CMSC 858M: Algorithmic Lower Bounds
Spring 2021
Chapter 10: Inapproximability

Instructor: Mohammad T. Hajiaghayi
Scribe: Nathan Hayes

Feb 27, 2021

1 Overview

2 10.1

This section is well-structured, and I think the definitions and conventions given are very clear and well-organized, especially the definition of an NPO problem. The examples given along the way help a great deal. The only not of improvement I suggest would be to add a sentence or two motivating why we need the definition of NPO to approach the problem of inapproximability, so that we have motivation going into the definition section.

3 10.2

In definition 10.2.1, you may want to specify that ALG outputs a solution to A. Also, I believe you have left out the runtime restrictions on algorithms used to show a problem is in APX. If there are no such I would also like to see examples of problems which are not in APX and the other classes, since it seems to me that since APX has no runtime restrictions, the problems not in APX must be very unnatural. Otherwise, I think the content of the section is good.

4 10.3

This section is quite good, in my opinion, and the use of exercises to help encourage the reader to thoroughly understand how the reductions are constructed is excellent. I'm surprised that the machinery for these reductions is this much more complex than NP reductions, but in retrospect it makes a lot of sense.

5 10.4

This section seems to me not to flow very well. It would benefit from a brief introduction along the lines of "These are variants of MAX-3SAT which are useful for important reductions," and perhaps should be reorganized such that results about MAX-3SAT come before variations.

6 10.5

This section is very short, but otherwise seems fine to me. I think the parallel in the results about NP and about APX is very intuitive.

7 10.6

I think this section is very natural, and the proof is well written and structured. I was surprised to encounter expander graphs in this section, but their introduction seems quite natural in retrospect. I was also surprised to learn that there are conflicting definitions of expander graph. I'm also curious - has anyone created a direct reduction from MAX-3SAT-E-3 to MAX-3SAT? If so, it might be a worthy inclusion to mention.

8 10.7

This section is quite dense, but very interesting. I also quite like the title. I'm honestly amazed that such a chain of reductions turns out to be useful, as I figured formula-space and graph-space wouldn't be quite so intertwined. The proofs you give are well-written and comprehensible, and I think on the whole there is no way to avoid the density.

9 10.8

It's interesting to me how incredibly general this is - the fact that this applies to any constraint satisfaction problem is astounding. I think this chapter is well written, and I like the inclusion of previously discussed examples to show how this applies.

10 10.9

This chapter seems fine, although it seems a little sparse on material. That said, I think that's appropriate, since it serves its purpose well. Overall this seems to be a standard addition.

11 10.10

I think the graphic used to demonstrate the APX reduction from Set Cover to Node Weighted Steiner Tree should be created in software - the image is shaded and seems unprofessional to me. The placement of the image is also strange, as it appears before being referenced. Otherwise, the chapter seems well-written, although it might benefit from slightly more rigorous definitions of the various classes. That said, this is a matter of personal preference.

12 Relevant Problems

- Knapsack Problem - This well-known problem has a PTAS, although it appears to be folklore rather than a cited result. An algorithm may be found on Wikipedia.
- Euclidean TSP - This problem is a restriction of TSP to Euclidean graphs. It was shown to have a PTAS by Arora [1].
- Multiterminal Cut - This was shown to be APX Hard by Dahlhaus et al. [2] To quote the abstract, "In the multiterminal cut problem one is given an edge-weighted graph and a subset of the vertices called terminals, and is asked for a minimum weight set of edges that separates each terminal from all the others." A generalization of the problem was shown to be APX Hard by Avidor and Langberg [3]
- k-Burning Number - This problem is a generalization of the Burning Number problem, which is to find the burning number of a graph. It was shown by Mondal et al. [4] to be APX Hard.
- Nash Social Welfare - This problem involves allocating indivisible items to people in order to maximize Nash social welfare, a particular way of prioritizing utility states. Lee [5] showed this problem to be APX Hard.
- MaxLeaf - The problem of finding the spanning tree with the most leaves. It was shown by Galbiati et al. [6] to be APX Hard.

The following problems apply to cubic graphs. A cubic graph is a 3-regular graph.

- Cubic MaxLeaf - Maxleaf on cubic graphs. Shown by Bonsma [7] to be APX Hard.
- Cubic Min Vertex Cover - Min vertex cover on cubic graphs. Shown by Alimonti and Kann [8] to be APX Hard.
- Cubic Max Independent Set - Max independent set on cubic graphs. Shown by Alimonti and Kann [8] to be APX Hard.

- Cubic Min Dominating Set - Min dominating set on cubic graphs. Shown by Alimonti and Kann [8] to be APX Hard.
- Cubic Max Cut - Max cut on cubic graphs. Shown by Alimonti and Kann [8] to be APX Hard.

References

[1] Sanjeev Arora. 1998. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. J. ACM 45, 5 (Sept. 1998), 753–782. DOI:https://doi.org/10.1145/290179.290180

[2] Dahlhaus, Elias et al. (1994). The Complexity of Multiterminal Cuts. SIAM J. Comput.. 23. 864-894. 10.1137/S0097539792225297.

[3] Avidor, Adi and Langberg, Michael. (2004). The multi-multiway cut problem. Theoretical Computer Science. 377. 35-42. 10.1016/j.tcs.2007.02.026.

[4] Debajyoti Mondal, N. Parthiban, V. Kavitha, & Indra Rajasingh. (2021). APX-Hardness and Approximation for the k-Burning Number Problem.

[5] Euiwoong Lee. (2015). APX-Hardness of Maximizing Nash Social Welfare with Indivisible Items.

[6] Galbiati, F. (1995). On the approximability of some maximum spanning tree problems. In LATIN '95: Theoretical Informatics (pp. 300–310). Springer Berlin Heidelberg.

[7] Paul Bonsma (2012). Max-leaves spanning tree is APX-hard for cubic graphs. Journal of Discrete Algorithms, 12, 14-23.

[8] Paola Alimonti, & Viggo Kann (2000). Some APX-completeness results for cubic graphs. Theoretical Computer Science, 237(1), 123-134.