Phase Transition in a Random NK Landscape Model

Kyomin Jung[*]
Department of Mathematics
MIT
77 Massachusetts Avenue
Cambridge, MA02139, USA
kmjung@mit.edu

Sung-Soon Choi[*]
School of Computer Science
and Engineering
Seoul National University
Seoul, 151-742 Korea
sschoi@soar.snu.ac.kr

Jeong Han Kim
Microsoft Research
One Microsoft Way
Redmond, WA98052, USA
jehkim@microsoft.com

ABSTRACT

An analysis for the phase transition in a random NK landscape model is given. For the fixed ratio model, NK(n, k, z), Gao and Culberson [17] showed that a random instance generated by NK$(n, 2, z)$ with $z > z_0 = \frac{27-7\sqrt{5}}{4}$ is asymptotically insoluble. Based on empirical results, they conjectured that the phase transition occurs around the value $z = z_0$. We prove that an instance generated by NK$(n, 2, z)$ with $z < z_0$ is soluble with positive probability by providing a variant of the unit clause algorithm. Using branching process arguments, we also reprove that an instance generated by NK$(n, 2, z)$ with $z > z_0$ is asymptotically insoluble. The results show the phase transition around $z = z_0$ for NK$(n, 2, z)$. In the course of the analysis, we introduce a generalized random 2-SAT formula, which is of self interest, and show its phase transition phenomenon.

Keywords

NK landscape, phase transition, random k-SAT problem, unit clause algorithm, branching process

1. INTRODUCTION

A fitness landscape is a function that maps each genetic composition (genotype) to the fitness of the expression (phenotype) of the genetic composition in an environment. The fitness landscape sometimes refers its graphical representation as the word "landscape" indicates. The notion of fitness landscape dates back to Wright [37]. It is well known that, for an organism, the contribution of one gene to the overall fitness generally depends on other genes. Such an interaction between genes is called *epistasis*. Wright paid attention to how epistasis affects the shape of the fitness landscape. When there is no epistasis between genes, the fitness landscape has a unimodal shape since the contribution of each gene to the fitness can be optimized independently of other genes. In the case that epistases exist, the fitness landscape may have a number of local fitness optima. A population of organisms would evolve to reach one of those local optima in a hill-climbing manner. These phenomena first observed by Wright have had a great influence on the design of fitness landscape models afterwards.

Mathematical models to study the evolution on fitness landscape have been proposed by many researchers including Franklin and Lewontin [15], Lewontin [26], Ewens [12], Kauffman and Weinberger [23], Macken ad Perelson [27]. Among them, the *NK model* proposed by Kauffman [22] has attracted considerable attention. Kauffman devised the NK model of fitness landscapes to investigate how the "ruggedness" of a landscape changes according to the degree of epistasis. He assumed that the fitness landscapes of the realistic biological environments have some correlation structures, in that the fitness value of one genotype and those of similar genotypes are positively correlated. The NK model is a mathematical model that generates fitness landscapes with correlation structures in which we can control the degree of epistasis and so, indirectly, the ruggedness and correlation degrees of the landscapes.

An NK landscape is specified by two parameters n and k, where n represents the number of genes an organism has and k stands for the number of other genes that epistatically affect the contribution of each gene to the overall fitness value of the organism. For example, if k is zero, there is no interaction among genes. Generally, the parameter k plays a role in controlling the degree of epistases between genes. In other words, the larger the value of k is, the more genes interact one another in constructing the fitness landscape. Through experiments in a few types of the NK model, Kauffman suggested that the ruggedness of the landscape generally increases as k increases [22].

The NK model has been used in biology, physics, and so on. In biology, the NK model explains evolutions of biological objects including amino acid sequences [23] [24] [27], protein or RNA sequences [32] [4] [13] [14] [29], molecular quasispecies [10]. The NK model has been served as a reference point for understanding the properties of those biological objects. In statistical physics, models of spin-glasses are investigated from the viewpoint of NK models in [33]. The evolution of organizations in a business environment is modelled based on the NK model [25]. The NK model has been used as a benchmark for evaluating various encoding schemes and genetic operators on the evolutionary algorithm and comparing them in the evolutionary computation area [3] [20] [28]. It has been also served as a basis for the design of problem difficulty measures for evolutionary algorithms

[*]This work was partially carried in Microsoft Research and partially supported by Institute of Theory and Education for Computing (ITEC) at Seoul National University.

[21] [31] and the design of epistasis measures [30].

The NK model itself has been studied over years. Kauffman [22] analyzed various features of the NK model in terms of adaptive walks. Weinberger [33] and Fontana *et al.* [14] carried out more detailed analysis of such walks. Evans and Steinsaltz [11] showed the asymptotic number of local optima in NK landscapes. Weinberger [34] and, later, Wright *et al.* [36] studied the computational complexities of problems related to NK landscapes. Gao and Culberson [18] showed a treewidth result for NK landscapes in a probabilistic way.

Recently, Gao and Culberson [17] provided results about the phase transition in a random NK landscape model. A phase transition in probabilistic combinatorial theory refers to the phenomenon that the probability that an instance of the random model has a property rapidly changes as the order parameter of the random model changes around a certain value. Before describing the results about the phase transition, we present the original NK landscape model proposed by Kauffman [22] and the probabilistic models proposed by Gao and Culberson. An NK landscape $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i, \Pi(x_i))$ is a real-valued function defined on the set of binary n-tuples, $\{0,1\}^n$. It is a summation of local fitness functions f_i's, where each f_i has a main variable x_i and the variables in the neighborhood of x_i as inputs. Here the neighborhood $\Pi(x_i)$ is a subset of the set $\{x_1, x_2, \ldots, x_n\} \setminus \{x_i\}$ and its size $|\Pi(x_i)|$ is k. Kauffman considered the variables in the neighborhood $\Pi(x_i)$ in two different ways, adjacent neighborhood and random neighborhood. In the adjacent neighborhood model, $\Pi(x_i)$ consists of the closest k variables (with a certain tie-break) to the main variable x_i with respect to the indices modulo n. In the random neighborhood model, $\Pi(x_i)$ is composed of the k variables chosen uniformly at random from $\{x_1, x_2, \ldots, x_n\} \setminus \{x_i\}$. Each local fitness function is fully specified by assigning a real number uniformly distributed between zero and one for each of 2^{k+1} inputs independently of other inputs.

By restricting the fitness values of local fitness function to 0 and 1, one may consider the computational aspect of the NK model more easily. Given an NK landscape f, a decision problem called *solubility problem* is to ask whether the maximum of f is equal to n. An NK landscape f is called *insoluble* if there is no assignment having f equal to n. Weinberger [34] and Wright *et al.* [36] proved that, while the solubility problem for the NK landscape with adjacent neighborhood can be solved in polynomial time for a fixed k, the problem for the NK landscape with arbitrary neighborhood is NP-complete for $k \geq 2$.

