Probabilistic Methods in Landscape Analysis: phase transitions, interaction structures, and complexity measures

Yong Gao and Joseph Culberson

Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{ygao, joe}@cs.ualberta.ca

Abstract. In recent years, the study of the phase transition behavior and typical-case complexity of search and optimization problems have become an active research topic, drawing attention from researchers from computer science, mathematics, and statistical physics. The probabilistic method that has been developed since 1960s plays an important role in this development. We give a brief introduction to the problems, techniques, and results in the phase transition analysis of search and optimization problems, and present an overview on our recent work, illustrating how probabilistic methods can be used in landscape analysis and how the results give us insight into the behavior and complexity of evolutionary algorithms.

1 INTRODUCTION

In physics, the notion of a phase transition refers to the abrupt change of states (phases) of compounds at certain values of parameters such as the pressure and the temperature. Popular examples of phase transitions in the physical world are the liquid-to-gas transition of water, and the conductor-superconductor transition of electrical resistance of some materials.

Similar phenomena have also been observed in computational and AI systems. For example, a random graph generated by selecting each of the potential edges independently with an edge probability $p = p(n)$ experiences several abrupt changes in its combinatorial properties when the edge probability $p(n)$ increases from $o(n)$ to $\Theta(n^2)$ [1, 2]. For some NP-complete problems such as the Boolean Satisfiability problem(SAT) and the graph coloring problem, it has been observed that the probability of being satisfied also has a phase transition from zero to one at a certain clause-variable ratio [3–5]. See also [6–9] for a series of popular science articles for the history and recent development.

In their seminal work [3], Cheeseman at al. showed that for many NP-complete search problems and some standard algorithms, the typical hardness of the problems is closely related to the critical point, called the *threshold*, where the phase transition occurs. Recent studies on the phase transitions in combinatorial and AI search problems have given us much insight into the typical

complexity of these problems and help in tackling questions such as "where are the really hard problems?" and "why do these hardest problems seem to resist any intelligent algorithms?" [10–13].

Answers to these questions have already stimulated work on designing efficient algorithms and appropriate benchmark problems [9, 14, 15].

In the the study of the phase transitions, the probabilistic method [16, 2], which uses tools from probability theory to analyze properties of combinatorial nature, plays an important role. In fact, the probabilistic method was first developed in the work of P. Erdös and A. Renyi [1] on the phase transition in random graphs.

Fitness landscapes have been widely used in the past to facilitate the analysis of the behavior of algorithms in evolutionary computation. Specially-designed landscapes also comprise of a large class of benchmark problems in the performance evaluation of new algorithms.

In the past several years, we have worked on understanding the characteristics of fitness landscapes from a phase transition perspective. The purpose of the current paper is to give a brief introduction on the study of phase transitions in combinatorial search and AI, and overview our work on the phase transition analysis of fitness landscapes.

2 Probabilistic Method and Phase Transitions in Combinatorial Search

The use of the probabilistic method to study the combinatorial structures started from the work of P. Erdös and A. Renyi [1] on the evolution and threshold phenomena in random graphs. Over the years, the probabilistic method has gained more and more popularity in the analysis of algorithms in general, and in the study of phase transitions in particular. The probabilistic method is an approach that studies properties of combinatorial structures by using techniques and tools from the theory of probability. These techniques include the first moment method, the second moment method, martingales, etc [16]. In this section, we illustrate the use of the probabilistic method by introducing some basic results related to the phase transitions of random SAT problems.

SAT [17] is perhaps one of the most popular and important NP-complete problems in the theory of computational complexity and in artificial intelligence (AI). It is the first problem that was shown to be NP-complete, and many of the studies in the phase transition of search has been on random models of SAT.

Let $\{x_1, x_2, \cdots, x_n\}$ be a set of boolean variables. A literal is either a variable x or its negation $\overline{x}$. A clause is a disjunction of literals and a conjunctive normal form (CNF) formula is the conjunction of a set of clauses. An instance of SAT is a CNF formula and the question is to decide whether there is a truth assignment that satisfies the formula. The problem k-SAT, $k \geq 1$, is a problem where each

clause of the formula has exactly k literals. It is known that k-SAT is NP-complete for $k \geq 3$ and can be solved in linear time for $k = 2$.

One of the well-studied random models for k-SAT is the random k-SAT formula $F_k(n, m)$ which consists of m clauses, each selected independently and uniformly from the set of all $2^k \binom{n}{k}$ possible k-clauses. We call $r = \frac{m}{n}$ the clause-variable ratio.

