We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



1

Study of Genetic Algorithms Behavior for High Epitasis and High Dimensional Deceptive Functions

Gras Robin
University of Windsor
Canada

1. Introduction

Optimization is a widespread notion: a large number of concrete problems require, during the solving process, optimizing a set of parameters (or variables) with respect to a given objective function (or fitness function). These variables can take integer or real values. Such problems for which variables can only take integer values, are called *combinatorial optimization* problems. This paper focuses on combinatorial optimization problems. The set of all possible combinations of values for the variables of the problem represents the *search space* of the problem. *Constraint combinatorial optimization* problems - that is problems which define a set of constraints on the variables enabling a part of the search space - are not considered here. *Global combinatorial optimization* problems - for which the whole search space is available – are the main focus. In a large number of real problems, in particular in bioinformatics, the objective function is partially or entirely unknown. In this contest, it is however possible to calculate the value of the function in each point of the search space. This kind of problem is called a "black box" problem. Classical techniques of operations research are weakly or not at all fitted to black box problems. Consequently, evolutionist approaches of heuristic exploration strategies, have been developed specifically for black box problems. We are particularly interested in the complexity of such problems and in the efficiency of evolutionist methods to solve them.

1.1 Definitions and notations

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n binary variables and $C = 2^X$ the set of parts of X. X denotes an element of C. Let F be a function from C to R. A combinatorial optimization problem O is a problem of discovery of maxima, for every possible point X of the search space C, of the objective function $F(X)$. Such a problem is said to be of size n. Its complexity is the minimum number of X of C for which $F(X)$ has to be calculated in order to guarantee the discovery of maxima. This number obviously depends on n but also on a large number of other parameters which influence can be greater than the influence of n.

Let $M = \{X_m \in C \mid \forall X_i \in C, F(X_m) \geq F(X_i)\}$ the set of all the solutions of O (also called global maximums), that is the set of points C for which F is maximal.

Exploration algorithms parse C using one or more operators which, from one or several points of C, will lead to one or several points of C. Let Ec be a subset (or sample) of C. We

Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos,
ISBN 978-953-307-008-7, pp. 572, October 2009, I-Tech, Vienna, Austria

Open Access Database www.intechweb.org

define an operator o as a function from Ec_1 to Ec_2. Ec_2 is said to be the set of points of C, which are neighboring points of Ec_1 for operator o. We define a function Vo, called neighborhood function of operator o, such as $Vo(Ec_1) = Ec_2$. $P = \{C,F, Vo\}$ is called a landscape of O for operator o. Therefore, a landscape depends on the neighborhood associated to an operator.

It is now possible to define the notion of local maximum which depends on a neighborhood and consequently on an operator. The set of local maximums for a given operator o is the set $Mo = \{X_m \in C \mid \forall\ X_i \in Vo(X_m),\ F(X_m) \geq F(X_i)\}$.

We define the basin of attraction of a maximum according to an operator as the set $Bo(X_m) = \{X_0 \in C \mid \exists\ X_1,\ldots, X_n\ ,\ X_n = X_m \in Mo$ and $X_{i+1} \in Vo(X_i)\ \ \forall\ 0 \leq i \leq n\}$, that is the set of points from which it is possible to reach the maximum repetitively applying the operator o.

We call *strict hill-climber* a kind of exploration method that only uses operators of the form o: $Ec_1 \rightarrow Ec_2$ with $\forall\ X_i \in Ec_1$ and $X_j = o(X_i)$, $F(X_j) \geq F(X_i)$ and $X_j \in Vo(X_i)$. We call stochastic hill-climber a kind of exploration method that only uses operators of the form o: $Ec_1 \rightarrow Ec_2$ with $\forall\ X_i \in Ec_1,\ X_j \in Vo(X_i)$ and $X_j = o(X_i)$, $P(F(X_j) \geq F(X_i)) = g(Vo(X_i))$ where g is a non-uniformly distributed function on [0..1] with a bias towards high values. In general, we also have $|Ec_1| = |o(Ec_1)| = 1$. The most used neighborhood for hill-climbers is the Hamming neighborhood. In this case, the neighborhood of a point $X_i \in C$ is $Vo(X_i) = \{X_j \in C \mid H(X_i,X_j) = 1\}$ with $H(X_i,X_j)$ the Hamming distance between X_i and X_j.

We call *evolutionary algorithm* an exploration algorithm which maintains a sample Ec (called *population*) of points of C and uses an operator os, called *selection operator*, which, from Ec, produces another bias sample Ec' such as $Vos(Ec) = Ec$ and $\forall\ X_j \in Ec' = os(Ec)$ and $\forall\ X_i \in Ec$, $X_j \in Vos(Ec)$ and $P(F(X_j) \geq F(X_i)) = g(Vos(Ec))$ where g is a non-uniformly distributed function on [0..1] with a bias towards high values. Then, 0, 1 or several operators are probabilistically applied to each point of Ec' leading to a new population. The same process is recursively reapplied to this new population until the end of the algorithm.

1.2 Complexity of problems

The complexity of combinatorial optimization problems has been extensively studied. Mostly, NP-completion proofs were first realized for classical problems - such as that of the knapsack, graph coloring, search of a clique of size k…- (Garey & Johnson, 1979; Papadimitriou & Steiglitz, 1998). These proofs are however not very informative indices. Indeed, it is well known that for a problem with NP-complete proof, there are, in most cases, instances of the problem that can be solved in polynomial time. Unfortunately, the problem properties that contribute to its complexity are not clearly known in most cases. This information worth discovering since it is often the key to design an efficient solving algorithm. Notions like the breaking in subproblems or the number of solutions of the problem have important consequences on its complexity. The complexity of a problem is also obviously directly linked to the efficiency of the algorithms applied to solve it and to the capacity of these algorithms to exploit the properties of this problem (Jones, 1995).

The first property influencing the complexity is the number of maxima of F (Kallel et al., 2001). For example, when $|M|$ is large compared with $|C|$, an algorithm that evaluates points of C randomly and which has therefore an average complexity of $|M|/|C|$, is very efficient. In the extreme, the problem of "needle-in-a-haystack" (Forrest & Mitchell, 1993; Goldberg, 1989), in which $F(X)$ is identical for 2^n-1 points of C and has a higher value for the last point, is a problem of maximal complexity since no property can be exploited. A

landscape property that is more precise than the number of maxima is the number and the size of the basins of attraction. The higher the size of a basin of attraction of a maximum given an operator o is, the higher the probability of discovering this maximum from a random point of C and using an algorithm based on o is. Thus, the respective size of the basins of attraction of local and global maxima in a given landscape strongly influences the complexity of the exploration of this landscape.

