The Incident Edge Model

Roberto Santana, Eunice Ponce de León, Alberto Ochoa

Institute of Cybernetics Mathematics and Physics, La Habana, Cuba.
{*rsantana, eunice, ochoa* }*@cidet.icmf.inf.cu*

In this paper we introduce the incident edge model, a fitness function model for a large set of problems defined on graphs. The model can be used to define functions whose optimization led to the finding of different structures on graphs. The model can also be useful to determine the best optimization algorithm for a given problem. As an example of the application of our model we describe an optimization approach for the problem of finding the dissection of a graph. For the optimization of this additively decomposable function we use the Factorized Distribution Algorithm. We focus on the way that different factorizations of the probability distribution can influence the behavior of the Factorized Distribution Algorithm for the dissection problem .

Keywords: Graphs, population based search, additively decomposable functions, fitness model, EDA

1. Introduction

Problems defined on graphs have been extensively considered in the field of Evolutionary Computation. Research devoted to the solution of theoretical and real graphs problems using evolutionary algorithms is widely refered in the literature, these problems have been also employed as a test bed for the validation of several evolutionary techniques.

In this paper we introduce a model for a large set of problems defined on graphs. The problems under consideration are those where the goal is to find structures (such as hamiltonian paths, cliques or spanning trees) in the graph. A necessary condition is that the search of structures could be trasformed in the optimization of a function defined for a set of variables mapping the edges of the graph.

The model, that we have called Incident Edge Model (IEM), can be used to design functions whose optimization led to the finding of the structures, the analysis of the model can also help to decide which is the suitable optimization algorithm for the given problem. Finally the IEM could be useful as a conceptual framework for the study of the characteristics that make a problem hard for an optimization algorithm.

As an example of the application of our model we describe an optimization approach for the problem of finding the dissection of a graph. A function is defined for this problem using the Incident Edge Model as a framework. The dissection function is also an additive function, we show how the Factorization Distribution Algorithm (FDA) can be used for the optimization of this function. FDA takes advantage of information about the structure of the function to do an efficient sampling. Previous results show that the FDA outperforms other population based search methods in the optimization of several ADF functions.

The outline of the paper is as follows. In Section 2 we introduce the IEM. In Section 3 the dissection function is presented as an example of an IEM. The way of constructing an independence graph G' from a simple graph G is described in Section 4. Some particular characteristics of these independence graphs are pointed out. In Section 5 we briefly present the FDA algorithm that was used in the optimization. Experiments are shown in Section 7. Finally the conclusions of our work are presented .

2. The Incident Edge Model

The study of complex systems has often led to the creation of simplified models that allow to understand the dynamics that these systems exhibit. In Evolutionary Optimization this sort of models has been successfully employed for the analysis of the different components or processes that influence the behavior of the optimization algorithms, such as fitness landscapes, genetic operators and variables interaction. We introduce here a model that supports a theoretical framework for the study of some of the factors that can make of a problem defined on a graph difficult. We assume that the problems considered can be solved through the optimization of a fitness function.

In order to introduce our model we present first the NK model of adaptation developed by Kauffman[4]. Initially Kauffman's model was introduced to envision nonrandom fitness landscapes whose contours reflect the underlying nonlinear and complex dynamics among the components in a system or ecosystem. The ideas of NK fitness landscapes were later incorporated to Genetic Algorithms[3] to analyze the behavior of GA in the optimization of additive functions.

For the classical Kauffman's N-K model the total fitness contribution of the string $x = (x_1, ..., x_n)$ is defined as an averaged sum of fitness contributions, that is:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(s_i)$$

where s_i is the substring of x of length $k + 1$. It includes site i of the string x. The k sites are called the neighbors of site i. The simplest way of choosing the neighbors is to use the k sites adjacent to site i. Alternatively, one assigns the neighbors by randomly selecting for each site i, k other sites from the string x.