Intuitively, formulas with a large clause-variable ratio are hard to be satisfied, while formulas with a small clause-variable ratio could have many satisfying solutions. Experimental studies in [10, 5] indicated that at $r \approx 4.2$ there is an abrupt change in the probability that $F_3(n, m)$ has a solution—the probability is asymptotically one for $r < 4.2$ and zero for $r > 4.2$. This leads to the following famous conjecture:

SAT Threshold Conjecture: There is a constant r_k, called the satisfiability threshold such that

$$\lim_n \Pr\{F_k \text{ is satisfiable } \} = \begin{cases} 1, & \text{if } r < r_k; \\ 0, & \text{if } r > r_k \end{cases}$$

After more than ten years of work, the above conjecture is far from settled. However, much insight about the hardness and its algorithmic impact has been gained in the effort to try to understand the behavior around the threshold and to improve the upper and lower bounds on the threshold.

2.1 Upper and Lower Bounds

Upper bounds on the SAT threshold have been established by using Markov's inequality:

$$\Pr\{X > 0\} \leq E[X], \tag{1}$$

where X is a random variable and $E[X]$ is its expectation.

For example, let $I_a, a \in \{0, 1\}^n$, denote the indicator function of the event that a satisfies $F_3(n, m)$, and $I = \sum_a I_a$. We have

$$\Pr\{I > 0\} \leq E[I] = \sum_a E[I_a] = 2^n (\frac{7}{8})^m.$$

Thus, if $m/n > \log_{8/7} 2 = 5.191$, $F_3(n, m)$ is asymptotically unsatisfiable with probability one. This gives an upper bound for the SAT threshold.

By taking into consideration the intrinsic structure of the solution space, better upper bounds have been obtained. See [18] for an account on a series of hard work that improves the upper bound from 5.191 to 4.596.

Lower bounds on the threshold are usually obtained by analyzing polynomial algorithms based on the Unit Clause heuristic: (1) If there are any clauses containing only one literal, then picking one of them and satisfy it; If not, randomly pick an unset variable and assign it to TRUE (or FALSE) randomly and uniformly.

By analyzing conditions at which these algorithms succeed asymptotically with probability one, lower bounds can be established. Interested readers can consult [19] and the reference therein for further details.

Lower bounds can also be established by the second moment method based on Chebyshev's inequality:

$$\Pr\{X = 0\} \leq \frac{var[X]}{(E[X])^2} \tag{2}$$

where $var(X)$ is the variance of X. The difficulty in using the second moment method lies in the fact that X is usually a sum of a set of random variables that are only "close" to being independent. To bound the variance of such a variable, combinatorial structures intrinsic to the problem under consideration should be utilized in a smart way [16, 20].

2.2 The Sharpness of a Phase Transition

In addition to the location of the critical point of the phase transition, the sharpness of the phase transition is also interesting. Roughly speaking, a phase transition for a combinatorial property is said to be sharp if the transition interval tends to zero faster than the critical parameter itself. A sufficient condition for a property such as the satisfiability of SAT to have a sharp phase transition has been established [21]. The condition basically indicates that in order to have a sharp transition, there should not exist small signatures (properties that can be determined locally) that can approximate the property.

2.3 Backbones and Complexity

In the statistical mechanics approach to the random SAT phase transition, a boolean variable is identified with a binary variable, called a spin, that takes its values on $\{-1, 1\}$ (-1 for FALSE and 1 for TRUE). A CNF formula F is associated with an energy function $\mathbf{E}[F, S], S \in \{-1, 1\}^n$ defined on the possible assignments to the binary variables, indicating the number of clauses not satisfied by the assignment.

To investigate the behavior of the optimum of the energy function and the structure of the space of the optimal solutions, statistical physics views the SAT problem as a system of spins whose configuration is governed by the Boltzmann distribution

$$p(S) = \frac{1}{Z} e^{-\frac{1}{T}\mathbf{E}[F,S]}$$

and its low temperature limit as T tends to zero. Note that this distribution is just a vehicle to carry out statistical mechanics analysis and has nothing to do with the randomness in the random SAT formula, which, in physics, is called the *quenched disorder*.

Analytical techniques from statistical mechanics can be used to analyze the deep relations among the minimum of the energy function $\mathbf{E}[F, S]$, the Boltzmann distribution of the SAT system, and the probability distribution of the

random SAT. These analyses have revealed interesting structural properties of the space of the optimal solutions and help explain why problem instances are hard at phase transition. Among these is the notion of *backbone variables* [22, 15].

For each variable x_i, use m_i to denote the average value of the corresponding spin over all the optimal assignments. Note that $|m_i| = 1$ implies that the variable x_i is fully constrained, i.e., it has to be assigned to the same value in every optimal solution. In this case, the variable is called a backbone variable or frozen variable [13, 15].

For a random SAT, m_i is a random variable in $[-1, 1]$. Statistical mechanics analysis shows that the asymptotic behavior of the fraction of backbone variables is quite different at 2-SAT phase transition and 3-SAT phase transition. For random 2-SAT, it changes smoothly across the threshold, while for random 3-SAT, the fraction of backbone variables jumps discontinuously from zero to positive constant at the phase transition. That is to say, right above the clause-variable threshold, a constant fraction of the variables suddenly become fully constrained. There is also theoretical and empirical evidence showing that a close relation exists between the behavior of the backbone and backtracking-style search algorithms as well as random local search algorithms. See, for example, the work on the behavior of backbones in the $2 + p$-SAT problem where an instance of the problem consists of a mixture of 2-CNF clauses and 3-CNF clauses [22].