A second property is the epistasie level of a problem. It corresponds to the maximum number of other variables each of the n variables depends on. For example, a problem with zero epistasie is a problem in which all variables are independent[1]. The epistasie is a more general concept than that of decomposability. Indeed, an epistasie level of k does not imply that the problem is decomposable into n/k completely independent sub-problems. The sub-problems can share variables, and there can be, for example, n sub-problems of size k. k dependent variables are called a *block*. Blocks can overlap if they share some variables. A classical problem of size n, of epistasie level k with overlapping is the *NK-landscape* (Kauffman, 1989; Kauffman, 1993). The idea of the NK-landscape, or of its extensions the *NKp-landscape* (Barnett, 1998) and the *NKq-landscape* (Newman & Engelhardt, 1998), is to allow the easy building of problems with scalable difficulty and epistasie level. It has been demonstrated that the problem is NP-complete[2] as soon as k is equal to two or more and when the problem is not decomposable into independent sub-problems (Thompson & Wright, 1996).

Deceptive (or trap) functions are functions that have been defined to be difficult for hill-climber algorithms. Ackley has suggested one of these functions and has shown that it produces an exponential number of local maxima in the landscape (with a Hamming neighborhood) (Ackley, 1987). Deceptive functions have been intensively studied and are often used as benchmark functions to validate exploration algorithms (Goldberg, 2002; Gras et al., 2004; Gras, 2004; Martin et al., 2003; Pelikan, 2002; Sastry et al., 2004; van Nimwegen et al., 1997). They can be viewed as a special case of the problem of the local maxima number. Knowing that local maxima are, by definition, linked to the chosen neighborhood, the notion of deceptive function is also linked to the neighborhood. It is possible to give an informal definition of deceptive function. An entirely ordered landscape can be obtained considering the set of all the possible values of $F(X)$ for all $X \in C$ sorted by growing value. If all the points of M, that is all global maxima, are execluded from this set and assuming C' is the new set, a deceptive function can be informally defined by : a function for which the instantiation of variables x_i of the points of the sub-set of C' points with the higher values[3] is the most different from the instantiation of the variables x_i of M points. In case of a binary function, the notion of "different" corresponds to the Hamming distance. This definition is linked to the Hamming neighborhood but it does not involve the notion of local maximum and is therefore almost independent of the chosen landscape. Then, it is possible to define intrinsically complex functions since this definition of complexity does not directly refer to a

[1] The onemax problem, in which one searches the maximum of $F(X)$, where $F(X)$ is the sum of the value of variables X, is a classical problem with zero epistasie.

[2] Indeed, the problem can be transformed into MAX-2-SAT which is NP-complete.

[3] The size of this sub-set is not specified but the higher it is, the more deceptive the function is.

specific landscape. Among classical deceptive function, we can cite the Ackley $trap_5$ function (Ackley, 1987) or the Mitchell, Forrest and Holland *royal road* function (Mitchell et al., 1992).

2. Genetic algorithms

The complexity of combinatorial optimization problems is hard to define and to measure since numerous properties are involved and that these properties need a large amount of computation to be evaluated. The neighborhood, which produces a specific landscape for a function, is a major component of the complexity. It means that there is a great correlation between the exploration algorithm that is used (and then several neighborhoods) and the function to optimize: a given algorithm is efficient for a given class of function and vice versa a given function has a low complexity if an algorithm using its intrinsic properties is discovered. In this section, we are interested in the study of this link while uniquely considering evolutionary explorative methods (Michalewicz & Fogel, 2000; Baluja, 1996). For a wider study, including all the major exploration methods see (Gras, 2004).

2.1 The cannonical approach

The initiation step of a standard genetic algorithm is the random generation of a population, of a given size Tp of vectors X (the chromosomes). Each vector corresponds to a point of the search space and therefore to a given instantiation of all the variables. The algorithm consists in a repetition of three successive steps: (1) a fitness value equal to $F(X)$ is associated to each vector X; (2) Tp vectors are randomly selected in the population with a probability depending on the fitness value of each vectors; (3) a series of genetic operators is applied stochastically to these vectors and the transformed vectors are then inserted in a new population. The three steps (called generations) are re-applied to this new population of size Tp.

There are a large number of selection operators. The most frequent ones are the fitness proportionate selection, the elitist selection and the tournament selection. They allow the modulation of the selection pressure that is put on the population, that is the importance of the statistical bias towards the best solutions. Genetic operators of step (2) are also very varied. The most classical are the punctual mutation of a gene (the change of the instantiation of a variable) and the uniform one or two points crossovers (the instantiations of a series of variables are exchanged between two vectors). These operators allow the exploration of the search space by transforming one point into another.

Numerous operators are necessary for genetic algorithms, for example the size of the population, the probabilities that are associated with the genetic operators, the selection pressure, etc. A lot of experimental studies have been carried out to evaluate the influence of each parameter. The conclusions show that the influences vary considerably depending on the class of problem that is considered. Other studies focus on the comparison of exploration capacities of genetic algorithms and more specifically of crossover operators with those of multi-start strict hill-climbers (Jones, 1995; Baluja, 1996; Goldberg & Segrest, 1987; Sharpe, 2000). Results show that in most cases, and more specifically in deceptive problems, standard genetic algorithms do not have better and even sometimes worse results results than hill-climber algorithms. However, we present more complete results in section 3, showing in which circumstances genetic algorithms are efficient.

Several theoretical studies have been carried out to understand the behavior of genetic algorithms during the exploration process. They are based on either a modelization by a

Markovian process (Goldberg & Segrest, 1987; Moey C.J., Rowe, 2004), or on the study of the cumulants (average, standard deviation…) of the fitness value of the population of chromosomes during the successive generations (Prugel-Bennett & Rogers, 2001). These works are still limited to simple problems and it hardly seems realistic to apply them to more complex non-zero epistasie problems. The first attempt to analyze the convergence behavior of classical genetic algorithms involving the epistasie concept is linked to the notion of schemes (Holland, 1968). Goldberg (Goldberg, 1989) uses this notion to explain the behavior of classical genetic algorithms[4]. The idea is that the function to be optimized is composed of sub-functions, each of them based on a sub-set of variables that are represented by a scheme. The genetic algorithm selects and combines the best schemes to discover a good solution faster. Goldberg shows that a genetic algorithm handles Tp^3 schemes in parallel and that is why it is efficient to solve such problems.

Thierens et Goldberg have published several works on the evaluation of the capacities of genetic algorithms to solve deceptive problems (Thierens & Goldberg, 1993; Thierens & Goldberg, 1994; Sastry & Goldberg, 2002; Thierens, 1999). When functions that are composed of a juxtaposition of deceptive sub-functions of size k are considered, they have evaluated the time t (number of generations) that is needed for the schemes corresponding to variables of sub-functions to mix correctly and to build the complete solution. They deduce from these evaluations the size of the population that is needed to assure the convergence of the exploration process towards the global maximum. These calculations show that the time of convergence and the size of the population of the genetic algorithm with one or two points crossovers is exponential with the number of dependences and the number of sub-functions. They conclude from these results that classical genetic algorithms are not able to solve these difficult problems. These studies have initiated the development of a new evolutionary approach aiming at discovering dependences between variables and solving the corresponding problem. We present this approach in the next section.

