Exact Computation of the Expectation Curves of the Bit-Flip Mutation using Landscapes Theory

Francisco Chicano
University of Málaga, Spain
chicano@lcc.uma.es

Enrique Alba
University of Málaga, Spain
eat@lcc.uma.es

ABSTRACT

Bit-flip mutation is a common operation when a genetic algorithm is applied to solve a problem with binary representation. We use in this paper some results of landscapes theory and Krawtchouk polynomials to exactly compute the expected value of the fitness of a mutated solution. We prove that this expectation is a polynomial in p, the probability of flipping a single bit. We analyze these polynomials and propose some applications of the obtained theoretical results.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Theory, Algorithms

Keywords

Fitness Landscapes, Elementary Landscapes, Bit-flip Mutation

1. INTRODUCTION

Landscapes theory focuses on the analysis of the structure of the search space that is induced by the combined influences of the objective function of the optimization problem and the neighborhood operator [11]. This theory has applications not only in evolutionary computation [17] but also in Chemistry [12], Biology [16], and Physics [5].

A *landscape* for a combinatorial optimization problem is a triple (X, N, f), where $f : X \mapsto \mathbb{R}$ defines the objective function and the *neighborhood operator* function $N(x)$ generates the set of points reachable from $x \in X$ in a single application of the neighborhood operator. If $y \in N(x)$ then y is a neighbor of x. *Elementary landscapes* are a type of landscape which are of particular interest due to their special properties [17]. They are characterized by the *Grover's wave equation*:

$$\operatorname*{avg}_{y \in N(x)}\{f(y)\} = f(x) + \frac{\lambda}{d}\left(\bar{f} - f(x)\right)$$

where d is the size of the neighborhood, $|N(x)|$, which we assume is the same for all the solutions in the search space, $\bar{f}$ is the average solution evaluation over the entire search space, and λ is a characteristic constant. The wave equation makes it possible to compute the average value of the fitness function f evaluated over all of the neighbors of x using only the value $f(x)$; we denote this average by using $\operatorname{avg}\{f(y)\}_{y \in N(x)}$:

$$\operatorname*{avg}_{y \in N(x)}\{f(y)\} = \frac{1}{|N(x)|}\sum_{y \in N(x)} f(y) \qquad (1)$$

The previous average can be interpreted as the expected value of the objective function when a random neighbor of x is selected using a uniform distribution. This point of view can be used to design new search strategies. In fact, it has been done in recent works like the one of Sutton *et al.* [13] or Lu *et al.* [7]. One limitation of this approach is that only a uniform distribution can be considered when a neighbor solution is selected. However, in practice, this probability is not always the same for all the neighbors. In particular, the bit-flip mutation in genetic algorithms works on binary strings and flips each bit with probability p. This means that it is possible for the bit-flip mutation to reach any solution in the search space with a non-uniform probability distribution.

In this work we focus on the bit-flip mutation and our main goal is to find a closed-form formula for computing the expected value of the fitness of the mutated individual. That is, we want to generalize Grover's wave equation to the case of the bit-flip mutation operator. This generalization has been recently proposed in [15], where an algorithm is given for computing the mentioned expected value. However, we use here the Krawtchouk polynomials [4] to simplify the expression. The new formula allows us to deepen in the behavior of the mutation operator and suggests a more efficient algorithm to compute the expectation. We will also discuss some practical applications of our findings that can be used to improve the search process.

The remainder of the paper is organized as follows. In the next section the mathematical tools required to understand the rest of the paper are presented. In Section 3 we present our main contribution of this work: the landscape analysis of the bit-flip mutation. Section 4 discusses two practical applications of our results. Finally, Section 5 presents the conclusions and future work.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

2. BACKGROUND

In this section we present some fundamental results of landscapes theory. We will only focus on the relevant information required to understand the rest of the paper. The interested reader can deepen on this topic in [10].

Let (X, N, f) be a landscape, where X is a finite set of solutions, $f : X \to \mathbb{R}$ is a real-valued function defined on X and $N : X \to \mathcal{P}(X)$ is the neighborhood operator. The pair (X, N) is called *configuration space* and can be represented using a graph $G(X, E)$ in which X is the set of vertices and a directed edge (x, y) exists in E if $y \in N(x)$ [1]. We can represent the neighborhood operator by its adjacency matrix

$$A_{xy} = \begin{cases} 1 & \text{if } y \in N(x) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The degree matrix D is defined as the diagonal matrix

$$D_{xy} = \begin{cases} |N(x)| & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The Laplacian matrix of a configuration space is defined as:

$$\Delta = A - D \tag{4}$$

Any discrete function, f, defined over the set of candidate solutions can be characterized as a vector in $\mathbb{R}^{|X|}$. Any $|X| \times |X|$ matrix can be interpreted as a linear map that acts on vectors in $\mathbb{R}^{|X|}$. For example, the adjacency matrix A acts on function f as follows

$$A\ f = \begin{pmatrix} \sum_{y \in N(x_1)} f(y) \\ \sum_{y \in N(x_2)} f(y) \\ \vdots \\ \sum_{y \in N(x_{|X|})} f(y) \end{pmatrix} \tag{5}$$

The component x of this matrix-vector product can thus be written as:

$$(A\ f)(x) = \sum_{y \in N(x)} f(y) \tag{6}$$

which is the sum of the function value of all the neighbors of x. In this paper, we will restrict our attention to regular neighborhoods, where $|N(x)| = d > 0$ for a constant d, for all $x \in X$. We also focus only on connected configuration spaces (the underlying graph of the configuration space is connected). When a neighborhood is regular, $\Delta = A - dI$. Stadler defines the class of *elementary landscapes* where the function f is an eigenvector (or eigenfunction) of the Laplacian up to an additive constant [11]. Formally, we have the following