Analysis also reveals interesting characteristics about the structure of the space of the optimal solutions in the satisfiable-region as the clause-variables $r = \frac{m}{n}$ increases:

1. When r is well below the phase transition threshold, the optimal solutions form a single cluster and these solutions are all characterized by a common distribution;
2. When r is close to the phase transition threshold, the single cluster of optimal solutions break up into exponentially many smaller clusters. While the distances between solutions in different clusters remains constant, solutions in a single cluster become more and more similar to each other as r increases.

3 Threshold Phenomenon in NK Landscapes [23, 24]

The *NK landscape* is a fitness landscape model devised by Kauffman [25]. An appealing property of the NK landscape is that the "ruggedness" of the landscape can be tuned by changing some parameters. Over the years, the NK landscape model itself has been studied from the perspectives of statistics and computational complexity [26, 27]. In the study of genetic algorithms, NK landscape models have been widely used as a prototype and benchmark in the analysis of the performance of different genetic operators and the effects of different encoding methods on the performance of various genetic algorithms [28, 29].

In [24], we investigated the decision version of NK landscapes. One of the reasons that this NK landscape model attracts us is because the decision version

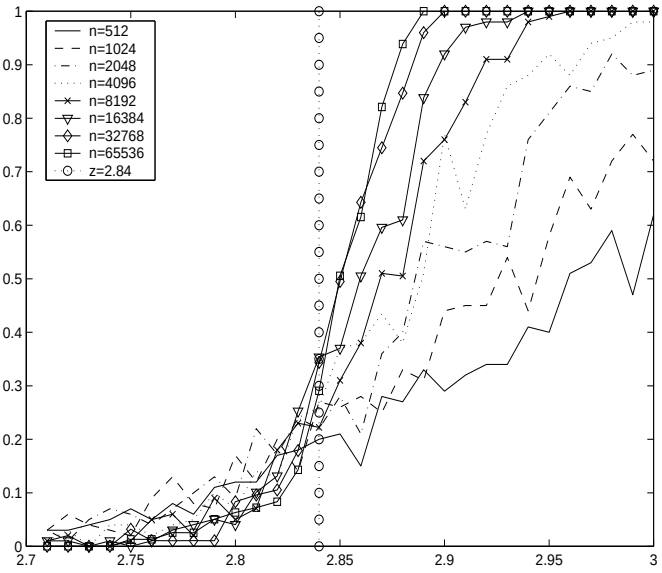

Fig. 1. Fractions of insoluble instances(Y-axis) as a function of z (X-axis).

defines a random SAT model that is quite different from those used in the study of the SAT phase transition. It is also our intention to investigate the typical complexity of the NK landscapes which have been widely used as benchmark problems in the GA community.

Consider an NK landscape $f(x) = \sum_{i=1}^{n} f_i(x_i, \pi(x_i))$ where

1. for each local fitness function $f_i(x_i, \pi(x_i))$, $\pi(x_i)$ is a set of $k-1$ variables randomly selected from $\{x_1, \cdots, x_n\} \setminus \{x_i\}$ and is called the neighborhood of x_i; and
2. each local fitness function takes its values in $\{0, 1\}$.

There are two ways to specify the distribution of the values of a local fitness functions, resulting in two random models for binary-valued NK landscapes:

1. The Uniform Probability Model $\overline{N}(n, k, p)$: In this model, the fitness value of the local fitness function $f_i(x_i, \pi(x_i))$ is determined as follows: For each assignment $y \in Dom(f_i) = \{0, 1\}^{k+1}$, let $f_i(y) = 0$ with the probability p and $f_i(y) = 1$ with the probability $1 - p$, and this is done for each possible assignment and each local fitness function independently; and
2. The Fixed Ratio Model $N(n, k, z)$: In this model, the parameter z takes on values from $[0, 2^{k+1}]$. If z is an integer, we specify the local fitness function $f_i(x_i, \pi(x_i))$ by randomly choosing without replacement z tuples of possible

assignments $Y = (y_1, \cdots, y_z)$ from $Dom(f_i) = \{0,1\}^{k+1}$, and defining the local fitness function as follows:

$$f_i(y) = \begin{cases} 0, \text{ if } \ y \in Y; \\ 1, \text{ else.} \end{cases}$$

For a non-integer $z = (1-\alpha)[z] + \alpha[z+1]$ where $[z]$ is the integer part of z, we choose randomly without replacement $[(1-\alpha)n]$ local fitness functions and determine their fitness values according to $N(n,k,[z])$. The rest of the local fitness functions are determined according to $N(n,k,[z]+1)$.