To investigate the difficulties of the solubility problems for typical NK landscapes with random neighborhood, Gao and Culberson proposed two probabilistic models of NK landscapes, the *uniform probability model* and the *fixed ratio model* inspired by the two famous random graph models $G(n,p)$ and $G(n,m)$, respectively. In the uniform probability model, the fitness value of each input for a local fitness function is independently assigned to zero with probability p and one with probability $1-p$. This process is independently repeated for each local fitness function. It was shown that an instance generated by this model is asymptotically insoluble or, if it is soluble, the solution can be found in polynomial time with high probability. However, unless p decreases very rapidly with n, it is easy to see that, with high probability, a random instance has a local fitness function that takes zero values for all inputs. This makes the random instance insoluble with high probability. For this reason, the model is not desirable as a model for representing typical instances.

The fixed ratio model overcomes the drawback of the uniform probability model by fixing the ratio of zero values for each local fitness function. The fixed ratio model $\mathrm{NK}(n,k,z)$ is as follows. The value of z ranges in $[0, 2^{k+1}]$. If z is an integer, for each local fitness function f_i, we choose z tuples of 2^{k+1} possible assignments uniformly at random and independently of other f_j's. Then $f_i = 0$ for those tuples and $f_i = 1$ for the other tuples. If z is not an integer so that $z = \lfloor z \rfloor + h$ $(0 < h < 1)$, we specify the fitness values of $\lfloor (1-h)n \rfloor$ local fitness functions as if they were local fitness functions in $\mathrm{NK}(n,k,\lfloor z \rfloor)$ and those of the rest of the local fitness functions as if they were in $\mathrm{NK}(n,k,\lfloor z \rfloor + 1)$. Another way to specify the fitness values of local fitness functions is that we regard each local fitness function as if it were a local fitness function in $\mathrm{NK}(n,k,\lfloor z \rfloor)$ with probability $1-h$ and in $\mathrm{NK}(n,k,\lfloor z \rfloor + 1)$ with probability h, independently of all others. For example, if $z = 2+h$, then each local fitness function has zero values for two random assignments with probability $1-h$ and three random assignments with probability h. This new model is denoted by $\overline{\mathrm{NK}}(n,2,z)$. It is easy to see that $\overline{\mathrm{NK}}(n,2,z)$ is essentially the same as $\mathrm{NK}(n,2,z)$.

For $k = 2$, it was proved [17] that an instance generated by the fixed ratio model with $z > z_0 = \frac{27-7\sqrt{5}}{4} \approx 2.837$ is almost always insoluble, where an event A_n *almost always* occurs if $\lim_{n\to\infty} \Pr[A_n] = 1$. And it was empirically suggested that the instances generated by the model with $z < z_0$ are soluble and the solutions are found in polynomial time with high probability. From these, Gao and Culberson conjectured that the phase transition takes place around $z = z_0$ in the fixed ratio model with $k = 2$.

In this paper, we prove that an instance generated by the model with $z < z_0$ is soluble with positive probability by providing a polynomial time algorithm. This settles the conjecture in an affirmative way. Using branching process arguments, we also reprove that an instance generated by the model with $z > z_0$ is almost always insoluble:

Theorem 1. *If $0 < z < z_0$, then there exists $\alpha > 0$ depending on z such that the probability of $NK(n,2,z)$ being soluble is at least α as n goes to infinity. If $z > z_0$, then $NK(n,2,z)$ is almost always insoluble.*

Though it is a very interesting question, we have no idea whether α can be arbitrarily close to 1 or not.

To prove Theorem 1, we reduce the solubility problem of an NK landscape to the $(k+1)$-SAT problem as in [17]. For given Boolean variables, the variables and their complements are called *literals*. Two literals are *strictly distinct* if their underlying variables are different. A *k-clause* is a disjunction of k strictly distinct literals and a *k-SAT formula* is a conjunction of k-clauses. Given a k-SAT formula F, the k-SAT problem is to ask whether there is a truth assignment satisfying F.

Let an NK landscape $f = \sum_{i=1}^{n} f_i(x_i, \Pi(x_i))$. For each local fitness function f_i, we construct the $(k+1)$-clauses with the literals of the main variable and neighborhood variables of f_i such that f_i is equal to zero only for the assignments that do not satisfy one of the clauses. For example, suppose that a local fitness function $f_i(x_i, x_j, x_k)$ has zero value only when (x_i, x_j, x_k) is one of $(0,0,0)$, $(0,1,0)$, and $(1,1,0)$. Then, we construct three 3-clauses $(x_i \vee x_j \vee x_k)$, $(x_i \vee \overline{x_j} \vee x_k)$, and $(\overline{x_i} \vee \overline{x_j} \vee x_k)$ for $f_i(x_i, x_j, x_k)$. We take the conjunction of the all $(k+1)$-clauses obtained from all the f_i's to construct a $(k+1)$-SAT formula F. It is easy to check that f is soluble if and only if F is satisfiable. Thus, it is sufficient to consider the phase transition for the satisfiability of the 3-SAT formula F.

There have been much studies for the phase transition of the satisfiability of the random 3-SAT formula in which the 3-clauses are chosen independently and uniformly at random [1] [2] [9] [16]. In verifying lower bounds of the threshold, many results were obtained by applying variants of the unit clause algorithm that were first analyzed by Chao and Franco [5] [6]. We will also apply a variant of the unit clause algorithm to the 3-SAT formula reduced from the random NK landscape $\mathrm{NK}(n, 2, z)$ in the subcritical region of the phase transition. In Section 2, we describe the unit clause algorithm and investigate some properties of the reduced 3-SAT formula when the unit clause algorithm is applied to it. The properties suggest that it is useful to consider four types of random 2-clauses or random equalities (of truth values of variables).

In Section 3, we introduce a generalized random 2-SAT formula consisting of the random 2-clauses and the random equalities presented in Section 2. It generalizes the well-known random 2-SAT formula in which the 2-clauses are chosen independently and uniformly at random [7]. After a parameter D is introduced, a threshold phenomenon result is obtained: A random 2-SAT formula generated by the model is satisfiable with positive probability if $D < 1$ and almost always unsatisfiable if $D > 1$. It turns out that the threshold is not sharp.

In Section 4, we provide the threshold phenomenon result for the satisfiability of the reduced 3-SAT formula, or equivalently, the proof of Theorem 1. To obtain the result for the subcritical region, we use similar approaches developed in Section 3. For the supercritical region, we introduce another random 2-SAT model, which is similar to the generalized random 2-SAT model presented in Section 2. The formula generated according to the model consists of random 2-clauses resolved from the 3-SAT formula reduced from $\mathrm{NK}(n, 2, z)$.

