

Local Search Techniques

Marijn J.H. Heule

**Carnegie
Mellon
University**

<http://www.cs.cmu.edu/~mheule/15816-f21/>

Automated Reasoning and Satisfiability

October 6, 2021

Random k -SAT

Stochastic Local Search

WalkSAT and ProbSAT

UnitWalk

Random k -SAT

Stochastic Local Search

WalkSAT and ProbSAT

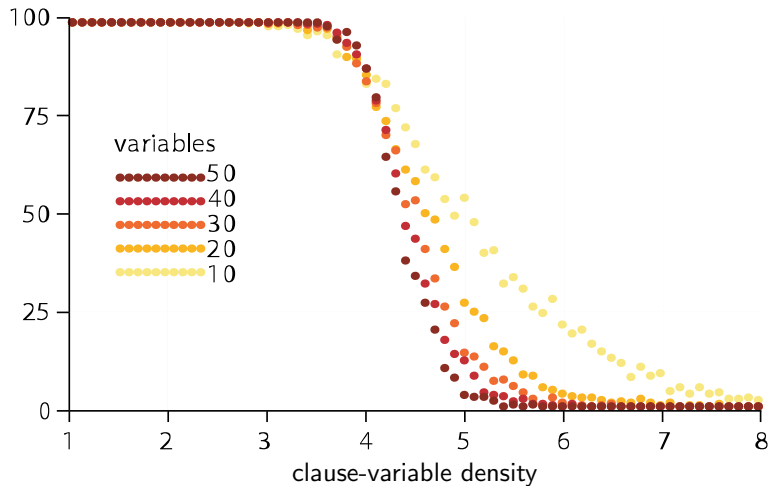
UnitWalk

Random k-SAT

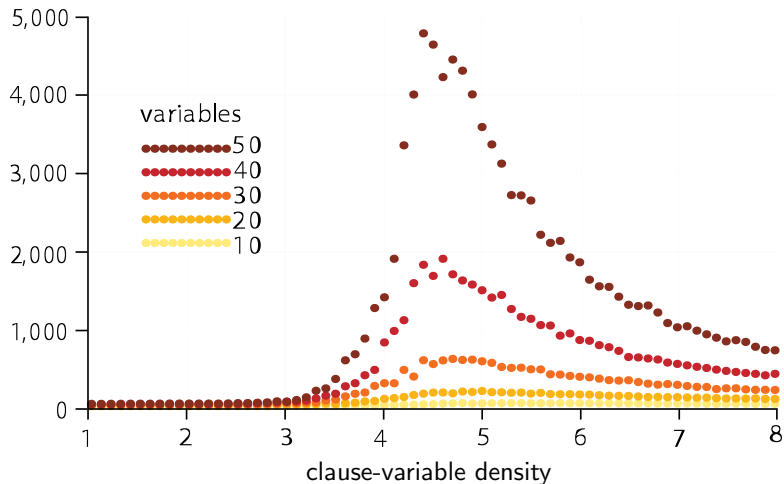
Local search solvers are particularly effective (and optimized) on hard uniform random (satisfiable) k-SAT problems

- All clauses have length k
- Variables have the same probability to occur
- Each literal is negated with probability of 50%
- Density is ratio Clauses to Variables

Random 3-SAT: % satisfiable, the phase transition



Random 3-SAT: exponential runtime, the threshold



Random k -SAT

Stochastic Local Search

WalkSAT and ProbSAT

UnitWalk

Local Search: Generic structure

Generic structure of local search SAT solvers

```
1: for i in 1 to MAX_TRIES do  
2:    $\alpha :=$  random initial assignment  
3:   for j in 1 to MAX_STEPS do  
4:     if  $\alpha$  satisfies  $\mathcal{F}$  then  
5:       return satisfiable  
6:     end if  
7:      $\alpha :=$  Flip ( $\alpha$ )  
8:   end for  
9: end for  
10: return unknown
```


Local Search: Global vs Local flips

Global flips

- Pro: Big improvements
- Neg: Probabilistic incomplete

Local flips

- Neg: Small improvements
- Pos: Probabilistic complete

Random k -SAT

Stochastic Local Search

WalkSAT and ProbSAT

UnitWalk

Local Search: Types of Flips

Select a random unsatisfied clause C

- Free flip
- Random flip
- Heuristic flip

Local Search: WalkSAT Code [Selman, Kautz, and Cohen '93]

FLIP_WALKSAT (α)

- 1: $C :=$ random falsified clause by $\alpha \circ \mathcal{F}$
- 2: **if** a variable in C can be flipped for free **then**
- 3: flip in α that variable
- 4: **else**
- 5: flip in α with p a random $x_i \in C$
- 6: flip in α with $1 - p$ the “optimal” $x_i \in C$
- 7: **end if**
- 8: **return** α

Local Search: ProbSAT [Balint and Schönig '12]

ProbSAT generalizes the WalkSAT code.

