Space Complexity of Estimation of Distribution Algorithms

Yong Gao ygao@cs.ualberta.ca
Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada, T6G 2E8

Joseph Culberson joe@cs.ualberta.ca
Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada, T6G 2E8

Abstract

In this paper, we investigate the space complexity of the Estimation of Distribution Algorithms (EDAs), a class of sampling-based variants of the genetic algorithm. By analyzing the nature of EDAs, we identify criteria that characterize the space complexity of two typical implementation schemes of EDAs, the factorized distribution algorithm and Bayesian network-based algorithms. Using random additive functions as the prototype, we prove that the space complexity of the factorized distribution algorithm and Bayesian network-based algorithms is exponential in the problem size even if the optimization problem has a very sparse interaction structure.

Keywords

Estimation of distribution algorithms, space complexity, additive fitness functions, graphical models and bayesian networks, treewidth.

1 Introduction

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 (Mühlenbein et al., 1999; Pelikan et al., 1999; Leung et al., 2001; Larrañaga et al., 2000; Larrañaga and Lozano, 2001).

Unlike many other stochastic local search algorithms, such as the standard genetic algorithm (Goldberg, 1989) and simulated annealing (Kirkpatrick et al., 1983), where the sampling distributions are implicitly defined by the random operators, the sampling distribution in EDAs is an explicit component of the algorithm. As a consequence, an EDA's success heavily depends on the representation and estimation of the probability distributions to capture the interaction information of variables of an optimization problem. An investigation of the complexity issues related to the representation of sampling distributions in EDAs will provide much insight into such problems as how to design efficient EDAs and what the limitations of the algorithm are.

In this paper, we report our research into the complexity issues in two typical implementation schemes of EDAs: the *factorized distribution algorithm* and the *Bayesian*

network-based algorithm. Using random additive functions as the prototype, we prove that both of the algorithms have a space complexity that is exponential in the problem size even if the optimization problem has a very sparse interaction structure.

The rest of the paper is organized as follows. In Section 2, we introduce the estimation of distribution algorithm and its typical implementations. In Section 3, we introduce additive fitness functions and their random models. We propose some graph-theory-based measures to capture the degree of interaction in an additive fitness function. We then analyze the graphical models used in EDAs to represent the sampling distributions, and identify criteria to characterize the space complexity of EDAs. In Section 4, we prove our results on the space complexity of two typical implementations of EDAs. Section 5 is the conclusion.

2 Estimation of Distribution Algorithms

A genetic algorithm (GA) is a population-based search algorithm that evolves a population of candidate solutions using so-called genetic operators such as selection, mutation, and crossover. The estimation of distribution algorithms (EDAs) are variants of genetic algorithms. Instead of maintaining a candidate population and using genetic operators, EDAs generate feasible solutions by iteratively sampling a probability distribution on the solution space and updating the probability distribution based on the information gathered from the candidate solutions.

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 over X; and (4) an algorithm to generate and update the sampling distribution P. In the rest of this paper, we will 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) (Mühlenbein and Mahnig, 1999).

2. Factorized distribution algorithm (FDA) (Mühlenbein et al., 1999): a multivariate distribution represented as a factorized product of low-dimensional distributions; and

3. Bayesian network-based algorithm (BNA): a multivariate distribution represented as a Bayesian network. In fact, this includes the Bayesian Optimization Algorithm (BOA) (Pelikan et al., 1999) and several classes of EDAs, such as the Estimation of Bayesian Network Algorithm (EBNA) (Etxeberria and Larrañaga, 1999; Larrañaga and Lozano, 2001), 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 mutation and selection operators (Leung et al., 1997). 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 BNA 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 BNA 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" (Wolpert and Macready, 1997; Culberson, 1998).

3 Additive Functions, Interaction Graphs, and Graphical Models of Sampling Distributions

In this paper, we use random models of additive fitness functions as our prototype and characterize the space complexity of EDAs by some graph-theory-based measures. In subsection 3.1, we introduce the random models for additive fitness functions. In subsection 3.2, we discuss some graph-theory-based measures that can be used to capture the variable interaction in an additive fitness function. These measures are based on the concept of *treewidth*. In subsection 3.3, we analyze the probability models used by EDAs to represent the sampling distributions, and identify natural criteria to characterize the space complexity of EDAs.

3.1 Random Models for Additive Functions

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

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

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. Many optimization problems studied over the years can be modelled as a special type of additive fitness functions. Examples include NK landscapes (Kauffman, 1989; Gao and Culberson, 2002; Kallel et al., 2001), the spin-glass model (Martin et al., 2001; Mezard and Parisi, 2003), deceptive functions (Goldberg et al., 1993), constraint satisfaction problems (CSPs) (Braunstein et al., 2003), etc.

Random models of search and optimization problems have been extensively used in the study of the typical behavior of search algorithms and in the generation of benchmark problems for performance evaluation (Cook and Mitchell, 1997; Gent et al., 1999; Martin et al., 2001). To define a random model for additive fitness functions, we need to describe how the variable set of each local fitness function is chosen and how the values of a local fitness function are assigned. Formally, we use

$$\mathcal{F}(n,k) = \sum_{c \in \mathcal{C}} f_c(x) \tag{1}$$

to denote the random model where

1. $\mathcal{C}$ consists of a collection of subsets of variables selected randomly according to a probability distribution from all the $\binom{n}{k}$ possible subsets of variables; and

2. the fitness values of each local fitness function are assigned randomly and independently according to a distribution on $[0, 1]$.

In this paper, we make no specific assumption about the distributions of the fitness value. For the selection of the subsets of variables, we focus on the following random model:

Definition 3.1. *The pure random model $\mathcal{F}(n, m, k)$,*

$$\mathcal{F}(n, m, k) = \sum_{c \in \mathcal{C}} f_c(x) \tag{2}$$

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.