The Incident Edge Model is defined from a simple graph $G =< V, E >$. To each edge belonging to E we map a binary variable x_i in the string $x = (x_1, ..., x_{|E|})$. The total fitness contribution $f(x)$ is the sum of n subfunctions defined on the substrings of x and another function $v(x)$ defined on x.

$$f(x) = u(x) + v(x)$$

$$f(x) = \sum_{i=1}^{n} u(s_i) + v(x)$$

The distinctive characteristic of the Incident Edge Model is the way the substrings s_i are selected: Each substring s_i includes those variables mapping the edges incident to the vertex i in G. As a consequence the length of the substring s_i is equal to the degree of vertex v_i in the graph. Subfunctions u_i are conveniently chosen.

The decomposition of $f(x)$ in the terms $u(x)$ and $v(x)$ reflects the case, commonly found in problems defined on graphs, when the fitness of a solution can be determined by considering first some neccesary conditions to be fulfilled by the subsolutions, and then analyzing a measure of the quality of the complete solution. We have presented examples of this kind of functions in [9], nevertheless in this paper we only consider the case when $v(x) = 0, \forall x$, and thus $f(x) = u(x)$, in this case $f(x)$ is an additive function (ADF). Additive functions have been extensively studied in Evolutionary Computation. We present now a formal introduction to ADF[2].

Let $I_n = \{1, ..., n\}$ be the set of all variable indices, $X = \{0, 1\}$ and let $f : X^n \rightarrow R$.$f(x)$ is called an additive function if:

$$f(x) = \sum_{l=1}^{m} f_l(x_{j_l})$$

where $J_l = \{l_1, ..., l_{n(l)}\} \subseteq I_n$, and $X_{j_l} := \{X_{l_1}, ..., X_{l_{n(l)}}\}$.

To avoid trivial cases we suppose that the sets J_l are minimal i.e. it is not possible to find K and M with $K \cup M = J_l$ and $K \cap M \neq J_l$, such that $f_l(x_{j_l}) = f_l(x_K) + f_l(x_M)$, the J_l are called definition sets.

When f(x)=u(x), as is the case we consider here, the number of definition sets is $m = |V|$ and each set X_{j_l} corresponds to a substring s_i.

In comparison to the NK model the IEM also includes overlapping variables, but due to the constraints imposed by the simple graph G, each variable belongs to only 2 different definition sets of $f(x)$. Furthemore the number of definition sets of the function is determined by the structure of graph G and not by the number of variables N.

The IEM explicitly represents some of the factors that make the optimization of a function difficult. Given a problem defined on a graph G it is possible to have a measure of its difficulty by analyzing the size of the definition sets, the number of overlapping variables and the characteristics of the subfunctions evaluated in each definition set. The knowledge that the model supports can be used to clarify the way in which different optimization algorithms behavior and to improve these algorithms.

¿From here when referring to instances of the IEM we will name them incident edge functions. In the next section we present an example of an incident edge function.

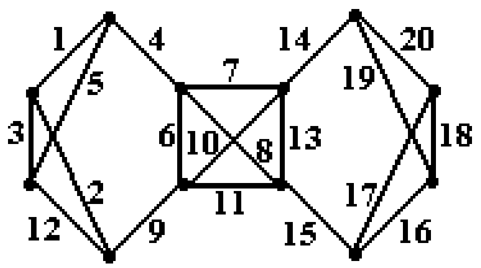

Figure 1. Simple graph G

3. The dissection function

Definition 1 .- A **dissection** of a graph G is defined to be a set of elementary paths of G such that each vertex of the graph is contained in exactly one path. We will represent one dissection as a graph $D =< V', E' >$ where $V' = V$ and $E \subseteq E$. We will call a path closed if it also defines a circuit, otherwise we will call it open. The value of a dissection D is defined to be the number of open paths present in it. A dissection of value 0 is a set of closed paths in G.[1]

Lemma 1. Theorem 1: Given a graph G, the value of a dissection D of G defines the number of edges the dissection has [8].