Our analytical analysis shows that the uniform probability model $\overline{N}(n,k,p)$ is asymptotically uninteresting:

Theorem 1. *[24] For any $p(n)$ such that $\lim_n p(n)n^{\frac{1}{2^{k+1}}}$ exists, k fixed, there is a polynomial time algorithm that successfully solves a random instance of $\overline{N}(n,k,p)$ with probability asymptotic to 1 as n tends to infinity.*

The key observations made in the analysis are that

1. if $p(n)$ decreases slowly, then there is asymptotically at least one local fitness function that always takes zero as its values;
2. if $p(n)$ decreases fast, then the problem defined by the model is asymptotically composed of a set of unrelated subproblems each of which involves a constant number of variables.

For the fixed ratio model $N(n,k,z)$, both theoretical and empirical analysis were carried out. First, for $z = 1$, the problem can be solved linearly—for each i, since f_i is zero only at one of the 2^{k+1} domain tuples, we can thus make an assignment to x_i to satisfy f_i without causing any conflicts. Second, it is shown that for $z \geq 2.83$, $N(n,2,z)$ asymptotically contains an unsatisfiable 2-SAT sub-problems:

Theorem 2. *If $z = 2 + \alpha > 2.837$, then $N(n,2,z)$ is asymptotically insoluble with probability 1.*

For $N(n,2,z)$ with $1 < z < 2.837$, experiments were carried out to investigate its typical complexity. We converted an instance of $N(n,2,z)$ to a SAT instance and then used R.J.Bayardo's SAT solver, called relsat without any parameter tuning. The solver is available at http://www.cs.ubc.ca/~hoos/SATLIB/. On a Linux machine of 450MHz, we were able to carry out experiments on problems with the number of variables up to 65 thousand (See Figure 1), and runtime statistics shows that time complexity is in the order of n^2.

In summary, for $N(n,2,z)$, there is a phase transition in the solution probability, but the problem is always easy without the corresponding peak in hardness as those observed in many other models of random SATs.

4 Treewidth of Fitness Landscapes and its Typical Size

In [26], it was shown that two types of NK landscape models, the adjacent neighborhood model and the random neighborhood model, share almost identical statistical characteristics such as the average number of local minimum and the average height of the local minimum. This is in contrast to the fact that the former model is a polynomial problem while the latter is NP-complete [26, 27]. The results presented in Section 3 of this paper indicate that NK landscapes with random neighborhoods are typically easy.

The work on the treewidth of fitness landscapes starts by asking the question: if NK landscapes are typically easy, what is the possible cause that makes them hard for algorithms like genetic algorithms? From the analysis, we believe that treewidth does provide a unique characteristic of fitness landscapes that cannot be captured by other existing measures, and thus may server as an appropriate GA-hardness measure.

A fitness function $f : X = \{0,1\}^n \to [0,\infty]$ is *additive* if it can be represented as a sum of lower dimensional functions

$$f(x) = \sum_{c \in \mathcal{C}} f_c(x), \quad x = \{x_1, \cdots, x_n\} \in X,$$

where $\mathcal{C}$ is a collection of subsets of $\{x_1, \cdots, x_n\}$. For each $c \in \mathcal{C}$, $f_c(x)$ only depends on the variables in c, and is thus called a *local fitness function*. The *order* k of an additive fitness function f is the size of the largest variable set in $\mathcal{C}$. Since we can always make the variable sets the same size by merging and/or adding dummy variables, we will assume throughout the rest of the paper that $\mathcal{C}$ consists of variable sets of size k. This gives us the *uniform additive fitness function* of order k.

Definition 1. *[30] The interaction graph of an additive fitness function is a graph $G(V,E)$ where the vertex set $V = \{x_1 \cdots, x_n\}$ corresponds to the set of variables in the additive fitness function and $(x_i, x_j) \in E$ if and only if x_i and x_j both appear in one of the local fitness functions.*

Definition 2. *Let f be an additive fitness function with an interaction graph G_f. The treewidth $w(f)$ of f is defined to be the treewidth of its interaction graph G_f*

The following results show that the treewidth indeed characterizes NK landscapes with adjacent neighborhoods and random neighborhoods.

Theorem 3. *[30] (1)Let $A(n,k)$ be the NK landscape model with adjacent neighborhoods. Then, the treewidth $w(A(n,k))$ is at most $2k$;*

(2) For an NK landscape with random neighborhoods $N(n,k)$. We have, for $k \geq 2$, there is a fixed constant $\delta > 0$ such that

$$\lim_n \Pr\{w(N(n,k)) \leq \delta n\} = 0.$$

The above analysis also applies to random additive fitness functions defined as follows.

