

VERTEX COVER IN COMPLEX NETWORKS

MARIANA O. DA SILVA

*DAINF, Universidade Tecnológica Federal do Paraná
Rua Sete de Setembro, 3164, Curitiba-PR, CEP 80230-901, Brazil
marianaoliveirasi@pos.utfpr.edu.br*

GUSTAVO A. GIMENEZ-LUGO

*DAINF, Universidade Tecnológica Federal do Paraná
Rua Sete de Setembro, 3164, Curitiba-PR, CEP 80230-901, Brazil
gustavogl@utfpr.edu.br*

MURILO V. G. DA SILVA

*DAINF, Universidade Tecnológica Federal do Paraná
Rua Sete de Setembro, 3164, Curitiba-PR, CEP 80230-901, Brazil
murilo@utfpr.edu.br*

Received Day Month Year

Revised Day Month Year

A *Minimum Vertex Cover* is the smallest set of vertices whose removal completely disconnects a graph. In this paper we perform experiments on a number of graphs from standard complex networks databases addressing the problem of finding a “good” vertex cover (finding an optimum is a NP-Hard problem). In particular, we take advantage of the ubiquitous power law distribution present on many complex networks. In our experiments we show that running a greedy algorithm in a *power law graph* we can obtain a very small vertex cover typically about 1.02 times the theoretical optimum. This is an interesting practical result since theoretically we know that: **(1)** In a general graph on n vertices a greedy approach cannot guarantee a factor better than $\ln n$; **(2)** The best approximation algorithm known at the moment is very involved and has a much larger factor of $2 - \Theta(\frac{1}{\sqrt{\log n}})$. In fact, in the context of approximation within a constant factor, it is conjectured that there is no $(2 - \epsilon)$ -approximation algorithm for the problem; **(3)** Even restricted to power law graphs and probabilistic guarantees, the best known approximation rate is 1.5.

Keywords: Complex networks; power law graphs; vertex cover; greedy algorithms.

PACS Nos.: 11.25.Hf, 123.1K

1. Introduction

The study of large real world graphs, also commonly called complex networks, grew enormously in last decade bringing together physicists, mathematicians, computer scientists and many other researchers¹. In this area many experimental work

has been performed by experimentalists ^{2,3,4,5,6,7} as well as a number of analytical models have been proposed by the community akin to mathematics and theoretical computer science ⁸.

A crucial point in this field is that many networks from different domains (social, economic, technological and biological networks, etc) share some common properties. Some of the most notable common properties among a variety of complex networks are small diameter (the “small world phenomenon”), the power law distribution (see Table 1 for a list with few examples of power law graphs according to ¹⁰), high clustering coefficient among a few others ^{10,9}.

In this paper, generally speaking, we are interested in the following (quite broad) question: can we explore these properties common to most real world networks in order to obtain more efficient algorithms for combinatorial optimization problems? This is an interesting practical question that still has to be explored more thoroughly. In our work we pick a particular optimization problem, the *Minimum Vertex Cover*, and conduct experiments taking advantage of one particular property of complex networks: the *Power Law Distribution*. We define these concepts in the next section.

Table 1. Examples of power law networks according to Barabasi and Bonabeau ⁸.

Networks	Nodes	# Links
Hollywood	Actors	Appearance in the same movie.
Internet	Routers	Optical and physical connections.
Protein regulatory network	Proteins	Interactions among proteins.
Research collaborations	Scientists	Co-atorship of papers.
Sexual relationships	People	Sexual contact.
World Wide Web	Web pages	URLs.
Celular Metabolism	Molecules involved in burning food for energy	Participation in the same biochemical reaction

The main point in our work is that since power law graphs contain a few very large “hubs”, i.e., vertices of very high degree, naturally one might expect that a greedy approach (i.e., choosing first these hubs for the covering) would perform well. Although this idea is obviously quite natural, the authors are not aware of any experimental or analytical work to confirm this expectation. More importantly, a question that has also not being answered – as far as we are aware – is how well (quantitatively) a greedy algorithm would perform on real world power law networks. We have run such experiments on a number of graphs from different domains from standard complex networks databases and we obtained coverings very close to the optimum, more precisely around 1.02 times the optimum (the larger obtained value was 1.05). We present this and other related results on Section 4. In Sections 2 and 3 we define the problem and discuss how the experimental results obtained in this paper relate to the theoretical guarantees for approximation algorithms for the vertex cover problem.