In [9] we have defined a family of functions for detecting dissections of different values on graphs. We have associated the i component of the vector to the i edge in E. For the component i, value 1 stands for "select the corresponding edge", whereas bit value 0 stands for "do not select the corresponding edge". Then, each vector will represent a subgraph G' of G such that all the edges of G are those components of the vector with value 1.

Let be $f(x) = \sum_{l=1}^{|V|}(S(x_{j_l}) - \alpha)^2$ where

$S(x_{j_l}) = \sum_{l=1}^{|x_{j_l}|}(x_{l_i})$.

$f(x)$ calculates the sum of the square's differences, taken coordinate to coordinate between a vector with the degree of each vertex of the subgraph G represented by the binary string x and α. For $\alpha = 1(\alpha = 2)$ $f(x)$ will reach its minimum for all those vectors W that represent dissections of the graph G with value $\frac{|V|}{2}$ (0) respectively. Demonstration could be found in [9]

It could be seen that $f(x)$ is an incident edge function which is also additive.

4. The independence graph of the Incident Edge Model.

In the previous section we have introduced two incident edge functions whose optimization leads to the detection of dissections on a graph. Now we address the question of how to optimize these functions using a population based search method that uses selection. We consider how the optimization algorithm can take advantage of the information store in the IEM.

Population Based Search Methods that use Selection (PBSMS) do the search of solutions using a set of points instead of a single one. These algorithms start generating a set of initial points, from this population another set of promising points is selected and a new population is generated. This process is repeated until a stop condition is satisfied.

It has been stated that a good search strategy for PBOMS is to generate new points with a similar probability distribution to that existing in the selected set, this is:

$P(x, t+1) \approx P^s(x, t)$[6].

The class of PBSMS which estimates a probability distribution of points in the selected set and uses this information to generate new points is called Estimation of Distribution Algorithms (EDA), nevertheless the estimation of distributions is a difficult problem.

The questions of when and how can a probability distribution be factorized arise from the fact that a straightforward implementation of the above equation is computationally prohibitive. The search of suitable factorizations has led to the study of the relation between the underlying structure of the fitness function and different probability distributions.

It has been shown that the general class of Additively Decomposable Functions can be mapped into a corresponding probability model, which captures the dependencies of the variables [7]. The following approximation has been proven to work in some situations: Two variables are dependent if they are contained in the same set s_i. The dependencies can be mapped into a conditional independence graph. From a conditional independence graphs it is possible to extract factorizations of the probability distribution. An EDA algorithm which uses this factorizations already exists[7], we explain it in the next section.

When $v(x) = 0$ our IEM is a particular case of the ADF class of functions, thus the assumption of con-

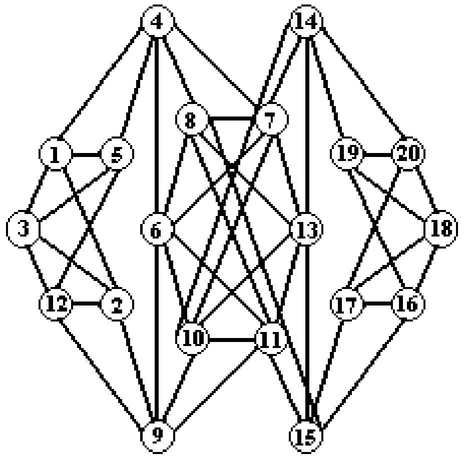

Figure 2. G', dependence graph of G

sidering as related those variables whose corresponding edges are incident to a common vertex could be taken as valid, as far as the definitions set for the functions we use are based on this characteristic of the graph structure.

Let $G =< V, E >$ be a simple graph and let the vector x be constructed from the mapping between edges in E and variables x_i. If we consider that two variables x_i, x_j are related if the edges that they represent are incident in a common vertex, then we can construct the dependency graph $G' < V', E' >$ of the variables xi where $|V'| = |E|$.

Figure 2 shows the dependency graph G' where variables are related in G' if their corresponding edges are incident to a common vertex of the graph G (Figure 1).