Another model that provides a restricted scope of interaction is also of interest for comparison purposes.

Definition 3.2. *The neighborhood model $\mathcal{N}(n, k)$ is defined as*

$$\mathcal{N}(n, k) = \sum_{j=1}^{n} f_j(x_j, \Pi(x_j)), \tag{3}$$

where $\Pi(x_j) \subset \{x_i, 1 \leq i \leq n, i \neq j\}$ is the neighborhood of x_j with the size $|\Pi(x_j)| = k - 1$.

If the neighborhood $\Pi(x_j)$ is selected randomly without replacement from $\{x_i, 1 \leq i \leq n, i \neq j\}$, we get the model of NK landscapes with random neighborhoods. If $\Pi(x_j)$ is defined to be x_j's nearest $k - 1$ neighboring variables, i.e.,

$$\Pi(x_i) = \{x_{((n+i-\frac{k}{2}) \mod n)}, \ldots, x_{((n+i+\frac{k}{2}) \mod n)}\},$$

we get the model of NK landscapes with adjacent neighborhoods. Details about these NK landscape models can be found in (Gao and Culberson, 2002; Gao and Culberson, 2003).

3.2 Interaction Graphs of Additive Functions

The interaction among different variables in a fitness function plays an important role in the study of the typical complexity of optimization problems. In additive functions, the interactions are encoded in the internal structures of the local fitness functions and the relations among the collection of subsets of variables of local fitness functions. These interactions can be represented as an interaction graph.

Definition 3.3. *The interaction graph of an additive fitness function*

$$f(x) = \sum_{c \in \mathcal{C}} f_c(x), \quad x = \{x_1, \ldots, x_n\}, \tag{4}$$

is a graph $G_f = G_f(V, E)$ where the vertex set $V = \{x_1, \ldots, x_n\}$ corresponds to the set of variables, and $(x_i, x_j) \in E$ if and only if there is a subset $c \in \mathcal{C}$ such that $x_i \in c$ and $x_j \in c$.

The interaction graph of an additive fitness function captures all the interactions among the variables. A knowledge about these interactions is critical in understanding the complexity and designing appropriate algorithms to solve the problems. For example, if the interaction graph is a tree, then a linear time algorithm readily exists to solve the problem. As yet another example, if the interaction graph can be decomposed

into several connected components, then a viable approach is to first solve the subproblems represented by the connected components and then combine the obtained partial solutions together.

The concepts of the treewidth and the tree decomposition of a graph generalize the concept of a tree and characterize the degree to which a graph has a tree-like structure (Kloks, 1994). These concepts provide a viable way to characterize the degree of interaction in an optimization problem. We discuss these concepts briefly below and refer interested readers to the works (Bodlaender, 1997; Kloks, 1994; Bouchitt and Todinca, 2001) for more details. The treewidth of a graph can be defined in terms of the l-tree.

Definition 3.4. *(Kloks, 1994) l-Trees are defined recursively as follows:*

1. *A clique with l+1 vertices is an l-tree;*
2. *Given an l-tree T_n with n vertices, an l-tree with $n+1$ vertices is constructed by adding to T_n a new vertex which is made adjacent to an l-clique of T_n and non-adjacent to the rest of the vertices.*

Definition 3.5. *(Kloks, 1994) A graph is called a partial l-tree if it is a subgraph of an l-tree. The treewidth of a graph G is the minimum value l for which G is a partial l-tree.*

The treewidth of a graph has an equivalent definition based on the concept of tree decomposition.

Definition 3.6. *(Kloks, 1994) A tree decomposition of a graph $G = (V, E)$ is a pair $\mathcal{D} = (\mathcal{S}, \mathcal{T})$ where $\mathcal{S} = \{S_i, i \in I\}$ is a collection of subsets of vertices of G and $\mathcal{T} = (I, F)$ is a tree with one node for each element in $\mathcal{S}$, such that*

1. *$\bigcup_{i \in I} S_i = V$,*
2. *for all the edges $(v, w) \in E$ there exists a subset $S_i \in \mathcal{S}$ such that both v and w are in S_i, and*
3. *for each vertex $v \in V$, the set of nodes $\{i \in I;\ \ v \in S_i\}$ forms a subtree of $\mathcal{T}$.*

The width of the tree decomposition $\mathcal{D} = (\mathcal{S}, \mathcal{T})$ is $\max_{i \in I}(|S_i| - 1)$. And the treewidth of a graph is the minimum width over all tree decompositions of the graph.

The treewidth has yet another equivalent definition based on the minimum width of a graph and the notion of vertex elimination in a graph. It is also called the *induced width* in the literature (See, for example, (Dechter, 1999)).

Definition 3.7. *Let $G = (V, E)$ be a graph and $\pi = (x_1, \cdots, x_n)$ be an ordering of the vertices.*

1. *The width $w(x, \pi)$ of a vertex x under the ordering π is the number of its preceding neighbors. The width $w(\pi)$ of the ordering π is the maximum width of all the vertices under the ordering, i.e.,*
$$w(\pi) = \max_{x \in V} w(x, \pi).$$
2. *The induced graph $G(\pi)$ of G under the ordering π is obtained by processing the vertices recursively according to the ordering π from x_n to x_1. At each step i, all the neighbors of x_i that precede x_i according to π are made adjacent and then x_i is marked as processed. This process is called the vertex elimination.*

3. *The* **induced width** $w^*(G, \pi)$ *of* G *under the ordering* π *is the width of the induced graph* $G(\pi)$ *of* G*. The induced width* $w^*(G)$ *of* G *is the minimum induced width over all the vertex orderings.*

Given a graph G and a vertex ordering π, one can obtain a tree decomposition by (1) forming the induced graph $G(\pi)$; (2) identifying all the (maximum) cliques of $G(\pi)$; and (3) building a tree of this set of cliques in linear time that satisfies all the requirements of a tree decomposition.

