Exponential Approximations to MAX-NEQ-4SAT

https://zack-lee.github.io/website/

1. Project Description

Constraint Satisfaction Problems (CSP's) are a generalization that covers many natural P/NP questions in the field of CS-Theory. The Dichotomy Theorem of CSP's guarantees that are CSP's are either in P or NP-hard. Now the nature of solving CSP's is well known, however the task of approximating then is much harder. For example, 2SAT is in P, through an implication graph reduction, but MAX-2SAT is NP and approximating it to a factor better than $\frac{21}{22}$ is known to be NP-hard.

Another example is 3SAT which is well known to be NP-hard, but there exists a simple $\frac{7}{8}$ approximation algorithm in P through randomization. It has been shown a $\frac{7}{8} + \epsilon$ approximation is NP-hard, however this can be done in $O(2^{\epsilon n})$ time. My project investigates extending these exponential time algorithms to gain better approximations for more CSP's. In more specificity, the problem of MAX-NEQ-4SAT is also easily approximated to a factor of $\frac{7}{8}$ by a randomization argument. My goal is to extend the results of the $\frac{7}{8} + \epsilon$ approximation to MAX-NEQ-4SAT.

MAX-NEQ-4SAT can also be thought of as coloring the 4-uniform hypergraph implied by the clauses. This study of hypergraphs is common in combinatorics in math, and hopefully relations like this will help shed insight to approximating the coloring problem.

I am working with Professor Venkatesan Guruswami of the CS Department at SCS.

2. Project Goals

75%: Finish all literature readings and find an approximation algorithm that solves in $o(2^n)$ time
100%: Able to find an approximation algorithm that solves in $O(2^{\epsilon n})$ time
125%: Able to show a lower bound or extend this is MAX-NEQ-N-SAT or other CSP's

3. Milestones

For the first technical milestone of 15-300, I plan on finishing reading all the literature listed in my literature session and meeting with Venkat again to discuss any plans/ideas on how to attack this approximation problem.

- February 15th: Brainstorm ideas, start implementing code for an actual solver and small cases
- March 1st: Finish solver for small cases, continue algorithm design

- March 15th: Continue algorithm design
- March 29th: Hopefully have finished 75% goal, will meet with Venkat to discuss when any nontrivial result is found
- April 12th: Attempt to have ideas to extend to 100% goal
- April 26th: Draft summary of results if have found anything significant
- May 10th: Finish editing paper and have codebase on GitHub and well-documented

4. Literature Search

The nature of this problem is decently introductory and covered in 15-251/451 and as such does not require too many background readings. The $\frac{7}{8}$ approximation algorithm for 3SAT only holds for EXACT-3SAT when each clause has exactly 3 variables. The Karloff-Swick algorithm resolves this for general 3SAT however this is more complicated than the standard random algorithm.

Additionally many proof techniques in this area of theory requires knowledge of taking Integer Linear Programs and relaxing them into approximations. Since I have not taken a class covering linear programming and semidefinite programming in detail, I have started watch Ryan O'Donnell's 15-751 playlist on YouTube as readings to understand these ideas.

Other attached readings are either notes from Professor Guruswami or the basic examples on randomized algorithms. I have already read these readings however, as of writing this.

H. Karloff and U. Zwick, "A 7/8-approximation algorithm for MAX 3SAT?," *Proceedings 38th Annual Symposium on Foundations of Computer Science*, Miami Beach, FL, USA, 1997, pp. 406-415, doi: 10.1109/SFCS.1997.646129.

https://www.youtube.com/playlist?list=PLm3J0oaFux3ZYpFLwwrlv_EHH9wtH6pnX
http://www.cs.cmu.edu/~venkatg/teaching/15252-sp20/notes/csp-notes.pdf
https://cstheory.stackexchange.com/questions/9350/super-polynomial-time-approximation-algorithms-for-max-3sat
https://arxiv.org/abs/2007.10592

5. Resources Needed:

Due to this being a theory research, and the large amount of free resources to access research papers, I will not need any software/hardware resources of note.