Definition 1. *Let (X, N, f) be a landscape and Δ the Laplacian matrix of the configuration space. The function f is said to be elementary if there exists a constant b, which we call* offset*, and an eigenvalue λ of $-\Delta$ such that $(-\Delta)(f - b) = \lambda(f - b)$. The landscape itself is elementary if f is elementary.*

According to the previous definition, every elementary function, f, can be written as the sum of an eigenfunction of $-\Delta$, denoted with g, and a constant b, i.e., $f = g + b$. The eigenvalue λ is a feature of the landscape: it depends on the objective function f and the configuration space (X, N). In connected neighborhoods (the ones we consider here) the offset b is the average value of the function over the whole search space: $b = \bar{f}$. In elementary landscapes, the average value $\bar{f}$ can be usually computed in a very efficient way using the problem data. That is, it is not required to do a complete enumeration over the search space.

Taking into account basic results of linear algebra, it is not difficult to prove that if f is elementary with eigenvalue λ, $af + b$ is also elementary with the same eigenvalue λ. Furthermore, in regular neighborhoods, if g is an eigenfunction of $-\Delta$ with eigenvalue λ then g is also an eigenvalue of A, the adjacency matrix, with eigenvalue $d - \lambda$. The average value of the fitness function in the neighborhood of a solution can be computed using the expression:

$$\operatorname*{avg}_{y \in N(x)}\{f(y)\} = \frac{1}{d}(A\ f)(x) \tag{7}$$

If f is an elementary function with eigenvalue λ, then the average is computed as:

$$\begin{aligned} \operatorname*{avg}_{y \in N(x)}\{f(y)\} &= \operatorname*{avg}_{y \in N(x)}\{f(y) - \bar{f}\} + \bar{f} \\ &= \frac{1}{d}(A\ (f - \bar{f}))(x) + \bar{f} = \frac{d - \lambda}{d}(f(x) - \bar{f}) + \bar{f} \\ &= f(x) + \frac{\lambda}{d}(\bar{f} - f(x)) \end{aligned} \tag{8}$$

and we get Grover's wave equation. In the previous expression we used the fact that $f - \bar{f}$ is an eigenfunction of A with eigenvalue $d - \lambda$.

A landscape (X, N, f) is not always elementary, but even in this case it is possible to characterize the function f as the sum of elementary landscapes [14], called *elementary components* of the landscape. The interested reader can find examples of elementary landscapes in [17, 18] and can deepen on the elementary landscape decomposition in [2].

The previous definitions are general concepts of landscapes theory. Let us focus now on the binary configuration spaces with the one-change neighborhood, which are the configuration spaces used by the bit-flip mutation. In these spaces the solution set X is the set of all binary strings of size n. Two solutions x and y are neighboring if one can be obtained from the other by flipping a bit. That is, if the Hamming distance between the solutions, denoted with $\mathcal{H}(x, y)$, is 1. We define the sphere of radius k around a solution x as the set of all solutions lying at Hamming distance k from x [13]. In analogy to the adjacency matrix we define the sphere matrix of radius k as:

$$S^{(k)}_{xy} = \begin{cases} 1 & \text{if } \mathcal{H}(x, y) = k \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The sphere matrix of radius one is the adjacency matrix of the one-change neighborhood, A, and the sphere matrix of radius zero is the identity matrix, I. The definition given in (9) is not useful for making computations with the sphere matrices. However, in [15] more useful expressions for $S^{(k)}$ can be found. Sutton *et al.* prove that the matrices $S^{(k)}$ can be defined using the recurrence:

$$S^{(0)} = I \tag{10}$$

$$S^{(1)} = A \tag{11}$$

$$S^{(k+1)} = \frac{1}{k+1}\left(A \cdot S^{(k)} - (n - k + 1)S^{(k-1)}\right) \tag{12}$$

Using the recurrence it is easy to check that each $S^{(k)}$ is a polynomial in A (the adjacency matrix). For example, the matrices $S^{(2)}$ and $S^{(3)}$ are:

$$S^{(2)} = \frac{1}{2}A^2 - \frac{n}{2}I \tag{13}$$

$$S^{(3)} = \frac{1}{6}A^3 + \left(\frac{1}{3} - \frac{n}{2}\right)A \tag{14}$$

As we previously noted, eigenvectors of the Laplacian matrix Δ are, in regular neighborhoods, eigenvectors of the adjacency matrix A, and this implies that they are also eigenvectors of the sphere matrices $S^{(k)}$. As a consequence, all the functions that are elementary in the one-change configuration space are eigenvectors (up to an additive constant) of $S^{(k)}$ and their eigenvalues can be computed using the same polynomial in A that gives the expression for $S^{(k)}$. For example, let g be an eigenvector of A with eigenvalue λ, then g is eigenvector of $S^{(2)}$ with eigenvalue:

$$\lambda_{S^{(2)}} = \frac{1}{2}\lambda^2 - \frac{n}{2} \tag{15}$$

We should notice that the previous polynomial is just the one of (13) replacing A by λ. We can combine both expressions into one defining the following series of polynomials:

$$S^{(0)}(x) = 1 \tag{16}$$

$$S^{(1)}(x) = x \tag{17}$$

$$S^{(k+1)}(x) = \frac{1}{k+1}\left(x \cdot S^{(k)}(x) - (n-k+1)S^{(k-1)}(x)\right) \tag{18}$$

We use the same name for the polynomials and the matrices related to the spheres. This is because we are going to focus on the polynomials and not on the matrices. The reader should notice, however, that the polynomials will be always presented with their argument and the matrices have no argument. That is, $S^{(k)}$ is the matrix but $S^{(k)}(x)$ is the polynomial.

