1 Global Heuristics

In the previous section we demonstrated local heuristics, that can be very efficiently implemented and one can rigorously analyze them. However, this benefits come with a cost of a significantly weaker performance on larger ratios of clauses to variables. In this section, we will present global heuristics, algorithms that in their free steps make decisions motivated by a global structure of the formula, rather than a local neighborhood. Currently, the most popular such heuristics are Belief Propagation (BP) and Survey Propagation (SP), both exhibiting a very good performance on large formulas with ratio close to the conjectured threshold of 4.2667. [1] (Maybe where to they come from, physics itp)

2 Belief Propagation

First, let's define a uniform distribution over the set of all satysfying assignments of given 3-CNF ϕ. Then, this distribution can be written down:

$$\mu(\mathcal{A}) = \frac{1}{Z}\mathcal{A}[\![\phi]\!] \tag{1}$$

where Z is the total number of satysfying assignments.

BP aims at computing marginals of this distribution, namely $\mu(\mathcal{A} \mid \mathcal{A}[\![x]\!] = b)$ for $b = 0, 1$. The algorithm itself is a message passing algorithm, that is, it procceds in rounds, where in each round the messages between clauses and variables are updated according to the equations below:

$$h^{r+1}_{i \to a} = \tanh\left\{ \sum_{b \in \partial_+ i(a)} u^r_{b \to i} - \sum_{b \in \partial_- i(a)} u^r_{b \to i} \right\} \tag{2}$$

$$u^{r+1}_{a \to i} = -\frac{1}{2}\log\left\{ 1 - \prod_{j \in \partial ai} \left(\frac{1 - h^r_{j \to a}}{2} \right) \right\} \tag{3}$$

In fact, Belief Propagation is a much more general algorithm, that can be applied to other combinatorial problems (Need for citation), however, in this report we will restrict ourselves to the 3-SAT case, which yields the relations above. Then, the message $u^r_{a \to i}$ is usually interpreted as a warning sent from clause a to variable i, and it informs variable i how much do the other variables inside clause a lean towards a polariry that violates a. With this interpretation in mind, the message $h^r_{i \to a}$ is treated as the probability that the variable i satisfies the clause a.

BP is presented in Algorithm 1. BP terminates when the messages converge, i.e. when the messages do not change significantly between two consecutive rounds, or when the number of rounds exceeds some predefined limit referred to by $maxiter$. Afterwards, the approximation to the marginals is computed as follows:

$$\mathcal{V}^r_x(b) = \frac{1}{2}\left(1 + (-1)^b \tanh\left\{ \sum_{a \in \partial_- i} u^r_{a \to i} - \sum_{a \in \partial_+ i} u^r_{a \to i} \right\} \right) \tag{4}$$

While for trees BP converges to the exact marginals, for general graphs it only approximates it[2]. Still, it provides us with a useful insight to the structure of the satysfying assignments, that we can exploit to make more informed decisions in the free steps, which is depicted in the Algorithm2.

Algorithm 1: Belief Propagation

Input: A k-SAT formula ϕ and messages $u^0_{a\to i}$ and $h^0_{i\to a}$
Output: Approximations $\{\mathcal{V}_{x_i}(b)\}_{i=1}^n$
for $r = 1, \ldots, maxiter$ **do**
  Calculate messages $u^r_{a\to i}$ and $h^r_{i\to a}$ according to equations 2 and 3;
  Compute the approximations $\mathcal{V}_x(b)$;
  if $\max_{x,b} |\mathcal{V}^r_x(b) - \mathcal{V}^{r-1}_x(b)| < \epsilon$ **then**
    Break
  end
end
return $\{\mathcal{V}_x(b)\}_{i=0}^n$

Algorithm 2: Belief Propagation Inspired Decimation

Input: A k-SAT formula ϕ
Output: *True* if ϕ is satisfiable, *False* otherwise
for $t = 1, \ldots, n$ **do**
  if *there exists pure literal* l *or unit clause* $\{l\}$ **then**
    satisfy l;
  else
    Compute the approximation of the marginals $\{\mathcal{V}_{x_i}(b)\}_{i=1}^n$ by running BP on ϕ ;
    Choose the most biased variable x^*, i.e. $x^* = \arg\max_x |\mathcal{V}_x(0) - \mathcal{V}_x(1)|$;
    Set x^* to $b \in \{0, 1\}$ that maximizes $\mathcal{V}_{x^*}(b)$ and adjust ϕ accordingly;
  end
end

Time Complexity

From previous section about PC and UC, we know that the steps taken by these heuristics run in $O(Nk\alpha)$, so we only need to analyze the complexity of Belief Propagation algorithm. First, we note that a single round of updates to the messages takes time $O(Nk\alpha)$, as we need to essentially iterate over all clauses. Hence, BP runs in $O(rNk\alpha)$, where r is the number of rounds. In fact, numerical experiments as well as theoretical considerations suggests that $r = O(\log N)$, which is motivated by the intuition that the expected distance between two nodes in our graph is $O(\log N)$. Hence, the overall time complexity of the heuristic is $O(N^2 k\alpha \log N)$.

In my implementation, I set *maxiter* to 200 and ϵ to 10^{-6}. Moreover, instead of running BP on each free step, I run it only having satisfied $\lceil f * N_t \rceil$ varaibles since previous propagation. It boosts the time complexity to $O(Nk\alpha \log^2 N)$, as BP is invoked only $O(\log N)$ times.

Empirical Results and Their Analysis

In this section, I present the empirical results of BP on large instances of 3-SAT.

References

[1] A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. 2002.

[2] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Solving constraint satisfaction problems through belief propagation-guided decimation. 2007.