An Empirical Analysis on Heuristic Local Search in 3-SAT Problems

Dong Xu
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
d26xu@uwaterloo.ca

***Abstract*—This work implemented the WalkSAT algorithm to search solutions for random 3-CNFs. Also, a greedy heuristic is applied for the choices from the WalkSAT algorithm. I discovered that a half-half policy for greediness has the best efficiency. When the number of clauses is about 4.25 times the number of variables, the probability of having a solution is about 0.45, and the probability drops steeply when the proportion is between 4.2 and 4.3.**

***Keywords*—random CNF, local search algorithm, SAT, WalkSAT, GSAT, density, heuristics, NP-Complete.**

I. Introduction

Constraint Satisfaction Problem (CSP) is one of the most important problem types in Artificial Intelligence, and CSP can be encoded into SAT. SAT is the first NP-Complete problem by Cook-Levin theorem [7] and has been studied for decades to improve the solver's performance [1]. 3-SAT is a subset of SAT where each clause has three literals. Any problem in SAT is reducible to a problem in 3-SAT. Thus 3-SAT is also a NP-Complete problem. If the density (proportion of the number of clauses to the number of variables) of the 3-CNF (Conjunctive Normal Form with 3 literals in each clause) reaches some threshold ratio [2], it is unlikely to be solvable. On the other hand, if a 3-CNF is solvable, we are interested in finding the solution (satisfiable assignment) with the highest efficiency. WalkSAT performs well on random 3-SAT instances of 5×10^5 variables with m to n ratio up to 4.2 [3].

In this project, I implemented a WalkSAT algorithm that applied different greediness to decide: 1. The most efficient ratio for greediness. 2. The (soft) threshold density for having solutions.

II. The Algorithm

$\boldsymbol{WalkSATRand}(N, M, K, P, L)$:

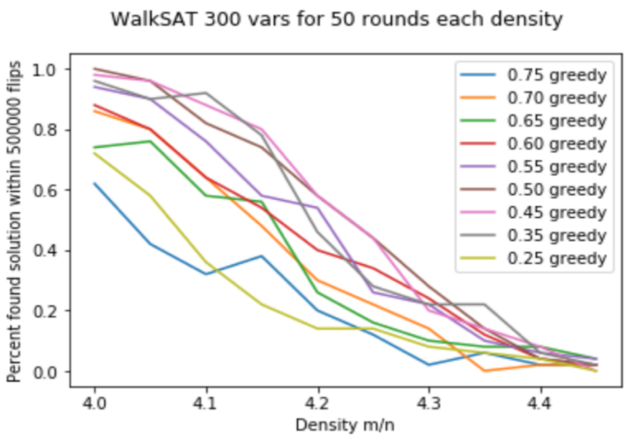
$\boldsymbol{Input}$: $N\ (number\ of\ varibles), M(number\ of\ clauses)$
$\boldsymbol{Parameters}$: $K = 3(literals\ per\ clause), P\ (probability\ to\ explore), L(limit\ on\ flips)$
$\boldsymbol{Output}$: $number\ of\ flips\ at\ termination, -1\ for\ exceeding\ the\ limit$
$\boldsymbol{begin}$
$\varphi \leftarrow \varphi_1\varphi_2 \cdots \varphi_M$
$\boldsymbol{for}\ m \leftarrow 1\ \boldsymbol{to}\ M\ \boldsymbol{do}$
$\varphi_{mk} \leftarrow x_n\ or\ \overline{x_n}\ with\ equal\ probability,\ \ k \leftarrow 1\ to\ K, n \leftarrow 1\ to\ N$
$\boldsymbol{for}\ n \leftarrow 1\ \boldsymbol{to}\ N\ \boldsymbol{do}$
$T(x_n) \leftarrow T\ or\ F\ with\ equal\ probability$
$C \leftarrow set\ of\ unsatisfied\ clauses\ in\ \varphi$
$flip \leftarrow 0$
$\boldsymbol{while}\ C\ is\ not\ empty\ \boldsymbol{do}$
$\boldsymbol{if}\ flip > L\ \boldsymbol{then}$
$\boldsymbol{return}\ flip$
$c \leftarrow a\ clause\ randomly\ chosen\ from\ C$
$next\ variable\ to\ flip: x_i$
$With\ probability\ p$:
$x_i \leftarrow a\ varible\ randomly\ selected\ in\ c$
$With\ probability\ p$:
$x_i \leftarrow a\ varible\ chosen\ in\ c\ whose\ flip\ will\ result\ in\ the\ smallest\ size\ of\ C$
$T(x_i) \leftarrow \neg T(x_i)$

