THE UNIVERSITY of ADELAIDE

A Parameterized Runtime Analysis of Evolutionary Algorithms for MAX-2-SAT

Jareth Day, Andrew M. Sutton and Frank Neumann

School of Computer Science
University of Adelaide, Australia

Genetic and Evolutionary Computational Conference
July 2012

Introduction

- Introduce MAX-2-SAT and FPT
- Parameterized complexity analysis on MAX-2-SAT
  - Identify fitness landscape of MAX-2-SAT
  - Produce parameterized algorithms for MAX-(2,3)-SAT.

MAX-2-SAT

- Maximum 2-Satisfiability Problem
- $\mathcal{C} = \{(\ell_{1,1} \vee \ell_{1,2}), (\ell_{2,1} \vee \ell_{2,2}), \ldots, (\ell_{m,1} \vee \ell_{m,2})\}$
- eg $\mathcal{C} = \{(\neg v_1 \vee v_2), (v_1 \vee v_3), \ldots, (v_6 \vee v_n)\}$
- m clauses, $\mathcal{C}_i = \{\ell_1, \ldots, \ell_m\}$
- n Boolean variables v_i = true/false

MAX-2-SAT

- $\mathcal{C} = \{(\neg v_1 \lor v_2), (v_1 \lor v_3), \ldots, (v_6 \lor v_n)\}$
- Given $x \in \{0,1\}^n$,
  $x_i = 1$ corresponds to $v_i = true$,
  $x_i = 0$ corresponds to $v_i = false$
- We want to maximize $f: \{0,1\}^n \to \{0\} \cup [m]$
- $f(x)$ = number of clauses satisfied by x

Parameterized Complexity

- MAX-2-SAT is NP-Hard.
- Standard algorithms: exp(|x|)
- Parameterized complexity: parameterization K
- XP algorithms: $|x|^{g(K(x))}$
  eg n^k
- Fixed-parameter tractable (FPT) algorithms: $g(K(x)) \cdot |x|^{O(1)}$
  eg $n \cdot 2^k$

MAX-2-SAT Algorithm

- We analyze the runtime of the (1+1) EA:

Algorithm 1: The (1+1) EA.

1 Choose x uniformly at random from $\{0,1\}^n$;
2 **repeat** *forever*
3 $\quad x' \leftarrow \texttt{mutate}(x)$;
4 $\quad$ **if** $f(x') \geq f(x)$ **then** $x \leftarrow x'$

- where mutate() negates some elements of x

Uniform-Complement Mutation

- Traditional uniform mutation creates offspring by flipping each bit of x with probability $1/n$
- Uniform-complement may, with uniform probability, produce the complement of x with probability $\Theta(1)$.
    - Complement under uniform mutation: probability $O(n^{-n})$

Uniform-Complement Mutation: Fitness Landscape

- MAX-2-SAT fitness function and uniform-complement operator corresponds to an elementary landscape.
- Can reach solutions of certain quality in polynomial time by making local improvements.

Uniform-Complement Mutation: Fitness Landscape

- Let $N(x)$ be union of the Hamming neighbors of x and the complement of x
- If the i-th clause is not satisfied by x, it is satisfied for three neighbors $y \in N(x)$:
  - The two Hamming neighbors of x that have the variables in the i-th clause negated, and
  - The complement of x.
- If the i-th clause is satisfied by x, at least one of its literals evaluates to true under x.
  - If only one true, clause is satisfied for all $y \in N(x)$ except for the negation of variable involved in the true literal.
  - If both true, clause is satisfied for all $y \in N(x)$ except for the complement.

Uniform-Complement Mutation: Fitness Landscape

- If clause i unsatisfied by x, clause satisfied by three neighbors.
- If clause i satisfied by x, clause satisfied by $|N(x)|$ - 1 neighbors.
- Let $c_i : \{0,1\}^n \rightarrow \{0,1\}$ if clause i is satisfied by x.

$$\sum_{y \in N(x)} c_i(y) = 3(1 - c_i(x)) + (|N(x)| - 1)c_i(x) = 3 + (n-3)c_i(x)$$

- Since $f(x)$ is the sum of the clauses satisfied in x

$$\sum_{y \in N(x)} f(y) = \sum_{i=1}^{m} (3 + (n-3)c_i(x)) = 3m + (n-3)f(x)$$

Uniform-Complement Mutation: Fitness Landscape

- Until no further improvements can be made, there are two cases in which an improvement is generated.
  - Complement is improving state.
    Probability ½ to choose.
  - Hamming neighbor is improving state.
    Probability $\frac{1}{2}(n^{-1}(1 - n^{-1})^{n-1}) \geq (2en)^{-1} = \Omega(n^{-1})$
- Number of improvements bounded by number of clauses.
- Reaches state with no improvements in expected time bounded by $O(mn)$