In many applications, it is desirable to find a tree decomposition with a minimum width. This problem is NP-complete and has been an interesting research topic in graph theory and artificial intelligence. See, for example, (Bodlaender, 1997; Kloks, 1994; Bouchitt and Todinca, 2001; Becker and Geiger, 1996) and the references therein for more details.

Example 3.1. *Figure 1 shows a graph G(V, E) with 5 vertices* (A, B, C, D, E) *and two of its tree decompositions. The tree decomposition to the right of the graph has a width 2 and the one below the graph has a width 3. The treewidth of the graph is 2.*

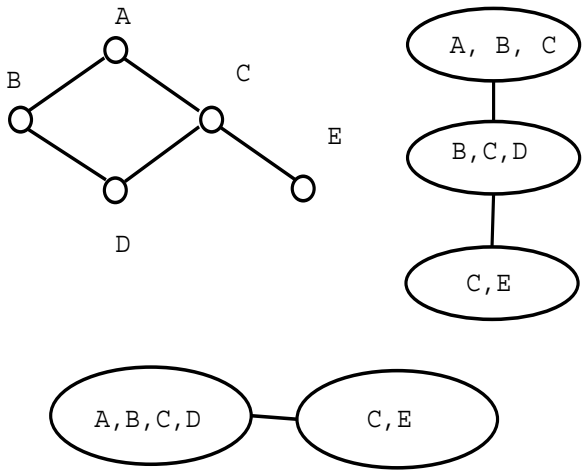

Figure 1: Examples of Tree Decompositions.

Based on the interaction graph, we can measure the degree of the variable interaction in an additive fitness function using the treewidth.

Definition 3.8. *Let* $f(x_1, \cdots, x_n)$ *be an additive fitness function with the interaction graph* G_f*.*

1. *The treewidth* $\omega(f)$ *of* f *is defined to be the treewidth of* G_f*.*

2. *Given an ordering* π *of the variables, the width* $w(f)$ *and the induced width* $w^*(f)$ *are defined respectively to be the width and induced width of* G_f *under* π*.*

3.3 Graphical Models for Sampling Distributions of EDAs

In stochastic search algorithms, new candidates are usually generated according to a sampling distribution. For some algorithms such as standard genetic algorithms and

simulated annealing, the sampling distribution is implicitly defined via random operators, and thus the representation is not an issue. For algorithms like EDAs where the sampling distribution is an explicit component of the algorithm, the representation of the sampling distribution is one of the most critical issues that one has to face. Without any assumptions and knowledge about the optimization problems to be modelled, the only way to represent a distribution on $X = \{0,1\}^n$ is to use a probability table of 2^n cells. This exponential representation is, of course, not a favorable one.

The independent distribution algorithm (IDA) is the simplest in terms of both the space of representation and computational cost. IDA models the sampling distribution as a product of n one-dimensional marginal distributions. However, IDA is inefficient in, if not incapable of, 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 BNA that can represent distributions with richer interaction structures.

In order to use sampling distributions to capture the knowledge about the interaction and guide the search, the representation of the sampling distribution has to be *effective* and *efficient*, which we may summarize as the following two basic requirements:

1. *(Effectiveness)* The higher the fitness $f(x)$, the higher the probability $p(x)$. This can be formulated as
$$p(x) \propto f(x). \tag{5}$$
2. *(Space Efficiency)* The representation of $p(x)$ should be as efficient as possible. A straightforward representation of $p(x)$ is a probability table of 2^n cells. This exponential space requirement, however, is not feasible even for a fitness function of moderate dimensions.

To fulfill these requirements, the modelling of the sampling distributions has to utilize the interactions depicted in the interaction graph of the fitness functions. This can be accomplished by using the *graphical model* developed in the study of probabilistic reasoning and multivariate statistics (Pearl, 1988; Whittaker, 1989).

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 3.9. *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 (Pearl, 1988)).

In the following, we will discuss how EDAs model the sampling distributions to capture the variable interactions, paying particular attention to the space complexity.

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 (Mühlenbein and Mahnig, 1999). 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

$$p(x) = \frac{\prod_{S \in \mathcal{S}} p_S(x)}{\prod_{S,T \in \mathcal{S}} p_{S \cap T}(x)} \tag{6}$$

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

In the original definition of the FDA (Mühlenbein et al., 1999), the factorized product representation of $p(x)$ can be either approximated or exact. In an approximated factorized product, the collection of subsets $\mathcal{S}$ does not necessarily form 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 $\mathcal{C}$ with a space requirement of $O(2^k)$ (Mühlenbein et al., 1999). However, as has also been mentioned in (Mühlenbein et al., 1999), 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. Our analysis in this paper will show that for a random additive fitness function, the space complexity of an exact factorization is exponential in the number of the variables even if the interaction structure of the function is sparse.

Below are a few examples to illustrate the concepts of tree decomposition and the factorized representation of a probability distribution.