In the domain of the problems defined on graphs there exist functions which are not additively decomposable.When $v(x) \neq 0$ for an incident edge function the conditional independence graph could have links between every pair of variables, nevertheless even in this case the indepedence graph associated to the additive function $u(x)$ could capture the most important dependencies determined by $f(x)$. We present now an algorithm which, by measuring the degree of interaction between variables, gives support to the previous statement .

The dependency algorithm introduced in [2]is a discrete version of the mixed partial derivative and starts from the following lemma:

Define $x \in X^n$and $\{\bar{i}, \bar{j}\} = I_n \setminus\{i, j\}$

$\Delta_{i,j}(f, x_{\{\bar{i},\bar{j}\}}) = [f(x_i^1, x_{j,}^1 x_{\{\bar{i},\bar{j}\}}) - f(x_i^0, x_{j,}^1 x_{\{\bar{i},\bar{j}\}})] - [f(x_i^1, x_{j,}^0 x_{\{\bar{i},\bar{j}\}}) - f(x_i^0, x_{j,}^0 x_{\{\bar{i},\bar{j}\}})]$

Then we have: $\Delta_{i,j}(f, x_{\{\bar{i},\bar{j}\}}) = 0$ for all $x \in X^n$ if and only if there is not l such that i and j do not belong to one same dependency set.

The algorithm begins with a random population X and calculates

$|\Delta_{i,j}(f)|\,(\mathrm{X}) = \sum_{x\in X} \left|\Delta_{i,j}(f, x_{\{\bar{i},\bar{j}\}})\right|$ for all pairs i, j . When $|\Delta_{i,j}(f)|\,(\mathrm{X}) = 0$ is intended that variables i and j are not related. As X does not contain all the vectors of the domain the dependencies are determined only with a certain probability.

If we apply the previous algorithm to a function for which interactions between all the variables exist, it will be obtained a unique dependency set containing all the variables of the problem.

Let us introduce the problem with an example:

Definition .- A hamiltonian cycle of G is a cycle through G that touches all vertices of V exactly once. A hamiltonian cycle is a dissection of value 0.

In [4] we have defined a function Fhc(x) able to detect a hamiltonian cycle in a graph G.

$Fhc(x) = \frac{1000}{f(x)}$ if $f(x) > 0$

$Fhc(x) = \frac{1000}{f(x)} + f_1(x)$ if $f(x) = 0$

where $f(x)$ is the dissection function, c$_i$ is the length of substring s_i, and

$f_1(x) = \sum_{i=1}^{m} c_i * (c_i - 1)$

When $f(x)$ reaches the value 0 then the vector x contains a set of cycles (m cycles). Function $f_1(x)$ measures the size of all the cycles in x. $f_1(x)$ reaches its maximum when there is a unique and maximum sized cycle in G, i.e. x contains a hamiltonian cycle. In this problem all the variables are related.

We apply the algorithm to function $Fhc(x)$ using a population X composed of all the feasible solutions.A solution is feasible for $f(x)$ and $Fhc(x)$ when there exist in the vector x exactly n components set to 1 (Any hamiltonian cycle connecting a graph with n vertices contains exactly n edges.). The selected base graph in the subgraph of G' composed by edges { 6,7,8,10,11,13 }, this simple graph has 12 edges.

Columns 1-2 of Table 1 show a mapping between edges of the selected graph and a set of 12 variables. There are four different values for $|\Delta_{i,j}(f)|\,(\mathrm{X})$, these are: {I1=14620,I2=14656,I3=18540,I4=20624}. The Table 1, columns 3-6, shows for each variable i, the remaining 11 variables clustered on their values $|\Delta_{i,j}(f)|$.