Using the previous polynomials, the matrix $S^{(k)}$ can be written as $S^{(k)}(A)$ (the polynomial $S^{(k)}(x)$ evaluated in the matrix A) and any eigenvector g of A with eigenvalue λ is also an eigenvector of $S^{(k)}(A)$ with eigenvalue $S^{(k)}(\lambda)$.

One relevant set of eigenvectors of the Laplacian in the one-change configuration space is that of the Walsh functions [14]. Furthermore, the Walsh functions form an orthogonal basis of eigenvectors in the configuration space. Thus, they have been used to find the elementary landscape decomposition of problems with a binary representation like the MAX-k-SAT [9]. The eigenvalue of a Walsh function is $2j$, where j is an integer number called *order* of the Walsh function and ranges from 0 to n. Thus, in the one-change configuration space there are $n+1$ possible values for the eigenvalues of the elementary landscapes. As a consequence, any function can be decomposed in a sum of at most n elementary landscapes. If we also take into account that the size of the neighborhood, d, is n in the one-change configuration space, we conclude that the only possible eigenvalues for the spheres are $S^{(k)}(n-2j)$ with $j \in \{0, 1, \ldots, n\}$. With the help of Eqs. (16) to (18) we can write the following recurrence formula for the eigenvalues of the sphere matrices:

$$S^{(0)}(n-2j) = 1 \tag{19}$$

$$S^{(1)}(n-2j) = n-2j \tag{20}$$

$$(k+1)S^{(k+1)}(n-2j) = (n-2j)\cdot S^{(k)}(n-2j) - (n-k+1)S^{(k-1)}(n-2j) \tag{21}$$

The previous recurrence formula is also satisfied by the Krawtchouk polynomials [3] and the solution of the recurrence is given by the elements of the Krawtchouk matrices $K^{(n)}$ [4]. The correspondence is as follows:

$$S^{(k)}(n-2j) = K^{(n)}_{kj} \tag{22}$$

where the (k,j) element of the n-th order Krawtchouk matrix, $K^{(n)}_{kj}$, is defined as:

$$K^{(n)}_{kj} = \sum_{l=0}^{\min(k,j)} (-1)^l \binom{n-j}{k-l}\binom{j}{l} \tag{23}$$

Equation (22) will be used in the next section to simplify some expressions.

3. ANALYSIS OF BIT-FLIP MUTATION

Our main goal is to obtain a closed-form formula for the expected fitness value of a solution after the bit-flip mutation operator has been applied to it. First of all, let us define the bit-flip mutation operator. Given a solution of size n (binary string), the operator changes the value of each bit with probability p. The parameter p is the only one of the mutation. In the literature it is common to use the value $p = 1/n$, which, on average, changes one bit in each solution. However, if $0 < p < 1$ the mutation operator can yield any solution of the search space with a different probability.

Let us now define a probability matrix P where the element P_{xy} is the probability of obtaining the solution y after a mutation to the solution x. This probability matrix contains all the information we need to compute the expectation of the fitness value of a mutated solution. This expectation for the solution x is:

$$\mathbb{E}[f]_x = \sum_{y \in X} P_{xy} f(y) \tag{24}$$

or using a vector equation we can write:

$$\mathbb{E}[f] = Pf \tag{25}$$

where we consider f and $\mathbb{E}[f]$ as vectors. In this case, $\mathbb{E}[f]$ is the vector of expectations, that is, the component x contains the expectation of the fitness value when the mutation is performed over x.

Now we need to compute the probability matrix for the bit-flip mutation. The probability of reaching the solution y from the solution x after an application of the mutation depends on the Hamming distance $\mathcal{H}(x,y)$. It is the probability of changing $\mathcal{H}(x,y)$ bits in the solution taking into account that the probability of flipping one single bit, p, is independent for each bit. Thus, we can compute the probability matrix P for the bit-flip mutation as follows:

$$P_{xy} = p^{\mathcal{H}(x,y)}(1-p)^{n-\mathcal{H}(x,y)} \tag{26}$$

The previous probability matrix can be expressed using the sphere matrices $S^{(k)}(A)$ of the previous section. We should notice here that all the solutions y that are at Hamming distance k from x have the same probability of being selected: $p^k(1-p)^{n-k}$. Thus, we can write the probability matrix in the following way:

$$P = \sum_{k=0}^{n} p^k (1-p)^{n-k} S^{(k)}(A) \tag{27}$$

The previous expression has an important implication: the sphere matrices $S^{(k)}(A)$ and the probability matrix P for the bit-flip mutation have the same eigenfunctions. We previously saw that the eigenfunctions of the sphere matrices are the eigenfunctions of the Laplacian matrix (elementary functions). Thus, we reach an important result that is stated in the following

THEOREM 1. *If f is an eigenfunction of the Laplacian matrix for the one-change binary configuration space with eigenvalue $2j$, then f is also an eigenfunction of P, the probability matrix of the bit-flip mutation operator, with eigenvalue*

$$\Lambda(p, j) = (1-2p)^j \tag{28}$$

where $\Lambda(p, j)$ is a two-variable function (p and j), j is an integer value in $\{0, 1, \ldots, n\}$ and $p \in [0, 1]$ is the probability of flipping one bit in the mutation.