Definition 3. *The pure random model* $\mathcal{F}(n, m, k)$,

$$\mathcal{F}(n, m, k) = \sum_{c \in \mathcal{C}} f_c(x) \tag{3}$$

is a random additive fitness function where $\mathcal{C}$ *consists of* m *subsets of variables selected randomly without replacement from* $\binom{n}{k}$ *possible size-*k *subsets of variables.*

Theorem 4. *[31] Let f(x) be an instance of the pure random model* $\mathcal{F}(n, m, k)$. *If* $\frac{m}{n} > \frac{\ln 2}{k \ln 3 - \ln(1+2^k)}$, *there is a fixed constant* $\delta > 0$ *such that*

$$\lim_n Pr\{w(f) \leq \delta n\} = 0.$$

Though, it remains to be analyzed regarding how the treewidth of a random additive fitness function changes when the ratio m/n increases, the following observation on the special case of $k = 2$ indicates that random additive fitness functions indeed have a phase transition in treewidth.

For $k = 2$, the interaction graph $G_f(n, m)$ of an random additive fitness function is just the standard random graph model. It follows that, for $m/n < 1/2$, G_f consists of a set of separated tree components and unicycle components [2], and thus has a treewidth 2. For $m/n > 1/2$, it is possible to show that the treewidth of the random graph is in the order n^ϵ with $\epsilon > 0$ a finite constant.

Several measures to characterize the complexity of fitness landscapes have been proposed before. Despite their success in characterizing various aspects of a fitness landscape, it is not clear if they are capable of distinguishing between landscape models that have similar statistical characterizations, but have totally different complexities.

The treewidth of a fitness landscape provides a new complexity measure that is capable of resolving these problems. For many NP-complete problems, polynomial-time algorithms exist if their underlying graphs have a bounded treewidth. This is also the case for the optimization problems on additive fitness functions, as we will illustrate in the following example:

Example 1. Consider an additive fitness function

$$\begin{aligned} f(x_1, x_2, x_3, x_4, x_5) = {} & f_a(x_1, x_2) + f_b(x_1, x_3) \\ & + f_c(x_1, x_5) + f_d(x_2, x_4) \\ & + f_e(x_2, x_5) + f_f(x_3, x_4) \end{aligned}$$

and the maximization problem

$$\max_{x_1 x_2 x_3 x_4 x_5} f(x_1, x_2, x_3, x_4, x_5).$$

Given an variable ordering $(x_1, x_4, x_5, x_2, x_3)$, an algorithm based on the idea of dynamic programming to solve this problem proceeds as follows:

$$
\begin{aligned}
&\max_{x_1x_2x_3x_4x_5} f(x_1, x_2, x_3, x_4, x_5) \\
&= \max_{x_1x_2x_4x_5} \{f_a(x_1, x_2) + f_c(x_1, x_5) + f_d(x_2, x_4) + f_e(x_2, x_5) + \\
&\qquad \underbrace{\max_{x_3}[f_b(x_1, x_3) + f_f(x_3, x_4)]}_{\lambda_3(x_1, x_4)}\} \\
&= \max_{x_1x_4x_5} \{f_c(x_1, x_5) + \lambda_3(x_1, x_4) + \underbrace{\max_{x_2}[f_a + f_d + f_e]}_{\lambda_2(x_1, x_4, x_5)}\} \\
&= \max_{x_1x_4}\{\lambda_3(x_1, x_4) + \underbrace{\max_{x_5}[f_c(x_1, x_5) + \lambda_2(x_1, x_4, x_5)]}_{\lambda_{25}(x_1, x_4)}\} \\
&= \max_{x_1}\{\max_{x_4}[\lambda_3(x_1, x_4) + \lambda_{25}(x_1, x_4)]\} \qquad (4)
\end{aligned}
$$

At each step, a variable is eliminated and a new function $\lambda_\cdot$ is created. The minimum, over all the possible variable orders, of the maximum number of arguments of these intermediate functions turns out to be the treewidth of the interaction graph of the additive fitness function f. And consequently, the cost of maintaining and computing these intermediate functions is exponential in the treewidth.

It is not obvious, however, that the measure of treewidth also applies to genetic algorithms, which of course is not a dynamic programming algorithm. In the next section, we will show that the treewidth complexity measure is indeed the right measure for a class of algorithms built on the basis of evolutionary computation.

5 Space Complexity of Estimation of Distribution Algorithms [31]

The *Estimation of Distribution Algorithms (EDAs)* are a class of sampling-based genetic algorithms that generate candidate solutions (individuals) by sampling some probability distributions on the solution space. The sampling probability distributions may be modelled as the product of independent marginal distributions, decomposable distributions obtained from the knowledge about the problem's interaction structures, or Bayesian networks constructed from existing samples of solutions [32–36].

In general, an EDA consists of four parts: (1) a search space X; (2) a fitness function $f : X \to [0, \infty]$; (3) a sampling probability distribution $P : X \to [0, 1]$; and (4) an algorithm to generate and update the sampling distribution P. Throughout this paper, we assume $X = \{0, 1\}^n$.