2. Power Law Graphs and Combinatorial Optimization

Barabasi and Albert¹⁰ and also others⁸ observed that in many real-world graphs, the vertex degree sequence has a power law distribution, i.e., the fraction of vertices with degree d is proportional to $d^{-\lambda}$, where λ is a constant independent of the size of the graph. The authors modelled such real world networks using a preferential attachment scheme which became well known in the field. Other researchers proposed a number of other different approaches to model such networks⁸. Experimental work in the field had pointed the λ mentioned above is 2.9 ± 0.1 and Bollobás et al¹¹ in 2001 presented an analytical argument pointing that such parameter is in fact 3. In our paper we will use the common designation “*Power Law Graph*” for graphs with such distribution for the vertex degree sequence. In Figure 1 we give two examples of the vertex degree distribution from networks used in our experiments.

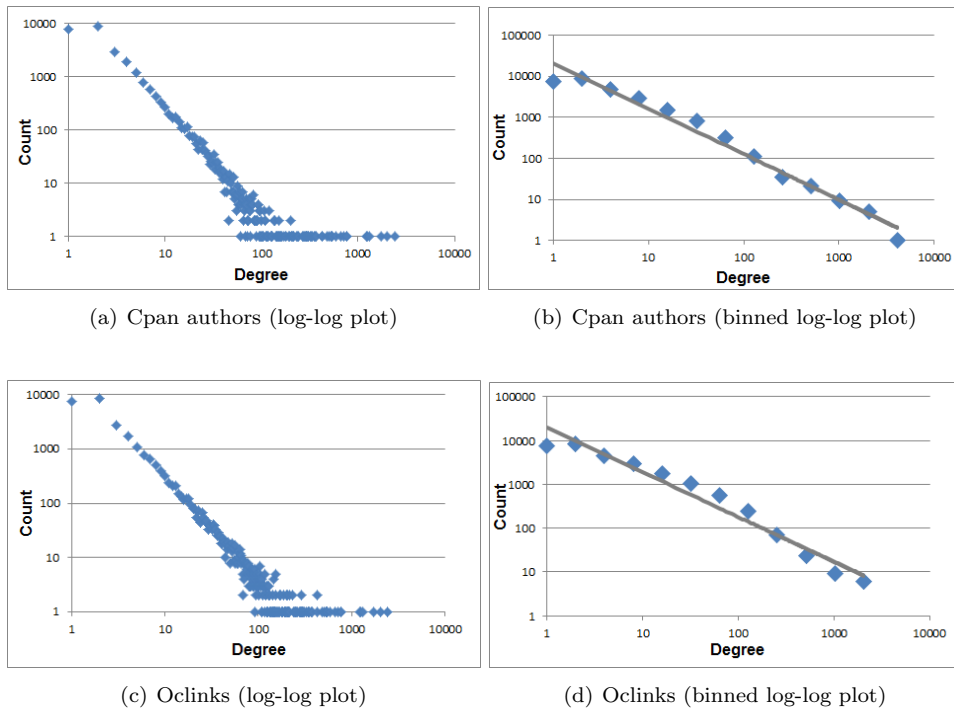


Fig. 1. Vertex degree distribution. The x -axis is the degree distribution from 0 to maximum value and the y -axis is the number of vertices of such degree.

In the light of the fact that many networks from real world are in fact power law graphs, many researchers in the field have been exploring this property structurally and algorithmically^{12,13,14}. Recently, some work also has been done on power law

graphs aiming to lower the average factor of approximation algorithms for NP-hard combinatorial problems^{15,16,17}. To be more precise, in the case of the vertex cover problem, although it is believed that 2 is the smallest constant factor for an approximation algorithm, in the particular case of power law graphs, Gast and Hauptmann proposes an algorithm which outputs an **expected** vertex cover not larger than 1.5 times the optimum¹⁵. These analytical results are exactly in line with what we propose experimentally in this paper. In particular, we show that in practice simpler algorithms can achieve much smaller factors of approximation.