2.2 The statisticall approach

In parallel with the continuous developments of the canonical approach, a new kind of genetic algorithm appeared: probabilistic model-building genetic algorithms (*PMBGA*) (Muhlenbein & Mahnig, 2001; Larranaga & Lozano, 2002). On principle the population of chromosomes is considered as a sample of the search space, and the algorithm builds a probabilistic model of this sample and uses the model to generate a new population. The general algorithm is:

1. $t = 0$
   Generate the initial population $P(t)$
2. Select $S(t)$ a set of promising solutions (chromosomes)
3. Construct M a statistical model of this set
4. Generate $O(t)$ a new set of solutions from this model
5. Create a new population $P(t+1)$ by replacing some solutions from $P(t)$ by solutions from $O(t)$
   $t = t + 1$
6. If the stop criterion is not reached then go to (2)

[4] In this case the used genetic algorithm consists in a fitness proportionate selection, a punctual mutation and a one point crossover.

The first works in this domain are the Bit Based Simulated Crossover (BSC) (Syswerda, 1993), the Population Based Incremental Learning (PBIL) (Baluja, 1994) and the Univariate Marginal Distribution Algorithm (UMDA) (Muhlenbein & Voigt, 1996). The idea of the last one was to replace the recombination step by a global recombination over the complete population. All these algorithms extract some statistics from the population for each variable independently. As a consequence, their performances are very poor (in particular less than the performances of a strict hill-climber) when the epistasis level of the problem is above zero. To get around this difficulty, others PMBGA have been proposed taking into account the probability distribution of more than two variables simultaneously. The factorized distribution algorithm (FDA) (Muhlenbein et al., 1999) for example, is based on the construction of a model *M* of the dependences between variables using a Bayesian network (Frez, 1998). It turns out that this algorithm is very efficient if it is fed with a sample of adequate size (function of *k* as we present later) and if the problem decomposition is already known (the network structure is an input of the algorithm). This efficiency is not surprising since when a problem is decomposable in independent sub-problems its complexity is considerably lowered. The main difficulty lies precisely in the discovery of these dependences for further exploitation.

The approach called linkage learning has been devised to treat this difficulty (Harik, 1997). We focus on a sub-class called *probabilistic linkage learning* (Pelikan, 2002) or *learning a Bayesian network* (LFDA) (Muhlenbein & Mahnig, 2001). The principle is to learn a bayesian network representing the dependences between the problem variables during the exploration of the search space by the genetic algorithm. It is the approach chosen for the *Bayesian Optimization Algorithm* (BOA) (Pelikan et al., 1999). Learning a Bayesian network implies two things: to discover the network structure, that is, to discover the dependences, and to determine the conditional probabilities associated to the edges. The conditional probabilities are easily estimated from the observed frequencies in the sample (the population) of the simultaneous occurrences of each instance of each variable. Learning the network is much more complex since it involves determining which one of 2^{n^2} possible networks best represents the data (the sample). It is a NP-complete problem (Chickering et al., 1997) and a non-exact heuristic method must be used to build the network. Therefore it is not possible to guarantee the quality of the network obtained. BOA uses a greedy strategy to build the network by adding edges successively, beginning with the empty network. The algorithm needs a metric to measure the quality of the network (how good the network to model the data is) at each step of the construction. Then the algorithm is: (1) Initialize the network without any edge; (2) Between the basic operations (add, remove or inverse) for all the possible edges, choose the operation which best increases the quality of the network; (3) If the stop criterion is not reached, go to (2). The metric used in BOA is the *Bayesian Information Criterion* (BIC) (Schwarz, 1978). There is no perfect measure of network quality. The BIC measure is sensitive to noisy data and can overestimate the number of dependences. The Bayesian network learning process is a non-exact heuristic based on a measure likely to produce errors, therefore it is not possible to guarantee that the final network is composed of all and only all the real dependences of the problem.

The global complexity is $O(k \cdot 2^k \cdot n^2 \cdot Tp + k \cdot n^3)$ (Sastry et al., 2004). It implies that the choice of the edge to add at each step is not made through computing the gain in measure for the global network for each possible edge but is only made through the value of gain from the empty network to the network with one edge (the values are pre-calculated

in $O(n^2 \cdot Tp)$). In this case, the choices are made only according to dependences between couples of variables and are therefore likely to be biased by deceptive problems in which dependences can be discovered only when *k* variables are observed simultaneously. Then, the Bayesian network is used to generate the new population. The total complexity of the generation of a new population is in $O(k \cdot n \cdot Tp)$.

Pelikan and Goldberg (Goldberg , 2002; Pelikan, 2002) propose a convergence proof for BOA based on the calculation of the size of the population. The size *Tp* of the population can be deduced: $O(2^k \cdot n^{1.05})$. However, these calculations are performed with several approximations. The value of $F_i(X)$[5] is supposed to have a normal distribution in the population. The probabilistic model is supposed to be exact, that is being a perfect model of the population statistics, but it is built with a drastic heuristic (greedy) algorithm. The quality measure is the BIC measure, but the BD measure is used in the complexity calculation. Finally, and this is also true for the calculation done on the canonical genetic algorithm, the mutation operator is never taken into account[6]. However, from these calculations, they give the convergence proof of BOA in $O(\sqrt{n})$ generations. Associated with the values reported about the building of the Bayesian network, the global complexity of BOA can be deduced: $O(k \cdot n^{7/2} \cdot (2^{2k}+1))$. Pelikan and Goldberg have proposed an extension of BOA called hierarchical bayesian optimization algorithm (hBOA) (Pelikan et al., 2001). Its goal is to solve multi-level deceptive problems in which there are blocks of blocks, each level having its own fitness sub-function. hBOA principle is similar to that of BOA except that hBOA uses decision graphs to store the frequency table of the dependent variables and uses ecological niches system within its populations.

Recent works bring several improvements to hBOA and others PMBGA (Sastry et al.., 2004a). A first one bears on the *mutation Building-Block-Wise* concept (Sastry et al.., 2004b; Sastry & Goldberg., 2004a; Sastry & Goldberg., 2004b). Its principle consists in the discovery of blocks from the probabilistic model of the population. Then, it uses a mutation operator acting as a strict hill-climber on the sub-space of all the possible instantiations of each block. Therefore, a total of $m \cdot 2^k$ evaluations are needed if the problem is composed of *m* blocks of size *k*. They have extend recently this approach to the BOA method (Lima et al., 2006). This approach is in fact strictly equivalent to the principle of dividing a problem into independent sub-problems and to solve each of them independently. This is, of course, much faster than solving directly the global problem. For that purpose, there is still a need for a population that should be large enough to discover the right model and a guarantee that the problem is decomposable into independent sub-problems. Another work is about the notion of *fitness inheritance* (Pelikan & Sastry, 2004). Here, the idea is to avoid explicit computing of the fitness of each new explored point of the search space using estimation. This estimation is done by inheritance of information from the previous populations. The probabilistic model is used to determine what the schemes are, then, for each of them, their contribution to the total fitness is estimated. This estimation is realized by studying the

[5] Which is the value of the sub-function corresponding to the block of *i*.

