Artificial Intelligence 173 (2009) 240–257

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

A comparative runtime analysis of heuristic algorithms for satisfiability problems

Yuren Zhou [a,c,*], Jun He [b], Qing Nie [c]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China
[b] Department of Computer Science, University of Wales, Aberystwyth, Ceredigion, SY23 3DB, UK
[c] Department of Mathematics, University of California, Irvine, CA 92697-3875, USA

ARTICLE INFO

Article history:
Received 18 March 2008
Received in revised form 23 July 2008
Accepted 1 November 2008
Available online 7 November 2008

Keywords:
Boolean satisfiability
Heuristic algorithms
Random walk
$(1+1)$ EA
Hybrid algorithm
Expected first hitting time
Runtime analysis

ABSTRACT

The satisfiability problem is a basic core NP-complete problem. In recent years, a lot of heuristic algorithms have been developed to solve this problem, and many experiments have evaluated and compared the performance of different heuristic algorithms. However, rigorous theoretical analysis and comparison are rare. This paper analyzes and compares the expected runtime of three basic heuristic algorithms: RandomWalk, $(1+1)$ EA, and hybrid algorithm. The runtime analysis of these heuristic algorithms on two 2-SAT instances shows that the expected runtime of these heuristic algorithms can be exponential time or polynomial time. Furthermore, these heuristic algorithms have their own advantages and disadvantages in solving different SAT instances. It also demonstrates that the expected runtime upper bound of RandomWalk on arbitrary k-SAT ($k \geqslant 3$) is $O((k-1)^n)$, and presents a k-SAT instance that has $\Theta((k-1)^n)$ expected runtime bound.

1. Introduction

The satisfiability problem (SAT) of a propositional formula plays a central role in computer science and artificial intelligence. It is the first proposed NP-complete problem [5,21] and one of the basic core NP-complete problems [10]. In addition to its theoretical importance, the SAT problem is also directly applied in VLSI formal verification, software automation, and so on.

Researchers have been trying to look for an effective algorithm for the SAT problem. Since the SAT problem is an NP-complete problem in nature, a polynomial algorithm is not currently available to solve it, although we cannot prove that such an algorithm does not exist. In fact, a basic conjecture of modern computer science and mathematics is that no polynomial algorithm exists for NP-complete problems. At present, the main methods for solving the SAT problems are complete algorithms [3,6,34] and incomplete algorithms [7,12,13,15,20,25,27,29,31,32]. There are several very successful complete algorithms (e.g., SATO [34]). A complete algorithm often explores the whole search space and can always determine whether a given propositional formula is satisfiable or not; however, its time complexity is usually exponential. An incomplete algorithm does not carry out a complete search on the search space; instead, it often explores some part of the search space using heuristic information within a limited time; however it does not give the correct answer with certainty.

Since the 1990s, the use of incomplete algorithm for solving the SAT problem has grown quickly. The basic incomplete heuristic methods are RandomWalk algorithm [25], GSAT algorithm [13,31], WalkSat algorithm [32], UnitWalk [15],

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China.
E-mail addresses: yrzhou@scut.edu.cn (Y. Zhou), jun.he@ieee.org (J. He), qnie@math.uci.edu (Q. Nie).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2008.11.002

population-search-based evolutionary algorithms [7,12,20] and so on. In recent years, some powerful concepts and techniques of statistical physics have been applied to the SAT problem. One of these incomplete algorithms, known as "survey propagation" [4,22], which is based on statistical physics methods, shows good performance on some difficult randomly generated SAT instances. It is well known that one of the earliest applications of statistical physics in the optimization problem is the simulated annealing algorithm [19]. WalkSat [32] used a probability selection mechanism similar to that of the simulated annealing algorithm.

For some heuristic algorithms for the SAT problem, theoretical results about computational complexities have been obtained to some extent. Papadimitiou [25] was the first to prove that the average time upper bound of RandomWalk for 2-SAT is $O(n^2)$. Schöning [29] presented a restarting local-search algorithm to show that, for any satisfiable k-CNF formula with n variables, the algorithm has to repeat $O((2(1-\frac{1}{k}))^n)$ times, on average, to find a satisfying assignment. Specially if $k=3$, the average time is $O(1.334^n)$ (the upper bound of an exhaustive search is $O(2^n)$). There have been several improvements on the upper bound by hybrid algorithms based on randomized algorithms by Paturi et al. [27] and Schöning [29], e.g. $O(1.324^n)$ [18] and $O(1.322^n)$ [28]. Alekhnovich et al. [2] proved that, when the clause density is less than 1.63, the average time complexity of RandomWalk for 3-SAT is linear.

Since there are many incomplete heuristic algorithms for SAT problems, comparing and understanding the working principals of these heuristic algorithms is useful. The first thing we have to accept is that no one algorithm beats all other algorithms on all problems. There have been many numerical experiments that compared various heuristic algorithms on SAT problems, but theoretical study has been rare. This paper analyzes and compares the expected running time of three basic heuristic algorithms: RandomWalk, $(1+1)$ EA, and hybrid algorithm. We use absorbing Markov chains to model search processes of these heuristic algorithms, and use explicit expressions of the first hitting time of a Markov chain to analyze and estimate their expected runtime. Through runtime analysis of three SAT instances, we show that the expected runtime of these heuristic algorithms can be exponential or polynomial. We also find that these heuristic algorithms have their own comparative advantage under different circumstances.

The rest of this paper is organized as follows. Section 2 introduces the concepts of the SAT problem, some heuristic algorithms for the SAT problem, and the first hitting time of an absorbing Markov chain. Section 3 discusses the worst-case bound and the worst-case example on RandomWalk. Section 4 analyzes and compares the expected runtime bounds of three heuristic algorithms on two 2-SAT instances. Section 5 presents our conclusions and suggestions for further research.

2. Heuristic algorithms for satisfiability and the first hitting time of the Markov chain

2.1. The SAT problem

We begin by stating some definitions and notations that will be used throughout the paper.

In Boolean logic, a literal is a variable or its negation, and a clause is a disjunction of literals. The formula $f=c_1\wedge c_2\wedge\cdots\wedge c_m$ is in k conjunctive normal form (k-CNF) if it is a conjunction of clauses with each clause as a disjunction of at most k literals. We view a CNF Boolean formula as both a Boolean function and a set of clauses. Satisfiability is the problem of determining whether the variables of a given Boolean formula can be assigned truth values in such a way as to make the formula evaluate to true.

SAT is originally stated as a decision problem. In this paper we consider the more general MaxSAT, so, our goal is to look for an assignment that satisfies the maximum number of clauses.

Evolutionary algorithms (EAs) are the heuristic algorithms that have been applied to SAT and to many other NP-complete problems. EAs usually use a fitness value to guide the search process. In the MaxSAT formulation, the fitness value is defined as the number of satisfied clauses, i.e.

$$fit(x)=c_1(x)+c_2(x)+\cdots+c_m(x) \tag{1}$$

where $c_i(x)$ $(1\leqslant i\leqslant m)$ represents the true value of the ith clause. This fitness function is used in most EAs for SAT problems.

Throughout this paper, for $x=(x_1\cdots x_n)$, $y=(y_1\cdots y_n)\in\{0,1\}^n$, we denote by $H(x,y)$ the Hamming distance between two points x and y, i.e. $H(x,y)=\sum_{i=1}^n|x_i-y_i|$. We also denote $|x|=x_1+\cdots+x_n$, and let $S_i=\{x\mid x\in S=\{0,1\}^n,\ |x|=i\}$ $(i=0,1,\ldots,n)$ be a partition of search space $S=\{0,1\}^n$.

2.2. Heuristic algorithms for the SAT problem

RandomWalk, first introduced by Papadimitiou [25], is one of the most basic incomplete algorithms, and many other heuristics have been developed based on the improvement of this algorithm, e.g. the Walk-SAT [32], combines RandomWalk with a greed bias towards assignments that satisfy more clauses. RandomWalk algorithm first randomly selects a clause that is not satisfied with the CNF, then randomly selects a flip in the clause (see Algorithm 1).

Algorithm 1 *(The RandomWalk algorithm).*

begin
  initialization: Select an initial bit string x at random;
  while (termination-condition does not hold) **do**
    Select $c :=$ an unsatisfied clause chosen at random;
    Select $x_i :=$ a variable in c chosen at random;
    Flip the value of x_i;
  od
end

Evolutionary algorithms are inspired from modeling the processes of natural selection and genetic evolution. Here we consider a simple EA using mutation and selection approaches with population size of 1 denoted as $(1+1)$ EA [9]. $(1+1)$ EA is a simple but effective random hill-climbing EA. Its general description is:

Algorithm 2 *($(1+1)$ EA).*

begin
  initialization: Choose randomly an initial bit string x;
  while (termination-condition does not hold) **do**
    Mutation: $y :=$ mutate(x);
    Selection: If fitness(y) $>$fitness(x), $x := y$;
  od
end

$(1+1)$ EA generally uses two kinds of mutation, called local mutation and global mutation:

(1) Local mutation randomly chooses a bit x_i $(1 \leqslant i \leqslant n)$ from the individual $x = (x_1 \cdots x_n) \in \{0,1\}^n$ and flips it.
(2) Global mutation flips each bit of individual $x = (x_1 \cdots x_n) \in \{0,1\}^n$ independently with the probability of $\frac{1}{n}$. The expected number of bit flips for the global mutation is 1.