According to the internal representation of the probability distribution, EDAs can be categorized into three classes.

1. Independent distribution algorithm (IDA): a multivariate distribution of an independent product of one-dimensional distributions; IDA is also called the univariate marginal distribution algorithm (UMDA) [37].
2. Factorized distribution algorithm (FDA) [32]: a multivariate distribution represented as a factorized product of low-dimensional distributions; and
3. Bayesian Optimization Algorithm (BOA) [33]: a multivariate distribution represented as a Bayesian network. In fact, BOA is only one of the several classes of EDAs, such as the Estimation of Bayesian Network Algorithm (EBNA) [36], that learn and use Bayesian networks to represent the probability distributions.

Among the three types, IDA is the simplest in terms of both the space and computational complexity. Furthermore, the formula used to update the sampling distributions can be derived explicitly based on the original selection and mutation operators [38]. However, IDA is inefficient in, if not incapable of at all, capturing and utilizing the interactions among the variables of the fitness functions. This is the primary reason why recent research has focused on FDA and BOA that can represent distributions with richer interaction structures.

The use of distributions with richer correlation structures, however, comes with a cost. First, both FDA and BOA require more space to represent the distribution; and second, we need to determine the correct distribution that faithfully represents the interaction among the variables in the fitness functions. An incorrect representation might be much worse than the simple distribution of independent products of one-dimensional distributions. In this regard, we are in a situation quite similar to those discussed in the famous "no free lunch theorem" [39, 40].

The study of probabilistic reasoning in artificial intelligences is concerned with reasoning (inference) in probability models that are built around the idea of representing a probability distribution by graphical objects. The concepts of a dependency map and an independency map play important roles in the theory of graphical models. We present these concepts below in the context of interaction graphs of additive functions.

Definition 4. *Let f be an additive fitness function with the interaction graph $G_f(V, E)$ and let P be a probability distribution.*

- *G_f is said to be a dependency map (*or D-map*) of P if for all disjoint subsets of variables X, Y, Z, we have that X and Y are conditionally independent given Z only if Z separates X and Y in G_f.*
- *G_f is said to be an independency map (*or I-map*) of P if for all disjoint subsets of variables X, Y, Z, we have that Z separates X and Y in G_f only if X and Y are conditionally independent given Z;*
- *G_f is a perfect map of P if it is both a D-map and an I-map.*

It has been proved that for any graph G, there exists a probability distribution P such that G is a perfect map (see Section 3.2.3 of [41]).

A Bayesian network [41] is a directed acyclic graph $B = B(V, E)$ where V corresponds to the set of variables and a directed edge from x_i to x_j indicates that the variable x_j depends on the variable x_i.

In order to understand the Bayesian networks, the following concept of d-separation is essential.

Definition 5. *(Section 3.3.1, [41]) Let X, Y, and Z be three disjoint subsets of vertices in a directed acyclic graph D. Z is said to d-separate X from Y if along every undirected path between a vertex in X and a vertex in Y, there is a vertex w satisfying one of the following two conditions: (1) w has converging edges, i.e., edges on the path that meet head-to-head at w, and none of w or its descendants are in Z, or (2) w does not have converging edges and w is in Z.*

A directed acyclic graph $B = B(V, E)$ is called an I-map of a probability distribution P if for any disjoint subsets of variables X, Y, Z, the d-separation of X and Y by Z in B implies the conditional independence of X and Y given Z. A directed acyclic graph is a minimal I-map if no edge can be deleted without destroying the I-mapness.

The Factorized Distribution Algorithm (FDA)

FDA directly uses the interaction graph or an estimated interaction graph of the additive fitness function to model the sampling distribution [37]. For arbitrary fitness functions of which the exact interaction structure is usually unknown, an estimated interaction graph can also be used. Given an additive fitness function f and its interaction graph $G_f = G_f(V, E)$ with $V = (x_1, \cdots, x_n)$, FDA constructs a probability distribution $p(x)$ satisfying

1. G_f is an I-map of $p(x)$; and
2. $p(x)$ can be represented as a factorized product of the form

$$p(x) = \frac{\prod_{S \in \mathcal{S}} p_S(x)}{\prod_{S,T \in \mathcal{S}} p_{S \cap T}(x)} \tag{5}$$

where $\mathcal{S}$ is the collection of subsets of variables in a tree decomposition of the interaction graph G_f and $p_S(x)$ is the marginal distribution over the variables in $S \in \mathcal{S}$.

In the original definition of the FDA [32], the factorized product representation of $p(x)$ can be either approximated or exact. In an approximated factorized product, the set of subsets $\mathcal{S}$ is not necessarily a tree decomposition of the interaction graph. For the purpose of investigating the space complexity, we require that the factorization is always exact.