[6] The mutation operator does not explicitly exist in BOA but it has its equivalent. The process of generation of solution from the model is able to generate block instantiations that are not present in the initial population.

fitness distribution, in the previous population, of the solutions containing these schemes. The estimated values are used to calculate the fitness for a fraction of the new population. In spite of the loss in precision due to the estimation, the global improvement in total computing time can be significant.

3. Efficiency of genetic algorithms

3.1 Definition of the Problem

We have carried out a study on the effects of the different properties presented in section 1.2 on the efficiency of several evolutionary algorithms. We focus on the epistasie level, the relative position of variables in the representation of the problem and deceptive functions. We have studied the classical genetic algorithm behavior on these problems, of hBOA and of several simple PMBGAs using different probabilistic models. The functions that we have used in our study are all built according to a simple model: the sum of independent sub-functions deceptive or not. Thus, the fitness function F is defined by :

$$F(X) = \sum_{i=1}^{m} F_i(X_i) \tag{1}$$

with m the number of sub-functions F_i of size $k = n / m$ and X_i the k variables of X involved in the sub-function F_i. Therefore, k corresponds to the epistasie level of F. There are two kinds of F_i sub-functions:

$$F_i(X_i) = \sum_{x_i \in X_i} x_i \tag{2}$$

if F_i is not deceptive and :

$$\begin{cases} F_i^t(X_i) = 1 - \dfrac{\left(\sum_{x_i \in X_i} x_i\right) + 1}{k} & if \sum_{x_i \in X_i} x_i < k \\ F_i^t(X_i) = 1 & if \sum_{x_i \in X_i} x_i = k \end{cases} \tag{3}$$

if F_i^t is deceptive.

The position of variables of each sub-function in the complete representation of a solution (chromosome) can be in two different configurations: adjacent and non-adjacent. In the first configuration the numbering of variables of Fi is as follows:

$$X_i = \left\{x_{(i-1)\cdot k+1}, x_{(i-1)\cdot k+2}, \ldots, x_{(i-1)\cdot k+k}\right\} \quad with\ i \in \{1, \ldots, m\} \tag{4}$$

In the non-adjacent configuration, the numbering of variables of F_i is as follows:

$$X_i = \left\{x_i, x_{i+m}, \ldots, x_{i+(k-1)\cdot m}\right\} \quad with\ i \in \{1, \ldots, m\} \tag{5}$$

All tests have been realized with Athlon 1.5 Ghz processors.

3.2 The canonical genetic algorithm

The genetic algorithm we have used contains a punctual mutation with a probability of occurrence of 0.05, a one point crossover with a probability of occurrence of 0.45 and a two points crossover with a probability of occurrence of 0.5. One operator at most is applied to a given chromosome. It uses also a tournament selection of size 10. The value of these parameters has few consequences on the results. The first test function is composed of 120 variables ($|X|$ =120) and of $m = 120/k$ deceptive sub-functions F_i^t of size k. The values of k vary from 4 to 12. We use the adjacent configuration. The size of the population is calculated with the formula proposed by Pelikan and Goldberg, $Tp = 2^k \cdot n^{1.05}$, in order to compare the performances of each algorithm in the same condition. For each k value, the presented results correspond to the number of run, over 60 performed each time, for which the global maximum has been reached in less than 100 generations. Table 1 gives the number of failures over 60 runs.

k	4	6	8	10	12
Nb failures	58	7	0	0	0

Table. 1. Number of failures of the classical genetic algorithm among 60 runs for each k value and for the adjacent deceptive function.

The results that are obtained in term of computation time and number of generations are given in figure 1. Because of the huge memory that is needed to store the population for the higher k value (Tp = 622 592 for k = 12), we limited the size of the population to Tp = 480 000. We observe that the classical genetic algorithm can easily discover the right solution even in problems that are considered to be very difficult ones (n = 120, $k \geq 10$) if the population is large enough. It seems also that for high epistasie level the size of the population that is necessary to find the maximum is smaller than the size that is required by hBOA. However the population size of 2400 that is used for k = 4 is too small to discover the maximum in most cases.

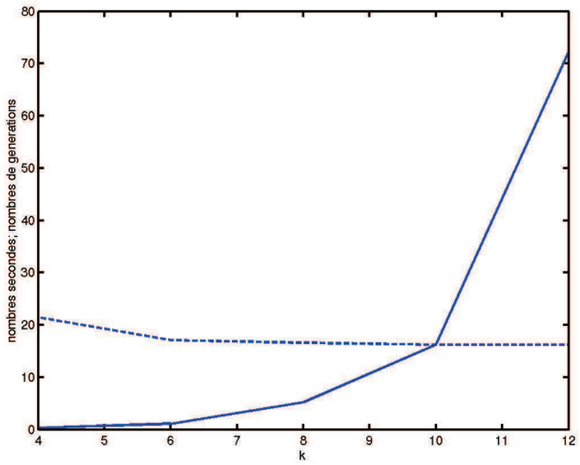


Fig. 1. Average time in seconds (plain line) and average number of generations (dot line) to discover the global maximum depending on the size k of the sub-functions (epistasie level).

The second test function is the same as the first one but with a non-adjacent configuration. The results that are obtained are very different from those of the first function. Indeed, the classical genetic algorithm is unable to discover the solution (100% fails) for k values, even if the population size is much more important than the necessary size for hBOA. For example, when k = 4 the population size should be 2432 and we have used a size of 480 000! It can be deduced that the operators that are used to explore the search space (mutation and crossover) are unsuited to the structure of the problem. They produce a neighborhood that makes the landscape particularly hard to explore. Crossovers destroy every block, or parts of blocks, that were previously built since the operator inevitably separates the variables of a given block. Thus, two neighboring points of the landscape, that is distant of one operator application, are totally different considering the function to optimize. On the other hand, in the adjacent configuration, crossovers are optimal neighborhood operators since they preserve and mix most of the blocks. That explains why classical genetic algorithm performances are extremely good for this problem. In general, it can be concluded that, if the problem has unknown dependences, the configurations used are very likely to be non-adjacent and therefore the problem is unsolvable for classical genetic algorithm. Solvable problems for classical genetic algorithm are those that have no or few dependences, or those for which a deep knowledge about the problem structure can be used to design specifically adapted and efficient operators. (a) The manuscript must be written in English, (b) use common technical terms, (c) avoid abbreviations, don't try to create new English words, (d) spelling: Follow Merriam Webster´s Collegiate Dictionary, Longman or Oxford Dictionaries.

3.2 The hBOA algorithm

The population size is computed with $Tp = 2^k \cdot n^{1.05}$ and the maximum number of generations is 100. All the other parameters are hBOA default parameters. The first test function is composed of 100 or 96 variables and of $m = 100/k$ or $m = 96/k$ (in order that m is an integer) deceptive sub-functions F_i^t of size k. The k value varies from 4 to 12 and the variables are in an adjacent configuration. The running time of hBOA is much higher than the one of the classical genetic algorithm, so only 12 runs have been performed for each problem. The results vary much more, so each result is detailed independently.