Although values in Table 1 are all higher than 0 indi-

Table 1
$|\Delta_{i,j}(f)|\,(X)$ values for the function $Fhc(x)$

Var	$Edge$	I_1	I_2	I_3	I_4
1	$6-7$	$8,9,10,12$	11	$2,4,5,6$	$3,7$
2	$7-8$	$6,7,10,11$	12	$1,3,5,8$	$4,9$
3	$7-13$	$5,6,9,12$	7	$2,4,8,10$	$1,11$
4	$7-10$	$5,7,9,11$	8	$1,3,6,10$	$2,12$
5	$6-8$	$3,4,8,11$	9	$1,2,7,10$	$6,12$
6	$6-10$	$2,3,8,11$	10	$1,4,7,12$	$5,9$
7	$6-11$	$2,4,8,10$	3	$5,6,9,12$	$1,11$
8	$8-13$	$1,4,7,12$	6	$2,3,9,11$	$5,10$
9	$8-11$	$1,3,6,10$	4	$5,7,8,11$	$2,12$
10	$10-13$	$1,2,7,9$	5	$3,4,11,12$	$6,8$
11	$11-13$	$2,4,5,6$	1	$8,9,10,12$	$3,7$
12	$10-11$	$1,3,5,8$	2	$6,7,10,11$	$4,9$

cating that all variables are related. we can exclude for a general independence graph those edges corresponding to relations between variables for which the second differences are under a defined threshold. For instance, if we set the threshold in Table 1 to 14620, and consider as related variables whose entries are over this value, then the interactions could be represented using the same independence model associated to the dissection function for this graph.

5. The Factorized Distribution Algorithm (FDA)

We use the Factorized Distribution Algorithm for the optimization of functions described in this paper. The FDA is a particular case of the Estimation Distribution Algorithms[5] which uses a factorization of the joint probability, in [7] the factorization is constructed from an independency graph based on the definition sets of the function. In that paper is assumed that two variables are dependent if they are contained in the same definition set. In this way the algorithm is supposed to capture the structure of the given function. An exact correspondence between the joint distribution and its factorization has been theoretically demonstrated only for the Boltzman distribution. This correspondence has been used for the implementation of a conceptual EDA with infinite population and Boltzman selection (BEDA).

FDA is very similar to BEDA but uses a finite population. Any selection method can be employed, but the truncation selection has been the norm for current FDA implementations. It has been shown that to use a factorization constructed from an independency graph based on the definition sets is convenient also for the FDA. Previous experimental results show that a FDA that uses this strategy to factorize the joint distribution outperforms other population based search methods in the optimization of several ADF functions.

Now we describe the FDA we have used in this paper. The algorithm inputs a junction tree where the factorization is represented:

The Factorized Distribution Algorithm

○ STEP 0: Set $t \Leftarrow 0$. Generate N points randomly among the feasible solutions

○ STEP 1: Truncation Selection

○ STEP 2: Update the densities associated to the junction tree.

○ STEP_3: Generate the new population using the junction tree.

○ STEP_4: If termination criteria is met, FINISH.

○ STEP_5: Set $t \Leftarrow t+1$. Go to STEP 2.

In order to optimize an incident edge function the independence graph can be constructed from the graph were the problem is defined.When the independence graph of variables is chordal, the factorization can be obtained immediatly. Otherwise different techniques can be used to obtain an approximate factorization, one that has shown good results[7] is to find one triangulation of the independence graph by adding new edges. In this paper we explore a new approach. Instead of finding a triangulation of the independence graph by the addition of new edges, we eliminate a certain number of edges (dependencies between variables) in order to get another graph, very close to the independence graph but with the desirable property of being chordal.