2. THE UNIT CLAUSE ALGORITHM

In the subcritical region, we will apply a variant of unit clause algorithm to the 3-SAT formula $F(n, 2, z)$ reduced from a random instance of $\mathrm{NK}(n, 2, z)$, and show that the algorithm finds a satisfying assignment with positive probability. The 3-clauses in the formula are to be regarded as ordered 3-tuples and (copies of) literals came from main variables are placed in the first coordinate of the corresponding 3-clauses. Those (copies of) literals are called main (copies of) literals.

We now consider unit clause algorithm (UC). UC takes as input a formula F and outputs a satisfying assignment of the input, or "cannot determine." In each iteration of the loop of UC, it chooses the literal l contained in a unit clause chosen uniformly at random among all the unit clauses. If there is no unit clause, it chooses a literals l uniformly at random among all the literals not assigned truth values. And it sets l to be true. Then all the clauses containing l are satisfied and all the clauses containing $\bar{l}$ are shortened to the clauses without $\bar{l}$. UC fails to produce a satisfying assignment if and only if 0-clause, a clause with no literal, is created.

```
UC($F$):
    $V \leftarrow \{x_1, x_2, \ldots x_n\}$;
    $S \leftarrow \varnothing$;
    For $t = 0, 1, \ldots, n-1$
        If $|C_1(t)| \neq 0$
            Choose $l$ uniformly at random from $C_1(t)$;
            $V \leftarrow V - \{var(l)\}$;
        Else
            Choose $l$ uniformly at random from $L(V)$;
            $V \leftarrow V - \{var(l)\}$;
        Satisfy all clauses of $F$ containing $l$;
        Remove $\bar{l}$ from all clauses of $F$;
        $S \leftarrow S \cup \{l\}$;
    If $C_0(n) = \varnothing$ Output solution $S$;
    Else Output "Cannot determine.";
```

Figure 1: Pseudo code of the unit clause algorithm

Figure 1 describes the pseudo code of UC. For a literal l, let $var(l)$ be the underlying variable of l. For a set $V = \{x_1, \ldots, x_n\}$ of Boolean variables, let $L(V)$ denote the set of $2|V|$ literals on the variables of V. For $i \geq 0$, let $C_i(t)$ denote the collection of all the i-clauses of F at the end of the t^{th} iteration. If $F = F(n, 2, z)$, $C_3(0)$ is the collection of all the clauses of F and the other $C_i(0)$'s are empty. In general, it is easy to see that

$$C_i(t+1) = \{c | c \in C_i(t) \ , \ l \notin c \ , \ \bar{l} \notin c \ \ or \ \ (c \wedge \bar{l}) \in C_{i+1}(t))\}.$$

When we apply UC to $F(n, 2, z)$, there are three main distinctive properties to be considered. First, there may be a pair of 3-clauses of the form $(l_1 \vee l_2 \vee l_3)$ and $(\overline{l_1} \vee l_2 \vee l_3)$. If $\bar{l}_3$ is set to be 1, then two clauses $(l_1 \vee l_2)$ and $(\overline{l_1} \vee l_2)$ would be created. The conjunction of the two clauses is equivalent to the unit clause (l_2). So we will regard it as the unit clause. This property is called *sublimation*. If we apply UC without sublimation, UC almost always fails to satisfy F. Note that there are $\Theta(n)$ number of pairs of 3-clauses of the form $(l_1 \vee l_2 \vee l_3)$ and $(\overline{l_1} \vee l_2 \vee l_3)$ in $C_3(0)$. In the process of UC if $\overline{l_2}$ is set to be true, then two clauses $(l_1 \vee l_3)$ and $(\overline{l_1} \vee l_3)$ would be created. Then if $\overline{l_3}$ is set to be true, they would be reduced to a pair of unit clauses (l_1) and $(\bar{l}_1)$.

Second, there may be a pair of 3-clauses of the form $(l_1 \vee \overline{l_2} \vee l_3)$ and $(\overline{l_1} \vee l_2 \vee l_3)$. Again, if $\bar{l}_3$ is set to be true, then two clauses $(l_1 \vee \overline{l_2})$ and $(\overline{l_1} \vee l_2)$ would be produced. The conjunction of the two clauses is equivalent to the *equality* $l_1 = l_2$.

Third, the main (copies of) literals from different local fitness functions are strictly distinct. This fact turns out to increase the threshold value.

In the process of UC, 2-clauses are produced. Some pair of 2-clauses will become equalities by the second property. Due to the third property, 2-clauses with main variables will appear so that the literals in their first places are strictly distinct. Similarly, equalities with main variables will appear too. Pairs of two clauses like $(l_1 \vee l_2)$ and $(\overline{l_1} \vee l_2)$ do not appear because of the sublimation property. Motivated by these facts, we will separately consider a generalized random 2-SAT formula consisting of random 2-clauses and equalities,

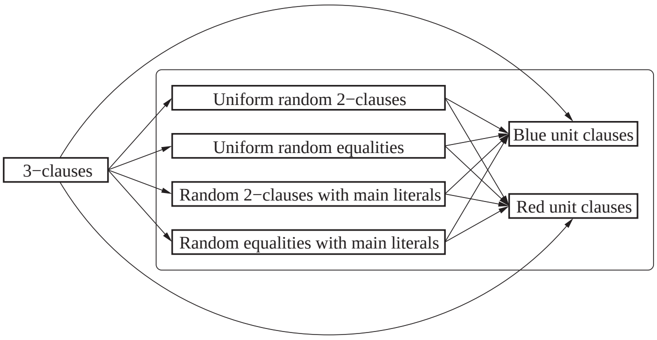
3−clauses
Uniform random 2−clauses
Uniform random equalities
Random 2−clauses with main literals
Random equalities with main literals
Blue unit clauses
Red unit clauses

Figure 2: Flow diagram of clauses in the process of UC

both with and without main variables.

As mentioned below, unit clauses consisting of main (copies of) literals and unit clauses consisting of other (copies of) literals have different properties. So we will consider two types of unit clauses. Unit clauses consisting of main literals and the (copies of) literals therein are to be colored red. The other unit clauses and the (copies of) literals therein are to be colored blue. Then Figure 2 is the flow diagram of clauses in the process of UC.

3. A GENERALIZED RANDOM 2-SAT FORMULA

In this section, we define a generalized random 2-SAT formula and examine its satisfiability. As mentioned in Section 2, the generalized random 2-SAT formula has four types of random 2-clauses or equalities. Here 2-clauses and equalities are to be regarded as ordered pairs. The first type consists of typical uniform random clauses, that is, clauses chosen uniformly at random among all the 2-clauses. The second type consists of uniform random equalities over all the literals. The third and fourth types are the same as the first and the second types, respectively, except that the copy of literals in the first places of the clauses or the equalities are pairwise strictly distinct. Those copies of literals are called *main literals*. Let c_1, c_2, c_3 and c_4 be non-negative real numbers with $c_3 + c_4 \leq 1$. Denote $F_i = F_i(n, c_i)$ the conjunction of $c_i n$ 2-clauses or equalities of type i with repetition, $1 \leq i \leq 4$. Denoted by $F(n, c_1, c_2, c_3, c_4)$ is the conjunction of the four random formulae with pairwise strictly distinct main literals.

If $c_2 = c_3 = c_4 = 0$, it is well known [7, 8, 19] that $F(n, c_1, 0, 0, 0)$ is almost always satisfiable if $c_1 < 1$ and almost always unsatisfiable if $c_1 > 1$. It turns out that the parameter

$$D = c_1 + 2c_2 + c_3 + 2c_4 - \frac{(c_3 + 2c_4)^2}{4} \tag{1}$$

plays a similar role in the general case, as D essentially determines the branching ratio. Roughly speaking, the branching ratio is the expected number of unit clauses produced when a literal is set to be true. This is why a variant of UC succeeds with positive probability if $D < 1$, and it almost always fails if $D > 1$. More precisely, we have the following theorem.

THEOREM 2. *If $D < 1$, then there exists $\alpha > 0$ depending on c_i's so that the probability of $F(n, c_1, c_2, c_3, c_4)$ being satisfiable is at least α as n goes to infinity. If $D > 1$, then the random formula is almost always unsatisfiable.*

It is worth to note that α may not be close to 1 if $c_2 > 0$ or $c_4 > 0$ as it is not hard to see that $(c_2 + c_4)n$ equalities imply $l = \bar{l}$ for a literal l with positive probability.

Theorem 2, in particular, says that the existence of main literals makes the random formula easier to be satisfied. For example, if there are $0.1n$ uniform random 2-clauses and n random 2-clauses with main literals, then the random formula is satisfiable with positive probability. On the other hand, if there are $1.1n$ uniform random 2-clauses, the random formula is almost always unsatisfiable. If one equality is regarded as its corresponding two 2-clauses then $c_1 + 2c_2 + c_3 + 2c_4$ represents the total number of 2-clauses. The extra term $-(c_3 + 2c_4)^2/4$ is the effect of the existence of main literals.

3.1 Subcritical Region

Now we prove the first part of Theorem 2. Without loss of generality, we may assume $c_1 + c_2 > 0$ and $c_3 + c_4 > 0$. Otherwise, some uniform random 2-clauses or random 2-clauses with main literals might be added to F while the conditions $D < 1$ and $c_3 + c_4 \leq 1$ are kept. Here we define some notations. "At time t" means after t times of iteration of UC, or equivalently, after t literals have been set. Let $V(t)$ denote the set of variables not assigned truth values at time t. For $1 \leq i \leq 4$, let $F_i(t)$ denote the conjunction of remaining 2-clauses or equalities of F_i at time t. Define $|F_i(t)|$ to be the number of 2-clauses or equalities in $F_i(t)$. Let $F(t) = F_1(t) \wedge F_2(t) \wedge F_3(t) \wedge F_4(t)$.

As in Section 2, unit clauses consisting of main (copies of) literals and the main (copies of) literals themselves are colored red. The other unit clauses and the (copies of) literals therein are colored blue. Let $B(t)$ and $R(t)$ denote the set of blue unit clauses and red unit clauses at time t, respectively. Let $V_M(t)$ denote the set of the underlying variables of the main literals of $F_3(t)$ and $F_4(t)$.

As mentioned, we apply a variant of UC that uses a different literal selection policy. Think of UC as an imaginary server whose task is satisfying one unit clause, if any, at each time. We regard $B(t)$ and $R(t)$ as two task queues that the server works for. The server will work for one queue at a time and the queue selection is made randomly with a given probability p, which will be specified later. We call this modified UC *UC with switching server policy (UCS)*. Figure 3 describes the pseudo code of UCS. Note that if $c_3 + c_4 = 1$ and $p < 1$, then UCS may encounter the case that a literal in $L(V(t) - V_M(t))$ must be chosen while $V(t) - V_M(t)$ is empty, which is, of course, impossible. We first consider the case that $c_3 + c_4 < 1$. Let

$$p = \frac{c_1 + 2c_2 + \sqrt{(c_1 + 2c_2)^2 + 2(c_1 + 2c_2)(c_3 + 2c_4)}}{c_1 + 2c_2 + c_3 + 2c_4 + \sqrt{(c_1 + 2c_2)^2 + 2(c_1 + 2c_2)(c_3 + 2c_4)}}. \tag{2}$$

Note that $0 < p < 1$. We defined p so that the expected number of blue unit clauses produced at each time is less than p and the expected number of red unit clauses produced at each time is less than $1 - p$. Using these facts and by a coupling argument, we will show that, with positive probability, no 0-clause is produced until $(1 - \epsilon)n$ variables are assigned truth values, for a small constant $\epsilon > 0$. When $(1 - \epsilon)n$ variables are assigned truth values, the remaining formula is very sparse and it is easy to show that the formula is satisfiable with high probability.

```
UC with switching server policy ($F$):
  For $t = 0, \ldots, n-1$
    $\chi(t) \leftarrow 1$ with probability $p$, $\chi(t) \leftarrow 0$ otherwise;
    If $\chi(t) = 1$
      If $B(t) \neq \varnothing$
        Pick a unit clause $(l)$
        uniformly at random from $B(t)$;
      Else
        Pick a literal $l$
        uniformly at random from $L(V(t))$;
    If $\chi(t) = 0$
      If $R(t) \neq \varnothing$
        Pick a unit clause $(l)$
        uniformly at random from $R(t)$;
      Else
        If $V(t) - V_M(t) = \varnothing$  Exit;
        Pick a literal $l$
        uniformly at random from $L(V(t) - V_M(t))$;
    Satisfy clauses of $F$ containing $l$;
    Remove all the copies of $\bar{l}$ and sublimate if possible;
```

Figure 3: Pseudo code of UCS

Note that at each time t, $F(t)$ has the same distribution as $F(n-t, c_1(t), c_2(t), c_3(t), c_4(t))$, where $c_i(t) = |F_i(t)|/(n-t)$. This is a crucial property used in the analysis. The distributions of the numbers of blue and red unit clauses highly depend on the sizes of $F_i(t)$'s. So, we first show that $|F_i(t)|$'s are highly predictable using Wormald's theorem [35].

Lemma 1. *We have*

$$\begin{aligned} \mathrm{E}[|F_1(t+1)| - |F_1(t)|] &= -\frac{2|F_1(t)|}{n-t}, \\ \mathrm{E}[|F_2(t+1)| - |F_2(t)|] &= -\frac{2|F_2(t)|}{n-t}, \\ \mathrm{E}[|F_3(t+1)| - |F_3(t)|] &= -(1+p)\frac{|F_3(t)|}{n-t} + o(1), \\ \mathrm{E}[|F_4(t+1)| - |F_4(t)|] &= -(1+p)\frac{|F_4(t)|}{n-t} + o(1). \end{aligned} \tag{3}$$

Proof. Omitted. □

Using Lemma 1 and applying Wormald theorem, we may have

Lemma 2. *We almost always have*

$$\begin{aligned} \frac{|F_1(t)|}{n-t} &= c_1(1-\frac{t}{n}) + o(1), \frac{|F_2(t)|}{n-t} = c_2(1-\frac{t}{n}) + o(1), \\ \frac{|F_3(t)|}{n-t} &= c_3(1-\frac{t}{n})^p + o(1), \frac{|F_4(t)|}{n-t} = c_4(1-\frac{t}{n})^p + o(1). \end{aligned}$$

Proof. Omitted. □

For $1 \le i \le 4$, let $b_i(t)$ be the number of blue unit clauses coming from $F_i(t)$ at time t and let $r_i(t)$ be the number of red unit clauses coming from $F_i(t)$ at time t. We will obtain the expectations and distributions of $b_i(t)$'s and $r_i(t)$'s, conditioned on $|F_i(t)|$'s. Suppose that UCS sets a literal l to be true at time t. Then $b_1(t)$ is the number of 2-clauses in $F_1(t)$ that contain $\bar{l}$. And $r_1(t) = 0$ since there is no main literal in $F_1(t)$. Note that, for each 2-clause $(l_1 \vee l_2) \in F_1(t)$, $\Pr[\bar{l} = l_1 \text{ or } \bar{l} = l_2] = \frac{1}{n-t}$. And the 2-clauses in $F_1(t)$ are independent from one another. So $b_1(t)$ has a binomial distribution $\mathrm{Bin}[|F_1(t)|, \frac{1}{(n-t)}]$. The same argument can be applied to have that $r_2(t) = 0$ and $b_2(t)$ has a binomial distribution $\mathrm{Bin}[|F_2(t)|, \frac{2}{(n-t)}]$. For $b_3(t)$, observe that $b_3(t)$ is the number of the 2-clauses in $F_3(t)$ whose main literals are $\bar{l}$. Here we consider two cases according to the value of $\chi(t)$. First, suppose that $\chi(t) = 1$. Then $b_3(t)$ has a Bernoulli distribution with density $\frac{|F_3(t)|}{2(n-t)}$ since $\bar{l}$ may be equal to at most one of the main literals in $F_3(t)$. When $\chi(t) = 0$, $b_3(t) = 0$ since $\bar{l}$ cannot be equal to any of the main literals in $F_3(t)$ and hence only unit clauses with main literals are produced. For $r_3(t)$, observe that $r_3(t)$ is the number of 2-clauses in $F_3(t)$ whose second literals are equal to $\bar{l}$. Suppose that l is not strictly distinct with one of main variables of $F_3(t)$. Then, since exactly one 2-clause in $F_3(t)$ has l or $\bar{l}$ as main literal, $r_3(t)$ has a binomial distribution $\mathrm{Bin}[|F_3(t)| - 1, \frac{1}{2(n-t-1)}]$. Otherwise, $r_3(t)$ has a binomial distribution $\mathrm{Bin}[|F_3(t)|, \frac{1}{2(n-t-1)}]$. Thus, the distribution of $r_3(t)$ is a linear combination of $\mathrm{Bin}[|F_3(t)|-1, \frac{1}{2(n-t-1)}]$ and $\mathrm{Bin}[|F_3(t)|, \frac{1}{2(n-t-1)}]$. The same argument can be applied to obtain the distributions of $b_4(t)$ and $r_4(t)$. If $\chi(t) = 1$, $b_4(t)$ has a Bernoulli distribution with density $\frac{|F_4(t)|}{(n-t)}$ and if $\chi(t) = 0$, then $b_4(t) = 0$. And $r_4(t)$ has a binomial distribution $\mathrm{Bin}[|F_4(t)|-1, \frac{1}{(n-t-1)}]$ if l is not strictly distinct with one of the main variables of $F_4(t)$. Otherwise, $r_4(t)$ has a binomial distribution $\mathrm{Bin}[|F_4(t)|, \frac{1}{(n-t-1)}]$. Thus, the distribution of $r_4(t)$ is a linear combination of $\mathrm{Bin}[|F_4(t)| - 1, \frac{1}{(n-t-1)}]$ and $\mathrm{Bin}[|F_4(t)|, \frac{1}{(n-t-1)}]$.

The expectations of $b_i(t)$ and $r_i(t)$ are as follows;

$$\begin{bmatrix} \mathrm{E}[b_i(t)] \\ \mathrm{E}[r_i(t)] \end{bmatrix} = T_i(t) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix} + o(1),$$

where

$$\begin{aligned} T_1(t) &= c_1(1-\frac{t}{n}) \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \\ T_2(t) &= c_2(1-\frac{t}{n}) \begin{bmatrix} 2 & 2 \\ 0 & 0 \end{bmatrix}, \\ T_3(t) &= c_3(1-\frac{t}{n})^p \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \\ T_4(t) &= c_4(1-\frac{t}{n})^p \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \end{aligned}$$

Then, for $b(t) = \sum_{i=1}^4 b_i(t)$, $r(t) = \sum_{i=1}^4 r_i(t)$, and $T(t) = \sum_{i=1}^4 T_i(t)$, we have that

$$\begin{bmatrix} \mathrm{E}[b(t)] \\ \mathrm{E}[r(t)] \end{bmatrix} = T(t) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix} + o(1).$$

Lemma 3. *(Main Lemma) We have*

$$\begin{bmatrix} \mathrm{E}[b(t)] \\ \mathrm{E}[r(t)] \end{bmatrix} < \begin{bmatrix} p \\ 1-p \end{bmatrix},$$

where the inequality for the vectors means that the inequality holds for each pair of the entries.

Proof. Since $T(t) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix} \leq T(0) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix}$, it suffices to show that

$$T(0) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix} < \begin{bmatrix} p \\ 1-p \end{bmatrix}.$$

Clearly, $T(0) = \begin{bmatrix} c_1 + 2c_2 + \frac{1}{2}c_3 + c_4 & c_1 + 2c_2 \\ \frac{1}{2}c_3 + c_4 & \frac{1}{2}c_3 + c_4 \end{bmatrix}$ has two nonnegative eigenvalues,

$$\frac{c_1 + 2c_2 + c_3 + 2c_4 \pm \sqrt{(c_1 + 2c_2)^2 + 2(c_1 + 2c_2)(c_3 + 2c_4)}}{2}.$$

Let λ be the larger one. Since $D < 1$ implies $\lambda < 1$ and $\begin{bmatrix} p \\ 1-p \end{bmatrix}$ is an eigenvector of $T(0)$ corresponding to λ,

$$T(0) \cdot \begin{bmatrix} p \\ 1-p \end{bmatrix} = \lambda \begin{bmatrix} p \\ 1-p \end{bmatrix} < \begin{bmatrix} p \\ 1-p \end{bmatrix},$$

as desired. □

Then using these facts and a coupling argument, we show that the sizes of $\sum |B(t)|$ and $\sum |R(t)|$ are $O(n)$. In the course of that, we use a simplified version of *Lazy-server lemma*, which was introduced by Achlioptas [1]. Suppose that there is a server so that the probability that the server would work at time t is $w(t)$ and, if it works, it can handle one task per unit time. And the expected number of tasks that arrive to the server at time t is $z(t)$. Then Lazy-server lemma says that if $z(t)$ is bounded above by $w(t)$ uniformly for all t, then the sum of sizes of the task queue over all t would not become excessively large.

Lemma 4. *We almost always have*

$$\sum_{t=0}^{(1-\epsilon)n} |B(t)| < Cn, \quad \max_{0 \leq t \leq (1-\epsilon)n} |B(t)| < \log^K n.$$

$$\sum_{t=0}^{(1-\epsilon)n} |R(t)| < Cn, \quad \max_{0 \leq t \leq (1-\epsilon)n} |R(t)| < \log^K n,$$

for some constants C, K.

Proof. Omitted. □

Now we prove that with positive probability no 0-clause is produced until $t = (1-\epsilon)n$. Under the condition that no 0-clause is produced until time $t-1$, the probability that the same holds until time t is at least

$$(1 - \frac{1}{2(n-t-1)})^{|B(t)|}(1 - \frac{|R(t)|}{2(n-t-1)}) \geq (1 - \frac{2}{\epsilon n})^{|B(t)|+|R(t)|}$$

So the probability that no 0-clause is produced until $t = (1-\epsilon)n$ is at least

$$(1 - \frac{2}{\epsilon n})^{\sum_{t=0}^{n-\epsilon n}(|B(t)|+|R(t)|)} + o(1) \geq (1 - \frac{2}{\epsilon n})^{Cn} + o(1) = e^{-\frac{2C}{\epsilon}} + o(1).$$

Now consider the case that $c_3 + c_4 = 1$. As mentioned above, since $V(0) - V_M(0)$ is empty in this case. Unless $p = 1$, UCS may encounter the case that $\chi(t) = 0$ but $V(t) - V_M(t)$ is empty. However, when we set p as in (2), UCS may not encounter the case that $\chi(t) = 0$ but $V(t) - V_M(t)$: Initially, $|V(t) - V_M(t)| = 0$. At each step t of the first δn steps, if $\chi(t) = 1$, then the expected change of $|V(t) - V_M(t)|$ is $1 + O(\delta)$, as one uniform random literal eliminates $1 + O(\delta)$ 2-clauses or equalities with main literals, in expectation. The other effects are small enough if δ is small enough. If $\chi(t) = 0$, then the expected change is $O(\delta)$, as a non-main literal eliminates one 2-clause or equality with main literal, in average. Thus, at each step, $|V(t) - V_M(t)|$ increases by $p + O(\delta)$, in average, and hence $|V(t) - V_M(t)| > 0$ for $t \geq 1$ with positive probability. Notice that UCS produces 0-clause in the first δn steps with probability $O(\delta)$ (cf. Lazy-Server Lemma). Therefore, with positive probability, UCS proceeds to the first δn steps without encountering $\chi(t) = 0$ and $V(t) - V_M(t) = \emptyset$. After $t = \delta n$ steps, it is easy to see that $c_3(t) + c_4(t) \leq (1-a)(n-t)$ for some constant $a > 0$, which is covered in the previous case.

3.2 Supercritical Region

Omitted by space limitation.

4. SOLUBILITY OF NK$(N, 2, Z)$

In this section, we prove Theorem 1 for the model $\overline{\text{NK}}(n, 2, z)$. This is enough as $\overline{\text{NK}}(n, 2, z)$ is essentially the same as $\text{NK}(n, 2, z)$. Recall $z_0 = \frac{27 - 7\sqrt{5}}{4} \approx 2.837$. In the first subsection, the result for the subcritical region $z < z_0$ is proven. The next subsection is for another proof for the supercritical region. By the monotonicity of the solubility of $\overline{\text{NK}}(n, 2, z)$, it is enough to consider cases $2 < z < z_0$ and $z_0 < z < 3$.

4.1 Subcritical Region

As in $\text{NK}(n, 2, z)$, a 3-SAT formula F can be reduced from a random instance f of $\overline{\text{NK}}(n, 2, z)$. More precisely, a 3-SAT formula L_j is reduced from each local fitness function f_j of f and F is the conjunction of L_j's. We call L_j a *local formula*. Then a local formula consists of two 3-clauses with probability $1 - h$ and three 3-clauses with probability h, where $z = 2 + h$. Main variables or its negations appeared in a local formula are called main (copies of) literals. Note that any pair of main literals came from different L_j's are strictly distinct. UCS is to be applied to F as in the generalized random 2-SAT problem. In the process of UCS, there appear four types of 2-clauses or equalities as presented in the previous section. Denoted by $F_i(t)$ $(1 \leq i \leq 4)$ is the 2-SAT formula consisting of the 2-clauses or equalities of type i at time t. Denoted by $F_5(t)$ is the 3-SAT formula consisting of remaining local formulae at time t. Let $|F_i(t)|$ be the number of the 2-clauses or equalities in $F_i(t)$ and $|F_5(t)|$ be the number of the local formulae in $F_5(t)$ at time t. It is clear that $F_i(0)$ is empty for $1 \leq i \leq 4$ and $|F_5(0)| = n$. The unit clauses consisting of main literals and the copies of literals therein are colored red. The other unit clauses and the copies of literals therein are colored blue. As in Section 3, we let $B(t)$ ($R(t)$, resp.) be the set of blue (red, resp.) unit clauses at time t.

We run UCS with

$$p = p(t) = p_0 - \frac{t}{10n}, \quad \text{where } p_0 = \frac{(\sqrt{5}-1)}{2} \approx 0.618. \quad (4)$$

We defined $p(t)$ so that the expected number of blue (red, resp.) unit clauses produced at time t is uniformly bounded above by $p(t)$ ($1 - p(t)$, resp.) for $1 \leq t \leq (1-\epsilon)n$, where ϵ is a small constant. Then by a coupling argument and Lazy-server lemma, we will show that the sizes of $\sum_{t=0}^{(1-\epsilon)n} |B(t)|$ and $\sum_{t=0}^{(1-\epsilon)n} |R(t)|$ are $O(n)$. It is easy to show that, with positive probability, no 0-clause is produced until $t = (1 - \epsilon)n$. As in Section 3, at $t = (1-\epsilon)n$, the remaining formula is sparse enough that it is satisfiable with positive probability.

Note that at each time t, $F_1(t)$ consists of uniform random 2-clauses over $V(t)$, and $F_2(t)$ consists of uniform random equalities over $V(t)$. The formula $F_3(t)$ consists of random 2-clauses with main literals over $V(t)$, and $F_4(t)$ consists of random equalities with main literals over $V(t)$, where the main literals in $F_3(t)$ and $F_4(t)$ are pairwise strictly distinct. Let $b_i(t)$ and $r_i(t)$ be the numbers of blue and red unit clauses coming from $F_i(t)$ at time t, respectively. As mentioned in Section 2, during the execution of UCS, there occurs some sublimations. So we also need to consider the number $b_5(t)$ ($r_5(t)$, resp.) of blue (red, resp.) unit clauses produced by the sublimations.

As in Section 3, we investigate $\mathrm{E}[|F_i(t+1)|-|F_i(t)|]$, $\mathrm{E}[b_i(t)]$ and $\mathrm{E}[r_i(t)]$ $(1 \le i \le 5)$ and use Wormald theorem to obtain approximations of $|F_i(t)|$, $\mathrm{E}[b_i(t)]$, and $\mathrm{E}[r_i(t)]$. For $1 \le i \le 4$, let $u_i(t)$ be the number of 2-clauses or equalities that come from $F_5(t)$ to $F_i(t)$ at time t, and $d_i(t)$ be the number of 2-clauses or equalities that is removed from $F_i(t)$ at time t. Then for $1 \le i \le 4$, $\mathrm{E}[|F_i(t + 1)| - |F_i(t)|] = \mathrm{E}[u_i(t)] - \mathrm{E}[d_i(t)]$. As we already obtained the equations for $\mathrm{E}[d_i(t)]$, $\mathrm{E}[b_i(t)]$ and $\mathrm{E}[r_i(t)]$ $(1 \le i \le 4)$ in Section 3, we only need to consider $\mathrm{E}[u_i(t)]$ $(1 \le i \le 4)$, $\mathrm{E}[b_5(t)]$, $\mathrm{E}[r_5(t)]$ and $\mathrm{E}[|F_5(t + 1)| - |F_5(t)|]$.

Lemma 5. *We have*

$$\begin{aligned}
\mathrm{E}[|F_1(t+1)| - |F_1(t)|] &= -\frac{2|F_1(t)|}{n-t} + p(t)(\frac{4}{7} - \frac{h}{7})\frac{|F_5(t)|}{n-t},\\
\mathrm{E}[|F_2(t+1)| - |F_2(t)|] &= -\frac{2|F_2(t)|}{n-t} + p(t)(\frac{1}{14} + \frac{h}{14})\frac{|F_5(t)|}{n-t},\\
\mathrm{E}[|F_3(t+1)| - |F_3(t)|] &= -(1+p(t))\frac{|F_3(t)|}{n-t} + (\frac{8}{7} - \frac{2h}{7})\frac{|F_5(t)|}{n-t} + o(1),\\
\mathrm{E}[|F_4(t+1)| - |F_4(t)|] &= -(1+p(t))\frac{|F_4(t)|}{n-t} + (\frac{1}{7} + \frac{h}{7})\frac{|F_5(t)|}{n-t} + o(1),\\
\mathrm{E}[|F_5(t+1)| - |F_5(t)|] &= -(2+p(t))\frac{|F_5(t)|}{n-t} + o(1).
\end{aligned}$$

Proof. Omitted. □

We apply Wormald theorem to approximate $|F_i(t)|$ for $1 \le t \le (1-\epsilon)n$.

Lemma 6. *For the solution $\varphi_i(x) : [0, 1-\epsilon] \to \mathbb{R}$ of the following system of differential equations,*

$$\begin{aligned}
\frac{d\varphi_1}{dx} &= -\frac{2\varphi_1(x)}{1-x} + (p_0 - 0.1x)(\frac{4}{7} - \frac{h}{7})\frac{\varphi_5(x)}{1-x} & \varphi_1(0) = 0,\\
\frac{d\varphi_2}{dx} &= -\frac{2\varphi_2(x)}{1-x} + (p_0 - 0.1x)(\frac{1}{14} + \frac{h}{14})\frac{\varphi_5(x)}{1-x} & \varphi_2(0) = 0,\\
\frac{d\varphi_3}{dx} &= -(1 + p_0 - 0.1x)\frac{\varphi_3(x)}{1-x} + (\frac{8}{7} - \frac{2h}{7})\frac{\varphi_5(x)}{1-x} & \varphi_3(0) = 0,\\
\frac{d\varphi_4}{dx} &= -(1 + p_0 - 0.1x)\frac{\varphi_4(x)}{1-x} + (\frac{1}{7} + \frac{h}{7})\frac{\varphi_5(x)}{1-x} & \varphi_4(0) = 0,\\
\frac{d\varphi_5}{dx} &= -(2 + p_0 - 0.1x)\frac{\varphi_5(x)}{1-x} & \varphi_5(0) = 1,
\end{aligned}$$

we almost always have

$$|F_i(t)| = \varphi_i(\frac{t}{n}) \cdot n + o(n)$$

uniformly for all $1 \le t \le (1-\epsilon)n$ and $1 \le i \le 5$.

Proof. Omitted. □

Note that expectations of $b_i(t)$ and $r_i(t)$ are as follows;

$$\begin{bmatrix} \mathrm{E}[b_i(t)] \\ \mathrm{E}[r_i(t)] \end{bmatrix} = T_i(t) \cdot \begin{bmatrix} p(t) \\ 1 - p(t) \end{bmatrix} + o(1),$$

where

$$\begin{aligned}
T_1(t) &= \frac{|F_1(t)|}{(n-t)} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},\\
T_2(t) &= \frac{|F_2(t)|}{(n-t)} \begin{bmatrix} 2 & 2 \\ 0 & 0 \end{bmatrix},\\
T_3(t) &= \frac{|F_3(t)|}{(n-t)} \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},\\
T_4(t) &= \frac{|F_4(t)|}{(n-t)} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},\\
T_5(t) &= \frac{(\frac{1}{7} + \frac{2h}{7})|F_5(t)|}{(n-t)} \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.
\end{aligned}$$

Let $b(t) = \sum_{i=1}^{5} b_i(t)$ and $r(t) = \sum_{i=1}^{5} r_i(t)$. Then for

$$T(t) = \sum_{i=1}^{5} T_i(t),$$

$$\begin{bmatrix} \mathrm{E}[b(t)] \\ \mathrm{E}[r(t)] \end{bmatrix} = T(t) \cdot \begin{bmatrix} p(t) \\ 1 - p(t) \end{bmatrix} + o(1).$$

Then as in Section 3 with $D < 1$ and $c_3 + c_4 = 1$, by a coupling argument and Lazy-server lemma, we obtain that almost always $\sum_{t=1}^{(1-\epsilon)n} |B(t)|$ and $\sum_{t=1}^{(1-\epsilon)n} |R(t)|$ are bounded by $O(n)$. Then we see that with positive probability no 0-clause is produced until $t = (1 - \epsilon)n$. It is easy to see that the remaining formula after t steps is satisfiable with positive probability.