1. n = 100, k = 4, Tp = 2 016. The global maximum (25) has been reached for each run with an average number of generations of 25 and an average running time of 20 seconds.
2. n = 96, k = 6, Tp = 7 720. The global maximum (16) has never been reached during the 12 runs. The average value is 15.7, which is much higher than the "trap" value (14.4) and close to the global maximum. It is reasonable to suppose that the maximum should have been reached with a few more generations. The average running time was 13 minutes.
3. n = 96, k = 8, Tp = 30 720. The global maximum (12) has never been reached during the 12 runs. The value that has been reached was 10.8 for each run, which is the "trap" value. It is reasonable to suppose that the exploration process was trapped for a long time in a local maximum and was not able to reach the global maximum. The average running time was 1 hour and 53 minutes. A second experimentation (12 runs) has been done with the same function but with a double size population (Tp = 61 440). The

average running time was 4 hour and 50 minutes. The global maximum has never been reached during the 12 runs and the "trap" value has been slightly overtopped with a value of 10.9 each time.

4. n = 100, k = 10, Tp = 129 000. The global maximum (10) has never been reached during the 12 runs. The value that has been reached was 9.1 for each run, which is the "trap" value. It is reasonable to suppose that the exploration process was trapped for a long time in a local maximum and was not able to reach the global maximum. The average running time was 12 hours 40 minutes.
5. n = 96, k = 12, Tp = 491 520. The average running time was 61 hours, so we have only done 6 runs. The global maximum (8) has never been reached during the 6 runs. The value that has been reached was 7.4, which is slightly above the "trap" value (7.2). It reasonable to suppose that the process was not definitively trapped in a local maximum and that the result should have been slightly improved with more generations.

One can already observe that, contrary to what has been published, in each situation in which it is supposed to be efficient, hBOA cannot discover the global maximum using the parameters established in the convergence proofs. hBOA results, presented in the literature, have always been obtained with function containing few dependences (maximum 6 or 8). The running time became prohibitive when the level of epistasie increased and there is no guarantee that the global maximum is discovered or that local maximum traps can be escaped. Heuristics that are used during the construction of the Bayesian network and not evaluated in the convergence proofs possibly have dramatic consequences on the efficiency of the exploration process.

During our tests, we have observed a particular sensitiveness of hBOA to the structure of the fitness function. To our knowledge, this phenomenon is not described in the literature and does not appear in the convergence calculation of Pelikan and Goldberg. We have therefore carried out another series of tests (12 in each case) in order to have a better understanding of this phenomenon. In the previous tests, the function F_i^t for n=96, k=8 and Tp=30 720 was the function presented in figure 2. We have done several tests in which we change this function by those which are presented in figure 3 to 5. These new functions correspond to change in the value of the difference between the global maximum value and the "trap" value of the function. Once we have changed this function but with the same n=96, k=8 and Tp=30 720, very different results are obtained. For the function of the figure 3, the same results were obtained, that is the global maximum was never reached and the process was trapped in a local maximum corresponding to the "trap" value 10.8. For the function, of the figure 4, five global maxima were obtained for twelve runs with an average 56 generations. For the seven failed runs the average score was 11.8, which is close to the global maximum. For the function, of the figure 5, the global maximum was obtained for the twelve runs with an average 15 generations.

During our tests, we have observed a particular sensitiveness of hBOA to the structure of the fitness function. To our knowledge, this phenomenon is not described in the literature and does not appear in the convergence calculation of Pelikan and Goldberg. We have therefore carried out another series of tests in order to have a better understanding of this phenomenon. In the previous tests, the function F_i^t for n = 96, k = 8 and Tp = 30 720 was the following the one of figure 2.