PROOF. We have previously claimed that if f is an eigenfunction of the Laplacian with eigenvalue $2j$, then it is also an eigenfunction of the sphere matrix $S^{(k)}(A)$ with eigenvalue $S^{(k)}(n-2j)$. According to (27) the probability matrix P is a linear combination of the $S^{(k)}(A)$ matrices. Thus, f is also an eigenfunction of P with eigenvalue:

$$\begin{aligned}\Lambda^{(n)}(p, j) &= \sum_{k=0}^{n} p^k (1-p)^{n-k} S^{(k)}(n-2j) \\ &= \sum_{k=0}^{n} p^k (1-p)^{n-k} K_{kj}^{(n)}\end{aligned} \tag{29}$$

where we have used the equality (22). The previous equation can be simplified with the help of some properties of the Krawtchouk matrices. In particular, the elements of a Krawtchouk matrix satisfy the following equation [4]:

$$(1+x)^{n-j}(1-x)^j = \sum_{k=0}^{n} x^k K_{kj}^{(n)} \tag{30}$$

Thus, we can write:

$$\begin{aligned}\Lambda^{(n)}(p, j) &= \sum_{k=0}^{n} p^k (1-p)^{n-k} K_{kj}^{(n)} \\ &= (1-p)^n \sum_{k=0}^{n} \left(\frac{p}{1-p}\right)^k K_{kj}^{(n)} \\ &= (1-p)^n \left(1 + \frac{p}{1-p}\right)^{n-j} \left(1 - \frac{p}{1-p}\right)^j \\ &= (1-p)^n \left(\frac{1}{1-p}\right)^{n-j} \left(\frac{1-2p}{1-p}\right)^j \\ &= (1-2p)^j\end{aligned} \tag{31}$$

where we can observe that $\Lambda^{(n)}(p, j)$ does not depend on n, so we omit the superscript (n). □

Now we can give an efficient formula for the vector of expectations as defined in (25) for an elementary landscape. This formula is presented in the following

THEOREM 2. *Let f be an elementary function in the one-change binary configuration space with eigenvalue $2j$, then the vector of expectations after the application of the bit-flip mutation operator to the solutions in the search space can be computed as:*

$$\mathbb{E}[f] = \bar{f} + (1-2p)^j (f - \bar{f}) \tag{32}$$

where $\bar{f}$ is the average value of f in the search space.

PROOF. If f is an elementary function $f - \bar{f}$ is an eigenfunction of the Laplacian (by definition). With the help of Theorem 1 we can write:

$$\begin{aligned}\mathbb{E}[f] &= \bar{f} + \mathbb{E}[f - \bar{f}] \\ &= \bar{f} + P(f - \bar{f}) \\ &= \bar{f} + \Lambda(p, j)(f - \bar{f}) \\ &= \bar{f} + (1-2p)^j (f - \bar{f})\end{aligned}$$

and we get (32). □

The previous theorem is the main contribution of this work and it says that for elementary landscapes we can compute the expectation using $\bar{f}$ (which is a constant) and the value of the objective function in the solution we are interested, $f(x)$. Equation (32) is a vector equation, but we can express it for a particular solution in the following way:

$$\mathbb{E}[f]_x = \bar{f} + (1-2p)^j (f(x) - \bar{f}) \tag{33}$$

Some conclusions can be obtained just analyzing the previous expression. Let us focus on the factor $\Lambda(p, j) = (1-2p)^j$, since it is the only one that contains the probability p of flipping a bit. If we make the variable change $z = 1-2p$, then the factor is written z^j, which is a monomial in z, so the shape of $\Lambda(p, j)$ is that of a polynomial of degree j with one only root in $p = 1/2$ (with multiplicity j). If j is even then $\Lambda(p, j)$ is symmetric with respect to $p = 1/2$ and the function has a minimum in that value. If j is odd, then $\Lambda(p, j)$ is strictly decreasing. In Figures 1 and 2 we show some $\Lambda(p, j)$ functions for even and odd values of j, respectively. Since the value of p is restricted to the interval $[0, 1]$ we conclude that if j is even $\Lambda(p, j)$ has two maxima in $p = 0$ and $p = 1$ with value 1. Otherwise, if j is odd it has one maximum in $p = 0$ with value 1 and one minimum in $p = 1$ with value -1.

With the help of (33), let us now translate these observations to an arbitrary solution x of the search space. We distinguish three cases depending on the value $f(x)$:

- Case $f(x) > \bar{f}$. The shape of the expectation curve, $\mathbb{E}[f]_x$, is similar to that of $\Lambda(p, j)$ but it is scaled and vertically moved by $\bar{f}$ (see Figure 3). If $p = 1/2$ the expected value is $\bar{f}$. This is common sense because in this case the bit-flip mutation can reach any solution of the search space with the same probability. If j is even $\mathbb{E}[f]_x$ takes the same value for $p = 0$ (no mutation) and

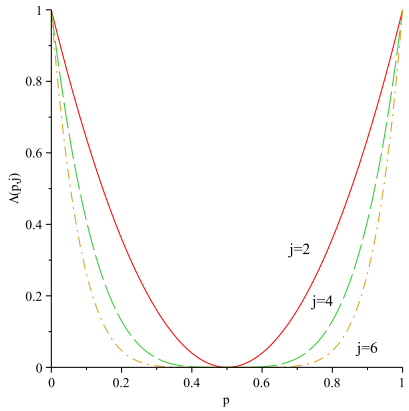

Figure 1: Polynomial $\Lambda(p, j)$ for j even.

