Problem Set 1 for CS 540/495

Chen-Fu Chiang (please put all members here) *

September 17, 2018

Abstract

Briefly give the abtract of your work and what you aim to achieve in this report. Due on Oct. 3 and I require both the pdf file (in hard copy and the soft copy (the latex code and I will compile)). Please turn in one copy per group.

1 Introduction

In this problem set, your job is to explore research papers that used **simulated annealing (or any variant ones)** algorithm to attack problems that are of your interest. If you use Linux/Unix based system, the LaTeX to pdf package might have already been installed. If you are running windows, you might want to install MiKTex (http://miktex.org). It is expected the report should be typed up in LaTex[1]. Give an introduction to what is being solved and current status of this problem (how well it is being solved; and where the difficulty is).

2 The Background

Problem. *Formal description of the problem that is being solved, preferably the problem should be expressed mathematically.*

A brieif description of what has been used to solve this problem. **For instance:**

Problem. *A 3-SAT is defined as a formula F with M clauses and n binary variables. Let $N = 2^n$ be the number of possible assignments for n boolean variables. Let $V = \{v_1, v_2, \cdots, v_n\}$ be the set of boolean variables and $\bar{V} = \{\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_n\}$ be the complement set. A 3-SAT formula is described as*

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m \quad and \quad C_i = (l_1^i \vee l_2^i \vee l_3^i)$$

where (1) $\forall i \in \{1, M\}$, literal $l_j^i \in V$ or literal $l_j^i \in \bar{V}$ and (2) $\forall k \in \{1, n\}$, v_k or $\bar{v}_k$ appears at least once in F. The task is to find an assignment to $v_1, v_2, \cdots, v_n$ such that F is evaluated to 1.

*Department of Computer Science, State University of New York Polytechnic Institute, Utica, NY 13502, USA. Email: chiangc@sunyit.edu

3 The Algorithm

Briefly describe the algorithm used in the paper and connection to other existing others methods. And describe how they **translate** the problem into a max/min engery problem (that is either maximizing the gain or minimizing the loss; min and max are the same problem for simulated annealing, since just putting a '-' sign can easily turn max to min, min to max).

3.1 The Procedure

Describe how the algorithm is used (for instance) ; and maybe how it differs from the regular implementation (any clever ideas, such as nice data structure desgined for this problem for efficient storage purpose; or any interesting reduction in the algorithm steps;).

Algorithm 1 G: Naive instance generation algorithm

Require: (m, n) where n is the number of variables and m is the number of clauses
Ensure: A 3-SAT instance with certain probability that has a unique solution
  Start of algorithm
  Choose truth assignment $t \in \{0, 1\}^n$ randomly
  $F = \emptyset$
  for $i = 1, \cdots, m$ **do**
    Choose a clause C_i that can be satisfied by t randomly such that $F = F \wedge C_i$
  Output F
  End of algorithm

4 The Complexity & Discussion

The gain in complexity, theoretically or heuristically such as where it excels. Furthermore, can you describe where this can be further **improved** or any related work that can be benefit from this line of research?

5 Acknowledgments

C. C. gratefully acknowledges the support from XYZ.

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.