3. The vertex cover problem

A cover in a graph is a set of vertices whose removal completely disconnects a graph^a. A Minimum Vertex Cover is a smallest set with such property. Figure 2 depicts an example of a minimum vertex cover in a graph. This problem is a classical NP-Hard problem¹⁸ which remains intractable even for cubic graphs and planar graphs with maximum degree at most three¹⁸. Therefore approximation algorithms for finding “good” solutions in polynomial time are of great interest. We discuss in the next section the two most common approximation approaches in this direction.

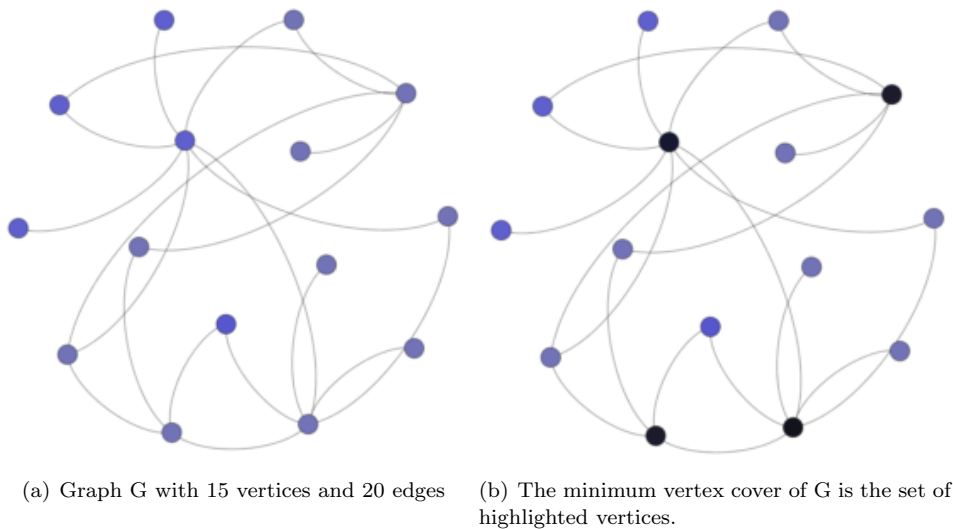


Fig. 2. Example of a Minimum Vertex Cover of a graph G : If the set of vertices highlighted in (b) are removed from G , the resulting graph has no edges.

^aA completely disconnected graph is a graph with no edges. Alternatively a vertex cover S is a set of vertices such that every edge in the graph has an end in S .

3.1. Simple ideas for approximation algorithms

- **Greedy Algorithm:** Given a graph G , pick the vertex v with the highest degree and insert it in the cover S and remove v from G . Repeat the same strategy in the remaining graph iteratively until there are no more edges. Such approach always finds a vertex cover S such that $|S|$ is at most $\ln n$ times the optimum, where n is the number of vertices in input graph G . For a general graph this upper bound is tight ¹⁹.
- **2-Approximation Algorithm:** Given a graph G , pick an arbitrary edge uv , insert uv in the set M and remove both u and v from G . Repeat the same strategy in the remaining graph iteratively until there are no more edges ^b. In the next step include in the vertex cover S both ends of each edge in the set X . This approach leads to a cover S of size at most twice the optimum ¹⁹ (i.e., this is an approximation algorithm with factor 2).

The 2-approximation algorithm presented above is known for about 40 years and it is one of the best known until now for general graphs ²⁰. It is known that no algorithm can guarantee a factor better than 1.36 ²¹ (unless P=NP). In fact it is conjectured an even stronger assertion: For any constant ϵ , there is no $(2 - \epsilon)$ -approximation algorithm for the problem ²². Furthermore, even for non constant guarantees, very small improvements on the factor 2 require very involved algorithms. At the moment the best known approximation algorithm, proposed by Karakostas ²³, has a factor of $2 - \Theta(\frac{1}{\sqrt{\log n}})$ which is much larger than the numbers that we obtained experimentally using the greedy approach.

In the domain of power law graphs, Gast and Hauptmann recently proposed an algorithm and proved that it outputs on average a vertex cover of size 1.5 times the optimum on the (α, β) -model of power law graphs proposed by Aiello ¹⁵. This result is the only one that we found on the topic of vertex cover in power law graphs.