The hill-climbing algorithm is usually trapped in a region which is a local optimum and needs to be restarted with a random new assignment. Another widely-used mechanism for escaping such a local optimum of the maximization problem is to permit the search to make occasional downhill moves. The following hybrid strategy, which combines $(1+1)$ EA and RandomWalk, is closely related to WalkSat [32] in that it allows for the possibility of downhill moves.

Algorithm 3 *(The hybrid algorithm of local $(1+1)$ EA and RandomWalk).*

begin
  initialization: Set parameters, choose randomly an initial bit string;
  while (termination-condition does not hold) **do**
    With probability p, follow the RandomWalk scheme;
    With probability $1-p$, follow the Local $(1+1)$ EA scheme;
  od
end

2.3. The absorbing Markov chain

Most heuristic algorithms are memory-less in the sense that the processes of selecting the next point in the search space depend only on the current point. This allows us to model these search processes as absorbing Markov chains whose absorbing set is the optimal solution (s). Such models are widely used in two heuristic algorithms: Simulated Annealing [1] and Genetic Algorithms [14,24]. Basic knowledge of absorbing Markov chains can be found in any literature regarding random processes, e.g. [17].

Let $(X_t;\ t = 0, 1, \ldots)$ denote a discrete homogeneous absorbing Markov chain in a finite state space S. T is the transient state set, $H = S - T$ is the absorbing set. Assume there are r absorbing states and t transient states, i.e. $|T| = t$ and $|H| = r$ where $|\cdot|$ denotes the cardinality of a set, then the transition matrix can be written in the canonical form as

$$\mathbf{P} = \begin{pmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{R} & \mathbf{Q} \end{pmatrix}$$

where $\mathbf{I}$ is an r-by-r identity matrix, $\mathbf{O}$ is an r-by-t zero matrix, $\mathbf{R}$ is a nonzero t-by-r matrix, and $\mathbf{Q}$ is a t-by-t matrix. For the power of $\mathbf{P}$, a standard matrix argument shows that the region $\mathbf{I}$ remains $\mathbf{I}$. This corresponds to the fact that once the Markov chain reaches an absorbing state, it will never leave that absorbing state.

Definition 1. Let $(X_t;\ t = 0, 1, \ldots)$ be an absorbing Markov chain. The first hitting time from status i $(i \in S)$ to the absorbing status set H is:

$$\tau_i = \min\{t\colon\ t \geqslant 0,\ X_t \in H \mid X_0 = i\}$$

if the right-hand side involves the empty set, let $\tau_i = \infty$.

We are interested in the question: Given that the chain starts in state i, what is the expected number of steps before the chain is absorbed? Theorem 1 provides an answer.

Theorem 1. *Given that the absorbing Markov chain X_t starts in transient state i, let m_i be the expected number of steps before the chain is absorbed, i.e. $m_i = E[\tau_i]$. Denote $\mathbf{m} = [m_i]_{i \in T}$. Then*

$$\mathbf{m} = (\mathbf{I} - \mathbf{T})^{-1}\mathbf{1} \tag{2}$$

where **1** *represents the column vector all of whose entries are* 1.

Proof. See Ref. [17]. □

Several corollaries can be derived directly from Theorem 1.

Corollary 1. *Let $\{X_t \mid t = 0, 1, \ldots\}$ be an absorbing Markov chain with finite state space $S = \{0, 1, \ldots, n, n+1\}$ and absorbing state set $\{0, n+1\}$, and its transition probabilities are defined as follows*:

(1) *For $i = 0$ or $n+1$,*

$$p_{ij} = \begin{cases} 1, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(2) *For $1 \leqslant i \leqslant n$,*

$$p_{ij} = \begin{cases} a_i, & j = i-1, \\ b_i, & j = i+1, \\ 1 - a_i - b_i, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Then for this absorbing Markov chain, its mean first hitting time to the absorbing state is given by

$$\begin{cases} m_{n+1} = 0, \\ m_n = \dfrac{\frac{1}{a_1} + \sum_{h=1}^{n-1} \frac{1}{a_{h+1}} (\sum_{j=1}^{h} \prod_{i=j}^{h} \frac{b_i}{a_i} + 1)}{1 + \sum_{j=1}^{n} \prod_{i=j}^{n} \frac{b_i}{a_i}}, \\ m_{n-1} = m_n (1 + \frac{b_n}{a_n}) - \frac{1}{a_n}, \\ m_{k-1} = m_k + m_n \prod_{i=k}^{n} \frac{b_i}{a_i} - \sum_{j=0}^{n-(k+1)} \frac{1}{a_{j+k+1}} \prod_{i=0}^{j} \frac{b_{k+i}}{a_{k+i}} - \frac{1}{a_k}, \quad k = n-1, \ldots, 2, \\ m_0 = 0. \end{cases}$$

Corollary 2. *Let $\{X_t \mid t = 0, 1, \ldots\}$ be an absorbing Markov chain with finite state space $S = \{0, 1, \ldots, n\}$ and absorbing state set* 0, *and its transition probabilities are defined as follows*:

(1) *For $i = 0$ or $n+1$,*

$$p_{ij} = \begin{cases} 1, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(2) *For $1 \leqslant i < n$,*

$$p_{ij} = \begin{cases} a_i, & j = i-1, \\ b_i, & j = i+1, \\ 1 - a_i - b_i, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

(3) *For* $i = n$,

$$p_{ij} = \begin{cases} a_i, & j = i-1, \\ 1-a_i, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Then for this absorbing Markov chain, its mean first hitting time to the absorbing state is given by

$$\begin{cases} m_0 = 0, \\ m_i = m_{i-1} + \frac{1}{a_i} + \sum_{j=0}^{n-(i+1)} \frac{1}{a_{j+i+1}} \prod_{h=0}^{j} \frac{b_{i+h}}{a_{i+h}}, \quad i = 1, \ldots, n-1, \\ m_n = m_{n-1} + \frac{1}{a_n}. \end{cases}$$

The difference between the two corollaries lies in the fact that the Markov chain of Corollary 1 has two absorbing states while the Markov chain of Corollary 2 has only one. He et al. [14] used Corollary 2 to estimate the expected running time of evolutionary algorithms.

For our time complexity discussion and analysis, Theorem 1 and its corollaries play a key role. These methods are close to the Markov chain analysis of the stochastic local search algorithms by Schöning [29,30]. The main difference is: We use linear system (2) to estimate the absorbing time of the Markov chain while Schöning calculated the "success probability".

Now we introduce two vector norms, the average vector norm and the maximum vector norm, both of which are often used in the vector analysis. For vector $\mathbf{m} = [m_i]_{i \in S}$, let $\mu_0(i) = P(X_0 = i)$ be the initial distribution, the average vector norm $\|\mathbf{m}\|_1$ and the maximum vector norm $\|\mathbf{m}\|_\infty$ are defined as

$$\|\mathbf{m}\|_1 = \sum_{i \in S} \mu(i) m_i$$

and

$$\|\mathbf{m}\|_\infty = \max_{i \in S} \{m_i\}.$$

Specially, if the initial distribution is the uniform distribution on S, i.e. $\mu_0(i) = \frac{1}{|S|} (i \in S)$, then we have

$$\|\mathbf{m}\|_1 = \frac{1}{|S|} \sum_{i \in S} m_i.$$

Norms $\|\mathbf{m}\|_1$ and $\|\mathbf{m}\|_\infty$ present average case and worst case performance measures respectively in the time complexity analysis.

3. Bounds on RandomWalk

It is well known that the most simple algorithm, complete enumeration, needs $\Theta(2^n)$ steps to find the satisfying assignment of the SAT problem with n variables. In the following, we shall show that the general upper bound of the average iteration number of RandomWalk for k-SAT is $O((k-1)^n)$. We also construct a SAT instance for which this bound is tight, i.e. for which the expected runtime of RandomWalk is $\Theta((k-1)^n)$.

Proposition 1. *The expected runtime of RandomWalk for any k-SAT* ($k \geqslant 3$) *instance is at most* $O((k-1)^n)$.

Proof. Let $S = \{0,1\}^n$ be the search space, and S^* the satisfying assignment set for given k-CNF formula ω. Let $d(x) = \min_{y \in S^*} \{H(x,y)\}$ denote the distance between a point $x \in S$ and the set S^*.

Define D_i to be $D_i = \{x \in S \mid d(x) = i\}$, $i = 0, 1, \ldots, n$. Then the search space S is partitioned into $n+1$ subspaces: $S = \bigcup_{i=0}^n D_i$.

Suppose we are given a string $x \in D_i$ $(1 \leqslant i \leqslant n)$, for any clause σ that is not satisfied, there exists at least one of the k bits to be flipped to decrease $d(x)$ by 1. Since RandomWalk picks a variable in some unsatisfied clause and flips its truth assignment, the probability that RandomWalk transfers x to some string $y \in D_{i-1}$ is at least $\frac{1}{k}$, and the probability that it transfers x to some string $y \in D_{i+1}$ is at most $1 - \frac{1}{k}$.