¿From the chordal graph a junction tree is constructed. In our implementation the FDA employs the junction tree to estimate the joint distribution of the selected set and do the sampling.The example shown in Figure 3 represents a junction tree corresponding to a subgraph of G'' obtained by extracting edges. The choice of which edges to extract is critical, here we have

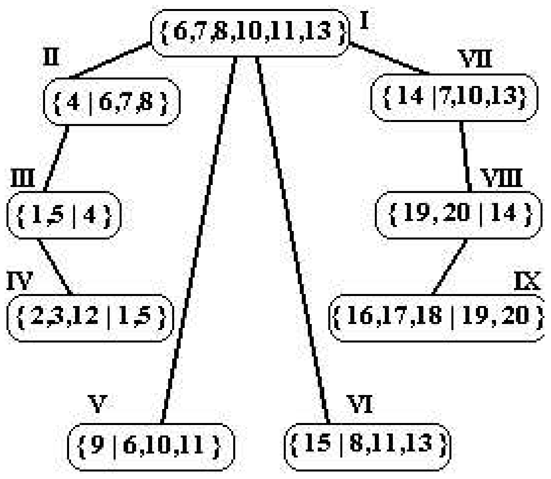

Figure 3. Junction tree of graph of G'

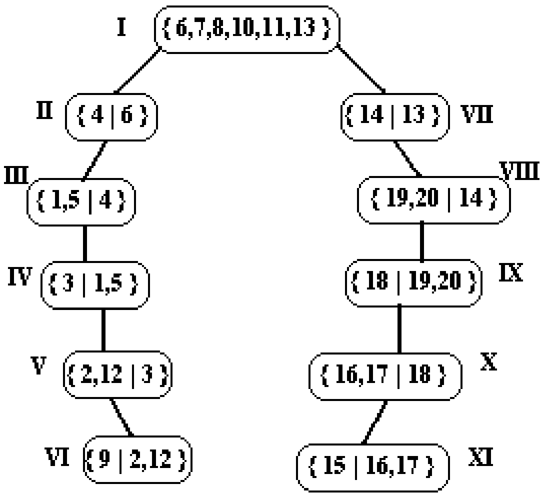

Figure 4. Chain shaped junction tree of graph G'

used the criteria of eliminating as few edges as possible in order to have a graph very similar to the original graph of dependencies. It can be seen that also some edges have been added, these edges make the function of completing those subgraphs that are near to be cliques.

6. Experiments and discussion of results

In this section we analyze the effectiveness of the FDA when it is used in the optimization of the dissection problem. We enphasize how the choice of different factorizations influences in the efficiency of the optimization process. The test problems that have been used for our experiments are the dissections of order $\frac{|V|}{2}$ and 0. The complexity of these problem depends on the graph under consideration.We have chosen the simple graph G as a benchmark for the optimization of these functions in order illustrate how the application of different factorizations of G' incides in the numerical results achieved by the FDA.

For each experiment, 100 independent runs were performed. We do not address in this paper the question of finding the optimal setting of parameters for the dissection problems. In order to test the performance of the different algorithms in similar conditions the parameters were set to the same value in all the experiments. Values of population size and the coefficient of truncation were respectively 300 and 0.15.

6.1. Experiment 1:

In order to evaluate the effectiveness of the FDA in the search of dissections, we compare the performance of the FDA and an UMDA in our base graph G. The FDA used the junction tree shown in figure 3.

Tables 2 and 3 show results of both algorithms for dissections of order $\frac{|V|}{2}$ and 0 respectively. In the tables the entry Clique indicates the clique that is taken as the root in the junction tree, Success is the number of experiments where the optimum was reached. Gen is the average generation where the optimum was reached. The algorithm exhibits different performance for both functions. The second function is more difficult to optimize. This evidence agrees with previous results obtained by applying other heuristic operators to the same problems.[9]

Table 2
Results for Dissection of order $\frac{|V|}{2}$

Algorithm	*Clique*	*Success*	*Gen*
FDA	4	85	3.43
FDA	5	91	3.32
FDA	6	91	3.29
UMDA		76	5.59

6.2. Experiment 2:

Now we consider other different junction trees with similar structure (The same number and size of the

Table 3
Results for Dissection of order 0

Algorithm	*Clique*	*Success*	*Gen*
FDA	4	71	3.50
FDA	5	72	3.44
FDA	6	62	3.58
UMDA		61	6.93

cliques) but where the variables belonging to each clique were randomly selected .