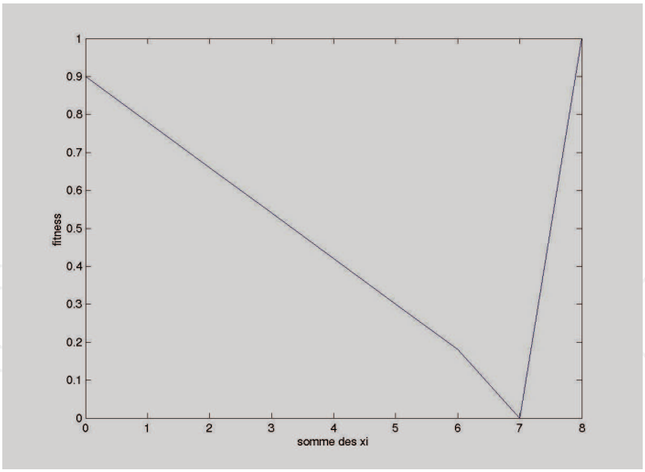


Fig. 2. The initial trap function

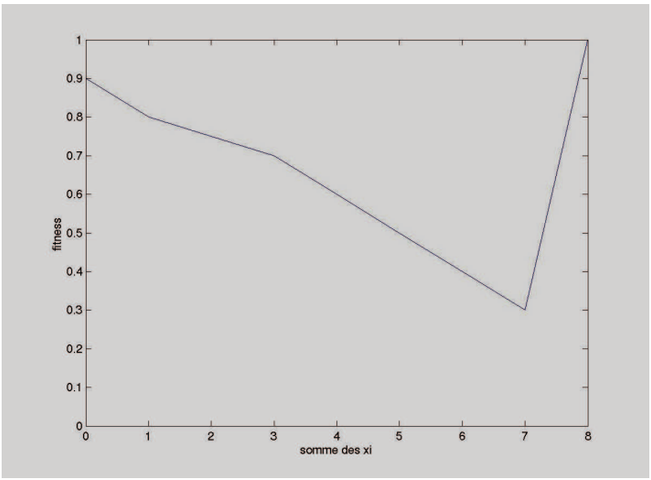


Fig. 3. The second trap function

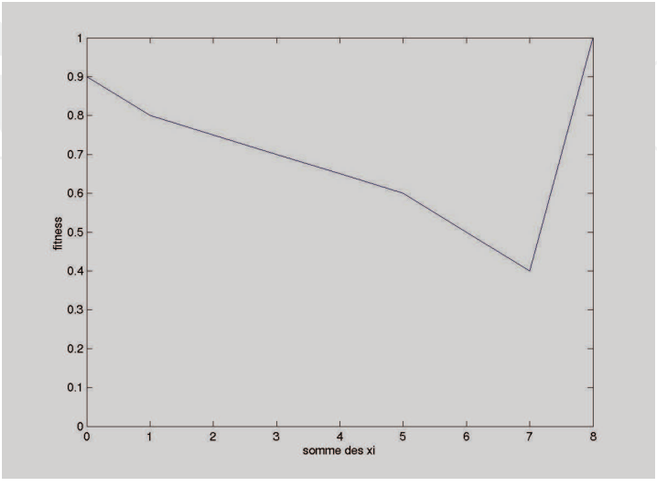


Fig. 4. The third trap function

Even though these results need to be confirmed with more of experiments, the phenomenon seems to be relatively clear. It seems that a too large difference value between the global maximum and the solutions the most similar to the global maximum (in Hamming distance) makes the probabilistic model harder to construct and disables the discovery of the right dependences. Even though the heuristic for the bayesian network of hBOA construction has not been described in detail, results that are obtained tend to show that it rests in, at least in part, univariate frequencies. Indeed, the previous series of functions increasingly favor the selection of vectors that are composed of a majority of '1'. The global maximum is the vector entirely composed of '1'. Since the functions that favor the selection of '1' also favor the construction of a model that predicts '1' at each position, these functions are logically easier to solve. Another possible explanation for this phenomenon can be linked to the system that is used in hBOA to generate the population from the model. As it is emphasized in section 2.2, Pelikan and Goldberg's algorithm that generates the new population is not clear as it does not give any clue about the initialization process. It is therefore possible that this initialization is done from uni or bivariate frequencies and consequently introduces a bias in the population in favor of the values that are instantiated in the previous generation.

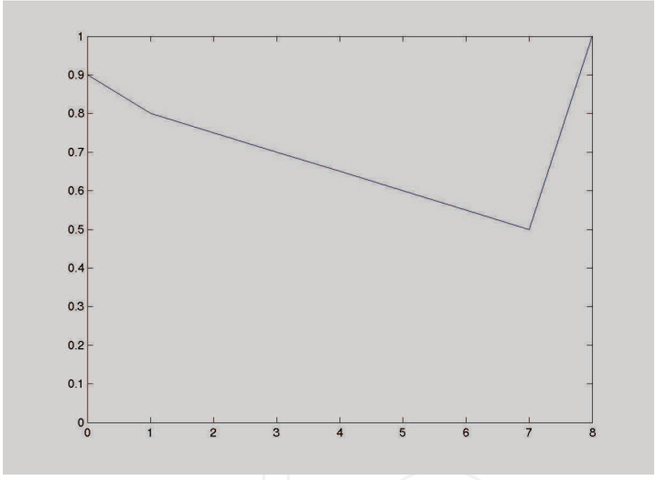


Fig. 5. The fourth trap function.

All these experimentations have been realized with the adjacent configuration. We have done the same experimentations with the non-adjacent configuration and the results are practically identical. That means that hBOA is not disrupted by the relative position of variables that are involved in a same block. It is to be expected because the position of variable in the chromosome/vector in the hBOA concept is not taken into account. It is an important result, which has not been published, because, as we have seen in the previous section, the classical genetic algorithm is totally unable to escape from the local maximum traps. The result demonstrates that hBOA is able to discover and exploit the problem dependences in a complex situation if the epistasie level is reasonable and if the structure of the function does not prevent the heuristics of the probabilistic model building and that of population generation from guiding the exploration towards the global maximum.

4. Tables and figures

We have tried to understand what is possible to discover if only one piece of information about the problem is exploited. We have studied other probabilistic models to discover the dependences and designed new operators based on these models. We have limited our model to the representation of dependences of maximum size two and we have designed an algorithm, a variant of a classical genetic algorithm with a tournament selection, using specific operators based on our model. Then, we have tested our approach on problems with epistasie level above two (the problems presented in section 3.1 and 3.2). The principle of our algorithm is as follows. The frequencies of occurrence of all the couples of variables instantiations that are encountered in the population are computed at each generation. Then, we use these statistics to measure correlations between instantiations of variables. Operators that are based on these measurements are used to generate new solutions for the new population.

We have compared different measures to study their capacities to discover the couples of dependent variables among the blocks of size above two (in our test problems, the sizes of the blocks range from 4 to 12). We have used the following measures:

- The conjoint frequencies of two variables instantiations:

$$f(x_i = \phi_1, x_j = \phi_2)\ \forall i, j \in \{1, \cdots, n\}\ and\ \phi_1, \phi_2 \in \{0,1\} \tag{6}$$

- The conditional probability:

$$P(x_i = \phi_1 \mid x_j)\ \forall i, j \in \{1, \ldots, n\}\ and\ \phi_1 \in \{0,1\} \tag{7}$$

- The mutual information $M(x_i, x_j)$:

$$M(x_i, x_j) = \sum_{x_i=\phi_1, x_j=\phi_2} f_{x=\phi_i x_j=\phi_2} \cdot \log \frac{f_{x_i=\phi x_j=\phi_2}}{f_{x_i=\phi_1} \cdot f_{x_j=\phi_2}} \quad \forall i, j \in \{1, \cdots, n\} \tag{8}$$
$$and\ \phi_1, \phi_2 \in \{0,1\}$$

- The statistical implication (Blanchard et al., 2003; Kuntz et al., 2002)

$$q(x_i = \phi_1, x_j = \overline{\phi_2}) = \frac{n_{x_i=\phi_1 \cap x_j=\overline{\phi_2}} - \frac{n_{x_i=\phi_1} \times n_{x_j=\overline{\phi_2}}}{Tp}}{\sqrt{\frac{n_{x_i=\phi_1} \times n_{x_j=\overline{\phi_2}}}{Tp}}} \quad \forall i, j \in \{1, \cdots, n\} \tag{9}$$

with $n_{x_i=\phi_1}$ et $n_{x_j=\overline{\phi_2}}$ the number of times where x_i is instantiated to ϕ_1 and x_j is instantiated to non ϕ_2 respectively in the whole sample of size Tp and:

$$\varphi\left(x_i=\phi_1, x_j=\phi_2\right)=\frac{1}{\sqrt{2\pi}}\int_{q(x_i=\phi_1, x_j=\overline{\phi_2})}^{\infty} e^{-\frac{t^2}{2}}dt \tag{10}$$

This last measure is a value of the degree of implication between two variable instantiations. It has been shown as being much more sensitive than the conditional probability.

These tests have shown that none of these four measures is able to detect the instantiations of two variables corresponding to their instantiation in their own block in the global maximum. For example, in the case of blocks of size 8, if the variables x_1 to x_8 belong to the same block and have an instantiation of '1' in the global maximum and an instantiation of '0' in the local deceptive maximum, the four measures detect the strongest dependences for couples of variables instantiated to '00' and the weakest for those instantiated to '11'. Thus, we confirm that it seems to be impossible to discover the right dependences when all the variables of a same block are not considered together (in the example, it would be necessary to measure dependences between the 8 variables simultaneously). When dependences of a problem are not known, it is therefore necessary to discover them completely to find the global maximum. The number of combination corresponding to the total number of possible dependences is the number of partitions of *n* variables and is therefore exponential with *n*. Consequently, either a heuristic is needed, for example the one that is used in hBOA for the construction of the bayesian network, or some information known about the problem structure has to be used to determine the dependences and then solve the problem (for example, sort variables so that they are in adjacent configuration).