Let $\text{break}(x, \alpha)$ denote the number of clauses that are **only satisfied** by x or \bar{x} under the assignment α

- $C :=$ random falsified clause by $\alpha \circ \mathcal{F}$
- randomly pick a variable x in C using **weights** $c^{-\text{break}(x, \alpha)}$
- an effective constant for random 3-SAT: $c = 2.5$
- update α by flipping x

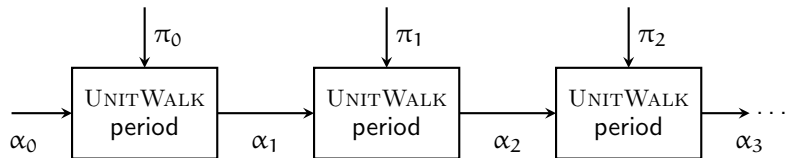
Random k -SAT

Stochastic Local Search

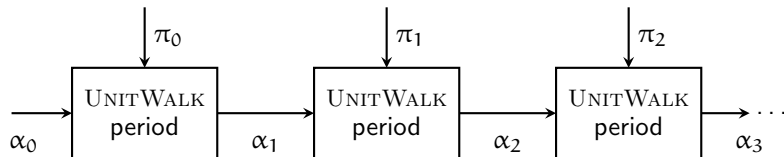
WalkSAT and ProbSAT

UnitWalk

The UnitWalk Algorithm



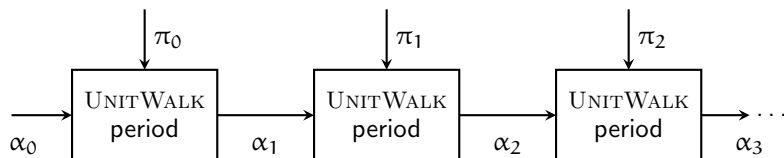
The UnitWalk Algorithm



The general idea of a UNITWALK period:

- Within each unsatisfied clause in $\alpha_i \circ \mathcal{F}$ the assignment to the least important variable (based on π_i) is flipped

The UnitWalk Algorithm



The general idea of a UNITWALK period:

- Within each unsatisfied clause in $\alpha_i \circ \mathcal{F}$ the assignment to the least important variable (based on π_i) is flipped

For example:

- $\mathcal{F} = (x \vee \bar{y})$, $\alpha_0 = \{x = 0, y = 1\}$, $\pi_0 = \{y, x\}$

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = *, \chi_2 = *, \chi_3 = *, \chi_4 = *\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

extend α_{active} with most important free variable according to π

while α_{active} contains *'s

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = *, \chi_2 = *, \chi_3 = *, \chi_4 = *\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

→ iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

extend α_{active} with most important free variable according to π

while α_{active} contains *'s

Action:

- no unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = *, \chi_2 = 1, \chi_3 = *, \chi_4 = *\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

→ extend α_{active} with most important free variable according to π

while α_{active} contains *'s

Action:

- extend α_{active} with $\chi_2 := 1$ (the truth value in α_{master})

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = *, \chi_2 = 1, \chi_3 = 0, \chi_4 = *\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

→ iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

extend α_{active} with most important free variable according to π

while α_{active} contains *'s

Action:

- detected unit clause $\bar{\chi}_3 \rightarrow \chi_3 := 0$

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = *, \chi_2 = 1, \chi_3 = 0, \chi_4 = 0\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

→ iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

extend α_{active} with most important free variable according to π

while α_{active} contains *'s

Action:

- detected unit clauses χ_4 and $\bar{\chi}_4$ → conflict
- assign χ_4 to truth value in α_{master} → $\chi_4 := 0$

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 0, \chi_4 = 0\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

→ extend α_{active} with most important free variable according to π

while α_{active} contains *'s

Action:

- extend α_{active} with $\chi_1 := 0$ (the truth value in α_{master})

UnitWalk Period Example

$$\begin{aligned}\mathcal{F}_{\text{example}} &:= (\chi_1 \vee \chi_2) \wedge (\bar{\chi}_1 \vee \chi_2 \vee \chi_3) \wedge (\bar{\chi}_2 \vee \bar{\chi}_3) \wedge \\ &\quad (\bar{\chi}_2 \vee \chi_3 \vee \bar{\chi}_4) \wedge (\bar{\chi}_2 \vee \chi_3 \vee \chi_4) \wedge (\bar{\chi}_3 \vee \bar{\chi}_4) \\ \alpha_{\text{master}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 1, \chi_4 = 0\} \\ \alpha_{\text{active}} &:= \{\chi_1 = 0, \chi_2 = 1, \chi_3 = 0, \chi_4 = 0\} \\ \pi &:= (\chi_2, \chi_1, \chi_4, \chi_3)\end{aligned}$$

do

iterative propagate unit clauses in $\alpha_{\text{active}} \circ \mathcal{F}_{\text{example}}$

extend α_{active} with most important free variable according to π

→ **while** α_{active} contains *'s

Action:

- end of period because all variables are assigned in α_{active}

Local Search Techniques

Marijn J.H. Heule

**Carnegie
Mellon
University**

<http://www.cs.cmu.edu/~mheule/15816-f21/>

Automated Reasoning and Satisfiability

October 6, 2021