$flip \leftarrow flip + 1$
$update\ C$
end while
return $flip$
end

III. HEURISTICS OF GREEDINESS

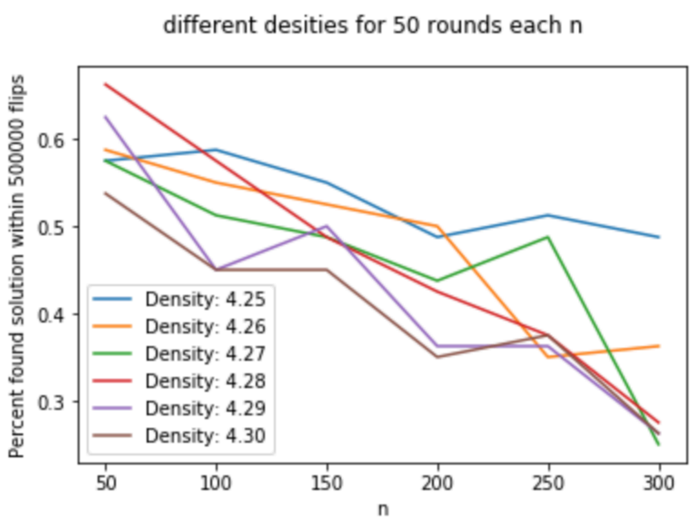
Plain WalkSAT is stochastic. Without any strategy, it cannot be optimal. GSAT is the purely greedy version of WalkSAT, which minimizes the set of unsatisfied clauses each iteration. However, plain GSAT performs poorly because it is easy to repeatedly fall into a locally best solution without exploring. The experiment enables WalkSAT to behave greedily at some probability (or some noise in GSAT) between 0.25 and 0.75. The instances are random 3-CNFs of 300 variables from density 4 to 4.45 with 0.05 increment. They were compared by the number of solutions found within 500,000 flips over 50 runs each.

The best heuristic is 45 percent of the time being greedy (0.55 noise), slightly better than 50 percent, which is close to 0.57 as the optimal noise level [4]. This heuristic is applied in the next experiment.



IV. THRESHOLD DENSITY FOR 3-SAT

With the knowledge from related work and the evidence in the curves from the previous experiment, we can find the threshold between 4.25 and 4.3 with high probability. This time, I focused on the behavior of curve when the density is fixed while n keeps increasing. The other parameters stay the same as before.



The flip limit kept constant at 500,000. Therefore, the curves are downward sloped as expected since more flips are required for a larger n. While the curves should behave differently for densities below and above the threshold (the curve may drop much more rapidly if the density is above threshold).

Densities 4.27, 4.28, 4.29 and 4.30 behaved similarly (dropped to a very low level when n goes to 300). The 4.25 curve dropped slowly within expectation. The 4.26 curve went down more than 4.25 but still deviated from 4.27 to 4.30. Thus, a possible deduction on the density threshold is in [4.25, 4.27] or [4.26, 4.27].

V. Summary and future work

The work above aimed to deduce a threshold density where random 3-SAT instances have s solution below that density. Due to the computation power and time limit, I could only conduct experiments on small instances with small number of trials. Algorithm and data-collection process are implemented in C++. It takes several minutes to exceed 500,000 flips per instance when n is 300. Comparing and analyzing the curve behaviors is the most effective way when working on small data. The optimal heuristic local search algorithm is necessary with limited time. So, the optimal noise level is investigated at the first hand.

Further research can be conducted on the density threshold for random k-SAT problems where k > 3. For sparse k-SAT instances, related works have shown that $exist\ a\ constant\ k_0 > 3\ such\ that\ for\ all\ k > k_0\ and\ all\ r \leq \frac{2^k}{6k^2}, stochastic\ WalkSAT\ will\ find\ a\ satisfying\ assignment\ within\ n\ steps\ with\ high\ probability$ [5]. This property still holds for even higher density $exist\ a\ constant\ k_0 > 3\ such\ that\ for\ all\ k > k_0\ and\ all\ r \leq \frac{2^k}{25k}, it\ can\ be\ done\ within\ n/k\ steps$ [6]. The density increases subexponentially with respect to k when WalkSAT still takes sublinear time. However, for any truth assignment, each clause has probability 2^{-k} to be unsatisfied. When $m \in \Omega(2^k)$, the expected number of unsatisfied clauses becomes $\Omega(1)$, which means the instance cannot be solve with high probability. Thus, the threshold should be sub-exponential somewhere between $\theta\left(\frac{2^k}{k}\right)$ and $\theta(2^k)$.

References

[1] Alouneh, S., Abed, S., Al Shayeji, M.H. et al. Artif Intell Rev (2018). https://doi.org/10.1007/s10462-018-9628-0

[2] Balint, A., and Schöning, U. 2012. Choosing probability distributions for stochastic local search and the role of make versus break. In Proc. of SAT-12, 16–29.

[3] Cai, S., Su, K., & Luo, C. (2013). Improving WalkSAT for Random k-Satisfiability Problem with k > 3. *AAAI*.

[4] Kroc, Lukas & Sabharwal, Ashish & Selman, Bart. (2010). An Empirical Study of Optimal Noise and Runtime Distributions in Local Search. 346-351. 10.1007/978-3-642-14186-7_31.

[5] A. Coja-Oghlan, U. Feige, A. Frieze, M. Krivelevich, D. Vilenchik: On smoothed k-CNF formulas and the Walksat algorithm. Proc. 20th SODA (2009) 451–460.5

[6] Coja-Oghlan, Amin & Frieze, Alan. (2011). Analyzing Walksat on Random Formulas. SIAM Journal on Computing. 43. 10.1137/12090191X.

[7] Wikipedia contributors. (2019, March 8). Boolean satisfiability problem. In *Wikipedia, The Free Encyclopedia*.