In this perspective, we have chosen to use an information about the problem structure, i.e. the fact that the sub-functions are deceptive, to try to build an efficient algorithm for theses problems. We use a PMBGA based on the four previously defined measures. We have designed an extremely simple operator to generate the new population. Knowing that the sub-functions to be optimized are deceptive, we have chosen to favour the less frequent instantiations in the population after a tournament selection. Indeed, with deceptive sub-functions, instantiations corresponding to or similar to those of the deceptive local maxima are present in the population and selected with higher probabilities. Consequently, instantiations corresponding to or similar to those of the global maxima are largely under represented in the sample population. Therefore, our operator consists in the random selection of two variables x_i and x_j, then in choosing among the four possible instantiations of these two variables the one, $\phi_i\phi_j$, with the lowest value for the measure (that is the one with the lowest correlation). Finally, for the instantiation ϕ_i of the variable x_i, the instantiation of all other variables x_k are chosen in the following way:

$$choice(x_k)=\arg\min_{\phi_k} measure(x_i=\phi_i, x_k=\phi_k) \qquad \forall k\neq i \tag{11}$$

with $measure(x_i=\phi_i, x_k=\phi_k)$ one of the four measures previously defined. A new solution is then built from these instantiations. We have used this algorithm for all the problems that are presented in section 3.1. We have observed that, using the same

population size than the one that is calculated for hBOA[7], the global maximum is found in only one generation (that is, the computation of the first random population and its fitness evaluation, the tournament selection, the computation of the value of one of the four measures and the construction of one new solution) for all the tested problems, even those of size 120 with blocks of size 12 ! The optimal solution is therefore discovered in few minutes (few seconds for the smallest problem and 5 minutes for the largest ones, the global complexity is in $O(n^2)$) for problems for which hBOA does not find the solution in several days of computation! Thus, with no information about the dependences and their number, but uniquely using the information that the function is deceptive, we have designed a strategy that discovers the solution in 100% of the cases in few minutes even for very complex problems.

To establish the limits of our approach, we have constructed a problem composed of half deceptive sub-functions and half non-deceptive sub-functions. In this case, our approach is totally unable to discover the solution. We have modified our method to handle such mixed problems. We have used several operators: classical mutation and crossover, our previously defined operator, and several operators doing mixed instantiation choices between high and low value of the measures (that is choosing very highly or very lowly correlated instantiations). Even with these new operators our method fails to discover the optimal solution. It can be easily understood because the discovery of the variables that participate to deceptive sub-functions and those that participate to non-deceptive sub-functions is it-self an NP-complete problem. We have verified that those kinds of mixed problem do not perturb hBOA. Indeed, hBOA reaches the same performances with this kind of function as with complete deceptive function in section 3.2.

5. Conclusion

A large number of properties are involved in the intrinsic complexity of a global combinatorial optimization problem. We have focused our studies on two particularly important properties: epistasis and deceptiveness. We first show that the canonical genetic algorithm is unable to solve problems (even simple ones) with epistasie when no information is available on the nature of the dependencies. We then propose to distinguish two classes of strategies: (1) those based on an algorithm that is dedicated to a unique problem using expert knowledge on the structure of this problem - we call it the expert strategy - and (2) those based on the discovery, through modeling a bias sample of the search space, of the structure of the function to be optimized to determine a pertinent neighborhood for exploration - we call it the automated strategy. We study the capability of these two strategies using one representative of each class: hBOA for the automated class and a very simple algorithm, exploiting the unique knowledge that the problem is deceptive, for the expert class. Depending on the nature of the problem, either one or the other of these classes can be efficient. If the first one is highly dependent to the information exploited and therefore to the expert's knowledge, the second one is limited if the number of

[7] We have as well noticed that these results are still true with population of size 2 or 4 times smaller.

dependences between the problem variables is high. However we have shown that hBOA can really discover dependences even in non-adjacent and mixed configurations. We have also shown how the use of partial information on the problem structure can lead to the definition of a highly efficient algorithm, solving very complex problems in a few minutes.
Future work should focus on the study of new heuristic strategies building an informative probabilistic model of the problem and incorporating new probabilistic measures in this model and on how they can be coupled with an expert strategy. Such approaches should be applied to real complex problems, in particular in bioinformatics (Hernandez et al., 2008; Armañanzas et al., 2008), to assess their efficiency to solve real problems and to examine what new knowledge about the problem itself the probabilistic model is able to reveal.

6. References

Ackley, D.H. (1887). An empirical study of bit vector function optimization, In *Genetic Algorithms and Simulated Annealing*. pp 170-204, Morgan Kaufmann

Armañanzas, R.; Inza, I.; Santana, R.; Saeys, Y.; Flores, J.L.; Lozano, J.A.; Van de Peer, Y.; Blanco, R.; Robles, V.; Bielza, C. & Larrañaga P. (2008). A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 1, 6

Baluja, S. (1994). Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. CMU-CS-94-163, Carnegie Mellon University

Baluja, S. (1996). An Empirical Comparison of Seven Iterative and Evolutionary Function Optimization Heuristics. *Proceedings of the Eleventh International Conference on Systems Engineering*

Barnett, L. (1998). Ruggedness and neutrality - the NKp family of fitness landscapes, *Proceedings of the sixth international conference on artificial life*, pp 18-27

Blanchard, J.; Kuntz, P.; Guillet, F. & Gras, R. (2003). Implication intensity: from the basic statistical definition to the entropic version. In *Statistical data Mining and Knowledge Discovery*. Chapman H. (Ed.), Washington, pp 473-485

Chickering, D.M.; Heckerman, D. & Meek, C. (1997). Learning Bayesian Network is NP-Hard, MSR-TR-97-07. Redmond, Microsoft Research

Forrest, S. & Mitchell, M. (1993). Relative building-block fitness and building-block hypothesis, In *Foundations of Genetic Algorithms 2*, pp 109-126, Whitley L.D. (Ed.), Morgan Kaufmann

Frez, B.J. (1998). *Graphical Models for Machine Learning and Digital Communication*, MIT Press

Garey, M.R. & Johnson, D.S. (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman

Goldberg, D.E. & Segrest, P. (1987). Finite Markov chain analysis of genetic algorithms, *Proceedings of the Second International Conference on Genetic Algorithms,* Grefenstette, J.J. (Ed.)

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

Goldberg, D.E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers

Gras, R.; Hernandez, D.; Hernandez, P.; Zangger, N.; Mescam, Y.; Frey, J.; Martin, O.; Nicolas, J. & Appel, R.D. (2004). Cooperative metaheuristics for exploring proteomic data, In *Artificial Intelligence Methods and Tools for Systems Biology*. Dubitzky, W.; Azuaje, F. (Ed.), Springer

Gras, R. (2004). Structure des espaces de recherche, complexité des algorithmes d'optimisation combinatoire stochastique et application à la bioinformatique, *Habilitation à diriger les recherches*, University of Rennes, Thesis/Dissertation

Harik, G.R. (1997). Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms, *PhD Thesis,* University of Michigan

Hernandez, D.; Gras, R. & Appel, R. (2008). Neighborhood Functions and Hill-Climbing Strategies dedicated to the Generalized Ungapped Local Multiple Alignment, *European Journal of Operational Research*, 185(3), pp 1276-1284

Holland, J.H. (1968). Hierarchical descriptions of universal spaces and adaptive systems, Report, University of Michigan

Jones, T. (1995). Evolutionary Algorithms, Fitness Landscapes and Search, *PhD Thesis*, University of New Mexico.