Example 3.2. *Consider three additive fitness functions defined on the variables* $x = \{x_1, x_2, x_3, x_4\}$*:*

$$f_A(x) = f_1(x_1, x_2) + f_2(x_2, x_3) + f_3(x_3, x_4) \tag{7}$$
$$f_B(x) = f_1(x_1, x_2) + f_2(x_2, x_3) + f_3(x_3, x_4) + f_4(x_4, x_1) \tag{8}$$
$$f_C(x) = f_1(x_1, x_2, x_3) + f_2(x_1, x_2, x_4) + f_3(x_2, x_3, x_4) \tag{9}$$

(1) The interaction graph G of f_A *is simply a path over four vertices*

$$x_1 - x_2 - x_3 - x_4.$$

G has a treewidth of 1 and an optimal tree decomposition of G is $\mathcal{T} = \{(x_1, x_2), (x_2, x_3), (x_3, x_4)\}$. A probability distribution p(x) defined on G can thus be represented as a factorized product of the form

$$p(x) = \frac{p(x_1, x_2)p(x_2, x_3)p(x_3, x_4)}{p(x_2)p(x_3)}$$

(2) The interaction graph of f_B is a cycle

$$x_1 - x_2 - x_3 - x_4 - x_1$$

and has a treewidth 2. A tree decomposition is

$$\mathcal{T} = \{(x_1, x_2, x_4), (x_2, x_3, x_4)\}$$

and a probability distribution defined on the interaction graph can be represented as a factorized product of the form

$$p(x) = \frac{p(x_1, x_2, x_4)p(x_2, x_3, x_4)}{p(x_2, x_4)}$$

(3) The interaction graph is a 4-clique. Its treewidth is three with the only possible tree decomposition $\mathcal{T} = \{(x_1, x_2, x_3, x_4)\}$, and consequently, the only possible factorized representation of a probability distribution $p(x)$ is

$$p(x) = p(x_1, x_2, x_3, x_4).$$

For the function $f_A(x)$, the factorized representation of $p(x)$ truthfully reflects the conditional independence in the interaction graph. For the function $f_B(x)$, the best factorized distribution one can have only partially reflects the conditional independence in the interaction graph. Also, for $f_B(x)$, we need to use three-dimensional distributions, while for $f_A(x)$, two-dimensional distributions are enough. For $f_C(x)$, there is no conditional independence available and one is forced to represent a distribution by enumerating the probability of each possible configuration of the variables.

The space complexity of FDAs depends exponentially on the width of the tree decomposition used in the factorized representation of $p(x)$. Recall that the width of a tree decomposition is defined as $\max_{S \in \mathcal{S}}(|S| - 1)$ where $\mathcal{S}$ is the collection of subsets of variables in the tree decomposition. This is because FDAs need $\Omega(2^{|S|})$ space to represent each factorized component $p_S(x)$. For an interaction graph, there are many different tree decompositions with different width, and the *treewidth* of the interaction graph is defined to be the minimum of the width of different tree decompositions. It follows that the space complexity of an FDA is exponential in the treewidth of the interaction graph. Therefore, we can have

Definition 3.10. *The space complexity F_f of an FDA for an additive function f is defined to be*

$$F_f = 2^{\omega(f)} \tag{10}$$

where $\omega(f)$ is the treewidth of f as in Definition 3.8.

Bayesian network-based algorithm (BNA)

BNA models the sampling distribution by a Bayesian network (Larrañaga et al., 2000; Pelikan et al., 1999). A Bayesian network 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 (Pearl, 1988). In order to identify an appropriate complexity measure for BNAs, we need to formalize the notion of Bayesian networks that can be used to capture the variable interaction in an additive fitness function.

Let us start with the concept of *d-separation* in a directed graph.

Definition 3.11. *(Section 3.3.1, (Pearl, 1988)) 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 the graph $B(V, E)$ 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.

Definition 3.12. *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 G_f be the interaction graph of f and let $U_i(\pi) = (x_1, \cdots, x_{i-1})$. A *Markov boundary* $B_i(\pi)$ 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 (Pearl, 1988).

Theorem 3.1. *(Section 3.3.1, (Pearl, 1988)) 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*

$$Pa(x_i) = B_i(\pi), \quad i \geq 1, \tag{11}$$

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 π.

From Theorem 3.1, 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 BNAs using $\max_i |B_i(\pi)|$.

Definition 3.13. *The space complexity B_f of a BNA for a given additive function f is defined as*

$$B_f = 2^{\overline{w}(f)} \tag{12}$$

where $\overline{w}(f) = \min_{\pi} \max_{1 \leq i \leq n} |B_i(\pi)|$ and $B_i(\pi)$ is the Markov boundary of x_i under the ordering π.

It should be mentioned that the above definition of space complexity of BNAs is based on the assumption that Bayesian networks in a BNA are constructed to reflect the exact conditional independence in a fitness function, and we do not consider the situations where BNAs use some machine learning procedures to construct an approximating Bayesian network with less space requirement.

Example 3.3. *Consider the additive fitness function*

$$f(x) = f_a(x_1, x_2) + f_b(x_1, x_3) + f_c(x_2, x_4) + f_d(x_3, x_4).$$

Its interaction graph G_f is a cycle $x_1{-}x_2{-}x_4{-}x_3{-}x_1$. According to Theorem 3.1, the Bayesian network for the variable ordering (x_1, x_2, x_3, x_4) is shown in Figure 2(a). The Bayesian network in Figure 2(b) is not the true Bayesian network of f because it encodes the conditional independency of x_3 and x_2 given x_1 alone, which is not specified in the interaction graph of f. Of course, it is highly possible that BNAs with some machine-learning mechanisms may construct the Bayesian network of Figure 2(b).

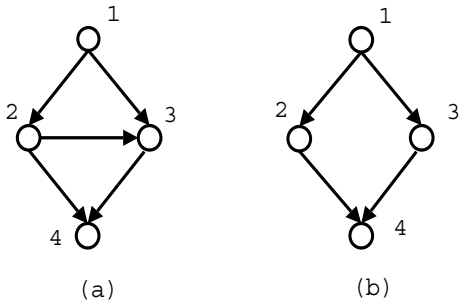

Figure 2: Examples of Bayesian networks of an additive fitness function.