Uniform-Complement Mutation: Fitness Landscape

- Reached solution *x'* s.t. *f*(*x'*) has best fitness in neighborhood
- Current state $x = x'$

$$\frac{1}{|N(x')|}\sum_{y \in N(x)} f(y) \leq f(x')$$

$$\frac{1}{|N(x')|}\sum_{y \in N(x)} (3m + (n-3) f(x')) \leq f(x')$$

$$\frac{3m}{(n+1)} + \frac{(n-3)}{(n+1)} f(x') \leq f(x')$$

$$f(x') \geq \frac{3}{4} m$$

MAX-(2,3)-SAT

- Restricted problem: MAX-(2,3)-SAT
- Each variable may only appear in at most 3 clauses
- eg $\mathcal{C} = \{(\neg v_1 \lor v_2), (v_1 \lor \neg v_3), (\neg v_1 \lor \neg v_5), \ldots\}$
- Still NP-hard

MAX-(2,3)-SAT

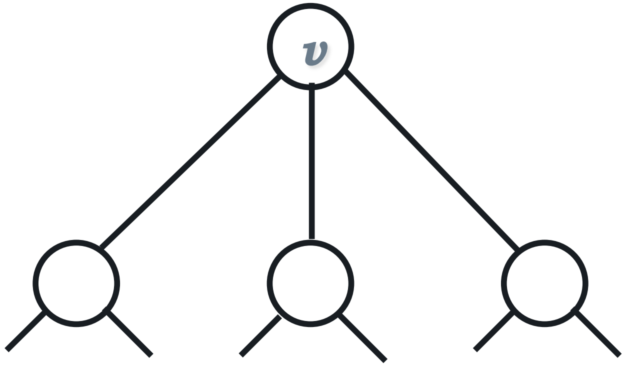
- Graph: $G(V,E)$

  $|V| = n$
  $E = \{\{u,v\} \subset V \mid u \text{ and } v \text{ appear together in a clause}\}$

  - As long as there are two variables in a clause, there's an edge.

- Diameter of G: maximum shortest-path distance in any of the connected components
- Parameter: diameter of G is bounded by k

MAX-(2,3)-SAT

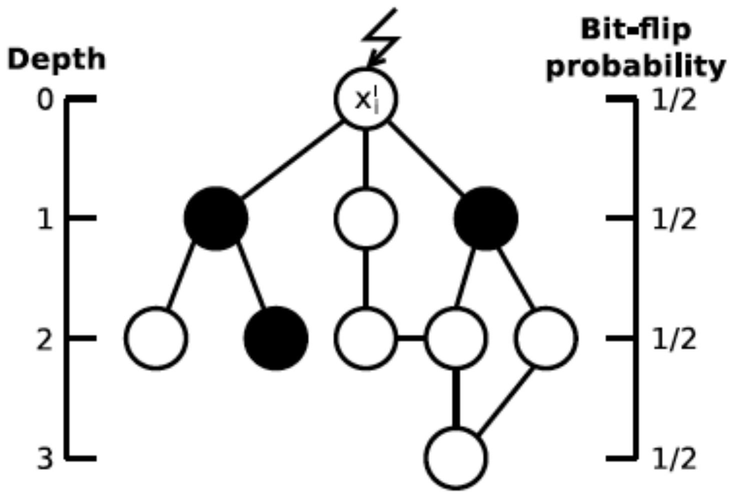


- $C(v)$ is a connected component containing node v
- With diameter bounded by k, the number of nodes in $C(v)$ is bound by:

$$1 + \sum_{i=0}^{k-1} 3 \cdot 2^{i} = 3 \cdot 2^{k} - 2 \geq |C(v)|$$

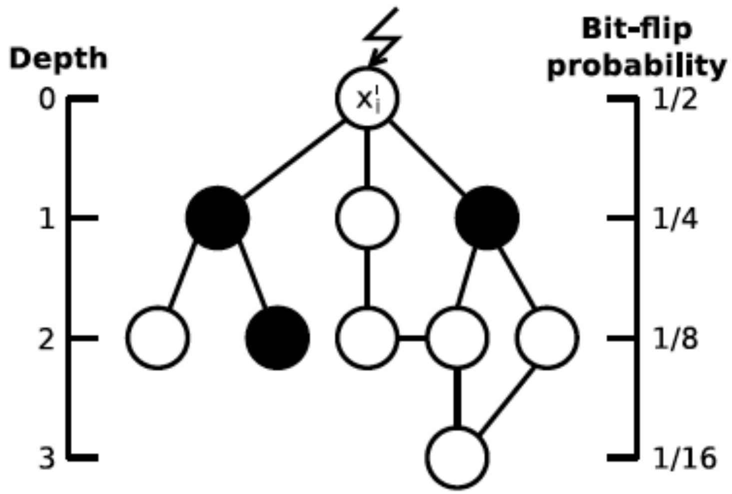
Basic FPT Algorithm

- Select v_i uniformly at random.
- Flip all bits in x associated with Boolean variables in $C(v_i)$ with probability 1/2



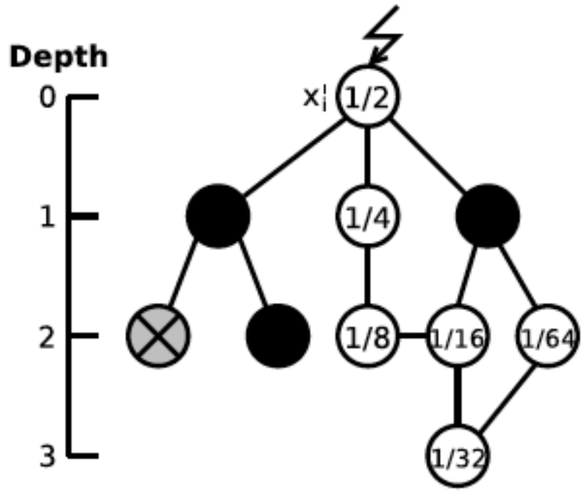
Modified FPT Algorithm

- Select v_i uniformly at random.
- Flip all bits in x associated with Boolean variables in $C(v_i)$ with decreasing probability the further the distance from v_i



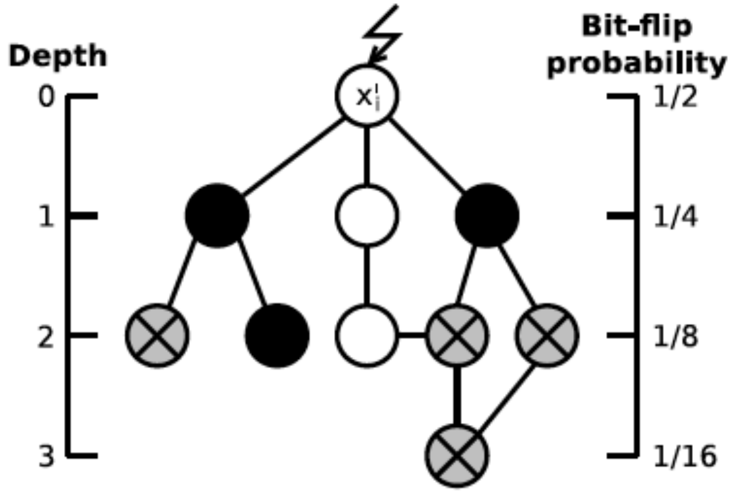
Propagation FPT Algorithm

- Select v_i uniformly at random.
- Flip all bits in x associated with Boolean variables in $C(v_i)$ with decreasing probability the further the distance from v_i. Only propagates to next bit if current bit flipped.



Modified Propagation FPT Algorithm

- Select v_i uniformly at random.
- Flip all bits in x associated with Boolean variables in $C(v_i)$ with decreasing probability the further the distance from v_i. Only propagates to next bit if current bit flipped. Only visits bits at increasing distance from v_i.



Expected Time to Completion

- Number of nodes in connected component bound by $g(k) = 3 \cdot 2^k - 2$

XP (in paper)	$O(n^{g(k)})$
Simple FPT	$O(n \log n \cdot 2^{g(k)})$
Modified FPT	$O(n \log n \cdot 2^{k \cdot g(k)})$
Propagation FPT	$O(n \log n \cdot 2^{g(k)^2})$
Modified Propagation FPT	$O(n \log n \cdot 2^{k \cdot g(k)})$

Conclusion

- We analyzed MAX-2-SAT and MAX-(2,3)-SAT and determined:
  - Elementary landscape of MAX-2-SAT
  - XP evolutionary algorithm for MAX-(2,3)-SAT
  - FPT evolutionary algorithms for MAX-(2,3)-SAT