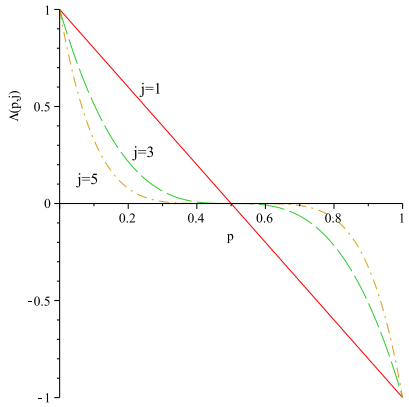

Figure 2: Polynomial $\Lambda(p, j)$ for j odd.

$p = 1$ (all bits flipped). This implies that any solution x and its complement $\tilde{x}$ (built by flipping all the bits in the solution) have the same fitness value: $f(x) = f(\tilde{x})$. If j is odd then the polynomial is antisymmetric with respect to $p = 1/2$ and the complement of any solution have fitness value: $f(\tilde{x}) = 2\bar{f} - f(x)$.

- Case $f(x) < \bar{f}$. The expectation curve is like $\Lambda(p, j)$ but inverted with respect to a horizontal line (see Figure 4). Thus, if j is even the expectation reaches a maximum in $p = 1/2$ with value $\bar{f}$. As in the previous case, depending on the parity of j every solution x satisfies $f(x) = f(\tilde{x})$ (j even) or $f(\tilde{x}) = 2\bar{f} - f(x)$ (j odd).

- Case $f(x) = \bar{f}$. In this case the factor $\Lambda(p, j)$ is multiplied by 0 and the expectation does not depend on p, it is $\mathbb{E}[f]_x = \bar{f}$.

Once we have analyzed $\mathbb{E}[f]_x$ for the elementary landscapes we now generalize the results to arbitrary functions. We mentioned in the previous section that it is always possible to express the objective function f as a sum of, at most, n elementary landscapes with eigenvalues from 2 to $2n$ (at steps of 2). For each of these elementary landscapes we can compute the expectation using (32), so we just have to sum all the expectations of the components to get the expectation of f itself. Let us formally present this idea in the following

THEOREM 3. *Let f be an arbitrary function whose elementary decomposition in the one-change binary configuration space is:*

$$f = \sum_{j=1}^{n} \Omega_{2j} \tag{34}$$

where Ω_{2j} denotes the elementary component with eigenvalue $2j$. The vector of expectations after the application of the bit-flip mutation operator to the solutions in the search space is:

$$\mathbb{E}[f] = \overline{f} + \sum_{j=1}^{n} (1-2p)^j (\Omega_{2j} - \overline{\Omega_{2j}}) \tag{35}$$

where $\overline{\Omega_{2j}}$ denotes the average value of the function $\Omega_{2j}(x)$ in the entire search space.

PROOF. With the help of (32) we can write:

$$\begin{aligned}
\mathbb{E}[f] &= \mathbb{E}\left[\sum_{j=1}^{n} \Omega_{2j}\right] = \sum_{j=1}^{n} \mathbb{E}[\Omega_{2j}] \\
&= \sum_{j=1}^{n} \left(\overline{\Omega_{2j}} + \Lambda(p, j)(\Omega_{2j} - \overline{\Omega_{2j}})\right) \\
&= \overline{f} + \sum_{j=1}^{n} \Lambda(p, j)(\Omega_{2j} - \overline{\Omega_{2j}}) \\
&= \overline{f} + \sum_{j=1}^{n} (1-2p)^j (\Omega_{2j} - \overline{\Omega_{2j}})
\end{aligned}$$

and we get (35). □

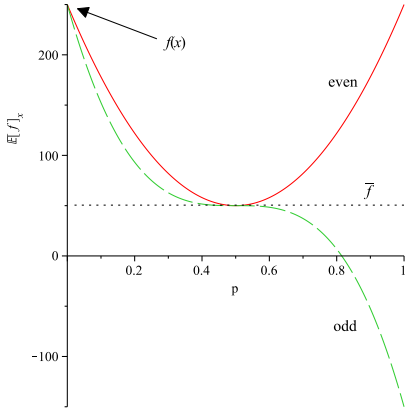

Figure 3: Expectation $\mathbb{E}[f]_x$ as a function of p when $f(x) > \bar{f}$.

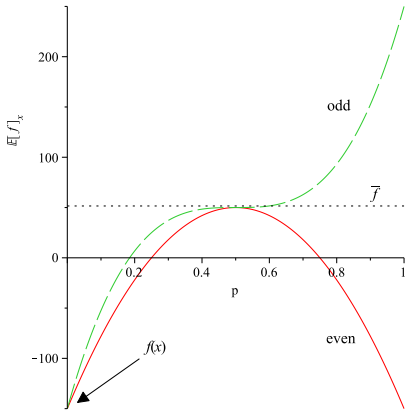

Figure 4: Expectation $\mathbb{E}[f]_x$ as a function of p when $f(x) < \bar{f}$.

In particular for a solution x we have:

$$\mathbb{E}[f]_x = \overline{f} + \sum_{j=1}^{n} (1-2p)^j (\Omega_{2j}(x) - \overline{\Omega_{2j}}) \qquad (36)$$

With this expression we can efficiently compute the expected value after the application of the bit-flip mutation operator to solution x for an arbitrary function f. The complexity of this operation is the sum of the complexities of the evaluation of the elementary components. The average value $\overline{\Omega_{2j}}$ is a constant that depends on the parameters of the particular instance we are solving and can be efficiently precomputed before the search process. It is our experience that usually we can find an algorithm for computing the component Ω_{2j} and the value $\overline{\Omega_{2j}}$ that has the same complexity as the original function f. If this is true for a problem the complexity of computing $\mathbb{E}[f]_x$ is at most n times the complexity of computing a particular component Ω_{2j}. One interesting observation is that in many problems the number of elementary components is a fixed number lower than n, independently of the instance. For example, in the MAX-k-SAT problem the objective function can be written as a sum of k elementary landscapes [9]. In these cases the computation of $\mathbb{E}[f]_x$ will have the same complexity as the computation of the hardest elementary component Ω_{2j}.

The curves of $\mathbb{E}[f]_x$ for general functions f are not so easy to analyze like the ones of the elementary landscapes. In general, the shape of such curves can be almost arbitrary. The only limitations are that they must be a polynomial of degree at most n and $\mathbb{E}[f]_x = \bar{f}$ at $p = 1/2$. We can state that an elementary component with order j (and eigenvalue $2j$) is related to a polynomial of degree j in the expectation curve. As an example we show in Figure 5 the curve $\mathbb{E}[f]_x$ for a function which can be decomposed into three elementary landscapes. We show in the figure the expectation (solid line) and the contribution of each elementary component (dashed lines). We can observe in this case that the function value $f(x)$ is $\bar{f}$ (see the value for $p = 0$). However, in spite of this, the expectation does depend on p because we are not dealing with an elementary landscape. Furthermore, it has the maximum value at $p = (4 - \sqrt{7})/6 \approx 0.226$. We can observe that for $p = 1/2$ the expectation crosses $\bar{f}$ again.

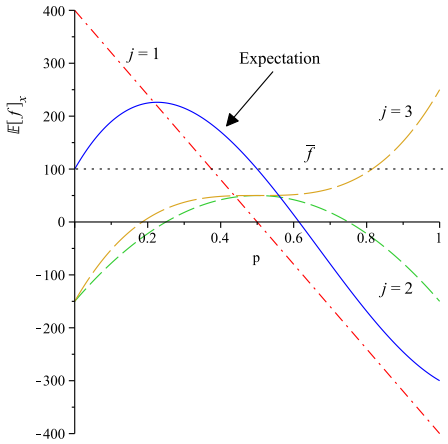

Figure 5: Expectation $\mathbb{E}[f]_x$ for a function with three elementary components.

From a theoretical point of view, the results presented in this section shed some light on the behavior of the bit-flip mutation operator. This knowledge could be used, for example, in the theoretical analysis of the runtime of search algorithms in which this operator is used. However, these results can also have practical applications. We devote the next section to provide some examples of such applications.

4. FROM THEORY TO PRACTICE

In this section we discuss two possible applications of the theoretical results obtained. The first one is related to the selection operator in population-based metaheuristics whilst the second one is related to the bit-flip mutation operator.

4.1 Selection operator

One of the main advantages of the landscape decomposition of a problem is that we can efficiently compute the average value of the fitness function in the neighborhood of a solution. In a search algorithm this fact can be used to select the solutions according to this average value instead of basing the selection on the fitness value of the solution itself. This idea has been used to escape from plateaus in [13] and to reduce the computational effort in [7].

Equation (32) allows us to compute the expected fitness value after the bit-flip mutation with the same efficiency as the average value in the neighborhood. Thus, if bit-flip mutation is used in the search algorithm, it seems that using the expected fitness value is a better alternative than using the fitness value of the solution itself or the average in the neighborhood.

However, the efficacy of this approach depends on the number of times in which the proposed expectation-based selection selects an individual that is different from the one selected by a traditional fitness-based selection. Another issue that could limit the success of this approach is the change in the search strategy itself. An expectation-based selection could speed up the improvement of the population. This could guide the population to local optima.

4.2 Mutation operator

In Section 3 we saw that the expectation depends on the probability p of flipping a bit. Furthermore, for a given function and solution the expectation depends only on p. We discuss here a strategy for deciding the value of p in order to maximize the expectation of the fitness value of the new solution. In this way we can provide a self-adaptive strategy in which the parameter p is optimally selected according to the individual.

The method consists in the following operation (we assume maximization). Before mutating a solution, we use equation (36) to obtain a polynomial in p that gives the value of $\mathbb{E}[f]_x$. This is always possible, since we assume that we can evaluate the elementary components of f in the solution x. After that, we find the maximum of p in the interval $p \in [0, 1]$. In theory, we can do this analytically, since we know the exact expression of the polynomial $\mathbb{E}[f]_x$. Finally, we select the value p for which $\mathbb{E}[f]_x$ is maximum and apply the bit-flip mutation using this value of p.

In general, the previous method can be computationally expensive because we need to solve an optimization problem to decide p. It is supposed that the new optimization problem is much easier to solve than the one we are interested in. It is a continuous problem with one single variable: p.

There is a special case in which the problem of deciding p can be simplified. This happens when we are optimizing an elementary landscape.

In elementary landscapes we can use (33) and we really do not need to take into account the value of $f(x)$ in the computations. We only need to know whether $f(x) > \bar{f}$ or not. The suggested values for p depends on the relative position of $f(x)$ and $\bar{f}$. If $f(x) > \bar{f}$ then we have two cases:

- If j is odd, then the maximum is in $p = 0$ and the expected value of the mutated individual will be lower if $p > 0$. Strictly applying the strategy would mean to stop the search. Perhaps a better solution is to use a low value of p (say, the traditional $p = 1/n$).
- If j is even, then the maxima are in $p = 0$ or $p = 1$, which means that the best option is to keep the same solution or to go to the complementary solution. As in the previous case, it is possible that the search is stopped if we strictly apply the strategy so we should use a low (or high) value for p.

If $f(x) < \bar{f}$ in an elementary landscape, then we need to find a value of p that minimizes $(1-2p)^j$. We can also distinguish two cases:

- If j is odd, the minimum is in $p = 1$, so we should go to the complementary solution.
- If j is even, then the minimum is in $p = 1/2$, which means start from a new random solution.

Third, if $f(x) = \bar{f}$ the expectation is $\bar{f}$ and the value of p is irrelevant.

If the landscape is not elementary we have to deal in general with Eq. (36). In order to find the optima we need to compute the derivative of (36) and find the roots of the derivative polynomial. We can use numerical methods, like the Newton method, for finding these roots. Furthermore, we can use exact methods to solve the derivative polynomials up to degree four (Ferrari method). This means that it is always possible to provide closed-form values for p in the case of functions composed of up to five elementary landscapes (with eigenvalues up to 10).

In the rest of this section we are going to analyze the case in which at most the three lower order elementary landscapes are present. Some important combinatorial optimization problems fulfill this requirement. Some examples are the MAX-3-SAT [14] and the Unconstrained Quadratic Optimization [6]. In this case, (36) can be written as:

$$\mathbb{E}[f]_x = \overline{f} + (1-2p)\Delta\Omega_2 + (1-2p)^2\Delta\Omega_4 + (1-2p)^3\Delta\Omega_6 \quad (37)$$

where we introduced the notation $\Delta\Omega_{2j} = \Omega_{2j}(x) - \overline{\Omega_{2j}}$ to simplify the following expansions. The derivative is:

$$\frac{d\mathbb{E}[f]_x}{dp} = -2\Delta\Omega_2 - 4(1-2p)\Delta\Omega_4 - 6(1-2p)^2\Delta\Omega_6 \quad (38)$$

Now we have to distinguish three cases:

- If $\Delta\Omega_6 \neq 0$, then the previous equation is a second order formula and the roots are:

$$p = \frac{3\Delta\Omega_6 + \Delta\Omega_4 \pm \sqrt{(\Delta\Omega_4)^2 - 3\Delta\Omega_2\Delta\Omega_6}}{6\Delta\Omega_6} \quad (39)$$

  If $(\Delta\Omega_4)^2 - 3\Delta\Omega_2\Delta\Omega_6 \geq 0$ then the previous equation will have one or two real values. However, these values could be out of the interval $[0, 1]$. The solutions in the interval $[0, 1]$ must be compared with the limits of the interval in order to select the optimum value.

- If $\Delta\Omega_6 = 0$ but $\Delta\Omega_4 \neq 0$ then Eq. (38) is a linear function whose root is:

$$p = \frac{2\Delta\Omega_4 + \Delta\Omega_2}{4\Delta\Omega_4} \quad (40)$$

  As in the previous case we must check that this value is in the interval $[0, 1]$ before comparing it with the limits.

- If $\Delta\Omega_6 = \Delta\Omega_4 = 0$ then the expectation is monotone (if $\Delta\Omega_2 \neq 0$) or constant (if $\Delta\Omega_2 = 0$). In any case the optimum value is in the limits of the interval $[0, 1]$ ($p = 0$ or $p = 1$).

The complexity of this strategy is the same as evaluating the elementary components Ω_{2j} of the objective function. It is our experience that these evaluations do not take longer than the evaluation of f itself. Thus, in the case of a function composed of three elementary landscapes, the complexity is usually the same as the function evaluation.

The efficacy of the proposed method depends on several issues. First, the strategy could be really useful for the solutions in which the maximum expectation is not in the limits of the interval $[0, 1]$. In this case we gain information from the landscape decomposition. One necessary condition for this to happen is that at least one of the elementary components has a value that is below its average $\overline{\Omega_{2j}}$. We can easily check this from Eq. (36). If it happens that $\Omega_{2j}(x) > \overline{\Omega_{2j}}$ for all the elementary components, then those components with odd order are all decreasing and those with even order reach their maximum at $p = 0$ and $p = 1$. This means, that no value $p \in (0, 1)$ exists such that the expectation is higher than in $p = 0$.

In order to study how often does our proposed strategy give a useful value for the probability p we have selected a problem, we have generated random solutions for that problem and we have counted the number of solutions for which the probability is in the open interval $(0, 1)$. The problem is the 0-1 Unconstrained Quadratic Optimization (UQO) [6], which can be decomposed as a sum of two elementary landscapes with order $j = 1$ and $j = 2$. We have randomly generated 30 instances of UQO using the generator by Palubeckis [8]. The size of the instances ranges from $n = 500$ to $n = 3000$. The density δ of the instances is taken from the set $\{10, 30, 50, 70, 90\}$. The number of generated random solutions is 10000. In Table 1 we show for each instance the number of solutions for which the maximum expectation is found when $p \in (0, 1)$. We can observe that all the numbers are around 3000, which means that 30% of the search space is composed of solutions that could profit from the proposed adaptive mutation.

A second important issue for the success of this approach is the appearance during the search process of solutions for which the strategy is useful. During the search process, the solutions in the population will improve their fitness value and, thus, the probability of finding a solution for which the strategy is useful decreases as the search progresses. Following with the UQO example, we have solved the instance

Table 1: Number of solutions (from a total of 10000) for which the maximum expectation is in $p \in (0,1)$.

	δ				
n	**10**	**30**	**50**	**70**	**90**
500	2975	2932	3010	3125	3041
1000	3024	3003	3070	2984	3095
1500	3050	3154	3088	3024	2978
2000	2939	3048	3074	3055	3083
2500	3079	3014	3084	3025	3051
3000	3107	3062	3052	3022	3001

with $n = 500$ and density $\delta = 10$ using a genetic algorithm equipped with our proposed strategy and only for a 1% of the solutions found during the search the maximum expectation happened for a value $p \in (0,1)$.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have used some results of landscapes theory to provide a closed-form formula for the expectation of the fitness value of a solution that has changed using the bit-flip mutation. The formula uses the elementary components of the objective function and can be computed in polynomial time if the components can also be computed in polynomial time. We have also discussed some applications of the expectation formula.

The findings of this paper can be extended in multiple directions. First, it is possible to provide closed-form formulas not only for the expectation of the fitness function but also for the variance or higher order moments. In this sense, the work by Sutton *et al.* [15] is a guide to follow on this topic.

Second, using a methodology similar to that presented in this paper we can extend the results to other configuration spaces and operators. For example, the permutation space with the swap mutation, the sequence space with the one-change mutation, etc.

Third, an interesting question that remains open is whether we can apply landscapes theory to analyze combinations of operators. For example, crossover followed by mutation. If this analysis were possible and provided a simple formula for the expectation, we could consider the entire search process as an operator and we would have a closed-form formula for the expected performance of an algorithm. This last idea would connect the theoretical runtime analysis with landscapes theory.

6. ACKNOWLEDGMENTS

This research has been partially funded by the Spanish Ministry of Science and Innovation and FEDER under contract TIN2008-06491-C04-01 (the M* project) and the Andalusian Government under contract P07-TIC-03044 (DIRICOM project).

7. REFERENCES

[1] T. Biyikoglu, J. Leyold, and P. F. Stadler. *Laplacian Eigenvectors of Graphs*. Lecture Notes in Mathematics. Springer-Verlag, 2007.

[2] F. Chicano, L. D. Whitley, and E. Alba. A methodology to find the elementary landscape decomposition of combinatorial optimization problems. *Evolutionary Computation*, 2011. DOI: 10.1162/EVCO_a_00039.

[3] R. Coleman. On krawtchouk polynomials. http://arxiv.org/abs/1101.1798, January 2011.

[4] P. Feinsilver and J. Kocik. Krawtchouk polynomials and krawtchouk matrices. In R. Baeza-Yates, J. Glaz, H. Gzyl, J. Hüsler, and J. Palacios, editors, *Recent Advances in Applied Probability*, pages 115–141. Springer US, 2005.

[5] R. García-Pelayo and P. Stadler. Correlation length, isotropy and meta-stable states. *Physica D: Nonlinear Phenomena*, 107(2-4):240–254, Sep 1997.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[7] G. Lu, R. Bahsoon, and X. Yao. Applying elementary landscape analysis to search-based software engineering. In *Proc. of the 2nd Intl. Symposium on Search Based Software Engineering*, 2010.

[8] G. Palubeckis. An algorithm for construction of test cases for the quadratic assignment problem. *Informatica (Lithuanian Academy of Sciences)*, 11(3):281–296, 2000.

[9] S. Rana, R. B. Heckendorn, and D. Whitley. A tractable walsh analysis of SAT and its implications for genetic algorithms. In *Proceedings of AAAI*, pages 392–397, Menlo Park, CA, USA, 1998.

[10] C. M. Reidys and P. F. Stadler. Combinatorial landscapes. *SIAM Review*, 44(1):3–54, 2002.

[11] P. F. Stadler. Toward a theory of landscapes. In R. López-Peña, R. Capovilla, R. García-Pelayo, H.Waelbroeck, and F. Zertruche, editors, *Complex Systems and Binary Networks*, pages 77–163. Springer-Verlag, 1995.

[12] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[13] A. M. Sutton, A. E. Howe, and L. D. Whitley. Directed plateau search for MAX-k-SAT. In *Proceedings of SoCS*, Atlanta, USA, 2010.

[14] A. M. Sutton, L. D. Whitley, and A. E. Howe. A polynomial time computation of the exact correlation structure of k-satisfiability landscapes. In *Proceedings of GECCO*, pages 365–372, New York, USA, 2009.

[15] A. M. Sutton, L. D. Whitley, and A. E. Howe. Computing the moments of k-bounded pseudo-boolean functions over hamming spheres of arbitrary radius in polynomial time. *Theoretical Computer Science*, 2011. in press (doi: 10.1016/j.tcs.2011.02.006).

[16] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, 1990.

[17] D. Whitley, A. M. Sutton, and A. E. Howe. Understanding elementary landscapes. In *Proceedings of GECCO*, pages 585–592, New York, USA, 2008.

[18] L. D. Whitley and A. M. Sutton. Partial neighborhoods of elementary landscapes. In *Proceedings of GECCO*, pages 381–388. Montreal, Canada, 2009.