4. Results for Real Networks and Discussion

In this section we present our experimental results addressing the minimum vertex cover problem on 25 power law complex networks from well known graph databases ^{24,25,26}. The size of the networks range from a few hundred to about 75000 vertices. In 15 of these networks we had the value of the optimum cover for comparison. In these 15 networks our experiments show that a greedy approach outputs a typical vertex cover of size around 1.02 times the optimum. The size of the coverings found by the greedy algorithm are shown in the column named *Greedy* in Table 2 and the rate of approximation in the column named *Greedy/Opt* in Table 3. Theoretically a greedy approach can output results with a factor as bad as $\ln n$, so the point here

^bNote that at the end of this process M is a *maximal matching* in G , i.e., M is a set of edges in G such that for every distinct $e, e' \in M$, edges e and e' have no endnodes in common and there is no other set of edges M' in G such that $M \subset M'$.

is of course that such “obstructions” which would bring to a larger approximation factor are not typical in power law graphs. So for that reason in practice the greedy approach does very well.

We also implemented the 2-approximation algorithm discussed in the section 3.1 and obtained significantly worst results than the greedy algorithm. A typical vertex covering obtained by the 2-approximation algorithm was around 1.56 times the optimum. Again, these results were obtained from the 15 graphs for which we had the optimum for comparison. These results are presented in Tables 2 and 3. We note that we ran both the greedy algorithm and the 2-approximation algorithm on all 25 networks (i.e., not only on the networks where we had the optimum for comparison) and the results pointed that the greedy approach obtained a typical vertex cover of size 0.66 times the size of the competing algorithm. These results are presented in the column named *Greedy / 2-App* in Table 3. We included Table 4 to refer to the well known databases from where the networks used in our experiments were obtained. For the sake of completeness we also included the number of edges for each network.

In order to obtain the optimum value for the vertex cover, we used the well known SAGE Library ²⁷. We managed to obtain minimum coverings for graphs up to about 10000 vertices, but for the 10 largest graphs we were not able to obtain these values due to the exponential nature of the algorithm. It is still somewhat surprising that the optimum value could be computed for graphs sizing almost 10000 vertices. So it is very important to point out that although the optimum algorithm is exponential, due to the intricate process in which particular graphs are explored, successful executions for relatively large data were possible ^c.

Finally, we would like to add two remarks: (1) In the literature of complex networks, in particular in the context of network robustness, some authors refer to the sequence of vertices obtained by the greedy strategy described in this paper as *high degree adaptive attack* ²⁸. (2) In a final note, as we have mentioned before, in our experiments we also computed the ratio between the size of the cover obtained by the greedy algorithm divided by the size of the cover cover obtained by the 2-approximation algorithm and the results show that the cover obtained by the greedy algorithm is about 66% of the size of the other approach (Table 3). It should also be pointed that this number seems to be invariant to the size of the graph. This fact might give us some indication that even for the 10 largest graphs (for which the optimum values were not available for comparison) the size of the coverings obtained by the greedy algorithm might still be very close to the optimum as well.

^cWe ran our experiments on a R710 DELL sever running a Intel Xenon Quad Core 5500 y 5600 and a 60GB Swap Memory + 12GB Physical Memory. The smallest graph for which the SAGE exponential approach “crashed” was running for about 12 hours and stopped due to lack of memory

Table 2. Algorithms for Vertex Cover in Complex Networks: (1) First column is the ID of the graph. We use it to refer to the same graphs on other tables. (2) Second column is the name of the file available on the databases. (3) Third column is the number of vertices of the network. (4) Forth column is the size of the minimum vertex cover. We used the algorithm from SAGE Library for computing these values. Data for larger graphs could not be obtained due to the exponential nature of the algorithm (see section 4 for details). (5) Fifth column is the vertex cover obtained by the greedy algorithm implemented by us. (6) Sixth column is the vertex cover obtained by the 2-approximation algorithm implemented by us. (7) Seventh column indicates the type of the network.

Net ID	Network file	# of vertices	Opt	Greedy	2-Aprox	Network type
1	Airlines	235	96	97	146	Technological
2	US Air	332	149	151	230	Technological
3	Codeminer	724	191	196	334	Social Net.
4	Cpan authors	839	116	117	196	Social Net.
5	EuroSis	1285	597	608	896	Information
6	Oclinks	1899	749	763	1100	Information
7	YeastS	2284	763	773	1240	Biologic
8	CA-GrQc	5241	2783	2795	3960	Citation
9	p2p-Gnutella08	6301	2054	2070	3366	Communication
10	Wiki-Vote1	7115	2249	2370	3460	Social Net.
11	p2p-Gnutella09	8114	2574	2589	4238	Information
12	p2p-Gnutella06	8717	3405	3484	5352	Information
13	p2p-Gnutella05	8846	3428	3475	5412	Information
14	CA-HepTh	9875	4981	5003	7240	Citation
15	p2p-Gnutella04	10876	4348	4428	6624	Collaboration
16	CA-AstroPh	18771	n.a	12044	15194	Collaboration
17	p2p-Gnutella25	22687	n.a	6055	9800	Information
18	CA-CondMat	23133	n.a	13561	18150	Collaboration
19	p2p-Gnutella24	26518	n.a	7250	11624	Information
20	Cit-HepTh	27769	n.a	18225	23396	Citation
21	p2p-Gnutella30	36682	n.a	9321	15096	Information
22	Email-Enron	36692	n.a	14477	20674	Communication
23	Brightkite-edges	58228	n.a	22177	34814	Social Net.
24	p2p-Gnutella31	62586	n.a	15864	25582	Information
25	soc-Epinions1	75879	n.a	22418	35964	Social Net.

Table 3. (1) First column is the graph ID. (2) Second column shows the rate of approximation obtained by the greedy algorithm (i.e., the value obtained by the algorithm divided by the optimum). (3) Third column shows the rate of approximation obtained by the 2-approximation. (4) Forth column shows the size of the cover obtained by the greedy algorithm divided by the size of the cover obtained by the 2-approximation. In the bottom of the table we include the average and the standard deviation for each column.

ID	Greedy /Opt	2-App /Opt	Greedy /2-App
1	1,01	1,52	0,66
2	1,01	1,54	0,66
3	1,03	1,75	0,59
4	1,01	1,69	0,60
5	1,02	1,50	0,68
6	1,02	1,47	0,69
7	1,01	1,63	0,62
8	1,00	1,42	0,71
9	1,01	1,64	0,61
10	1,05	1,54	0,68
11	1,01	1,65	0,61
12	1,02	1,57	0,65
13	1,01	1,58	0,64
14	1,00	1,45	0,69
15	1,02	1,52	0,67
16	n.a	n.a	0,79
17	n.a	n.a	0,62
18	n.a	n.a	0,75
19	n.a	n.a	0,62
20	n.a	n.a	0,78
21	n.a	n.a	0,62
22	n.a	n.a	0,70
23	n.a	n.a	0,64
24	n.a	n.a	0,62
25	n.a	n.a	0,62
average	1,02	1,56	0,66
st. dev.	0,01	0,09	0,05

Table 4. In this table we indicate the databases from where networks were obtained. For the sake of completeness we also include the size of the vertex set and edge set for each network.

ID	Database	Vertices	Edges
1	Gephi	235	1295
2	Pajek	332	2126
3	Gephi	724	1015
4	Gephi	839	2112
5	Gephi	1285	6462
6	Gephi	1899	13821
7	Pajek	2284	6646
8	Stanford	5241	14484
9	Stanford	6301	20776
10	Stanford	7115	100729
11	Stanford	8114	26013
12	Stanford	8717	31523
13	Stanford	8846	31837
14	Stanford	9875	25973
15	Stanford	10876	39993
16	Stanford	18771	198050
17	Stanford	22687	54705
18	Stanford	23133	93439
19	Stanford	26518	65368
20	Stanford	27769	352285
21	Stanford	36682	88328
22	Stanford	36692	183811
23	Stanford	58228	214023
24	Stanford	62586	147890
25	Stanford	75879	404953

References

1. D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press (2010)
2. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. S. Tomkins, and J. Wiener. *Graph structure in the Web*. Computer networks, 33(1-6):309-320, June 2000.
3. J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan and A. S. Tomkins. *The Web as a graph: measurements, models and methods*. Proceedings of the 5th annual international Conference on Computing and Combinatorics, 1-17, 1999.

4. M. Faloutsos, P. Faloutsos and C. Faloutsos. *On power-law relationships of the internet topology*. ACM SIGCOMM Computer Communication Review, 29(4):251-262, 1999.
5. M. Jovanović, F. S. Annexstein and K. A. Berman. *Modeling peer-to-peer network topologies through "small-world" models and power laws*. In IX Telecommunications Forum, TELFOR, 2001
6. N. Guelzim, S. Bottani, P. Bourguin and F. Kps. *Topological and causal structure of the yeast transcriptional regulatory network*. Nature Genetics, 31(1):60-63, 2002.
7. S. Eubank, V. S. A. Kumar, M. V. Marathe, A. Srinivasan and N. Wang. *Structural and algorithmic aspects of massive social networks*. In Proceedings of the 15th annual ACM-SIAM Symposium on Discrete Algorithms, pages 718-727. SIAM, 2004.
8. L. Lu and F. Chung. *Complex Graphs and Networks*. American Math. Society (2006).
9. A.L. Barabási and E. Bonabeau. *Scale-Free Networks*. Scientific American 288, 50-59, 2003.
10. A.L. Barabási and R. Albert. *Emergence of scaling in random networks*. Science, 286(5439):509, 1999.
11. B. Bollobás, O. Riordan, J. Spencer, G. Tusndy *The Degree Sequence of a Scale-Free Random Graph Process*. Random Structures & Algorithms. Vol. 18 Issue 3, 279-290 (2001)
12. K. Park and H. Lee. *On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets*. ACM SIGCOMM Computer Communication Review, 31(4):15-26, October 2001.
13. C. Gkantsidis, M. Mihail and A. Saberi. *Conductance and congestion in power law graphs*. SIGMETRICS Performance Evaluation Review, 31:148-159, 2003.
14. M. Mihail, C. H. Papadimitriou and A. Saberi. *On certain connectivity properties of the internet topology*. Journal of Computer and System Sciences, 72(2):239-251, 2006.
15. M. Gast and M. Hauptmann. *Approximability of the Vertex Cover Problem in Power Law Graphs*. Computing Research Repository (CoRR), arXiv:1204.0982, 2012.
16. M. Hauptmann, M. Gast and M. Karpinski. *Inapproximability of Dominating Set in Power Law Graphs*. Computing Research Repository (CoRR), arXiv:1212.3517, 2012.
17. M. Hauptmann, M. Gast and M. Karpinski *Improved Approximation Lower Bounds for Vertex Cover on Power Law Graphs and Some Generalizations*. Computing Research Repository (CoRR), arXiv:1210.2698, 2012.
18. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co. New York, NY (1979).
19. C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice Hall (1982).
20. S. Dasgupta, C.H. Papadimitriou and U.V. Vazirani. *Algorithms*. MG-Hill, 2008.
21. I. Dinur and S. Safra. *On the hardness of approximating vertex cover*. Annals of Mathematics, Pages 439-485 from Volume 162 (2005).
22. S. Khot, O. Regev. *Vertex cover might be hard to approximate to within $2 - \epsilon$* . Journal of Computer and System Sciences Volume 74, Issue 3, May 2008, Pages 335-349.
23. G. Karakostas. *A better approximation ratio for the vertex cover problem*. ACM Transactions on Algorithms, Volume 5 Issue 4, Article No. 41, 2009.
24. GEPHI Datasets, <http://wiki.gephi.org/index.php/Datasets>
25. PAJEK Datasets, <http://vlado.fmf.uni-lj.si/pub/networks/data>
26. Stanford Large Network Dataset Collec. <http://snap.stanford.edu/data>
27. SAGE, <http://www.sagemath.org/library.html>
28. C. M. Schneider, A. A. Moreira, J. S. Andrade Jr., S. Havlin and H. J. Herrmann. *Onion-like Network Topology Enhances Robustness against Malicious Attacks*. Journal of Statistical Mechanics: Theory and Experiment (2011) P01027