Table 4 shows the results of experiment 2. It can be appreciated that the FDA performance critically degrades when the variables that are in each clique do not correspond to those present in the definition sets of the function. This example also confirms that the way in which the IEM is defined is valid and consistent with the optimization algorithm. The knowledge about the dependencies determined by the function contributes to make a better sampling of the search space.

Table 4
Results using an invalid Junction Tree

Algorithm	*Clique*	*Success*	*Gen*
FDA	4	11	2.90
FDA	5	14	2.64
FDA	6	28	2.68

6.3. Experiment 3:

In this experiment we construct another dependency graph G'''close to G' but not as close as G''.We eliminate a higher number of edges in order to construct a chain shaped junction tree with smaller cliques (Figure 4). Reducing the size of cliques contribute to increase the speed of the algorithm. As results shown in Table 5 demonstrate, this reduction of the computational time spent in the search has a cost in terms of efficient. Nevertheless the new junction tree created from G''' is still better for the FDA than that used in the Experiment 2, this could be explained by the fact that this junction tree still captures a high number of those dependencies determined by the function.

Table 5
Results using an an approximated Junction Tree

Algorithm	*Clique*	*Success*	*Gen*
FDA	4	64	3.38
FDA	5	60	3.53
FDA	6	62	3.69

7. Conclusions and further work

In this paper we have introduced the Incident Edge Model that could be used for the detection of structures in graphs through the fuction optimization . We have shown how to construct the independency graph corresponding to this model. The independency graph has been used for the optimization of the function evidencing that to consider the structure of the ADF could be helpful in the optimization.

Additional experiments show that there exist a class of non additive functions for which it is possible to detect a strong contribution of definition sets. We conjecture that a factorization based on these definition sets could be a good approximation to the factorization of such non additive functions.Our results validate the idea that the kind of structural information that the FDA uses is a critical factor in the reduction of the amount of function evaluations needed to find the optimum.

Additional work is in progress related with the search of criteria for the determination of approximated factorizations straight from the initial graph G where the IEM has been defined.

References

[1] Claude Berge (1985): Graphs, North-Holland, Mathematical Library, New York.

[2] D. Cvetkovic ,Heinz Mühlenbein, and Gerard Paaß (1997): Optimization of Additively Decomposable Functions.

[3] Inman Harvey(1992): Species Adaptation Genetic Algorithms, in Toward a Practice of Autonomous Systems, Proc. of First ECAL, F. J. Varela and P. Bourgine(eds), MTT Press.

[4] Stuart Kauffman(1993): Origins of Order. Oxford University Press.

[5] Mühlenbein H. and Paaß G.(1996): From recombination of genes to the estimation of distributions I. Binary parameters, in Voigt et al., pp. 178-187.

[6] Mühlenbein H.(1998): The equation for response to selection and its use for prediction, Numb. 5, pp. 303–346.

[7] Mühlenbein H., Mahnig Th. and Rodŕiguez Ochoa A.(1998): Schemata, Distributions and Graphical Models in Evolutionary Optimization, to appear in Journal of Heuristics Vol 5, No. 2.,Also at http://set.gmd.de/AS/ga.publineu.html#index

[8] Santana R. and Ponce de León E.(1998): An evolutionary optimization approach for detecting structures on graphs, Smart Engineering System Design: Neural Networks, Fuzzy Logic, Rough Sets adn Evolutionary Programming, Dagli,Akay,Buczak,Ersoy and Fernandez Edtors, ASME press, pp. 371–376.

[9] Santana, R. and Ponce de León, E.(1998): A Conceptual Model for Detecting Structures on Graphs Using Evolutionary Optimization Algorithms",Technical Report ICIMAF, ICIMAF 98-69, CENIA 98-03. ISSN 0138-8916, December

[10] Whittaker, J.(1991): Graphical models in applied multivariate statistics. Wiley Series in Probability and Mathematical Statistics. New York : Wiley.