Construct an auxiliary homogeneous Markov chain which is defined on the state space $\{D_0, D_1, \ldots, D_n\}$ with the transition matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \frac{1}{k} & 0 & \frac{k-1}{k} & 0 & \ldots & 0 & 0 & 0 \\ 0 & \frac{1}{k} & 0 & \frac{k-1}{k} & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \ldots & \frac{1}{k} & 0 & \frac{k-1}{k} \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix}$$

According to Corollary 2, its mean first hitting time to the absorbing state D_0 is

$$\begin{cases} m_0 = 0, \\ m_i = m_{i-1} + k + k\sum_{j=0}^{n-(i+1)}(k-1)^{j+1}, \quad i = 1, \ldots, n-1, \\ m_n = m_{n-1} + 1. \end{cases}$$

By induction, we have

$$\begin{aligned} m_n &= m_{n-1} + 1 \\ &= (n-1)k + \frac{k(k-1)}{k-2}\left(\sum_{l=1}^{n-1}(k-1)^{n-l-1} - (n-1)\right) \\ &= (n-1)k + \frac{k(k-1)}{k-2}\left(\frac{(k-1)^{n-1}-1}{k-2} - (n-1)\right). \end{aligned}$$

Hence

$$\|\mathbf{m}\|_\infty = m_n = O\big((k-1)^n\big).$$

This completes the proof. □

We have shown that the expected running time of RandomWalk for any k-SAT instance is $O((k-1)^n)$, but this does not mean that RandomWalk needs, on average, $\Theta((k-1)^n)$ steps to find a satisfying assignment in every k-SAT instance. In the following we present a k-SAT instance $\varphi^{(k)}(x)$ $(k > 2)$, where the average case expected time complexity is $\Theta((k-1)^n)$.

Definition 2. The SAT instance $\varphi^{(k)}(x)$ $(k < n)$ has the following clauses:

$$x_i, \quad \text{and} \quad x_{i_1} \vee (\bar{x}_{i_2} \vee \cdots \vee \bar{x}_{i_k})$$

where $1 \leqslant i \leqslant n$, and $(i_1, \ldots, i_k)$ ranges over all k-element subsets of $\{1, \ldots, n\}$.

It is evident that the SAT instance $\varphi^{(k)}(x)$ has the unique satisfying assignment $x^* = (1 \cdots 1)$.

Papadimitiou [26] first proposed the special cases $k = 3$ of $\varphi^{(k)}(x)$ and claimed that it was a difficult instance for RandomWalk. Here we discuss the general situation and derive its worst case and average case bounds of the expected runtime on RandomWalk.

Proposition 2. *Given an integer $k \geqslant 3$, the expected runtime of RandomWalk for SAT instance $\varphi^{(k)}(x)$ is*

(1) $\|\mathbf{m}\|_\infty = O((k-1)^n)$,
(2) $\|\mathbf{m}\|_1 = \Theta((k-1)^n)$.

Proof.

(1) It follows immediately from Proposition 1.
(2) From the consequence of (1), it is sufficient to show $\|\mathbf{m}\|_1 = \Omega((k-1)^n)$.
Let $T_i = \{x \mid x \in S = \{0,1\}^n, |x| = n - i\}$ $(i = 0, 1, \ldots, n)$ be the partition of search space S. We denote by X_t $(t = 0, 1, \ldots)$ the random string describing at which point RandomWalk is during iteration t. Then X_t is a homogeneous Markov chain with the absorbing state set T_0. The transition probabilities among the subspaces can be described as follows.
When $i = 0$,

$$P(X_{t+1} \in T_0 \mid X_t \in T_0) = 1.$$

When $1 \leqslant i \leqslant n - (k-1)$,

$$P(X_{t+1} \in T_{i+1} \mid X_t \in T_i) = \frac{k-1}{k}\frac{\binom{n-i}{k-1}}{1+\binom{n-i}{k-1}},$$

$$P(X_{t+1} \in T_{i-1} \mid X_t \in T_i) = 1 - \frac{k-1}{k}\frac{\binom{n-i}{k-1}}{1+\binom{n-i}{k-1}}.$$

When $n - (k-1) < i \leqslant n$,

$$P(X_{t+1} \in T_{i-1} \mid X_t \in T_i) = 1.$$

Denote

$$\begin{cases} a_i = 1 - \frac{k-1}{k}\frac{\binom{n-i}{k-1}}{1+\binom{n-i}{k-1}}, & 1 \leqslant i \leqslant n-(k-1), \\ b_i = \frac{k-1}{k}\frac{\binom{n-i}{k-1}}{1+\binom{n-i}{k-1}}, & 1 \leqslant i \leqslant n-(k-1), \\ a_i = 1, & n-(k-1) < i \leqslant n, \\ b_i = 0, & n-(k-1) < i \leqslant n. \end{cases}$$

Construct an auxiliary homogeneous Markov chain Z_t $(t = 0, 1, \ldots)$ which is defined on the state set $S = \{0, 1, \ldots, n\}$ with the transition matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ a_1 & 1-a_1-b_1 & b_1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & a_2 & 1-a_2-b_2 & b_2 & \ldots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \ldots & a_{n-1} & 1-a_{n-1}-b_{n-1} & b_{n-1} \\ 0 & 0 & 0 & 0 & \ldots & 0 & a_n & 1-a_n \end{pmatrix} \tag{3}$$

For this absorbing Markov chain, according to Corollary 2, its mean first hitting time to the absorbing state 0 is given by

$$\begin{cases} m_0 = 0, \\ m_i = m_{i-1} + \frac{1}{a_i} + \sum_{j=0}^{n-(i+k-1)} \frac{1}{a_{j+i+1}} \prod_{h=0}^{j} \frac{b_{i+h}}{a_{i+h}}, & 1 \leqslant i \leqslant n-(k-1), \\ m_i = m_{i-1} + 1, & n-(k-1) < i \leqslant n. \end{cases} \tag{4}$$

Note that for $1 \leqslant i \leqslant n-(k-1)$,

$$\frac{1}{a_i} \geqslant \frac{1}{k} \tag{5}$$

and

$$\frac{b_i}{a_i} = (k-1)\frac{1}{1 + k/\binom{n-i}{k-1}} \geqslant (k-1)e^{-k/\binom{n-i}{k-1}}. \tag{6}$$

Here we use the inequality $\frac{1}{1+x} \geqslant e^{-x}$ $(x \geqslant 0)$.
From Eqs. (4), (5) and (6), we get

$$\begin{aligned} m_1 &= \frac{1}{a_1} + \sum_{j=0}^{n-k} \frac{1}{a_{j+2}} \prod_{h=0}^{j} \frac{b_{h+1}}{a_{h+1}} \\ &\geqslant \frac{1}{k} + \frac{1}{k}\sum_{j=0}^{n-k}\big((k-1)^{j+1}\big)e^{-\sum_{h=1}^{j+1}\frac{k}{\binom{n-h}{k-1}}}. \end{aligned}$$

Because

$$\begin{aligned} \sum_{h=1}^{j+1} \frac{k}{\binom{n-h}{k-1}} &= k!\sum_{h=1}^{j+1} \frac{1}{\binom{n-h}{k-1}(k-1)!} \\ &\leqslant k!\sum_{h=1}^{j+1} \frac{1}{(n-h)(n-(h+1))} \quad (k > 2) \\ &= k!\sum_{h=1}^{j+1}\left(\frac{1}{(n-(h+1))} - \frac{1}{n-h}\right) \leqslant k!, \end{aligned}$$

we have

$$\begin{aligned} m_1 &\geqslant \frac{1}{k} + \frac{1}{k}e^{-k!}\sum_{j=0}^{n-k}(k-1)^{j+1} \\ &= \frac{1}{k} + \frac{1}{k}e^{-k!}\frac{k-1}{k-2}\big((k-1)^{n-k+1} - 1\big) \\ &= \Omega\big((k-1)^n\big). \end{aligned}$$

Finally, by the monotonicity of the sequence m_i we obtain

$$\|\mathbf{m}\|_1 = \frac{1}{2^n}\left(\binom{n}{0}m_0 + \binom{n}{1}m_1 + \cdots + \binom{n}{n}m_n\right)$$
$$\geqslant \frac{2^n-1}{2^n}\Omega\big((k-1)^n\big) = \Omega\big((k-1)^n\big).$$

This completes the proof. □

From Proposition 2, it should be mentioned that even the exhaustive search (with upper bound of 2^n) behaves much better than RandomWalk on SAT instance $\varphi^{(k)}(x)$ $(k > 3)$.

Droste et al. [8] studied the expected running time of $(1+1)$ EA on the special case $\varphi^{(3)}(x)$ and demonstrated that the average time complexity is a exponential time. Wei et al. [33] presented a class of formulas involving a so-called ternary chain which is similar to $\varphi^{(3)}(x)$. They also showed the expected runtime of RandomWalk on the ternary chain formula is exponential and proposed the "accelerating random walk" on this problem.

For $\varphi^{(k)}(x)$, according to Eq. (1), the fitness function used in evolutionary algorithms is

$$\mathit{fit}_{\varphi^{(k)}}(x) = \sum_{1\leqslant i\leqslant n} x_i + \sum_{1\leqslant i_1\leqslant n}\ \sum_{\substack{1\leqslant i_2\leqslant n\\ i_2\neq i_1}} \cdots \sum_{\substack{1\leqslant i_k\leqslant n\\ i_k\neq i_1,\ldots,i_k\neq i_{k-1}}} \big(1-(1-x_{i_1})x_{i_2}\cdots x_{i_k}\big)$$
$$= s + k!\binom{n}{k} - (n-s)(k-1)!\binom{s}{k-1} \quad \big(|x| = s\big).$$

The fitness function $\mathit{fit}_{\varphi^{(k)}}(x)$ induces the MaxSAT problem $\varphi^{(k)}(x)$ into a polynomial in s (the number of 1 in x) of degree k. We might expect that the heuristic algorithms using fitness function have difficulty finding the all-one string because the non-monotone polynomial fitness function might give misleading hints regarding the all-one string.

4. Behavior of three heuristic algorithms on SAT instances

In this section, in order to obtain a theoretical understanding of the behavior of different heuristic algorithms, we construct two SAT instances and analyze the average time complexity of RandomWalk, $(1+1)$ EA and hybrid algorithm on these SAT instances.

Definition 3. For $x = (x_1 \cdots x_n) \in \{0,1\}^n$, the SAT instance $\psi_1(x)$ is defined as

$$\psi_1(x) = (x_1 \vee \bar{x}_2) \wedge (x_1 \vee \bar{x}_3) \wedge \cdots (x_1 \vee \bar{x}_n) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3) \wedge \cdots (\bar{x}_1 \vee x_n).$$

The satisfying assignments of $\psi_1(x)$ are $(0\cdots 0)$ and $(1\cdots 1)$.

We start from $(1+1)$ EA for solving the MaxSAT instance $\psi_1(x)$. According to Eq. (1), for $|x| = k$, the fitness function of $\psi_1(x)$ is given as

$$\mathit{fit}_{\psi_1}(x) = \begin{cases} 2(n-1)-k, & x = (0*\cdots*),\\ (n-1)+(k-1), & x = (1*\cdots*).\end{cases}$$

When $x = (0*\cdots*)$ (or $x = (1*\cdots*)$), the fitness function $\mathit{fit}_{\psi_1}(x)$ decreases (or increases) monotonously with an increase of the number of ones. The fitness function $\mathit{fit}_{\psi_1}(x)$ with $n = 20$ is shown in Fig. 1.

In the following, we consider local $(1+1)$ EA and global $(1+1)$ EA on $\psi_1(x)$ respectively. We shall show that they both find the satisfying assignment in time $\Theta(n\ln n)$.

For simplicity, in Proposition 3 below we assume that n is even.

We further divide the search space S into $2n$ subspaces:

$$S_{0,k} = \big\{x \mid x = (0*\cdots*) \in S,\ |x| = k\big\} \quad (k = 0,\ldots,n-1),$$
$$S_{1,k} = \big\{x \mid x = (1*\cdots*) \in S,\ |x| = k\big\} \quad (k = 1,\ldots,n).$$

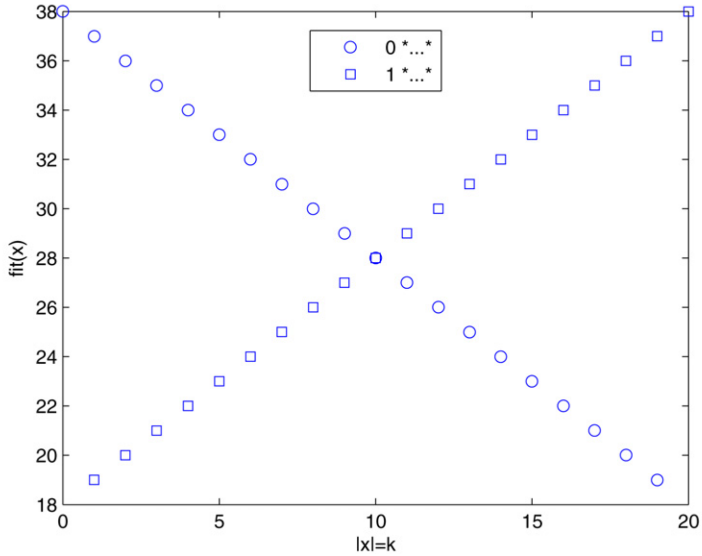

Fig. 1. The fitness function of $\psi_1(x)$ with $n = 20$.

Proposition 3. *For SAT instance $\psi_1(x)$, for any $x \in S_{u,k}$ $(u = 0, 1;\ k = 0, \dots, n)$, denote by $m_{u,k}$ the mean first hitting time of local $(1+1)$ EA starting from state x, then*

$$\begin{cases} m_{0,0} = 0, & \\ m_{0,k} = n(1 + \cdots + \frac{1}{k}), & 1 \leqslant k \leqslant \frac{n}{2} - 1, \\ m_{1,n} = 0, & \\ m_{1,k} = n(1 + \cdots + \frac{1}{n-k}), & \frac{n}{2} + 1 \leqslant k \leqslant n - 1, \\ m_{0,k} = \frac{1}{k+1}(n + k m_{0,k-1} + m_{1,k+1}), & \frac{n}{2} \leqslant k \leqslant n - 1, \\ m_{1,k} = \frac{1}{n-k+1}(n + (n-k) m_{1,k+1} + m_{0,k-1}), & 1 \leqslant k \leqslant \frac{n}{2}, \end{cases}$$

i.e. the expected runtime of the local $(1+1)$ EA is $\|\mathbf{m}\|_\infty = \Theta(n \ln n)$.

Proof. Let $X_t \in \{0,1\}^n$ $(t = 0, 1, \dots)$ be the random variable describing the state of local $(1+1)$ EA solving SAT instance $\psi_1(x)$ at time t, then the transition probabilities can be described as follows.

When $k = 0$,

$$P(X_{t+1} \in S_{0,k} \mid X_t \in S_{0,k}) = 1.$$

When $1 \leqslant k \leqslant \frac{n}{2} - 1$,

$$P(X_{t+1} \in S_{0,k-1} \mid X_t \in S_{0,k}) = \frac{k}{n},$$

$$P(X_{t+1} \in S_{0,k} \mid X_t \in S_{0,k}) = 1 - \frac{k}{n}.$$

When $\frac{n}{2} \leqslant k \leqslant n - 1$,

$$P(X_{t+1} \in S_{0,k-1} \mid X_t \in S_{0,k}) = \frac{k}{n},$$

$$P(X_{t+1} \in S_{1,k+1} \mid X_t \in S_{0,k}) = \frac{1}{n},$$

$$P(X_{t+1} \in S_{0,k} \mid X_t \in S_{0,k}) = 1 - \frac{k+1}{n}.$$

Similarly, when $k = n$,

$$P(X_{t+1} \in S_{1,k} \mid X_t \in S_{1,k}) = 1.$$

When $\frac{n}{2} + 1 \leqslant k \leqslant n - 1$,

$$P(X_{t+1} \in S_{1,k+1} \mid X_t \in S_{1,k}) = 1 - \frac{k}{n},$$
$$P(X_{t+1} \in S_{1,k} \mid X_t \in S_{1,k}) = \frac{k}{n}.$$

When $1 \leqslant k \leqslant \frac{n}{2}$,

$$P(X_{t+1} \in S_{1,k+1} \mid X_t \in S_{1,k}) = 1 - \frac{k}{n},$$
$$P(X_{t+1} \in S_{0,k-1} \mid X_t \in S_{1,k}) = \frac{1}{n},$$
$$P(X_{t+1} \in S_{1,k} \mid X_t \in S_{1,k}) = \frac{k-1}{n}.$$

Introduce an auxiliary homogeneous Markov chain Z_t $(t = 0, 1, \ldots)$ with the state space $\{z_{0,0}, z_{0,1}, \ldots, z_{0,n-1}; z_{1,1}, z_{1,2}, \ldots, z_{1,n}\}$, the transition probabilities are defined by

$$P(Z_{t+1} = z_{v,h} \mid Z_t = z_{u,k}) = P(X_{t+1} \in S_{v,h} \mid X_t \in S_{u,k})$$

where $u, v \in \{0, 1\}$, and $h, k \in \{0, \ldots, n\}$.

Then Z_t is an absorbing Markov chain with the absorbing state $z_{0,0}$ and $z_{1,n}$, and for any $x \in S_{u,k}$ $(u \in \{0, 1\}, k \in \{0, \ldots, n\})$, the mean first hitting time m_x equals $m_{z_{u,k}}$.

According to Theorem 1, the mean first hitting time of stochastic process Z_t is given by

$$\begin{cases} m_{0,0} = 0, & \\ -\frac{k}{n} m_{0,k-1} + \frac{k}{n} m_{0,k} = 1, & 1 \leqslant k \leqslant \frac{n}{2} - 1, \\ -\frac{k}{n} m_{0,k+1} + \frac{k+1}{n} m_{0,k} - \frac{1}{n} m_{1,k+1} = 1, & \frac{n}{2} \leqslant k \leqslant n - 1, \\ m_{1,n} = 0, & \\ -\frac{n-k}{n} m_{1,k+1} + \frac{n-k}{n} m_{1,k} = 1, & \frac{n}{2} + 1 \leqslant k \leqslant n - 1, \\ -\frac{n-k}{n} m_{1,k+1} + \frac{n-k+1}{n} m_{1,k} - \frac{1}{n} m_{0,k-1} = 1, & 1 \leqslant k \leqslant \frac{n}{2}. \end{cases}$$

The above linear equations can be solved as

$$\begin{cases} m_{0,0} = 0, & \\ m_{0,k} = n(1 + \cdots + \frac{1}{k}), & 1 \leqslant k \leqslant \frac{n}{2} - 1, \\ m_{1,n} = 0, & \\ m_{1,k} = n(1 + \cdots + \frac{1}{n-k}), & \frac{n}{2} + 1 \leqslant k \leqslant n - 1, \\ m_{0,k} = \frac{1}{k+1}(n + k m_{0,k-1} + m_{1,k+1}), & \frac{n}{2} \leqslant k \leqslant n - 1, \\ m_{1,k} = \frac{1}{n-k+1}(n + (n-k) m_{1,k+1} + m_{0,k-1}), & 1 \leqslant k \leqslant \frac{n}{2}. \end{cases} \tag{7}$$

In the following, we prove

$$m_{0,k} \leqslant n\left(1 + \cdots + \frac{1}{k}\right) \quad \left(\frac{n}{2} \leqslant k \leqslant n - 1\right) \tag{8}$$

by induction.

When $k = \frac{n}{2}$, from (7), we obtain

$$m_{0,n/2} = \frac{1}{n/2 + 1}\left(n + \frac{n}{2}\left(1 + \cdots + \frac{1}{n/2 - 1}\right) + n\left(1 + \cdots + \frac{1}{n - (n/2 + 1)}\right)\right)$$
$$\leqslant n\left(1 + \cdots + \frac{1}{n/2}\right).$$

Thus we have that (8) holds for $k = \frac{n}{2}$.

Assume it is true for some $k \geqslant \frac{n}{2}$, i.e. $m_{0,k} \leqslant n(1 + \cdots + \frac{1}{k})$, from (7), we have

$$m_{0,k+1} = \frac{1}{k+2}\big(n + (k+1) m_{0,k} + m_{1,k+2}\big)$$
$$\leqslant \frac{n}{k+2} + n\left(1 + \cdots + \frac{1}{k}\right)$$
$$\leqslant n\left(1 + \cdots + \frac{1}{k+1}\right).$$

Therefore (8) holds for all k.

Similarly, we have $m_{1,k} \leqslant n(1+\cdots+\frac{1}{n-k})$ $(1 \leqslant k \leqslant \frac{n}{2})$.
This proves the claim. $\square$

Proposition 4. *For SAT instance $\psi_1(x)$, the expected runtime of the global $(1+1)$ EA is*

(1) $\|\mathbf{m}\|_\infty = O(n\ln n)$,
(2) $\|\mathbf{m}\|_1 = \Theta((n\ln n)$.

Proof.

(1) We decompose the state space $S=\{0,1\}^n$ into n subspaces by the Hamming distance between the string in S and the satisfying assignment: $S=\bigcup_{k=0}^{n-1} U_k$ where

$$U_k = T_{0,k} \cup T_{1,k} \quad (k=0,1,\ldots,n-1),$$

and

$$T_{0,k} = \{x \mid x=(0*\cdots*)\in S,\ H\big(x,(0\cdots0)\big)=k\},$$
$$T_{1,k} = \{x \mid x=(1*\cdots*)\in S,\ H\big(x,(1\cdots1)\big)=k\}.$$

In contrast to the notations $S_{0,k}$ and $S_{1,k}$, both of which are used in the proof of Proposition 3, we see that $T_{0,k}=S_{0,k}$ and $T_{1,k}\neq S_{1,k}$. For $x=(1*\cdots*)\in S$, $x\in S_{1,k}$ means that the Hamming distance between x and $(0\cdots0)$ is k while $x\in T_{1,k}$ means that the Hamming distance between x and $(1\cdots1)$ is k.
For $x\in U_k(k=1\cdots n-1)$, note that $fit_{\psi_1}(x)=2n-k$. Thus the probability that the global $(1+1)$ EA leads x to some $y\in U_{k-1}\cup\cdots\cup U_0$ is greater than $\frac{k}{n}$.
Therefore we get $\|\mathbf{m}\|_\infty \leqslant (1+\cdots+\frac{1}{n-1}) = O(n\ln n)$.

(2) According to the result of (1), it is sufficient to prove that $\|\mathbf{m}\|_1=\Omega((n\ln n)$. The proof below is similar to that of the linear functions with nonzero weights by Droste et al. [9]. The main difference is that the linear function has only one optimum $(0\cdots0)$ while SAT instance $\psi_1(x)$ has two satisfying assignments $(0\cdots0)$ and $(1\cdots1)$.
By Chernoff bounds [23], for any $0<\epsilon<\frac{1}{2}$, the probability that the initial string x satisfies $(\frac{1}{2}-\epsilon)n \leqslant |x| \leqslant (\frac{1}{2}+\epsilon)n$ (for the simplicity of analysis, we assume that $(\frac{1}{2}-\epsilon)n$ is an integer) is exponentially close to one, i.e. $1-e^{-\Omega(n)}$. It is equivalent that with probability $1-e^{-\Omega(n)}$ the randomly initialized string has at least $(\frac{1}{2}-\epsilon)n$ zeros and $(\frac{1}{2}-\epsilon)n$ ones. In order to reach the satisfying assignment, each of these strings needs to flip its all zeros (or all ones) at least once.
Let X be a random variable defined to be the number of generations required to flip the all zeros (or ones) of the above initialized string at least once, then its expectation is

$$E(X)=\sum_{t=1}^{\infty} tP(X=t)=\sum_{t=1}^{\infty} P(X\geqslant t).$$

Since the probability that one bit does not flip at all in $t-1$ steps is $(1-\frac{1}{n})^{t-1}$, the probability for the event that at least one of the $(\frac{1}{2}-\epsilon)n$ bits never flips in $t-1$ steps is $1-(1-(1-\frac{1}{n})^{t-1})^{(\frac{1}{2}-\epsilon)n}$.
Hence we have

$$\begin{aligned}E(X) &\geqslant \sum_{t=1}^{\infty}\left(1-\left(1-\left(1-\frac{1}{n}\right)^{t-1}\right)^{(\frac{1}{2}-\epsilon)n}\right)\\ &\geqslant (n-1)\ln n\left(1-\left(1-\left(1-\frac{1}{n}\right)^{(n-1)\ln n}\right)^{(\frac{1}{2}-\epsilon)n}\right)\\ &\geqslant (n-1)\ln n\big(1-e^{-(\frac{1}{2}-\epsilon)}\big)=\Omega(n\ln n).\end{aligned}$$

In the above, we use $(1-\frac{1}{n})^n \leqslant e^{-1}(n>1)$.
Therefore

$$\|\mathbf{m}\|_1 \geqslant \big(1-e^{-\Omega(n)}\big)\Omega(n\ln n)=\Omega(n\ln n).$$

This completes the proof. $\square$

Papadimitiou [25] proved that RandomWalk on any satisfiable 2-SAT will reach a satisfying assignment in time $O(n^2)$ by the theory of random walks. In the following Proposition 5, we shall demonstrate that RandomWalk has the $\Theta(n^2)$ worst case and average case expected runtime bound on 2-SAT instance $\psi_1(x)$.

Now we introduce a lemma which provides the upper bound for the tail of the binomial distribution function. It will be used in Proposition 5 for estimating the average case expected runtime of RandomWalk.

Lemma 1. *Let X be a random variable following the binomial distribution with parameters n and p. Given a integer $0 \leqslant k \leqslant n$, the cumulative distribution is expressed as*

$$F(k) = P(X \leqslant k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}.$$

Then $F(k) \leqslant e^{-\frac{2(np-k)^2}{n}}$.

Proof. It follows immediately from Hoeffding's inequality [16]. $\square$

Proposition 5. *For SAT instance $\psi_1(x)$, the average case expected runtime of RandomWalk is $\|\mathbf{m}\|_1 = \Theta(n^2)$.*

Proof. According to the above discussion, it is sufficient to show that $\|\mathbf{m}\|_1 = \Omega(n^2)$.

We denote by X_t $(t = 0, 1, \ldots)$ the stochastic process of RandomWalk for SAT instance $\psi_1(x)$. Then X_t is a homogeneous Markov chain with two absorbing states $(0 \cdots 0)$ and $(1 \cdots 1)$. The transition probabilities among the subspaces can be described as follows.

When $i = 0$,

$$P(X_{t+1} \in S_i \mid X_t \in S_i) = 1.$$

When $1 \leqslant i \leqslant n-1$,

$$P(X_{t+1} \in S_{i+1} \mid X_t \in S_i) = 1/2,$$
$$P(X_{t+1} \in S_{i-1} \mid X_t \in S_i) = 1/2.$$

When $i = n$,

$$P(X_{t+1} \in S_i \mid X_t \in S_i) = 1.$$

Construct an auxiliary homogeneous Markov chain Z_t $(t = 0, 1, \ldots)$ which is defined on the state space $\{0, 1, \ldots, n\}$ with the transition probabilities

$$P(Z_{t+1} = j \mid Z_t = i) = P(X_{t+1} \in S_j \mid X_t \in S_i)$$

where $i, j = 0, \ldots, n$.

According to Corollary 1, the mean first hitting time of absorbing chain Z_t is given by

$$m_i = i(n-i) \quad (i = 0, 1, \ldots, n).$$

Hence

$$\begin{aligned}
\|\mathbf{m}\|_1 &= \frac{1}{2^n}\left(\binom{n}{0} m_0 + \binom{n}{1} m_1 + \cdots + \binom{n}{n} m_n\right) \\
&= \frac{1}{2^n} \sum_{i=1}^{n} i(n-i) \binom{n}{i} \\
&\geqslant \frac{3n^2}{16} \frac{1}{2^n} \sum_{i=n/4}^{3n/4} \binom{n}{i} \\
&\geqslant \frac{3n^2}{16}\left(1 - 2e^{-\frac{n^3}{8}}\right) \quad \text{(by Lemma 1)} \\
&= \Omega\left(n^2\right).
\end{aligned}$$

Finally, we reach the conclusion. $\square$

For SAT instance $\psi_1(x)$, from Propositions 3–5, we see that the $\Theta(n \ln n)$ expected runtime bound of $(1+1)$ EA is better than the $\Theta(n^2)$ expected runtime bound of RandomWalk. In the following, we construct another SAT instance $\psi_2(x)$, for which we shall have an opposite situation under some conditions.

Definition 4. The SAT instance $\psi_2(x)$ has the following clauses:

$$x_i, \quad 1 \leqslant i \leqslant n, \quad \text{and} \quad x_i \vee \bar{x}_j, \quad (i, j) \in \{1, \ldots, n\}^2, \quad i \neq j.$$

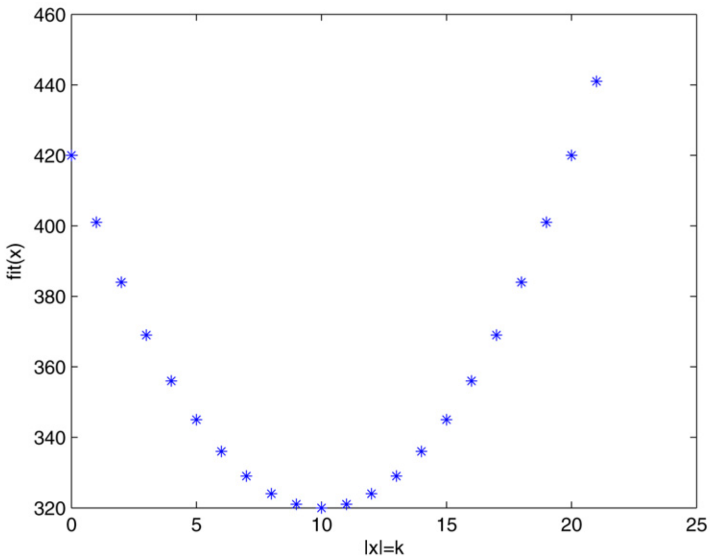

Fig. 2. The fitness function of $\psi_2(x)$ with $n = 21$.

The only satisfying assignment of SAT instance $\psi_2(x)$ is the all True assignment $(1 \cdots 1)$.

According to Eq. (1), for $|x| = k$, the fitness function of $\psi_2(x)$ is given as

$$\mathit{fit}_{\psi_2}(x) = n(n-1) + k(k-n+1),$$

which is a polynomial in k of degree 2. The fitness function $\mathit{fit}_{\psi_2}(x)$ with $n = 21$ is shown in Fig. 2. We note that this fitness function has its local minimum at $|x| = 10$ while all one string is the only global optimum. If the local hill-climbing $(1+1)$ EA starts from x $(|x| < 10)$, it will never reach the global optimum.

Proposition 6. *Let $T_k = \{x \mid x \in S = \{0,1\}^n,\ |x| = n-k\}$ $(k = 0, 1, \ldots, n)$ be the partition of search space $\{0,1\}^n$, and n an odd number. For SAT instance $\psi_2(x)$, for any $i \in T_k$ $(k = 1, \ldots, n)$ the expected runtime of the local $(1+1)$ EA is*

$$m_i = \begin{cases} +\infty & k < \frac{n-1}{2}, \\ O(n \ln n) & k \geqslant \frac{n-1}{2}. \end{cases}$$

Proof. When $k < \frac{n-1}{2}$, since the fitness function $\mathit{fit}_{\psi_2}(x)$ decrease monotonously as $|x|$ increases, the local $(1+1)$ EA starting from any initial string $i \in T_k$ will never reach the satisfying assignment $(1 \cdots 1)$.

When $k \geqslant \frac{n-1}{2}$, similar to that of OneMAX[11], the local $(1+1)$ EA starting from any initial string $i \in T_k$ will find the satisfying assignment in $O(n \ln n)$ on average. $\square$

RandomWalk on $\psi_2(x)$ has the same worst case expected runtime bound as that of $\psi_1(x)$:

Proposition 7. *For SAT instance $\psi_2(x)$, the expected runtime of RandomWalk is $\|\mathbf{m}\|_\infty = \Theta(n^2)$.*

Proof. It is sufficient to show that $\|\mathbf{m}\|_\infty = \Omega(n^2)$.

The following proof is similar to that of Proposition 2.

Let X_t $(t = 0, 1, \ldots)$ be the random variable describing at which point RandomWalk is during iteration t. Then X_t is an homogeneous Markov chain with the absorbing state set T_0. The transition probabilities among the subspace can be described as follows.

When $k = 0$,

$$P(X_{t+1} \in T_0 \mid X_t \in T_0) = 1.$$

When $1 \leqslant k \leqslant n-1$,

$$P(X_{t+1} \in T_{k-1} \mid X_t \in T_k) = 1 - \frac{1}{2} \times \frac{n-k}{1+(n-k)},$$

$$P(X_{t+1} \in T_{k+1} \mid X_t \in T_k) = \frac{1}{2} \times \frac{n-k}{1+(n-k)}.$$

When $k = n$,

$$P(X_{t+1} \in T_{k-1} \mid X_t \in T_k) = 1.$$

Denote

$$\begin{cases} a_k = 1 - \frac{1}{2} \times \frac{n-k}{1+(n-k)}, & 1 \leqslant k \leqslant n-1, \\ b_k = \frac{1}{2} \times \frac{n-k}{1+(n-k)}, & 1 \leqslant k \leqslant n-1, \\ a_k = 1, & k = n, \\ b_k = 0, & k = n. \end{cases}$$

Construct an auxiliary homogeneous Markov chain Z_t $(t = 0, 1, \ldots)$ which is defined on the state set $\{0, 1, \ldots, n\}$ with the transition matrix (3) (see the proof of Proposition 2).

For this absorbing Markov chain, according to Corollary 2, its mean first hitting time to the absorbing state 0 is given by

$$\begin{cases} m_0 = 0, \\ m_k = m_{k-1} + \frac{1}{a_k} + \sum_{j=0}^{n-(k+1)} \frac{1}{a_{j+k+1}} \prod_{i=0}^{j} \frac{b_{i+k}}{a_{i+k}}, & 1 \leqslant k \leqslant n-1, \\ m_n = m_{n-1} + \frac{1}{a_n} = \frac{1}{a_1} + \sum_{k=1}^{n-1} \frac{1}{a_{k+1}} (1 + \sum_{j=1}^{k} \prod_{i=j}^{k} \frac{b_i}{a_i}). \end{cases} \tag{9}$$

When $n \geqslant 2$ and $k \leqslant \frac{n}{2}$, we have

$$1 + \sum_{j=1}^{k} \prod_{i=j}^{k} \frac{b_i}{a_i} = 1 + \frac{n-k}{n-k+2} + \frac{(n-k)(n-k+1)}{(n-k+2)(n+1)}(k-1) \geqslant 1 + \frac{1}{4}(k-1).$$

It follows that

$$\begin{aligned} m_n &\geqslant \frac{1}{a_1} + \sum_{k=1}^{n/2-1} \frac{1}{a_{k+1}} \left(1 + \sum_{j=1}^{k} \prod_{i=j}^{k} \frac{b_i}{a_i}\right) \\ &\geqslant \frac{1}{2} \sum_{k=1}^{n/2-1} \left(1 + \frac{1}{4}(k-1)\right) = \Omega\big(n^2\big). \end{aligned}$$

This completes the proof. □

Contrary to the prior result that the expected runtime of (1 + 1) EA for SAT instance $\psi_1(x)$ is better than that of RandomWalk, here we demonstrate that for SAT instance $\psi_2(x)$, the worst case expected runtime of RandomWalk is better than that of local (1 + 1) EA.

In the following, we analyze the behavior of the hybrid algorithm of (1 + 1) EA and RandomWalk on SAT instance $\psi_2(x)$. We first set the selection probability $p = 0.5$.

Proposition 8. *If the selection probability $p = 0.5$, then for SAT instance $\psi_2(x)$, the expected runtime of the hybrid algorithm is*

$$\Omega\left(\frac{1}{n^2}\left(\frac{4}{e}\right)^{n/2}\right) \leqslant \|\mathbf{m}\|_\infty \leqslant O\left(\left(\frac{4}{e}\right)^{n/2}\right).$$

Proof. For simplicity, we assume that n is odd.

The proof is similar to that of Proposition 7 and the notations are the same as specified in the proof of Proposition 7.

The transition probability of the stochastic process X_t $(t = 0, 1, \ldots)$ introduced by the hybrid algorithm of local (1 + 1) EA and RandomWalk with a probability $p = 0.5$ can be described as follows.

When $k = 0$,

$$P(X_{t+1} \in T_0 \mid X_t \in T_0) = 1.$$

When $1 \leqslant k \leqslant (n+1)/2$,

$$\begin{aligned} &P(X_{t+1} \in T_{k-1} \mid X_t \in T_k) = p\left(1 - \frac{n-k}{2(n-k+1)}\right) + (1-p)\frac{k}{n}, \\ &P(X_{t+1} \in T_k \mid X_t \in T_k) = (1-p)\left(1 - \frac{k}{n}\right), \\ &P(X_{t+1} \in T_{k+1} \mid X_t \in T_k) = p\frac{n-k}{2(n-k+1)}. \end{aligned}$$

When $(n+1)/2 < k \leqslant n$,

$$P(X_{t+1} \in T_{k-1} \mid X_t \in T_k) = p\left(1 - \frac{n-k}{2(n-k+1)}\right),$$
$$P(X_{t+1} \in T_k \mid X_t \in T_k) = (1-p)\frac{k}{n},$$
$$P(X_{t+1} \in T_{k+1} \mid X_t \in T_k) = p\frac{n-k}{2(n-k+1)} + (1-p)\frac{n-k}{n}.$$

Denote

$$a_k = \begin{cases} p(1 - \frac{n-k}{2(n-k+1)}) + (1-p)\frac{k}{n}, & 1 \leqslant k \leqslant (n+1)/2, \\ p(1 - \frac{n-k}{2(n-k+1)}), & (n+1)/2 < k \leqslant n. \end{cases} \tag{10}$$

$$b_k = \begin{cases} p\frac{n-k}{2(n-k+1)}, & 1 \leqslant k \leqslant (n+1)/2, \\ p\frac{n-k}{2(n-k+1)} + (1-p)\frac{n-k}{n}, & (n+1)/2 < k \leqslant n. \end{cases} \tag{11}$$

Construct an auxiliary homogeneous Markov chain Z_t $(t = 0, 1, \ldots)$ which is defined on the state $\{0, 1, \ldots, n\}$ space with the transition matrix (3). Its mean first hitting time to the absorbing state 0 is given by Eqs. (9).

Now we show that the lower bound is $\Omega(\frac{1}{n^2}(\frac{4}{e})^{n/2})$.

By assumption $p = 0.5$, according to Eqs. (9), we have

$$\begin{aligned} m_n &= \frac{1}{a_1} + \sum_{i=1}^{n-1} \frac{1}{a_{i+1}}\left(1 + \sum_{j=1}^{i}\prod_{k=j}^{i}\frac{b_k}{a_k}\right) \\ &\geqslant \frac{1}{a_n}\prod_{k=(n+1)/2+1}^{n-1}\frac{b_k}{a_k} \\ &= 2\prod_{k=(n+1)/2+1}^{n-1}\frac{n-k}{n-k+2}\left(1 + \frac{2(n-k+1)}{n}\right) \quad \bigl(\text{from (10) and (11)}\bigr) \\ &= \frac{16}{(n-1)(n+1)}\prod_{k=1}^{(n-3)/2}\left(1 + \frac{2(k+1)}{n}\right). \end{aligned}$$

Note that

$$\begin{aligned} I &:= \left(\sum_{k=1}^{(n-3)/2}\left(1 + \frac{2(k+1)}{n}\right)\right)^2 \\ &= \left(\frac{n+4}{n}\frac{n+6}{n}\cdots\frac{2n-1}{n}\right)^2 \\ &\geqslant \frac{(n+3)(n+4)\cdots(2n-1)}{n^{n-3}} \\ &= \frac{1}{2}\frac{n^2}{(n+1)(n+2)}\frac{(2n)!}{n^n n!}. \end{aligned}$$

Using Stirling formula

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n < n! \tag{12}$$

and

$$n! < \left(1 + \frac{1}{12n-1}\right)\sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \tag{13}$$

we have

$$\begin{aligned} I &\geqslant \frac{1}{2}\frac{n^2}{(n+1)(n+2)}\frac{1}{n^n n!}\sqrt{2\pi 2n}\left(\frac{2n}{e}\right)^{2n} \quad \bigl(\text{by Stirling formula (12)}\bigr) \\ &= \frac{1}{2}\frac{n^2}{(n+1)(n+2)}\sqrt{2\pi 2n}\left(\frac{2}{e}\right)^{2n}\frac{n^n}{n!} \end{aligned}$$

$$\geqslant \frac{1}{2}\frac{n^2}{(n+1)(n+2)}\sqrt{2\pi 2n}\left(\frac{2}{e}\right)^{2n}\frac{1}{\sqrt{2\pi n}}\frac{12n-1}{12n}e^n \quad \big(\text{by Stirling formula (13)}\big)$$
$$= \Omega\big((4/e)^n\big).$$

Therefore $m_n \geqslant \Omega(\frac{1}{n^2}(\frac{4}{e})^{n/2})$. That proves the lower bound.

The upper bound can be proved as follows.

Similar to the above equations, we have

$$\begin{aligned}
I &= \left(\frac{n+4}{n}\frac{n+6}{n}\cdots\frac{2n-1}{n}\right)^2 \\
&\leqslant \frac{(n+4)(n+5)\cdots(2n)}{n^{n-3}} \\
&= \frac{n^3}{(n+1)(n+2)(n+3)}\frac{(2n)!}{n!n^n} \\
&\leqslant \frac{1}{n!n^n}\sqrt{2\pi 2n}\left(1+\frac{1}{24n-1}\right)\left(\frac{2n}{e}\right)^{2n} \quad \big(\text{by Stirling formula (13)}\big) \\
&\leqslant \sqrt{2\pi 2n}\left(1+\frac{1}{24n-1}\right)\left(\frac{2}{e}\right)^{2n}\frac{1}{\sqrt{2\pi n}}e^n \quad \big(\text{by Stirling formula (12)}\big) \\
&= O\big((4/e)^n\big).
\end{aligned} \tag{14}$$

Note that $\frac{1}{a_k} \leqslant 4$ $(1 \leqslant k \leqslant n)$ and $\frac{b_k}{a_k} \leqslant 1$ $(1 \leqslant k \leqslant (n+1)/2)$, we obtain

$$\begin{aligned}
m_n &= \frac{1}{a_1} + \sum_{i=1}^{n-1}\frac{1}{a_{i+1}}\left(1+\sum_{j=1}^{i}\prod_{k=j}^{i}\frac{b_k}{a_k}\right) \\
&\leqslant 4n + 4(n-1)^2\prod_{k=(n+1)/2+1}^{n-1}\frac{b_k}{a_k} \\
&= 4n + 32\frac{n-1}{n+1}\prod_{k=1}^{(n-3)/2}\left(1+\frac{2(k+1)}{n}\right) \\
&= O\big((4/e)^n\big) \quad \big(\text{by (14)}\big).
\end{aligned}$$

This completes the proof. □

In the above Proposition 8, we fixed the selection probability $p = 0.5$. In fact, when $p \leqslant 0.5$, the result of the upper bound still holds.

Proposition 9. *If the selection probability $p \leqslant 0.5$, then for SAT instance $\psi_2(x)$, the expected runtime of the hybrid algorithm is $\Omega(\frac{1}{n^2}(\frac{4}{e})^{n/2})$.*

Proof. The proof is essentially the same that of Proposition 8. □

Proposition 10. *If the selection probability $p \geqslant \frac{n-1}{n}$, then for SAT instance $\psi_2(x)$, the expected runtime of the hybrid algorithm is $O(n^2)$.*

Proof. The proof is similar to that of Proposition 8.

For $1 \leqslant k \leqslant \frac{n+1}{2}$, according to Eqs. (10) and (11), notice that $p \geqslant \frac{n-1}{n}$, we have $\frac{b_k}{a_k} \leqslant 1$.

For $\frac{n+1}{2} < k < n$, we have

$$\begin{aligned}
\frac{b_k}{a_k} &= \frac{n-k}{n-k+2}\left(1+\frac{2(1+(n-k))}{n}\frac{1-p}{p}\right) \\
&\leqslant \frac{n-k}{n-k+2}\left(1+\frac{2(1+(n-k))}{n}\frac{1}{n-1}\right) \\
&\leqslant 1+\frac{1}{n}.
\end{aligned}$$

Table 1
The expected runtime bounds of RandomWalk and $(1+1)$ EA on SAT instance $\psi_1(x)$.

Local $(1+1)$ EA (Prop. 3)	Global $(1+1)$ EA (Prop. 4)	RandomWalk (Prop. 5)
$\Theta(n \ln n)$	$\Theta(n \ln n)$	$\Theta(n^2)$

Table 2
The expected runtime bounds of three heuristic algorithms on SAT instance $\psi_2(x)$.

Local $(1+1)$ EA (Prop. 6)	RandomWalk (Prop. 7)	Hybrid algorithm (Prop. 9, 10)
$+\infty$ ($\lvert x\rvert < \frac{n-1}{2}$)	$\Theta(n^2)$	$\Omega(\frac{1}{n^2}(\frac{4}{e})^{n/2})$ ($p \leqslant 0.5$)
$O(n \ln n)$ ($\frac{n-1}{2} \leqslant \lvert x\rvert \leqslant n$)		$O(n^2)$ ($\frac{n-1}{n} \leqslant p \leqslant 1$)

Then, for $1 \leqslant j \leqslant k < n$, we get

$$\prod_{i=j}^{k} \frac{b_i}{a_i} \leqslant \left(1+\frac{1}{n}\right)^{k-j+1} \leqslant \left(1+\frac{1}{n}\right)^{n} \leqslant e.$$

Note that $\frac{1}{a_k} \leqslant \frac{2}{p}$ $(1 \leqslant k \leqslant n)$, we have

$$\begin{aligned} m_n &= \frac{1}{a_1} + \sum_{i=1}^{n-1} \frac{1}{a_{i+1}} \left(1 + \sum_{j=1}^{i} \prod_{k=j}^{i} \frac{b_k}{a_k}\right) \\ &\leqslant \frac{2}{p}\left(1 + \sum_{i=1}^{n-1} (e \times i + 1)\right) = O\left(n^2\right). \end{aligned}$$

This completes the proof. □

Remark 1. For the selecting probability $p \leqslant 0.5$ or $p \geqslant \frac{n-1}{n}$, we obtain lower and upper bounds of the expected runtime for the hybrid algorithm on $\psi_2(x)$. Can we do better for it, i.e. can we get the tight bounds?

Remark 2. For $0.5 < p < \frac{n-1}{n}$, the time complexity analysis of the hybrid algorithm on $\psi_2(x)$ is much more complicated. However, we conjecture that the expected runtime changes from an exponential time to a polynomial time.

Tables 1 and 2 summarize the expected runtime bounds of three heuristic algorithms solving two SAT instances $\psi_1(x)$ and $\psi_2(x)$. From Table 1, we see that for instance $\psi_1(x)$, $(1+1)$ EA is faster than RandomWalk. From Table 2, for instance $\psi_2(x)$, starting from the initial string x satisfying $|x| \leqslant (n-1)/2$, $(1+1)$ EA can never reach the satisfying assignment and RandomWalk finds the satisfying assignment in $\Theta(n^2)$ on average. But $(1+1)$ EA is still faster than RandomWalk when the initial string x satisfies $|x| \leqslant (n-1)/2$.

For instance $\psi_2(x)$, our analysis demonstrates the hybrid algorithm may help local $(1+1)$ EA to escape from the local optimum. It also shows that the expected runtime of the hybrid algorithm changes from a exponential time bound to a polynomial time bound as the selection probability varies. However, it remains unclear how this phase transition gradually happens, and it is worth further investigating.

5. Conclusion

Incomplete heuristic algorithms are now among the most prominent and frequently applied techniques for SAT problems. Many experimental comparisons with different heuristic algorithms have been reported, although theoretic comparisons are rare. This paper contributes to the theory of heuristic algorithms for SAT problems. We derive the expected runtime bounds of RandomWalk on k-SAT problem. We construct two 2-SAT instances and provide analytic comparisons among RandomWalk, $(1+1)$ EA, and hybrid algorithm on these instances. It is shown that these heuristic algorithms have their own advantages and disadvantages in solving two SAT instances and their expected runtime ranges from a polynomial time to an exponential time. Our analysis provides an insight into the runtime behavior among these heuristic algorithms. Admittedly, two SAT instances $\psi_1(x)$ and $\psi_2(x)$ considered in this paper are relatively simple. Future investigation should be extended to a broader class of SAT problems and more heuristic algorithms, such as UnitWalk [15] and PPSZ [27].

Theoretic runtime analysis and comparison for heuristic algorithms on the SAT problem lag far behind experimental comparisons. Effort should be made to fill in the gap between theoretical studies on SAT and the design and application of practical algorithms, even though such an investigation will be challenging.

Acknowledgements

The authors are very grateful to the anonymous reviewers for their valuable comments and helpful suggestions, which improved the paper's quality greatly. This work is partially supported by National Natural Science Foundation of China (Nos. 60673062, 60873078) and Natural Science Foundation of Guangdong Province of China (No. 06025686) to YZ; and NIH grants P50GM76516 and R01GM75309, and NSF grants DMS0511169 to QN.

References

[1] E.H.L. Aarts, J.H.M. Korst, Simulated Annealing and Boltzmann Machines, Wiley, 1989.
[2] M. Alekhnovich, E. Ben-Sasson, Linear upper bounds for random walk on small density random 3-CNF, SIAM Journal on Computing 36 (5) (2006) 1248–1263.
[3] P. Beame, H. Kautz, A. Sabharwal, Towards understanding and harnessing the potential of clause learning, Journal of Artificial Intelligence Research 22 (2004) 319–335.
[4] A. Braunstein, M. Mezard, R. Zecchina, Survey propagation: An algorithm for satisfiability, Random Structures & Algorithms 27 (2) (2005) 201–226.
[5] S. Cook, The complexity of theorem-proving procedures, in: Proc. 3rd Ann. ACM Symp. on Theory of Computing, Assoc. Comput. Mach., New York, 1971, p. 151.
[6] M. Davis, H. Putnam, A computer procedure for quantification theory, Journal of the ACM 7 (3) (1960) 202–215.
[7] K.A. DeJong, W.M. Spears, Using genetic algorithm to solve NP-complete problems, in: Proceedings of the 3rd International Conference on Genetic Algorithms, Virginia, USA, 1989, pp. 124–132.
[8] S. Droste, T. Jansen, I. Wegener, A natural and simple function which is hard for all evolutionary algorithms, in: Proceedings of Third Asia–Pacific Conference on Simulated Evolution and Learning, Nagaya, Japan, 2000, pp. 2704–2709.
[9] S. Droste, T. Jansen, I. Wegener, On the analysis of the $(1+1)$-evolutionary algorithm, Theoretical Computer Science 276 (1–2) (2002) 51–81.
[10] M.R. Garey, D.S. Johnson, Computers and Intractability—A Guide to the Theory of NP-completeness, Freeman, New York, 1979.
[11] J. Garnier, L. Kallel, M. Schoenauer, Rigorous hitting times for binary mutations, Evolutionary Computation 7 (2) (1999) 167–203.
[12] J. Gottlieb, E. Marchiori, C. Rossi, Evolutionary algorithms for the satisfiability problem, Evolutionary Computation 10 (1) (2002) 35–50.
[13] J. Gu, Efficient local search for large-scale satisfiability problems, ACM SIGART Bulletin 3 (1) (1992) 8–12.
[14] J. He, X. Yao, Towards an analytic framework for analyzing the computation time of evolutionary algorithms, Artificial Intelligence 145 (1–2) (2003) 59–97.
[15] E.A. Hirsch, A. Kojevnikov, UnitWalk: A new SAT solver that uses local search guided by unit clause elimination, Annals of Mathematics and Artificial Intelligence 43 (2005) 91–111.
[16] W. Hoeffding, Probability inequalities for sums of bounded random variables, Journal of the American Statistical Association 58 (301) (1963) 13–30.
[17] M. Iosifescu, Finite Markov Processes and Their Applications, John Wiley & Sons, Chichester, 1980.
[18] K. Iwama, S. Tamaki, Improved bounds for 3-SAT, in: Proceedings of the Fifteenth Annual ACM–SIAM Symposium on Discrete Algorithms, 2004, pp. 321–322.
[19] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[20] F. Lardeux, F. Saubion, J.K. Hao, GASAT: A genetic local search algorithm for the satisfiability problem, Evolutionary Computation 14 (2) (2006) 223–253.
[21] L.A. Levin, Universal search problems, Problemy Peredachi Informatsii 9 (1973) 115–116 (in Russian), translation: Problems of Information Transmission 9 (3) (1975) 265–266.
[22] M. Mezard, G. Parisi, R. Zecchina, Analytic and algorithmic solution of random satisfiability problems, Science 297 (2002) 812–815.
[23] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge Univ. Press, Cambridge, UK, 1995.
[24] A.E. Nix, M.D. Vose, Modeling genetic algorithms with Markov chain, Annals of Mathematics and Artificial Intelligence 5 (1) (1992) 79–88.
[25] C.H. Papadimitiou, On selecting a satisfying truth assignment, in: Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science, 1991, pp. 163–169.
[26] C.H. Papadimitiou, Computational Complexity, Addison Wesley, 1994.
[27] P. Paturi, P. Pudlak, M.E. Saks, F. Zane, An improved exponential-time algorithm for k-SAT, Journal of the ACM 52 (3) (2005) 337–364.
[28] D. Rolf, Improved bound for the PPSZ/Schöning-algorithm for 3-SAT, Electronic Colloquium on Computational Complexity (ECCC), Report No. 159, 2005.
[29] U. Schöning, A probabilistic algorithm for k-SAT and constraint satisfaction problems, in: Proceedings of the 40th Annual Symposium on Foundation of Computer Science, 1999, pp. 410–414.
[30] U. Schöning, Principles of stochastic local search, in: Proceedings of Sixth Conference on Unconventional Computation, Kingston, Canada, 2007, in: Lecture Notes in Computer Science, vol. 4618, 2007, pp. 178–187.
[31] B. Selman, H. Levesque, D. Mitchell, A new method for solving hard satisfiability problems, in: Proceedings of Tenth National Conference on Artificial Intelligence (AAAI-92), San Jose, 1992, pp. 440–446.
[32] B. Selman, H.A. Kautz, B. Cohen, Noise strategies for improving local search, in: Proceedings of Twelfth National Conference on Artificial Intelligence (AAAI-94), vol. 1, 1994, pp. 337–343.
[33] W. Wei, B. Selman, Accelerating random walks, in: Proceedings of 8th Intl. Conference on the Principles and Practice of Constraint Programming (CP-2002), in: Lecture Notes in Computer Science, vol. 2470, 2002, pp. 216–232.
[34] H. Zhang, M. Stickel, Implementing the Davis Putnam method, Journal of Automated Reasoning 24 (12) (2000) 277–296.