Let $f(x) = \sum_{c \in \mathcal{C}} f_c(x)$ be an additive fitness function with $\max_{c \in \mathcal{C}} |c| < k$, i.e., each local fitness function depends on at most k variables. If the collection of subsets of variables, $\mathcal{C}$, satisfies the *running intersection property*, or equivalently it forms a tree decomposition of the interaction graph, then an exact factorized

representation can be built on $\mathcal{C}$ [32], and only $O(2^k)$ space is required to represent it. However, as has also been mentioned in [32], such a class of additive fitness functions is very limited. Otherwise, to get an exact factorized representation, one has to find a tree decomposition of the interaction graph. And the resulting exact factorization will have a space complexity exponential in the width of the tree decomposition.

Bayesian Optimization Algorithm(BOA)

BOA models the sampling distribution by a Bayesian network [35, 33].

Definition 6. *Let f be an additive fitness function with the interaction graph $G_f(V, E)$ and let P_f be the probability distribution such that G_f is a perfect-map of P_f. A directed acyclic graph B is called a Bayesian network for f if it is a minimal I-map of P_f.*

The following theorem shows how to construct a Bayesian network under a given variable ordering $\pi = (x_1, \cdots, x_n)$. Let $U_i(\pi) = (x_1, \cdots, x_{i-1})$. A *Markov boundary* [41] $B_i(\pi)$ of x_i with respect to $U_i(\pi)$ is a minimal subset such that (1) $B_i(\pi) \subset U_i(\pi)$; and (2) $B_i(\pi)$ separates x_i and $U_i(\pi) \setminus B_i(\pi)$ in G_f.

Theorem 5. *(Section 3.3.1, [41]) Let $G_f = G_f(V, E)$ be an interaction graph of an additive fitness function $f(x)$. For each $i \geq 1$, let $B_i(\pi)$ be a Markov boundary of x_i with respect to $U_i(\pi)$. Then the directed acyclic graph specified by the parent sets*

$$Pa(x_i) = B_i(\pi), \quad i \geq 1, \tag{6}$$

is a Bayesian network of f. Furthermore, if the probability distribution P_f is strictly positive, then the Bayesian network given above is unique under the given order.

From Theorem 5, we can see that for a given ordering of variables, there is a unique Bayesian network that captures the conditional independence depicted in the interaction graph of the fitness function. To represent this Bayesian network, we need a table for each variable x_i to store the conditional probabilities $P(x_i|Pa(x_i))$. It follows that the space complexity to represent this Bayesian network is $\Omega(\max_i |B_i(\pi)|)$.

Similar to the case of the treewidth in FDA, there are many different orderings of the variables, each of which gives us a different value of $\max_i(|B_i(\pi)|)$. Since a Bayesian network is a minimal I-map, we may define the space complexity of BOAs using $\max_i |B_i(\pi)|$.

It can be proved that $\min_\pi \max_{1 \leq i \leq n} |B_i(\pi)|$ is also equal to the treewidth of the interaction graph of a give additive fitness function [31].

The above analysis indicates that the treewidth of an additive fitness function indeed characterizes the space complexity of both FDA and BOA. From the results on the typical-case size of the treewidth of random additive fitness functions in Section 4, we see that both FDA and BOA have a space complexity exponential in the number of variables even for random additive functions that are still sparse.

6 Concluding Remarks

The analysis of phase transitions in combinatorial search and optimization problems has been proven to be quite fruitful in helping us understand when and how an algorithm works in the problem spaces. In this paper, we have reviewed some of existing work on the analysis of fitness landscapes, including the phase transition of NK landscapes, the treewidth measure of fitness landscapes, and the space complexity of EDAs.

References

1. Erdos, P., Renyi, A.: On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci. **5** (1960) 17–61
2. Bollobas, B.: Random Graphs. Cambridge University Press (2001)
3. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the *really* hard problems are. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1991) 331–337
4. Crawford, J., Auton, L.: Experimental results on the crossover point in satisfiability problems. In: Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93), AAAI Press / The MIT Press (1993) 21–27
5. Kirkpatrick, S., Selman, B.: Critical behavior in the satisfiability of random boolean expressions. Science **264** (1994) 1297–1301
6. Hayes, B.: Can't get no satisfaction. American Scientist **85** (1997) 108–112
7. Hayes, B.: The easiest hard problem. American Scientist **90** (2002) 113–117
8. Hayes, B.: On the threshold. American Scientist **91** (2003) 12–17
9. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: Determining computational complexity from characteristic 'phase transitions'. Nature **400** (1999)
10. Mitchell, D., Selman, B., Levesque, H.: Hard and easy distributions of sat problems. In: Proceedings of the 10th Natl. Conf on Artificial Intelligence, AAAI Press (1992) 459–465
11. Freeman, J.W.: Hard random 3-sat problems and the davis-putman procedure. Artificial Intelligence **81** (1996) 183–198
12. Cook, S., Mitchell, D.G.: Finding hard instances of the satisfiability problem: A survey. In Du, Gu, Pardalos, eds.: Satisfiability Problem: Theory and Applications. Volume 35 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society (1997)
13. Culberson, J., Gent, I.: Well out of reach: Why hard problems are hard. Technical Report APES-13-1999, APES Research Report (1999)
14. Achlioptas, D., Gomes, C., Kautz, H., Selman, B.: Generating satisfiable problem instances. In: Seventeenth National Conference on Artificial Intelligence (AAAI'00). (2000) 256–261
15. Martin, O., Monasson, R., Zecchina, R.: Statistical mechanics methods and phase transition in optimization problems. Theoretical Computer Science **265** (2001) 3–67
16. Alon, N., Spencer, J., Erdös, P.: The Probabilistic Method. John Wiley & Sons (1992)