In the next section, we will investigate the space complexity of FDAs and BNAs by evaluating the size of the treewidth and minimum width of random additive functions. We close this section by providing a discussion to justify why EDAs have to truthfully encode the interaction structures in the interaction graph. For a given additive fitness function $f : X = \{0, 1\}^n \to [0, 1]$, consider the following Boltzman-like distribution

$$p_f(x) = \frac{1}{Z} \exp(f(x)), \quad Z = \int_X \exp(f(x))dx. \tag{13}$$

Distributions similar to $p_f(x)$ are the building blocks in many studies of search and optimization problems. Concrete examples include the statistical physics approach, the simulated annealing algorithm, and some selection schemes in genetic algorithms (Mezard and Parisi, 2003; Goldberg, 1989; Kirkpatrick et al., 1983). It turns out that this distribution exactly encodes the interaction structures in the interaction graph.

Theorem 3.2. *Let $f(x)$ be an additive fitness function with the associated interaction graph $G_f = G_f(V, E)$. Then, variables x_i and x_j are not adjacent in G_f if and only if the two corresponding variables in the distribution $p_f(x)$ are conditionally independent given the rest of the variables.*

Proof. Two variables x_i and x_j are not adjacent in G_f if and only if they do not appear in a local fitness function at the same time. Let

$$f(x) = \sum_{c \in \mathcal{C}} f_c(x)$$

and let $\mathcal{C}_1$ be the set of variable sets that contain the variable x_j. Then, x_i and x_j are not adjacent if and only if x_i does not appear in any local fitness functions indexed by $\mathcal{C}_1$. The result follows because two components x_i and x_j are conditionally independent if and only if $p_f(x)$ can be written as $g(x_i, y) * h(x_j, y)$ where $y = \{x_1, \cdots, x_n\} \backslash \{x_i, x_j\}$. □

In (Mühlenbein et al., 1999), a factorization theorem for Boltzmann distributions is proved [1]. The above Theorem 3.2 is slightly different and serves a different purpose—to demonstrate why it is important to capture the independency depicted in the interaction graph.

4 Space Complexity of EDAs

In this section, we study the space complexity of three variants of EDAs using random additive functions as our prototypical model. As we have discussed in Section 3.3, the space complexity of FDAs and BNAs are characterized by some graph-theory-based measures. We will focus on how these measures are related to the degree of interaction of the additive fitness functions. Let us start with the simplest case of the independent distribution algorithm (IDA).

Theorem 4.1. *For any fitness function, the independent distribution algorithm (IDA) has a space complexity $O(n)$.*

Proof. This is true because for IDAs, one only needs to represent n one-dimensional marginal distributions. □

For the NK landscape model with adjacent neighborhoods, we have the following results.

Theorem 4.2. *Let f(x) be an instance of the neighborhood model with adjacent neighborhood $\mathcal{N}(n, k)$. Then, the space complexity of FDA is $F_f = 2^{\Theta(k)}$.*

Proof. Since the interaction graph f has cliques of size $k + 1$, its treewidth should be no less than k. We prove the theorem by constructing a tree decomposition with a treewidth $2k$. Let $V = \{x_1, \cdots, x_n\}$ be the set of vertices, and let $V_0 = \{x_1, \cdots, x_k\}$. We construct $S = \{X_i, i \geq 1\}$, a collection of subsets of the variables, as follows:

$$\begin{aligned}
X_1 &= \{x_1, \cdots, x_{k+1}\} \cup V_0, \\
X_2 &= \{x_2, \cdots, x_{k+2}\} \cup V_0, \\
&\cdots \\
X_{N-k} &= \{x_{N-k}, \cdots, x_N\} \cup V_0, \\
X_{N-k+1} &= \{x_{N-k+1}, \cdots, x_N, x_1\} \cup V_0, \\
X_{N-k+2} &= \{x_{N-k+2}, \cdots, x_N, x_1, x_2\} \cup V_0, \\
&\cdots \\
X_N &= \{x_N, x_1, x_2, \cdots, x_k\} \cup V_0,
\end{aligned}$$

[1] We thank one of the referees for pointing out this paper and the theorem.

and define a tree structure on S by assigning an edge between each of the pairs $(X_i, X_{i+1}), 1 \leq i \leq N-1$. It is easy to verify that the collection of subsets of variables and the tree structure specified in the above form a tree decomposition with a width $2k$. □

To investigate the space complexity of FDAs for NK landscapes with random neighborhoods and pure random model for additive fitness functions, we need some results from the study of treewidth.

In (Kloks, 1994), it is shown that a necessary condition for a graph to have a treewidth at most l is that the graph has a balanced l-partition. We only give the definition of balanced l-partitions. The establishment of the necessary condition can be found in (Kloks, 1994).

Definition 4.1. *(Kloks, 1994) Let $G(V, E)$ be a graph with $|V| = n$. A partition (S, A, B) of V is a balanced l-partition if the following conditions are satisfied:*

1. $|S| = l + 1$;
2. $\frac{1}{3}(n - l - 1) \leq |A|, |B| \leq \frac{2}{3}(n - l - 1)$; *and*
3. *S separates A and B, i.e., there are no edges between vertices of A and vertices of B.*

The theorem below deals with the space complexity of an FDA for NK landscapes with random neighborhoods.

