# On the Completeness of WalkSAT for 2-SAT[⋆]

Joseph Culberson[1] and Ian P. Gent[2]

[1] Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2H1. joe@cs.ualberta.ca

[2] School of Mathematical and Computational Science, University of St Andrews, St Andrews, Fife KY16 9SS, United Kingdom. ipg@dcs.st-and.ac.uk

**Report APES-15-1999**

September 1999

**Abstract.** WalkSAT is a highly successful local search algorithm for the Satisfiability problem. We show that for 2-SAT, it is complete in that it can reach a solution (if one exists) from any starting point. We leave open the more important case of 3-SAT.

Available from http://www.cs.strath.ac.uk/~apes/apesreports.html

WalkSAT is a highly successful local search algorithm for the Satisfiability problem [2]. The algorithm has been shown to work well on random 3-SAT formulae, and has been extensively studied, for example in [1]. We show that for 2-SAT, it is complete in that it can reach a solution (if one exists) from any starting point.

This is in response to a question posed by Holger Hoos. This 2-SAT case may be a known result, but it is new to us.

Let $\mathcal{S} = (\mathbf{U}, \mathbf{C})$ be an instance of SAT, and $\mathbf{T} : \mathbf{U} \rightarrow \{true, false\}$ be a truth assignment. The assignment $\mathbf{T}$ partitions $\mathbf{C}$ into unsatisfied and satisfied clauses, $\mathbf{C}_U$ and $\mathbf{C}_s$.

We use the notation $\mathbf{T}\widehat{u}$ to indicate the the truth assignment obtained from $\mathbf{T}$ by complementing the assignment of the variable $u$. A variable $u \in \mathbf{U}$ is said to be *free* (under an assignment $\mathbf{T}$) if $u$ or $\widehat{u}$ occurs in a clause $C \in \mathbf{C}_U$ and for all $d \in \mathbf{C}_S$ under $\mathbf{T}$, $d \in \mathbf{C}_S$ under $\mathbf{T}' = \mathbf{T}\widehat{u}$.

We can now describe the WALKSAT algorithm.

[⋆] Joseph Culberson's work was supported in part by an NSERC grant OGP8053 and an EPSRC visiting fellowship, GR/M54605. We thank members of APES for their help, and Holger Hoos for suggesting the problem to us. We are members of the APES research group, www.apes.cs.strath.ac.uk.

```
Assign $\mathbf{T}$ at random
While $\mathbf{C}_U \neq \emptyset$
    1. Select $C \in \mathbf{C}_U$
    2. if there is a free variable $u \in C$
        choose a free variable $u$
    else
        choose any $u \in C$
    3. $\mathbf{T} \leftarrow \mathbf{T}\widehat{u}$
end
```

The question we want answered is whether WALKSAT is *complete*; that is, is it true that for every satisfiable instance and for every initial assignment $\mathbf{T}$ there exists a sequence of selections which leads to some solution? If so then WALKSAT will succeed on satisfiable instances "eventually".

Notice that the only restriction on the algorithm's choices occurs when it selects an unsatisfied clause with a free variable. Without this restriction, it is easy to see that the algorithm is complete, since it can always set a variable to its final value until all clauses are satisfied.

Given a model (or solution) $\mathbf{T}_m$ (i.e. an assignment that satisfies all $\mathbf{C}$) we define the Hamming distance $h = h(\mathbf{T}, \mathbf{T}_m)$ to be the number of variables whose assignments differ under $\mathbf{T}$ and $\mathbf{T}_m$.

For purposes of simplifying the proof, we modify the selection rules by further constraining the algorithm to select $u$ such that $h$ is reduced in step 3 whenever possible. Clearly if this modified program is complete, then WALKSAT is complete.

To make the presentation more succinct we assume that $\mathbf{T}_m(u) = true, \forall u \in \mathbf{U}$. If there is an instance which shows WALKSAT incomplete, then we can create an equivalent instance with this property by complementing the appropriate literals throughout.

We introduce some notation. Each evaluated literal for a variable $u$ we refer to in any clause will be in one of the following forms

|  |  |
|---|---|
| *incorrect false* | $u_x^f$ |
| *correct true* | $u_c^t$ |
| *correct false* | $\widehat{u}_c^f$ |
| *incorrect true* | $\widehat{u}_x^t$ |

The correct ($c$) indicates the variable has the value assigned ($true$) in $\mathbf{T}_m$ while $x$ means it does not; $true$ ($t$) means the literal in the clause evaluates to $true$, while $f$ means it does not. Note there is a certain redundancy in the notation.

**Lemma 1.** *For all $C \in \mathbf{C}$, for all $\mathbf{T}$, there is some variable $u$ such that either $u_c^t \in C$ or $u_x^f \in C$. In particular, the latter case must hold for all $C \in \mathbf{C}_U$.*

*Proof.* $\mathbf{T}_m$ must make every clause true.

Note that $h$ is just the number of incorrect variables.

We define a *setup* as a solvable instance $S$ and an assignment $\mathbf{T}$ for which there is no selection sequence leading to a solution. If a setup exists, then every $\mathbf{T}$ reachable from it is also a setup. Furthermore, it a setup exists then there must be some minimum value $h'$ of $h$, where $h' > 0$, among all the setups. We designate one such minimal setup state by $\mathbf{T}_\alpha$.

We proceed to prove that WALKSAT is complete for 2-SAT by showing a contradiction on the minimality of $\mathbf{T}_\alpha$.

Consider the state $\mathbf{T}_\alpha$.

**Lemma 2.** *For each $C \in \mathbf{C}_U$ at $\mathbf{T}_\alpha$, there exists one $\widehat{u}_c^f \in C$ such that $u$ is free and the other $v_x^f \in C$ with $v$ not free.*

*Proof.* Since flipping the assignment of a variable changes $h$ and we start with $h'$ being the minimum over all states, every flip we are allowed to make must increase $h$. Thus, the flips must involve correct variables, and since $C$ in $\mathbf{C}_U$ the current values of these literals must be $false$. By lemma 1 the other literal must be $v_x^f$. If $v$ were also free, or $u$ was not free, then we could choose to flip $v$ reducing $h < h'$, a contradiction.

Lemma 2 says that we must climb up from the $h$ well. However, we cannot climb too far, or we would satisfy all the clauses in $\mathbf{T}_\alpha$, which would contradict the claim that it is a setup. In fact, the next lemma proves that we can be forced to make at most one increase in $h$ before having available a decreasing move which leads to another minimal state.

**Lemma 3.** *From a $\mathbf{T}_\alpha$ state, there exists free variables $\widehat{a}_c^f \in C_1$ and $\widehat{b}_c^f \in C_2$, $C_1, C_2 \in \mathbf{C}_U$, such that choosing to flip $a$ makes $b$ not free.*

*Proof.* Let $\mathbf{F} = \{a | \exists C \in \mathbf{C}_U, \widehat{a}_c^f \in C\}$, the set of free variables at $\mathbf{T}_\alpha$. If we were to flip all $a \in \mathbf{F}$ then we would satisfy all clauses that were in $\mathbf{C}_U$ at $\mathbf{T}_\alpha$. Since this is a setup, it must be that at some point in the sequence of flips we create another unsatisfied clause. By definition, this is only possible if at some point one of the free variables becomes not free.

Suppose after a (possibly empty) sequence of flips of variables in $\mathbf{F}$ a state $\mathbf{T}$ is reached where flipping free variable $a$ makes $b$ change from free to not free. This means there is a clause, call it $C'$ at $\mathbf{T}$ containing $a$ and $b$ which has two literals evaluating to $true$, but after the flip has only one. Since $\widehat{a}_c^f \in C_1$ and $\widehat{b}_c^f \in C_2$ at $\mathbf{T}$ it follows the clause $C'$ must be $\langle a_c^t \vee b_c^t \rangle$ at $\mathbf{T}$. Since we have only flipped free variables, and since after flipping the variables no longer occur in a false clause, this must also have been the status of $C'$ at $\mathbf{T}_\alpha$. But then at $\mathbf{T}_\alpha$ we could choose to flip $a$, making $b$ not free as claimed.

We are now in a position to complete the proof of

**Theorem 4.** *WALKSAT is complete for 2-SAT.*

*Proof.* Consider again $\mathbf{T}_\alpha$ and consider two clauses in $\mathbf{C}_U$, $C_1 = \left\langle \widehat{a}_c^f \vee d_x^f \right\rangle$ and $C_2 = \left\langle \widehat{b}_c^f \vee e_x^f \right\rangle$ with the property as described in lemma 3. First we flip $a$ making $b$ not free. Then, since we now have the choice, we flip $e$. (Note that $e$ is not $a$ since they have complementary settings at $\mathbf{T}_\alpha$). We now have a new state $\mathbf{T}_0$ with $h = h'$.

Since $e$ was not free at $\mathbf{T}_\alpha$ there must be a clause $C_3 \in \mathbf{C}_S, C_3 = \left\langle \widehat{e}_x^t \vee g_x^f \right\rangle$ at $\mathbf{T}_\alpha$. It must be $g_x^f \in C_3$ by lemma 1 and the fact that only the literal on $e$ made the clause *true* else $e$ would be free. Furthermore, since $g$ is originally incorrect, it could not be $a$. So the status of $C_3$ at $\mathbf{T}_2$ is $\left\langle \widehat{e}_c^f \vee g_x^f \right\rangle$ with $e$ not free. (The variable most recently flipped cannot be free since flipping is only possible if it occurred in a false clause.)

But this means we can now flip $g$ to its correct value, getting a new state with $h = h' - 1$, a contradiction.

The only critical usage we have made of 2-SAT is in lemma 3.

This proof fails for 3-SAT because at $\mathbf{T}_\alpha$ we can have three unsatisfied clauses with free variables, and one other clause containing all three free variables complemented. This forces at least two upward steps in $h$ before a downward step, and in this space many complications can be introduced. To prove completeness will require considerably more, and in fact we are not convinced even informally WALKSAT is complete on 3-SAT. Half or more of our time is currently being spent trying to find a setup for 3-SAT.

WALKSAT is only one of a set of possible random walk algorithms using a heuristic to attempt to improve the search. Another possibility is that after choosing an unsatisfied clause, select the variable in the clause that minimizes the number of satisfied clauses that become unsatisfied. Let us define $\kappa \in [-1, 0, 1, \ldots, \infty]$ as a parameter to the following algorithm.

```
Assign $\mathbf{T}$ at random
While $\mathbf{C}_U \neq \emptyset$
    1. Select $C \in \mathbf{C}_U$
    2. Set $chosen \leftarrow false$
    3. For $i = 0$ to $\kappa$ and not $chosen$ do
        if there is a $u \in C$ which causes
                $i$ clauses in $\mathbf{C}_S$ to move to $\mathbf{C}_U$
            choose $u$
            $chosen \leftarrow true$
    4. if not $chosen$
        choose $u$ at random from $C$
    5. $\mathbf{T} \leftarrow \mathbf{T}\widehat{u}$
end
```

Setting $\kappa = -1$ for this algorithm will allow a free choice of variable at every step. Since there is a variable $u_x^f$ in every clause in $\mathbf{C}_U$ the algorithm can just select this variable and move directly to the solution.

Setting $\kappa = 0$ is equivalent to the WALKSAT algorithm.

Setting $\kappa = 1$, the following is an example showing that the algorithm is incomplete on 2-SAT.

$$\begin{gathered}\langle a \vee c_x^f \rangle \\ \langle \widehat{a} \vee c_x^f \rangle \\ \langle \widehat{c}_x^t \vee d_x^f \rangle \\ \langle \widehat{c}_x^t \vee e_x^f \rangle\end{gathered}$$

Our proof shows that at any time WalkSAT can reduce the Hamming distance to a solution with non-zero probability independent of the time run so far. Thus at any time the algorithm has a non-zero probability of walking tothe solution. This implies the stronger property of 'probabilistically approximately completeness' defined by Hoos [1]. That is, the algorithm will almost certainly find a solution (if one exists) if left to run indefinitely. This property, directly only of theoretical interest, might have practical importance if it also means that in practice WalkSAT does not ever need to restart. Of course, our result will only be relevant to more practical application of WalkSAT if it can be extended to the NP-complete case of 3-SAT.

## References

1. Holger Hoos. On the run-time behaviour of stochastic local search algorithms for sat. In *Proceedings of AAAI-99*, 1999.
2. D. McAllester, B. Selman, and H. Kautz. Evidence for invariants in local search. In *Proceedings of AAAI-97*, pages 321–326, 1997.