17. Gu, J., Purdom, P.W., Franco, J., Wah, B.W.: Algorithms for satisfiability (sat) problem: A survey. In: Discrete Mathematics and Theoretical Computer Science: Satisfiability (SAT) Problem. American Mathematical Society (1997) 19–152
18. Dubois, O., Boufkhad, Y.: A general upper bounds for the satisfiability threshold of random r-SAT formelae. J. of Algorithms **24** (1997) 395–420
19. Achlioptas, D.: A survey of lower bounds for random 3-sat via differential equations. Theoretical Computer Science **265** (2001) 159–185
20. Achlioptas, D., Peres, Y.: The random k-SAT threshold is $2^k log 2 - o(k)$. In: Proceedings of the 35th Annual Symposium on Theory of Computing, STOC03. (2003) 223–231
21. Friedgut, E.: Sharp thresholds of graph properties, and the k-SAT problem. J. Amer. Math. Soc. **12** (1999) 1017–1054
22. Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B., Troyansky, L.: 2+p-sat: Relation of typical-case complexity to the nature of the phase transition. Random Structure and Algorithms **15** (1999) 414
23. Gao, Y., Culberson, J.: The phase transition in nk landscapes is easy. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001). (2001)
24. Gao, Y., Culberson, J.: An analysis of phase transition in NK landscapes. Journal of Artificial Intelligence Research **17** (2002) 309–332
25. Kauffman, S.: The Origins of Order: Self-organization and Selection in Evolution. Oxford University Press, Inc. (1993)
26. Weinberger, E.D.: NP completeness of Kauffman's NK model, a tunable rugged fitness landscape. Technical Report Working Papers 96-02-003, Santa Fe Institute, Santa Fe (1996)
27. Wright, A.H., Thompson, R.K., Zhang, J.: The computational complexity of N-K fitness functions. IEEE Transaction on Evolutionary Computation **4** (2000) 373–379
28. Hordijk, W.: A measure of landscapes. Evolutionary Computation **4** (1997) 335–360
29. Jones, T.C.: Evolutionary Algorithms, Fitness Landscapes and Search. PhD thesis, University of New Mexico, Albuquerque, NM (1995)
30. Gao, Y., Culberson, J.: On the treewidth of NK landscapes. In: Genetic and Evolutionary Computation Conference (GECCO-03). LNCS 2723, Springer-Verlag (2003) 848–954
31. Gao, Y., Culberson, J.: Space complexity of estimation of distribution algoritms. Submitted (2003)
32. Mühlenbein, H., Mahnig, T., Rodriguez, A.: Schemata, distributions and graphical models in evolutionary optimization. Journal of Heuristics **5** (1999) 215–247
33. Pelikan, M., Goldberg, D.E., Lobo, F.: A survey of optimization by building and using probabilistic models. Technical Report 99018, IlliGAL, University of Illinois (1999)
34. Leung, Y., Gao, Y., Zhang, W.: A genetic-based method for training fuzzy systems. In: Proc. of the 10th IEEE International Conference on Fuzzy Systems. Volume 1., IEEE (2001) 123–126
35. Larrañaga, P., Etxeberria, R., Lozano, J.A., Pena, J.M.: Combinatorial optimization by learning and simulation of bayesian networks. In Boutilier, C., Goldszmidt, M., eds.: Proceedings of the Sixteenth Conference on Uncertainty in Articial Intelligence, Morgan Kaufmann (2002) 343–352
36. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publisher (2001)

37. Mühlenbein, H., Mahnig, T.: Convergence theory and applications of the factorized distribution algorithm. Journal of Computing and Information Technology **7** (1999) 19–32
38. Leung, Y., Gao, Y., Xu, Z.: Degree of population diversity - a perspective on premature convergence in genetic algorithms and its markov chain analysis. IEEE Transactions on Neural Networks **8** (1997) 1165–1176
39. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **4** (1997) 67–82
40. Culberson, J.: On the futility of blind search: An algorithmic view of "no free lunch". Evolutionary Computation **6** (1998) 109–128
41. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausiable Inference. Morgan Kauffmann (1988)