Theorem 4.3. *Let f(x) be an instance of the neighborhood model with random neighborhood $\mathcal{N}(n, k)$. Then, with probability asymptotic to 1, the space complexity of FDA is $F_f = 2^{\Omega(n)}$.*

Proof. Similar to the proof of Theorem 4.4, and can be found in (Gao and Culberson, 2003). □

For the more general random models of additive functions, we have

Theorem 4.4. *Let f(x) be an instance of the pure random model $\mathcal{F}(n, m, k)$. Then, with probability asymptotic to 1, the space complexity F_f of FDA is $2^{\Omega(n)}$ if $\frac{m}{n} > \frac{\ln 2}{k \ln 3 - \ln(1+2^k)}$, i.e., $\frac{m}{n} > \frac{\epsilon}{k}$ for a constant $\epsilon > 0$.*

Proof. Let $G_f = G_f(V, E)$ be the interaction graph of $f(x)$ and $\omega(f)$ be the treewidth of the interaction graph G. We prove that there is a $0 < \delta < 1$ such that

$$\lim_n Pr\{\omega(f) \leq \delta n\} = 0. \tag{14}$$

From (Kloks, 1994), a necessary condition for a graph to have a treewidth at most l is that the graph must have a balanced l-partition. Let $\mathcal{P}$ be the set of all the $l-$partitions of the vertex set V that satisfies the first two conditions of the definition of balanced partition (Definition 4.1). For a given $P = (S, A, B) \in \mathcal{P}$, define a random variable I_P as follows:

$$I_P = \begin{cases} 1, & \text{if } P \text{ is a balanced partition;} \\ 0, & \text{otherwise.} \end{cases}$$

Note that $I_P = 1$ if and only if there are no edges between vertices in A and vertices in B in the interaction graph G_f.

Let $N = \binom{n}{k}$ be the number of possible subsets of variables for local fitness functions of size k, and let N_P be the number of possible local fitness functions whose k defining variables are either in $A \bigcup S$ or in $B \bigcup S$. Let $a = |A|$, we have

$$\begin{aligned} N_P &= \binom{a+l+1}{k} + \binom{n-a}{k} - \binom{l+1}{k} \\ &\leq \binom{a+l+1}{k} + \binom{n-a}{k}, \end{aligned}$$

and thus,

$$\begin{aligned} \frac{N_p}{N} &\leq \frac{(a+l+1)\cdots(a+l+1-k+1)}{n(n-1)\cdots(n-k+1)} \\ &\quad + \frac{(n-a)\cdots(n-a-k+1)}{n(n-1)\cdots(n-k+1)} \\ &\leq \left(\frac{a+l+1}{n}\right)^k + \left(\frac{n-a}{n}\right)^k \\ &= \frac{(a+l+1)^k + (n-a)^k}{n^k}. \end{aligned} \tag{15}$$

Write $y = \frac{l+1}{n}$ and consider the function $h(a) = (a+l+1)^k + (n-a)^k$ defined on the interval

$$[\frac{1}{3}(n-l-1), \frac{2}{3}(n-l-1)].$$

Since $h'(a) = 0$ at $a = \frac{1}{2}n(1-y)$. If $y = \frac{l+1}{n}$ is sufficiently small, then, $h(a)$ is maximized at $a = \frac{1}{3}(n-l-1)$. Therefore, we have

$$\begin{aligned} \frac{N_p}{N} &\leq \frac{1}{n^k}\left((\frac{1}{3}n+l+1)^k + (\frac{2}{3}n+l+1)^k\right) \\ &= (\frac{1}{3})^k(1+3y)^k + (\frac{2}{3})^k(1+\frac{3}{2}y)^k \\ &\leq \left((\frac{1}{3})^k + (\frac{2}{3})^k\right)(1+3y)^k \end{aligned} \tag{16}$$

It follows that

$$E\{I_P\} \leq \frac{\binom{N_P}{m}}{\binom{N}{m}} \leq \left(\left(\frac{1}{3}\right)^k + \left(\frac{2}{3}\right)^k\right)^m (1+3y)^{km}.$$

Let $I = \sum_{P \in \mathcal{P}} I_P$. By its definition, we have

$$\begin{aligned} |\mathcal{P}| &= \binom{n}{l+1} \sum_{\frac{1}{3}(n-l-1) \leq a \leq \frac{2}{3}(n-l-1)} \binom{n-l-1}{a} \\ &\leq \binom{n}{l+1} 2^n. \end{aligned}$$

It follows that the expectation of I satisfies

$$\begin{aligned} E\{I\} &= \sum_{P\in\mathcal{P}} E\{I_P\} \\ &\le \binom{n}{l+1} 2^n \left(\left(\frac{1}{3}\right)^k + \left(\frac{2}{3}\right)^k \right)^m (1+3y)^{km}. \end{aligned}$$

Recall that $0 < y = \frac{l+1}{n} < 1$. We obtain from Stirling's formula that

$$\binom{n}{l+1} \sim \frac{1}{\sqrt{2\pi y(1-y)n}} \left(\frac{1}{y^y(1-y)^{1-y}} \right)^n.$$

And hence,

$$\begin{aligned} E\{I\} \quad \le \quad & \frac{1}{\sqrt{2\pi y(1-y)n}} \left(\frac{2}{y^y(1-y)^{1-y}} \right)^n \\ & \cdot \left(\left(\frac{1}{3}\right)^k + \left(\frac{2}{3}\right)^k \right)^m (1+3y)^{km}. \end{aligned} \tag{17}$$

Notice that

$$\lim_{y\to 0} \frac{2}{y^y(1-y)^{1-y}} = 2.$$

For any $\frac{m}{n} > c$ with c satisfying

$$\left(\left(\frac{1}{3}\right)^k + \left(\frac{2}{3}\right)^k \right)^c < \frac{1}{2},$$

let $y = \frac{l+1}{n}$ be small enough so that

$$\frac{2}{y^y(1-y)^{1-y}} \left(\left(\frac{1}{3}\right)^k + \left(\frac{2}{3}\right)^k \right)^c (1+3y)^{kc} < 1$$

and let $\delta = y$, we have

$$\begin{aligned} \lim_n Pr\{\omega(f) \le \delta n\} \quad &\le \quad \lim_n Pr\{I > 0\} \\ &\le \quad \lim_n E[I] = 0, \end{aligned}$$

that is, (14) is true. The theorem is proved. □

For the space complexity of BNAs, we need to evaluate the value of $\overline{w}(f)$ (see Definition 3.13). We show in the following that $\overline{w}(f)$ is actually equal to the induced width $w^*(f)$ of f, and consequently the treewidth of f.

First, we establish two lemmas on the induced width of a general graph. Let $G = G(V,E)$ be a graph and $\pi = (x_1, \cdots, x_n)$ be an ordering of the vertices. For each $1 \le i \le n$, define

$$A_i = \{x_j;\ \ x_j \text{ is adjacent to } x_i,\ \ 1 \le j < i\},$$

$$\begin{aligned} C_i = \{x_j;\ \ &\text{There is a path } x_i x_{i_1} \cdots x_{i_k} x_j \\ &\text{s.t. } \{x_{i_1}, \cdots, x_{i_k}\} \subset \{x_{i+1}, \cdots, x_n\},\ \ 1 \le j < i\}, \end{aligned}$$

and let $B_i = A_i \cup C_i$.

Lemma 4.1. *For each $1 \leq i \leq n$, B_i separates x_i and $\{x_1, \cdots, x_{i-1}\} \setminus B_i$ and is minimal with respect to $\{x_1, \cdots, x_{i-1}\}$.*

Proof. For any $x_m \in \{x_1, \cdots, x_{i-1}\} \setminus B_i$, let $x_i x_{i_1} \cdots x_{i_k} x_m$ be a path connecting x_i and x_m. Assume x_{i_p} is the first variable in $\{x_{i_1}, \cdots, x_{i_k}\}$ that belongs to $\{x_1, \cdots, x_{i-1}\}$. If $p = 1$, then $x_{i_p} \in A_i$; Otherwise, $x_{i_p} \in C_i$. It follows that B_i separates x_i and $\{x_1, \cdots, x_{i-1}\} \setminus B_i$ in the graph G. Since removing any vertex x from B_i will either make x directly adjacent to x_i, or leave a path that connects x_i and x, but does not pass through $\{x_1, \cdots, x_{i-1}\}$, B_i is minimal. □

Lemma 4.2. *Let $w^*(G, \pi)$ be the induced width of G under the ordering π, then*

$$\max_{1 \leq i \leq n} |B_i| = w^*(G, \pi).$$

Proof. Let $G_i(\pi)$ be the graph after $(x_{i+1}, \cdots, x_n)$ have been eliminated, and let N_i be the set of neighbors of x_i in $G_i(\pi)$. We prove that $N_i = B_i$.

By the definition, we have $A_i \subset N_i$ and $C_i \subset N_i$, and consequently $B_i \subset N_i$. To establish the opposite inclusion, consider a variable $x \in N_i$. If x is adjacent to x_i in G, then $x \in A_i$; Otherwise, there must be a $j > i$ such that $x \in N_j$ and $x_i \in N_j$. Based on this fact and by induction on i, we can show that there is a path $x_i x_{i_1} \cdots x_{i_k} x$ such that $\{x_{i_1}, \cdots, x_{i_k}\} \subset \{x_{i+1}, \cdots, x_n\}$. It follows that $x \in C_i$. □

Based on the above two lemmas, we have the following result, showing that the space complexity of BNAs is also exponential in the treewidth.

Theorem 4.5. *For any additive fitness function f and a perfect map P_f of G_f that is strictly positive, $\overline{w}(f) = \min_{\pi} \max_{1 \leq i \leq n} |B_i(\pi)|$, as defined in Definition 3.13, is equal to the treewidth of f.*

Proof. Let π be a variable ordering and $B(V, E)$ be the unique Bayesian network with $Pa(x_i) = B_i(\pi)$ as given in Theorem 3.1. Based on the above two lemmas, we have $\max_{1 \leq i \leq n} |B_i(\pi)|$ is equal to the induced width $w^*(f, \pi)$. Taking the minimum over all the variable orderings completes the proof. □

From the above result and similar to the proof of Theorem 4.4, we have

Theorem 4.6. *Let f(x) be an instance of the pure random model $\mathcal{F}(n, m, k)$. Then, with probability asymptotic to 1, the space complexity B_f of BNA is $2^{\Omega(n)}$ if $\frac{m}{n} > \frac{\ln 2}{k \ln 3 - \ln(1+2^k)}$, i.e., $\frac{m}{n} > \frac{\epsilon}{k}$ for a constant $\epsilon > 0$.*

5 Conclusions

In this paper, we have discussed in detail the space complexity of two typical implementation schemes of the estimation of distribution algorithm. We identified criteria to characterize the space complexity of these implementations. Using random additive functions as our prototype, we prove that the space complexities of the factorized distribution algorithm and the Bayesian optimization algorithm are both exponential in the problem size even if the optimization problem has a very sparse interaction structure.

Our results should not be viewed as purely negative results. Similar to what has been shown in a related work on the tractability of constraint satisfaction and Bayesian network inference problems (Gao, 2003), results presented in the current paper only indicate that EDAs have their limitations and only work well in a part of the problem space where the interaction structure (i.e., the linkage structure) has a bounded

treewidth. In the literature, there have been many studies that emphasize the importance of linkage in the design of efficient genetic algorithms (Heckendorn and Wright, 2003; Yu and Goldberg, 2004). It would be interesting to further investigate the relations between the treewidth of interaction structures and the efficiency of general genetic algorithms that make use of linkage-identification methods.

The analysis in the current paper is based on pure random models of additive fitness functions. Similar random models have also been widely used in the recent study of the typical complexity and phase transitions of NP-complete problems (Kirkpatrick and Selman, 1994; Cook and Mitchell, 1997; Martin et al., 2001). In the past several years, it has been found that many real-world interaction structures such as the Internet and gene interaction can be better modelled as power-law random graphs (Dorogovtsev and Mendes, 2002). It is interesting to investigate the treewidth in these power-law graphs and to identify situations where the interaction graph of an optimization problem follows the power-law distribution. Also, for many real-world optimization problems, the interaction structure is actually unknown, and identifying these unknown interaction structures is also an interesting research topic (Heckendorn and Wright, 2003).

Acknowledgments

We thank the anonymous reviewers for their valuable comments.

References

Becker, A. and Geiger, D. (1996). A sufficiently fast algorithm for finding close to optimal junction trees. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 81–89. Morgan Kaufmann.

Bodlaender, H. L. (1997). Treewidth: algorithmic techniques and results. In *Lectures Notes in Computer Science 1295*, pages 19–36. Springer.

Bouchitt, V. and Todinca, I. (2001). Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing*, 31(1):212–232.

Braunstein, A., Mezard, M., Weigt, M., and Zecchina, R. (2003). Constraint satisfaction by survey propagation. Technical Report arXiv:cond-mat/0212451, http://arxiv.org/abs/cond-mat/0212451.

Cook, S. and Mitchell, D. G. (1997). Finding hard instances of the satisfiability problem: A survey. In Du, Gu, and Pardalos, editors, *Satisfiability Problem: Theory and Applications*, volume 35 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society.

Culberson, J. (1998). On the futility of blind search: An algorithmic view of "no free lunch". *Evolutionary Computation*, 6(2):109–128.

Dechter, R. (1999). Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113:41–85.

Dorogovtsev, S. and Mendes, J. (2002). Evolution of networks. *Adv. Phys.*, 51:1079–1187.

Etxeberria, R. and Larrañaga, P. (1999). Global optimization using Bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence and Adaptive Systems CIMAF 99. Special Session on Distributions and Evolutionary Computation. La Habana, Cuba*, pages 314–324.

Gao, Y. (2003). Phase transition of tractability in constraint satisfaction and Bayesian network inference. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI-2003)*, pages 265–271. Morgan Kaufmann.

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

Gao, Y. and Culberson, J. (2003). On the treewidth of NK landscapes. In *Genetic and Evolutionary Computation Conference (GECCO-03)*, LNCS 2723, pages 848–954. Springer-Verlag.

Gent, I. P., Hoos, H. H., Prosser, P., and Walsh, T. (1999). Morphing: Combining structure and randomness. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, pages 654–660, Orlando, Florida.

Goldberg, D., Deb, K., Kargupta, H., and Harik, G. (1993). Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In *Proceedings of the 5th Intern. Conf. on Genetic Algorithms*, pages 56–64. Morgan Kaufman.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Heckendorn, R. and Wright, A. (2003). Efficient linkage discovery by limited probing. In *Genetic and Evolutionary Computation Conference (GECCO-03)*, LNCS 2723, pages 1003–1014. Springer-Verlag.

Kallel, L., Naudts, B., and Reeves, C. R. (2001). Properties of fitness functions and search landscapes. In Kallel, L., Naudts, B., and Rogers, A., editors, *Theoretical Aspects of Evolutionary Computing*, pages 175–206. Springer, Berlin.

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

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Kirkpatrick, S. and Selman, B. (1994). Critical behavior in the satisfiability of random boolean expressions. *Science*, 264:1297–1301.

Kloks, T. (1994). *Treewidth: Computations and Approximations*. Springer-Verlag.

Larrañaga, P., Etxeberria, R., Lozano, J. A., and Peña, J. M. (2000). Combinatorial optimization by learning and simulation of Bayesian networks. In Boutilier, C. and Goldszmidt, M., editors, *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 343–352. Morgan Kaufmann.

Larrañaga, P. and Lozano, J. A. (2001). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publisher.

Leung, Y., Gao, Y., and Xu, Z. (1997). Degree of population diversity - a perspective on premature convergence in genetic algorithms and its Markov chain analysis. *IEEE Transactions on Neural Networks*, 8:1165–1176.

Leung, Y., Gao, Y., and Zhang, W. (2001). A genetic-based method for training fuzzy systems. In *Proc. of the 10th IEEE International Conference on Fuzzy Systems*, volume 1, pages 123–126. IEEE.

Martin, O., Monasson, R., and Zecchina, R. (2001). Statistical mechanics methods and phase transition in optimization problems. *Theoretical Computer Science*, 265:3–67.

Mezard, M. and Parisi, G. (2003). The cavity method at zero temperature. *J. Stat. Phys.*, 111:1–23.

Mühlenbein, H. and Mahnig, T. (1999). Convergence theory and applications of the factorized distribution algorithm. *Journal of Computing and Information Technology*, 7:19–32.

Mühlenbein, H., Mahnig, T., and Ochoa-Rodríguez, A. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausiable Inference*. Morgan Kauffmann.

Pelikan, M., Goldberg, D. E., and Lobo, F. (1999). A survey of optimization by building and using probabilistic models. Technical Report 99018, IlliGAL, University of Illinois.

Whittaker, J. (1989). *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 4:67–82.

Yu, T. and Goldberg, D. E. (2004). Toward an understanding of the quality and efficiency of model building for genetic algorithms. In *Genetic and Evolutionary Computation Conference (GECCO-04)*, LNCS 3103, pages 367–378. Springer-Verlag.