4.2 Supercritical Region

Omitted by space limitation.

5. CONCLUSION

In this paper, we analyzed the phase transition in NK landscape on the fixed ratio model, $\mathrm{NK}(n, 2, z)$. We also proposed a generalized random 2-SAT model and introduced a corresponding parameter D. Then a phase transition result for the model is obtained, that is, if $D < 1$, the formula is satisfiable with positive probability, and if $D > 1$, the formula is almost always unsatisfiable. For the proof for the subcritical region, we proposed a variant of unit clause algorithm, the unit clause algorithm with switching server policy, and analyzed it. For the supercritical region, a branching process argument was used.

Using the similar argument as in the generalized random 2-SAT model, it was proved that an instance generated by $\mathrm{NK}(n, 2, z)$ with $z < z_0 = \frac{27-7\sqrt{5}}{4}$ is soluble with positive probability. To the best of our knowledge, this is the first mathematical result that describes the behavior of $\mathrm{NK}(n, 2, z)$ with $z < z_0$. We also reproved that an instance generated by $\mathrm{NK}(n, 2, z)$ with $z > z_0$ is almost always insoluble using a branching process argument. This approach is a novel one and simpler than that of Gao and Culberson.

From these results, we established the threshold value, z_0, of the phase transition in $\mathrm{NK}(n, 2, z)$.

We believe that our approach used for $\mathrm{NK}(n, k, z)$ with $k = 2$ works for general $k \geq 3$ to obtain at least partial results for the phase transition phenomenon.

6. REFERENCES

[1] D. Achlioptas. Setting two variables at a time yields a new lower bound for random 3-SAT. In *32nd Annual ACM Symposium on Theory of Computing*, pages 28–37, 2000.

[2] D. Achlioptas. Lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265(1–2):159–185, 2001.

[3] L. Altenberg. NK fitness landscapes. In T. Bäck, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*. Oxford University Press, 1997.

[4] C. Amitrano, L. Peliti, and M. Saber. Population dynamics in a spin-glass model of chemical evolution. *Journal of Molecular Evolution*, 29:513–525, 1989.

[5] M. T. Chao and J. Franco. Probabilistic analysis of two heuristics for the 3-satisfiability problem. *SIAM Journal on Computing*, 15(4):1106–1118, 1986.

[6] M. T. Chao and J. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem. *Information Science*, 51(3):289–314, 1990.

[7] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33th Annual Symposium on Foundations of Computer Science*, pages 620–627, 1992.

[8] W. Fernandez de la Vega. *On random 2-sat.* unpublished manuscript, 1992.

[9] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-SAT formulae and the satisfiability threshold. In *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 126–127, 2000.

[10] M. Eigen, J. McCaskill, and P. Schuster. The molecular quasispecies. *Advanced Chemical Physics*, 75:149–263, 1989.

[11] S. N. Evans and D. Steinsaltz. Estimating some features of NK fitness landscapes. *Annals of Applied Probability*, 12:1299–1321, 2002.

[12] W. Ewens. *Mathematical Population Genetics*. Springer Verlag, 1979.

[13] H. Flyvbjerg and B. Lautrup. Evolution in a rugged fitness landscape. *Physical Review A*, 46:6714–6723, 1992.

[14] W. Fontana, P. F. Stadler, E. G. Bornberg-Bauer, T. Griesmacher, I. L. Hofacker, M. Tacker, P. Tarazona, E. D. Weinberger, and P. Schuster. RNA folding and combinatory landscapes. *Physical Review E*, 47:2083–2099, 1993.

[15] I. Franklin and R. Lewontin. Is the gene the unit of selection ? *Genetics*, 65:707–734, 1970.

[16] A. M. Frieze and S. Suen. Analysis of two simple heuristics on a random instance of k-SAT. *Journal of Algorithms*, 20(2):312–355, 1996.

[17] Y. Gao and J. Culberson. An analysis of phase transition in NK lanscapes. *Journal of Artificial Intelligence Research*, 17:309–332, 2002.

[18] Y. Gao and J. Culberson. On the treewidth of NK landscapes. In *Genetic and Evolutionary Computation Conference*, pages 948–954, 2003.

[19] A. Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53:469–486, 1996.

[20] W. Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4):335–360, 1997.

[21] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Sixth International Conference on Genetic Algorithms*, pages 184–192, 1995.

[22] S. A. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, pages 527–618. Addison Wesley, 1989. Santa Fe Institute Studies in the Sciences of Complexity.

[23] S. A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.

[24] S. A. Kauffman, E. D. Weinberger, and A. S. Perelson. Maturation of the immune response via adaptive walks on affinity landscapes. In A. S. Perelson, editor, *Theoretical Immunology I*. Addison Wesley, 1988. Santa Fe Institute Studies in the Sciences of Complexity.

[25] D. A. Levinthal. Adaptation on rugged landscapes. *Management Science*, 43:934–950, 1997.

[26] R. Lewontin. *The Genetic Basis of Evolutionary Change*. Columbia University Press, 1974.

[27] C. A. Macken and A. S. Perelson. Protein evolution on rugged landscapes. In *National Academic Science USA*, volume 86, pages 6191–6195, 1989.

[28] P. Merz and B. Freisleben. On the effectiveness of evolutionary search in high-dimensional NK-landscapes. In *IEEE International Conference on Evolutionary Computation*, pages 741–745, 1998.

[29] P. Schuster and P. F. Stadler. Landscapes: Complex optimization problems and biopolymer structures. *Computational Chemistry*, 18:295–324, 1994.

[30] D. I. Seo, Y. H. Kim, and B. R. Moon. New entropy-based measures of gene significance and epistasis. In *Genetic and Evolutionary Computation Conference*, pages 1345–1356, 2003.

[31] T. Smith, P. Husbands, P. Layzell, and M. O'Shea. Fitness landscapes and evolvability. *Evolutionary Computation*, 10(1):1–34, 2002.

[32] E. D. Weinberger. A more rigorous derivation of some properties of uncorrelated fitness landscapes. *Journal of Theoretical Biology*, 134:125–129, 1988.

[33] E. D. Weinberger. Local properties of Kauffman's NK model, a tuneably rugged energy landscape. *Physical Review A*, 44(10):6399–6413, 1991.

[34] E. D. Weinberger. NP completeness of Kauffman's NK model, a tuneably rugged fitness landscape. Technical Report 96-02-003, Santa Fe Institute, Santa Fe, 1996.

[35] N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5(4):1217–1235, 1995.

[36] A. H. Wright, R. K. Thompson, and J. Zhang. The computational complexity of NK fitness functions. *IEEE Trans. on Evolutionary Computation*, 4(4):373–379, 2000.

[37] S. Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Sixth International Congress on Genetics*, volume 1, pages 356–366, 1932.