v is denoted by $\delta(v)$.

The minimum vertex cover problem is the optimization problem of finding a minimum cardinality vertex cover for a given graph. An approximation algorithm is a polynomial-time algorithm that finds an approximate solution for an optimization problem. The performance of an approximation algorithm is usually measured by its approximation ratio. For a minimization problem, the approximation ratio is defined as the maximum ratio of the size of the solution found by the approximation algorithm and the size of the optimal solution, where the maximum is taken over all inputs.

An important tool in the analysis of approximation algorithms is a linear programming relaxation of the related integer programming problem, and was first used by Lovász [3]. Given a graph $G = (V, E)$, the following linear program is an LP-relaxation for the vertex cover problem.

$$
\begin{aligned}
\min \quad & \sum_{v \in V} x_v & (1)\\
\text{s.t.} \quad & x_v + x_u \geq 1 \quad \forall (u,v) \in E \\
& x_v \geq 0 \quad \forall v \in V
\end{aligned}
$$

For any vertex cover C of G, we can naturally define $\mathbf{x} = \{x_v\}_{v \in V}$ as follows.

$$
x_v = \begin{cases} 1 & v \in C \\ 0 & v \notin C \end{cases}
$$

$\mathbf{x}$ is a 0-1 valued feasible solution for linear program (1), and conversely every 0-1 feasible solution corresponds to a vertex cover for G. We define the size of $\mathbf{x}$ as $|\mathbf{x}| = \sum_{v \in V} x_v$, and clearly $|C| = |\mathbf{x}|$. A fractional valued solution corresponds to a fractional vertex cover: nonnegative weights assigned to vertices such that the sum of the weights on each edge is at least one. It follows that the optimum solution of the LP-relaxation is a lower bound on the size of the vertex cover.

The dual of (1) can be formulated as follows.

$$
\begin{aligned}
\max \quad & \sum_{e \in E} y_e & (2)\\
\text{s.t.} \quad & \sum_{e \in \delta(v)} y_e \leq 1 \quad \forall v \in V \\
& y_e \geq 0 \quad \forall e \in E
\end{aligned}
$$

Any $\mathbf{y} = \{y_e\}_{e \in E}$ that satisfies the conditions above is called a *fractional matching* of G. This is an assignment of non-negative weights to the edges of G such that the sum of the edge weights at any vertex is at most one. A matching in G thus corresponds to a 0-1 solution of the dual. The size of a fractional matching $\mathbf{y}$, denoted by $|\mathbf{y}|$, is defined as $\sum_{e \in E} y_e$. By weak LP-duality, for any feasible solution $\mathbf{x}$ of (1) and for any feasible solution $\mathbf{y}$ of (2), we have $|\mathbf{x}| \geq |\mathbf{y}|$. In summary, we have the following:

Fact 1. *For any graph G, let C be an minimum vertex cover of G and $\mathbf{y}$ be any fractional matching. Then $|C| \geq |\mathbf{y}|$.*

Using this fact we can obtain the bounds mentioned in the introduction for the matching, LP-rounding and Greedy approximation algorithms [2]. We can also obtain tight bounds for list heuristics.

In proving our main result we also make use of the following well known result.

Lemma 2. *Let $d_1, ..., d_s$ be positive integers with sum at most m, then*

$$\sum_{i=1}^{s} \frac{1}{d_i} \geq \frac{s^2}{m}.$$

Proof. Using the Cauchy-Schwarz inequality

$$\sum a_i^2 \geq \frac{(\sum a_i b_i)^2}{\sum b_i^2}$$

with $a_i = 1/\sqrt{d_i}$ and $b_i = \sqrt{d_i}$, we obtain

$$\sum_{i=1}^{s} \frac{1}{d_i} \geq \frac{s^2}{\sum_{i=1}^{s} d_i} \geq \frac{s^2}{m}.$$

□

3 An Upper Bound for the List Decreasing Heuristic

In this section we first describe the list-decreasing algorithm $\mathcal{A}_D$, and then give an analysis for the upper bound on the approximation ratio.

$\mathcal{A}_D$ works as follows. Given a graph $G = (V, E)$, let d_i denote the degree of vertex v_i, and assume the vertices are labeled such that $\Delta = d_1 \geq d_2 \geq \cdots \geq d_n$. It constructs a vertex cover C_D in at most $n-1$ steps by selecting certain vertices. Initially C_D is empty. At the i-th step, let R_i denote the set of edges incident to v_i that are not incident to any vertex already in C_D. If R_i is nonempty, vertex v_i is added to C_D.

Let *Opt* denote the size of the minimum vertex cover. An upper bound on the approximation ratio $|C_D|/Opt$ is obtained by getting a lower bound on *Opt* by constructing a dual feasible solution $\mathbf{y}$. An obvious dual feasible $\mathbf{y}$ is obtained by setting $y_e = 1/\Delta$ for each edge e. This gives the trivial lower bound $Opt \geq m/\Delta$ and an upper bound on the approximation ratio of Δ. To strengthen this bound we observe that any edge e in R_i could be given the weight $y_e = 1/d_i$, and the resulting vector $\mathbf{y}$ is still dual feasible. Vertices selected late by $\mathcal{A}_D$ will tend to have smaller degrees, so the corresponding edge weights will be higher. We have no control over the size of R_i except that it contains at least one edge. When $\mathcal{A}_D$ performs badly $R_i = 1$ will often be achieved for low degree vertices and thus few edges have higher weights.

It turns out that selecting just one edge from each of the R_i corresponding to vertices chosen later on by the heuristic suffices to give a tight bound on the approximation ratio.

Theorem 3. *The algorithm $\mathcal{A}_D$ has approximation ratio at most $\frac{\sqrt{\Delta}}{2} + \frac{3}{2}$.*

Proof. Let C_D be the solution obtained by $\mathcal{A}_D$ and denote by *Opt* the size of the optimal solution. Let p be the minimum index such that $\sum_{i=1}^{p} d_i \geq m$. Then

$$p \leq Opt$$

since the degree sum of any vertex cover is at least m, and $\{v_1, v_2, \ldots, v_p\}$ is a minimum cardinality set of vertices with this property. Since $\sum_{i=1}^{n} d_i = 2m$, it follows that $\sum_{i=p+1}^{n} d_i \leq m$. A vertex v_i such that $i \geq p+1$ is considered to be a low degree vertex.

We construct a fractional matching $\mathbf{y} = \{y_e\}_{e \in E}$ of G from C_D as follows. For each $i \geq p+1$ for which v_i is selected by $\mathcal{A}_D$, we choose an arbitrary edge e from R_i and give it weight $y_e = \frac{1}{d_i}$. Call such edges *special* and let s denote the number of special edges. We have

$$s \geq |C_D| - p \geq |C_D| - Opt.$$

Each of the remaining $m - s$ edges is given weight $\frac{1}{\Delta}$. It is easy to verify that this weight assignment is a fractional matching. See the example in Figure 1. (For this graph, the upper three vertices are chosen first by $\mathcal{A}_D$ and weight $1/\Delta = 1/9$ is given to all incident edges. Then the nine vertices in the middle are chosen and the rest of the edges obtain weight $1/4$.)

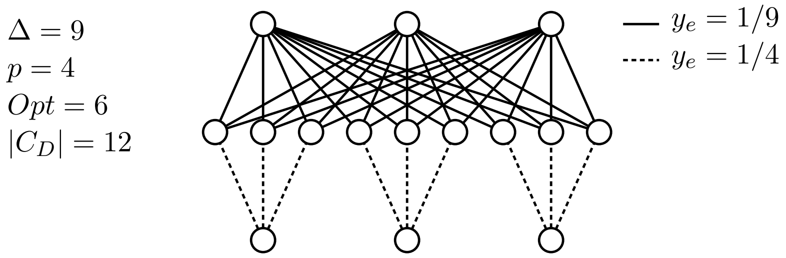

Figure 1: An example of the constructed fractional matching.

Using the lemma, we obtain a lower bound on the total edge weight assigned to the s special edges:

$$\sum_{v_i \in C_D, i \geq p+1} \frac{1}{d_i} \geq \frac{s^2}{m}.$$

Combining this with Fact 1 we deduce that

$$\begin{aligned}
Opt &\geq \sum_{e \in E} y_e \\
&\geq \frac{m-s}{\Delta} + \frac{s^2}{m} \\
&\geq 2\sqrt{\frac{m}{\Delta} \cdot \frac{s^2}{m}} - \frac{s}{\Delta} \quad \text{(since } a+b \geq 2\sqrt{ab}\text{)} \\
&= \frac{2s}{\sqrt{\Delta}} - \frac{s}{\Delta} \\
&= \frac{2s}{\sqrt{\Delta}} \left(1 - \frac{1}{2\sqrt{\Delta}}\right) \\
&\geq \frac{2(|C_D| - Opt)}{\sqrt{\Delta}} \left(1 - \frac{1}{2\sqrt{\Delta}}\right).
\end{aligned}$$

Multiplying each side by $\frac{\sqrt{\Delta}}{2\ Opt}$, we obtain

$$\frac{\sqrt{\Delta}}{2} \geq \left(\frac{|C_D|}{Opt} - 1\right) \left(1 - \frac{1}{2\sqrt{\Delta}}\right).$$

Since $1 - \frac{1}{2\sqrt{\Delta}} \geq \frac{1}{1+\frac{1}{\sqrt{\Delta}}}$ for any $\Delta \geq 1$, by rearranging we obtain

$$\frac{|C_D|}{Opt} \leq \frac{\sqrt{\Delta}}{2} \cdot \frac{1}{1 - 1/2\sqrt{\Delta}} + 1 \leq \frac{\sqrt{\Delta}}{2} \cdot \left(1 + \frac{1}{\sqrt{\Delta}}\right) + 1 = \frac{\sqrt{\Delta}}{2} + \frac{3}{2}.$$

This concludes the proof of Theorem 3.

□

4 A Lower Bound for List Heuristics

In the previous section we showed that the approximation ratio of the list decreasing heuristic $\mathcal{A}_D$ is at most $\frac{\sqrt{\Delta}}{2} + \frac{3}{2}$. In this section, we will show that this bound is tight up to the constant. A slight generalization gives a lower bound that applies to any list heuristic based on degree sequences. First we give a bad example for $\mathcal{A}_D$.

Let N be an integer. We construct a bipartite graph $G = (V, U, E)$ with bipartition V and U such that $|V| = 2N, |U| = N^2$. We divide V into two equal-size disjoint subsets V_1, V_2 and span edges between every vertex in V_1

and every vertex in U. Finally, we span edges between V_2 and U so that every vertex in V_2 has degree N and no two vertices in V_2 are adjacent to the same vertex in U. Thus every vertex in V_1 has degree N^2, every vertex in U has degree $N+1$, and every vertex in V_2 has degree N. Figure 1. shows the case $N = 3$.

Clearly G has $n = N^2 + 2N$ vertices and the minimum vertex cover consists of V, so $Opt = 2N$. For this graph, the algorithm $\mathcal{A}_D$ works very poorly. $\mathcal{A}_D$ first takes every vertex in V_1, then it takes all the vertices in U. In this case the approximation ratio is

$$\frac{|C_D|}{Opt} = \frac{N^2 + N}{2N} = \frac{N}{2} + \frac{1}{2} = \frac{\sqrt{\Delta}}{2} + \frac{1}{2},$$

since $\Delta = N^2$. As the upper bound we gave is $\frac{\sqrt{\Delta}}{2} + \frac{3}{2}$, our analysis is tight up to the final constant. We observe that after the N vertices in V_1 have been selected, each additional vertex selected is adjacent to exactly one uncovered edge, as suggested by the comment made before Theorem 3.

Recall that $\mathcal{A}_D$ processes each vertex in nonincreasing order of degree. A list heuristic may be defined for any order. For fixed permutation $\pi = (\pi_i)$ of length n, the algorithm $\mathcal{A}_\pi$ process π_i-th highest degree vertex at i-th step. Now we show that there is an n-vertex graph G such that, for any permutation π of length n, the approximation ratio of $\mathcal{A}_\pi$ on G is at least $\frac{\sqrt{\Delta}}{2}$. We modify the example above by setting $|V_1| = N - 1$, $|V_2| = N$ and $|U| = N^2$. We use the same edge construction as before. Now all vertices in both U and V_2 have degree N, so we have no control over the order they are processed. In the worst case the vertices in V_2 will be considered last among the vertices of degree N. This means that all of U will need to be selected in order to cover edges to V_2. Since $\Delta = N^2$, we have

$$\frac{|C_\pi|}{Opt} \geq \frac{N^2}{2N - 1} \geq \frac{\sqrt{\Delta}}{2}.$$

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. *Introduction to Algorithms.* MIT Press, 2001.

[2] V.V. Vazirani. *Approximation Algorithms.* Springer-Verlag, 2001.

[3] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383-390, 1975.