Kallel, L.; Naudts, B. & Reeves, C.R. (2001). Properties of Fitness Functions and Search Landscape. In *Theoretical Aspects of Evolutionary Computing*, pp 175-206, Kallel L., Naudts B., Rogers A. (Ed.), Springer

Kauffman, S.A. (1989). Adaptation on rugged fitness landscapes, *Lecture in Sciences of Complexity*, 1, pp 527-618

Kauffman, S.A. (1993). The Origins of Order. In *Self-Organization and selection in Evolution*, New York: Oxford University Press

Kuntz, P.; Gras, R. & Blanchard, L. (2002). Discovering Extend Rules with Implicative Hierarchies. *Proceedings of the conference on the new frontiers of statistical data mining and knowledge discovery*. Knoxville Tennesse

Larranaga, P. & Lozano, J.A. (2002). *Estimation of Distribution Algorithms: A new tool for evolutionary computation*, Kluwer Academic Publishers

Lima, C.; Pelikan, M.; Sastry, K.; Butz, M. & Goldberg, D.E. (2006). Structural neighborhoods for local search in Bayesian optimization algorithm, *proceeding of the Parallel Problem Solving from Nature*, pp 232-241

Martin, O.; Gras, R.; Hernandez, D. & Appel, R.D. (2003). Optimizing Genetic Algorithms Using Self-Adaptation And Explored Space Modelization, pp 291-294, *proceedings of 5th International Workshop on Frontiers in Evolutionary Algorithms*, North Carolina

Michalewicz, Z. & Fogel, D. (2000). *How to Solve It: Modern Heuristics*. Springer-Verlag

Mitchell, M.; Forrest, S. & Holland J.H. (1992). The royal road for genetic algorithms: Fitness landscapes and GA performance. *Proceedings of the First European Conference on Artificial Life,* pp 245-254, Varela, F.J. & Bourgine P. (Ed.), MIT Press

Moey, C.J. & Rowe, J.E. (2004). Population aggregation based on fitness. *Natural Computing*, 3, pp 5-19

Muhlenbein, H. & Voigt, H.M. (1996). Gene pool recombination in genetic algorithms, In *Metaheuritics: Theory and Applications*, pp 53-62, Kelly, J.P. & Osman, I.H. (Ed.), Kluwer Academic

Muhlenbein, H.; Mahnig, T. & Rodriguez. A.O. (1999). Schemata, distributions and graphical models in evolutionary optimization, *Journal of Heuristics*, 5, pp 215-47

Muhlenbein, H. & Mahnig, T. (2001). Evolutionary Algoritms: From Recombination to Search Distributions, In *Theoretical Aspects of Evolutionary Computing*, pp 135-173, Kallel, L.; Naudts, B. & Rogers, A. (Ed.), Springer

Newman, M. & Engelhardt R. (1998). Effect of neutral selection on the evolution of molecular species, *Proc. R. Soc. London*, 256, pp 1333-1338

Papadimitriou, C.H. & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*, Dover

Pelikan, M.; Goldberg, D.E. & Cantu-Paz, E. (1999). The Bayesian Optimization Algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference I*, pp 525-532, San Francisco, CA, Morgan Kaufmann

Pelikan, M.; Goldberg, D.E. & Cantu-Paz, E. (2000). Linkage Problem, Distribution Estimation, and Bayesian Networks, *Evolutionary Computation*, 8, pp 311-340

Pelikan, M. (2002). Bayesian Optimization Algorithm: From Single Level to Hierarchy, *PhD thesis*, University of Illinois

Pelikan, M. & Sastry, K. (2004). Fitness Inheritance in the Baysian Optimization Algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp 48-59

Prugel-Bennett, A. & Rogers, A. (2001). Modelling Genetic Algorithm Dynamics. In *Theoretical Aspects of Evolutionary Computing*, pp 59-85

Sastry, K. & Goldberg, D.E. (2002). Analysis of Mixing in Genetic Algorithms: A survey, 2002017 IlliGAL report

Sastry, K.; Goldberg, D.E. & Pelikan, M. (2004a). Efficiency Enhancement of Probabilistic Model Building Genetic Algorithms, *proceeding of Genetic and Evolutionary Computation Conference*

Sastry, K.; Pelikan, M. & Goldberg, D.E. (2004b). Efficiency Enhancement of Genetic Algorithms via building-block-wise fitness estimation, *proceeding of IEEE Conference on Evolutionary Computation,* pp 720 -727

Sastry, K. & Goldberg, D.E. (2004a). Designing Competent Mutation Operators via Probabilistic Model Building of Neighborhoods, *Proceedings of the Genetic and Evolutionary Computation Conference*

Sastry, K. & Goldberg, D.E. (2004b). Let's Get Ready to Rumble: Crossover Versus Mutation Head to Head, *Proceedings of the Genetic and Evolutionary Computation Conference*

Schwarz, G. (1978). Estimating the dimension of a model, *The Annals of Statis*tics, 6, pp 461-464

Sharpe, O.J. (2000). Towards a Rational Methodology for Using Evolutionary Search Algorithms, *PhD Thesis*, University of Sussex

Syswerda, G. (1993). Simulated Crossover in genetic algorithms, In *Foundations of Genetic Algorithms 2*, pp 239-255, Whitley, L.D. (Ed.), Morgan Kaufmann

Thierens, D. & Goldberg, D.E. (1993). Mixing in genetic algorithms, *Proceedings of the International Conference on Genetic Algorithms*, pp 38-45

Thierens, D. & Goldberg, D.E. (1994). Convergence models of genetic algorithm selection schemes. *Proceedings of Parallel Problem Solving from Nature*, pp 116-121

Thierens, D. (1999). Scalability Problems of Simple Genetic Algorithms. *Evolutionary Computation*, 7, pp 331-352

Thompson, R.K. & Wright, A.H. (1996). Additively decomposable fitness functions, University of Montana, Computer Science department, Report

van Nimwegen, E.; Crutchfield, J.P. & Mitchell M. (1997). Finite populations induce metastability in evolutionary search, *Physics Letters A*, 229, pp 144-150

IntechOpen

IntechOpen



Evolutionary Computation

Edited by Wellington Pinheiro dos Santos

ISBN 978-953-307-008-7
Hard cover, 572 pages
Publisher InTech
Published online 01, October, 2009
Published in print edition October, 2009

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Gras Robin (2009). Study of Genetic Algorithms Behavior for High Epitasis and High Dimensional Deceptive Functions, Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, Available from: http://www.intechopen.com/books/evolutionary-computation/study-of-genetic-algorithms-behavior-for-high-epitasis-and-high-dimensional-deceptive-functions

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen