DIRICOM

Diseño Inteligente de Redes Inalámbricas de Comunicación
Conserjería de Innovación, Ciencia y Empresa
P07-TIC-03044
http://diricom.lcc.uma.es

Málaga, 12 de Diciembre de 2011

Informe Ejecutivo

TÍTULO: SOFT-4.0-2011: Landscape Explorer: una herramienta para la experimentación en teoría de landscapes

RESUMEN: La teoría de *landscapes* es un marco que permite analizar funciones definidas sobre conjuntos de elementos conectados mediante un operador de vecindario. Esta teoría tiene aplicaciones en diversos dominios de conocimiento, entre los que se encuentra la optimización combinatoria. Este documento tiene un doble objetivo. Por un lado, presenta los fundamentos de la teoría y los resultados recientes más relevantes aplicados a los problemas de optimización combinatoria, dando al lector una visión global del estado del arte en este contexto. En segundo lugar, presenta una herramienta software, *Landscape Explorer*, que encapsula gran parte del conocimiento adquirido por los autores en teoría de *landscapes*. Esta herramienta pretende ser un primer punto de encuentro entre los investigadores interesados y la teoría. El diseño arquitectónico de la herramienta está pensado para su fácil extensión por parte de cualquier investigador de la comunidad científica.

OBJETIVOS:

1. Presentar el estado del arte en teoría de *landscapes*.
2. Introducir una herramienta, Landscape Explorer, para el estudio experimental de la teoría de *landscapes*.

CONCLUSIONES:

1. La teoría de *landscapes* permite analizar los problemas de optimización combinatoria y extraer información que puede usarse para guiar algoritmos de búsqueda.
2. La herramienta *Landscape Explorer* permite usar las técnicas y procedimientos propios de la teoría de *landscapes* sin conocer los detalles matemáticos de la misma.
3. Dicha herramienta está diseñada como ecosistema software para permitir la rápida ampliación de su funcionalidad.

RELACIÓN CON ENTREGABLES: No tiene dependencias.

DIRICOM

Diseño Inteligente de Redes Inalámbricas de Comunicación
Conserjería de Innovación, Ciencia y Empresa
P07-TIC-03044
http://diricom.lcc.uma.es

Málaga, December 12th, 2011

Executive Summary

TITLE: SOFT-4.0-2011: Landscape Explorer: a tool for landscape theory experimentation support

ABSTRACT: Landscape theory is a framework in which functions defined over search spaces connected by a neighborhood can be analyzed. This theory has applications in several knowledge domains, in which combinatorial optimization is included. This document has a two-fold contribution. First, it presents the foundations of the theory and the most relevant recent results applied to combinatorial optimization problems. This gives the reader a general view of the state of the art in this context. Second, it introduces a software tool, *Landscape Explorer*, which contains part of the acquired knowledge by the authors in landscape theory. The architectural design of the tool has been thought to be easy to extend by any researcher interested in landscape theory.

GOALS:

1. Present the state of the art in landscape theory.
2. Introduce a tool, Landscape Explorer, to support the experimental study of landscape theory.

CONCLUSIONS:

1. With landscape theory it is possible to analyze the combinatorial optimization problems and extract information from them that can be used to guide search algorithms.
2. The software tool *Landscape Explorer* provides techniques and procedures of landscape theory that can be used without a detailed knowledge of the mathematical details.
3. The tool is designed as a software ecosystem in order to ease a fast extension of its functionality.

RELATION WITH DELIVERABLES: No dependencies.

Landscape Explorer: a tool for landscape theory experimentation support

DIRICOM

December 12^{th}, 2011

1 Introduction

The theory of landscapes 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 [15]. In the field of combinatorial optimization, this theory has been previously used to characterize optimization problems [11], improve search algorithms [13], and obtain global statistics of the problems [21].

A *landscape* for a combinatorial optimization problem (COP) is a triple (X, N, f), where X is the set of *tentative solutions* of the COP, $f : X \mapsto \mathbb{R}$ defines the objective or *fitness function* and N is the *neighborhood operator*.

This theory has applications not only in evolutionary computation [24] but also in Chemistry [16], Biology [22] and Physics [11]. One of the objectives of this theory is to better understand the structure of an optimization problem. The knowledge obtained from the objective function using this theory could be used to improve the performance of EAs by proposing new operators or new values for the parameters.

There exists a special kind of landscapes, called *elementary landscapes* (EL), which are of particular interest due to their properties [24]. We define and analyze the elementary landscapes in Section 2, but we can advance that they are characterized by the *Grover's wave equation*:

$$\operatorname*{avg}_{y \in N(x)}\{f(y)\} = f(x) + \frac{k}{d}\left(\bar{f} - f(x)\right), \tag{1}$$

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 k is a characteristic constant. For a given problem instance whose objective function is elementary, the values $\bar{f}$ and k can be easily computed in an efficient way, usually from the problem data. Thus, 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 $f(x)$, without actually evaluating any of the neighbors.

The advantage of an expression like (1) is that it allows one to compute a statistic value from the value of f in x: the average of the value that the objective function takes in the neighborhood of x. We can observe that if (1) cannot be applied we need to evaluate all the solutions in the neighborhood $N(x)$ to obtain the average. Thus, we conclude that landscape theory provides an efficient way to compute non-trivial statistic values related to the distributions of the objective function. This property is inherent to most of the results on landscape theory, as we will see in Section 2.

In the last years several researchers have lead the research on landscape theory with applications to combinatorial optimization. We can highlight the work by Andrew Sutton and Darrell Whitley in the Colorado State University (USA), William Langdon in the University College London (UK), Guanzhou Lu and Xin Yao in the University of Birmingham (UK) and Francisco Chicano and Enrique Alba in the University of Málaga (Spain). Among the contributions of these researchers we find new search algorithms [13], mechanisms to escape from plateaus [17], algorithms to compute statistics over sets of solutions centered on a given one [20], a methodology to find the elementary landscape decomposition of a landscape [9], a method to estimate the probability distributions of the objective function in

the neighborhood of a solution [18], the elementary landscape decomposition of several problems [4, 10] and efficient algorithms to compute autocorrelation measures in NP-hard problems from the data of the instances [8].

In the previous success list we can observe that some of the contributions are practical or have a direct application inside a search algorithm. Those contributions yield an algorithm or method that can be implemented. The implementation of these procedures require a deep knowledge on the theory but the use of the procedures does not require such a knowledge. That is, it is possible in many cases to *encapsulate* the landscape theory knowledge in a small fragment of code that can be useful in practice, as it has been shown in several previous works. Furthermore, many of the results of landscape theory are general and can be applied to all the combinatorial optimization problems or big subsets of them. All this justifies the interest of this theory.

Along these years of research on landscape theory we have frequently implemented small code snippets and algorithms to empirically check the results of the theory and study the practical implications of it.We think that these algorithms can be useful for the rest of researchers in the domain. For this reason, we developed a software tool, called *Landscape explorer*, that includes these algorithms. This application has been designed to allow an easy extension. Our goal is to create a useful tool not only for the researchers of landscape theory, but also for thse researchers that want to discover the possibilities of the theory.

The rest of the document is organized as follows. Section 2 presents the foundations of landscape theory. Section 3 preents the most relevant results in the last years. Next, the software tool is introduced in Section 4. Finally, the document concludes in Section 5.

2 Background on Landscape Theory

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 [14].

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 $y \in N(x)$ [2]. 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 [15]. Formally, we have the following

Definición 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., $f = g + b$. In connected neighborhoods (the ones we consider here) the offset b is the average value of the function over the whole search space: $b = \bar{f}$. 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:

$$\underset{y \in N(x)}{\text{avg}}\{f(y)\} = \frac{1}{d}(A\ f)(x) \tag{7}$$

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

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

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 [21], called *elementary components* of the landscape. This decomposition is useful from a theoretical and practical point of view. In theory, the decomposition of an optimization problem can be used to compute the exact expression of the Weinberger autocorrelation function [22], the autocorrelation coefficient and the autocorrelation length [1]. These autocorrelation measures characterize the objective function in such a way that it is possible to predict the performance of some local search techniques. From a practical point of view the decomposition together with the Grover's wave equation can be used to compute the average of the fitness value in the neighborhood of a solution, what can be the base for new operators or search algorithms [6, 13, 17]. The reader interested in the elementary landscape decomposition is referred to [9].

3 Recent Results

The previous section presented a general background on landscape theory. In this section we focus on some particular configuration spaces. Once we fix the solution space and the neighborhood it is possible to derive some more results that are specific of the corresponding configuration space. In the following sections we present the main findings in three different configuration spaces: the binary hypercube, the generalized hypercube and the symmetric (permutation) space.

3.1 Binary Hypercube

We call *binary hypercube* to a configuration space in which the set of solutions X is the binary trings of length n, that is, $X = \mathbb{B}^n$; and two arbitrary solutions x and y are neighboring if the Hamming distance between them is 1, that is, there is one single bit which is different between the two strings. The graph associated to the binary hypercube is connected, symmetric and regular. As a consequence, there exists at least a basis in the space of objective functions $\mathbb{R}^{|X|}$ composed of eigenvectors of the Laplacian matrix.

One of these bases, perhaps the best-known one, is the one of the Walsh functions [21]. Given a binary string $w \in \mathbb{B}^n$, the Walsh function with parameter w is defined as:

$$\psi_w(x) = \prod_{i=1}^{n} (-1)^{w_i x_i} = (-1)^{\sum_{i=1}^{n} w_i x_i}. \tag{9}$$

This function is an eigenvector of the Laplacian matrix with eigenvalue $2|w|$, where $|w|$ is the number of ones of the string w. Given an arbitrary function f, we can always write it as a weighted sum of Walsh functions:

$$f = \sum_{w \in \mathbb{B}^n} z_w \psi_w, \tag{10}$$

where z_w are the so-called *Walsh coefficients*.

It is always possible to sum together all the Walsh functions having the same eigenvalue to produce an *elementary component* of f and, thus, we obtain:

$$f = \sum_{j=0}^{n} \sum_{\substack{w \in \mathbb{B}^n \\ |w|=j}} z_w \psi_w = \sum_{j=0}^{n} f_{[j]}, \tag{11}$$

where $f_{[j]}$ is the elementary component of f with eigenvalue $2j$.

Sutton *et al.* [20] analyze the *landscapes* in binary hypercubes. They prove that the elementary functions in that configuration space are also elementary when the neighborhood of a solution x is the set of solutions y at Hamming distance r from x, for an arbitrary r (sphere of radius r). This set of solutions is denoted with $S^{(r)}(x)$. The direct consequence is that it is possible to compute the average of the fitness function in $S^{(r)}(x)$. This average is:

$$\operatorname*{avg}_{y \in S^{(r)}(x)} \{f(y)\} = \sum_{j=0}^{n} \mathcal{K}_{rj}^{(n)} f_{[j]}(x), \tag{12}$$

where $\mathcal{K}_{rj}^{(n)}$ is the (r, j)-th element of the order n Krawtchouk matrix, defined by:

$$\mathcal{K}_{rj}^{(n)} = \sum_{l=\max(0,r+j-n)}^{\min(r,j)} (-1)^l \binom{n-j}{r-l} \binom{j}{l}. \tag{13}$$

The previous expression allows one to efficiently compute $\operatorname{avg}\{f(y)\}_{y \in S^{(r)}(x)}$ if we know the elementary landscape decomposition of f. If we do not use (12) the computation would be very inefficient, especially for high values of r. The previous expression can be extended to consider balls of arbitrary radius r, that is, solutions at Hamming distance r or lower from x. These results generalize the Grover's wave equation (1), extending the computation of the average to spheres and ball of arbitrary size.

But the results of Sutton *et al.* are deeper and allow one to compute higher order moments, not only the average. It is possible to compute the second, third, fourth and, in general, any moment in polynomial time. Using this results it is possible to compute the variance, the skewness, the kurtosis, etc. With the help of these statistics they are able to approximate the probability distribution of the fitness function around a solution [18]. From a practical point of view the computation of these statistics has been used to propose new search strategies [13], new operators [6] and strategies to escape from plateaus [17].

Using as a base the results by Sutton *et al.*, Chicano and Alba [5] proved that the expected value of the fitness function obtained after applying the bit-flip mutation operator to a solution x is given by the following formula:

$$\mathbb{E}[f]_x = \bar{f} + \sum_{j=1}^{n} (1-2p)^j f_{[j]}(x), \tag{14}$$

where p is the probability of flipping a bit in the string. We can observe that $\mathbb{E}[f]_x$ is a polynomial in p. If we fix x it is possible, in theory, to compute the value p to optimize the expected value of the fitness function after applying the mutation. This way, we could provide a value of p tuned for each particular solution x and optimal with respect to the expectation of the fitness. This adaptive mutation operator has been analyzed by Sutton *et al.* in [19] and the results suggest that it can be useful in the first generations of a genetic algorithm, speeding the search up at the beginning.

3.2 Generalized Hypercubes

We call *generalized hypercube* or q-ary *hypercube* to the configuration space in which the set of solutions X is composed of strings in a finite alphabet Γ with cardinality q; and two solutions x and y are neighboring when they differ in only one component of the solution. This configuration space is a generalization of the binary hypercube in which the components can take q different values instead of two.

There exist the corresponding generalized Walsh functions for this configuration space that form a basis of the function space [11]. However, in recent works we find that these bases are not used in the decomposition of the landscape in elementary components.

Whitley and Sutton [26] prove that the graph coloring problem is an elementary landscape. They use a component model of the problem. Using this model together with some algebraic results, Whitley *et al.* [25] prove that the frequency assignment problem (FAP) can be decomposed in a sum of two elementary components. Graph coloring is a particular case of FAP, Later, Chicano *et al.* [10] prove using algebraic arguments that the more general version of FAP is also a sum of two elementary landscapes and characterize the cases in which the problem is elementary. One of these cases is the one in which the problem represents an instance of the graph coloring problem.

Recent unpublished researches suggest that the results mentioned ni the previous section for the binary hypercube can be generalized to the q-ary hypercube. This way, it would be possible to compute the moments of any order over spheres and balls of arbitrary radius and compute the expected value of a fitness function after applying a mutation to an arbitrary solution.

3.3 Permutations

Last but not least, we consider the case in which the set of solutions X is the one of all the permutations of size n, $X = S_n$. Several configuration spaces have been defined over permutations. We highlight two of them here. The first one uses as neighborhood the set $N(x)$ composed of all the solutions that can be built from x applying a 2-opt move. The second considers as neighbors $N(x)$ the set of all the solutions obtained from x after swapping two positions in the permutation.

Whitley and Sutton [26] prove that the travelling salesman problem (TSP) if an elementary landscape under the 2-opt neighborhood. Later, Whitley and Ochoa [23] derive expressions to optimize the computation of the average fitness function in partial neighborhoods for this problem. They call partial neighborhood a subset of $N(x)$ (the whole neighborhood) for which the Grover's wave equation (1) is valid with small variations.

Chicano *et al.* study the Quadratic Assignment Problem (QAP) over the configuration space induced by the swap neighborhood. They prove in [7] that this problem can be decomposed in three elementary components and propose in [8] an efficient algorithm to compute the autocorrelation measures for the instances of that problem. TSP is a subproblem of QAP and, thus, all the results of QAP can be applied to TSP. In fact, Chicano *et al.* prove that TSP can be decomposed as a sum of, at most, two elementary components. In particular, TSP is elementary when the distances between any two cities are symmetric (or antisymmetric).

4 Landscape Explorer

The researches interested in analyzing a landscape using the previous results should have a deep knowledge of the landscape theory. The main goal of the software tool presented in this section is to hide, as far as possible, mathematical details to the researchers. It is our intention that our software tool, *Landscape Explorer*, to be an ally for these researchers. This tool is available at `http://neo.lcc.uma.es/software/landexplorer`.

The main requirement we considered for the architecture of Landscape Explorer was the extensibility. Landscape theory is not a closed domain, it is growing with contributions of different research groups. Thus, if we want the software tool to be useful we need it to be extensible. We also need the tool to be highly modular, in order to provide new variations or modifications of the tool with small changes in the modules.

The second requirement is the multi-platform support. With this requirement in mind, we could focus on the Java or .NET framework, which allows the developer to focus on the code without taking care of the details realted to the different operating systems or computer architectures.

In order to satisfy the two previous requirements we based Landscape Explorer on the Eclipse *Rich Client Platform* (RCP), which uses Java as programming language. This way, we can think in *Landscape Explorer* as a set of Java modules (*plug-ins*) with dependencies and interactions among them. In order to extend the application with a new feature we only need to implement the new feature as a plug-in and add it to the rest of the application. In the same way, it is possible to remove or replace modules. Some example of other applications based on the RCP platform are: the Eclipse IDE, Azureus, BaZaa and CCTVnet.

In the next sections we present a brief introduction to RCP, we describe the architecture of Landscape Explorer and we present the different procedures currently included in the tool.

4.1 Eclipse RCP

The Eclipse *Rich Client Platform* (RCP) is a set of modules offering services for the developers to build extensible software tools. RCP is implemented in top of *Equinox*, which is an implementation of the *OSGi R4 core framework specification*. The goal of the OSGi specification, and the Equinox implementation, is to offer a platform for the development of *software ecosystems*, which are a set of modules, called *bundles* in the OGSi jargon and *plug-ins* in the *Eclipse* jargon, with dependencies and interactions among them. Each module must specify which modules are required and what is the offered functionality to other modules. *Equinox* provides a runtime environment for all the modules to link among them and cooperate to perform their tasks.

In addition to the *Equinox* services, RCP provides two *plug-ins*, `org.eclipse.ui` and `org.eclipse.core.runtime`, that offer class libraries to create graphical user interfaces (using the SWT and JFace libraries), access the functionalities exported by other plug-ins, extend the functionalities of other plug-ins and run CPU-intensive tasks. An application developed using RCP is composed of a set of *plug-ins* (including the mentioned ones) and an executable file which is in charge of indicating the main class of the application (see Figure 1).

The main extension mechanism in RCP is based on the concepts of *extension point* and *extension*. An *extension* represents a new functionality that is added to the software. This extension must *extend* one existing *extension point*. For the extension to be correct, it must provide the information required by the extension point in the correct format. This information can be from text strings to Java classes implementing a given interface.

The plug-ins can define extensions and extension points. If a plug-in defines an extension point it helps to extend the software. On the other hand, the plug-ins that define extensions contribute to extend the functionality of the software. The description of the extensions and extension points of a plug-in can be found inside the `jar` file of the plug-in and the runtime environment uses the description file to provide the information to the interested plug-ins. However, it is the plug-in responsibility to check that other plug-ins extended the extension points.

In addition to the extensions and extension points, the plug-ins can export packages. This way, it is possible to include and export libraries in the plug-ins.

4.2 Architecture of Landscape Explorer

It is not possible to predict the kind of analysis that will be needed in the future to work with the landscape theory. For this reason, the architecture of Landscape Explorer admit almost any possible extension. Only a minimum set

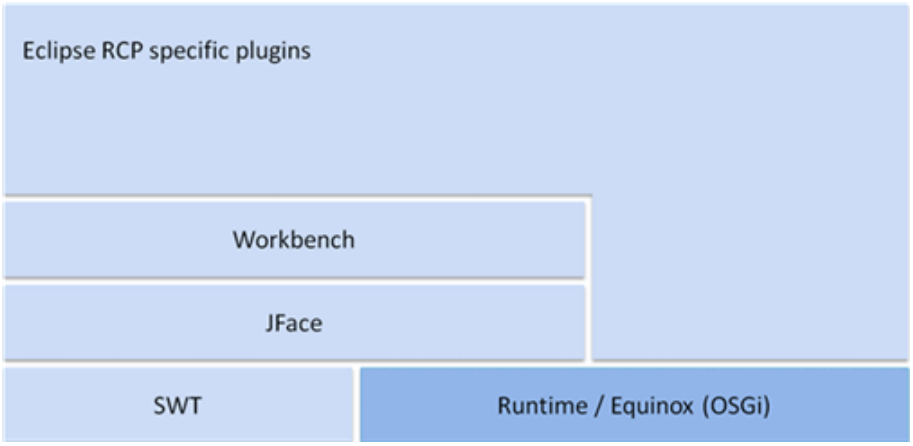

Figure 1: Architecture of a generic RCP application.

of classes and procedures have been defined. However, even this minimum set could change and the tool is ready for it.

The basic classes and procedures are distributed in two plug-ins (Figure 2):

- `neo.landscapes.theory.kernel`: provides the interfaces and classes defining the landscape (from the point of view of the implementation) and offers several sample implementations.
- `neo.landscapes.theory.tool`: is the *plug-in* containing the main class of the application and takes care of the graphical user interface (GUI). It also defined the main extension points of the application.

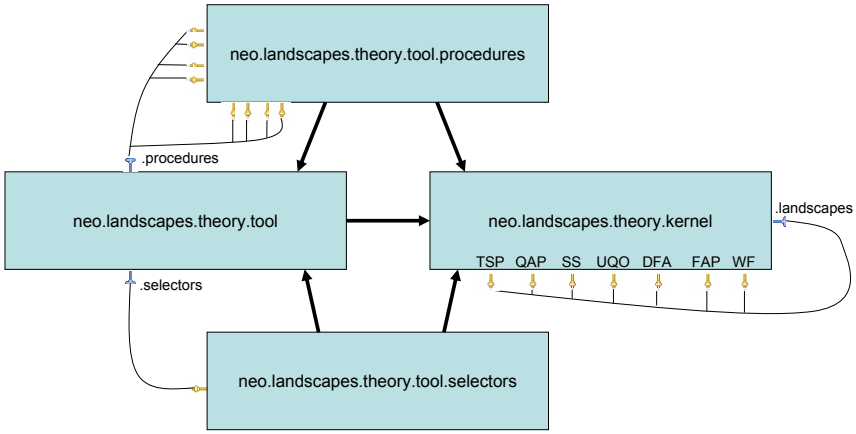

Figure 2: Plug-ins of the application.

The first *plug-in*, that we call `kernel`, defines the extension point `neo.landscapes.theory.kernel.landscapes`, which must be extended by the plug-ins defining landscapes. When a researcher want to apply all the analysis implemented in the tool to a concrete problem s/he must implement a plug-in providing an extension to this extension point. At this moment, the kernel plug-in defines six extensions, one for each optimization problem mentioned in Section 4.3.

The second plug-in, called `tool` for short, defines the next extension points:

- `neo.landscapes.theory.tool.procedures`: allows another plug-ins to define new analysis methods. The

plug-ins extending this extension point must provide a class implementing the interface `IProcedure`. This class is in charge of defining a GUI component for the user to interact with the procedure.

- `neo.landscapes.theory.tool.selectors`: one of the basic steps before applying a procedure to a landscape is to select the landscape. This extension point allows the developers to provide procedures for the selection of landscapes. Those developers interested in providing a selection procedure must provide a class implementing the interface `ISelector`.

The `tool` plug-in also exports a class, called `SelectionServices`, that offers public operations to select *landscapes*. This class should be used by the analysis procedures. When an analysis method needs the user to select a landscape, this procedure must obtain the landscape after a query to the `SelectionServices` class, which delegates this selection to one of the selection procedures registered in the application.

In order to add a new landscape, analysis procedure or selection procedure, the developer must create a plug-in defining the corresponding extension and put the `jar` file int he *plugins* folder of the application.

4.3 Built-in Functions

In this section we present the selection procedure and the analysis procedures included in the tool at this moment.

The third plug-in included in *Landscape Explorer* is `neo.landscapes.theory.tool.selectors`. This plug-in contains a basic selection procedure. This procedure shows the different landscapes defined as extensions in the `neo.landscapes.kernel.landscapes` extension point. These landscapes are seven:

- *Quadratic Assignment Problem* (QAP): is a classical optimization problem. The user must provide the path to a text file with the instance in the QAPLIB format [3].
- *Travelling Salesman Problem* (TSP): the user must provide the path to a file with the instance.
- *Unconstrained Quadratic Optimization* (UQO): this is an NP-hard optimization problem [12]. The user must provide the path to a text file with the instance, containing the coefficient matrix.
- *Subset Sum* (SS): the user must provide the size of the problem and a random seed. The instance of the problem will be randomly generated.
- *Frequency Assignment Problem* (FAP): this is a telecommunication problem related to the graph coloring problem. The user must provide the number of nodes and frequencies.
- *DNA Fragment Assembly* (DFA): is a problem in the domain of computation biology related to the QAP in which several objective functions are possible. The user must provide the size of the problem and the objective function to use.
- *Walsh Functions* (WF): controlled and weighted sum of Walsh functions. This landscape is not a concrete optimization problem. However, any problem defined over the binary hypercube can be written as a weighted sum of Walsh functions. The user must provide the size of the search space, the order of the Walsh functions and a random seed.

The last plug-in included in the application is `neo.landscapes.theory.tool.procedures`. This plug-ins defines several analysis methods. Some of them can be applied to all the landscapes. They are:

- **Empirical measure of the autocorrelation**. This method performs a random walk in the search space jumping from a solution x to a neighbor $y \in N(x)$. At the same time it collects some information to compute the values of the Weinberger autocorrelation function [22].
- **Elementary landscape check**. This method performs a random sample of a set of solutions and their neighboring ones to check if the landscpae is elementary. The method is not exhaustive. Thus, if the answer is "yes" we cannot be sure that the landscape is elementary. However, if the answer is "no", we can be sure that the landscape is not elementary. When the answer is positive it provides the tentative eigenvalue and the average over the whole search space of the fitness function.

- ***Mathematica* program**. Given an arbitrary *landscape*, this method prepares a script for Mathematica to decompose the landscape into elementary components. This method is a key piece in the methodology for the elementary landscape decomposition of combinatorial optimization problems [9].
- **Estimation of the number of elementary components**. Given a landscape defined over the binary hypercube, this method determines in an empirical way the number of elementary components of the function or any power of the function. The result is not an exact value but can be near the exact one as the number of random samples is increased.

In addition to the previous procedures, we also include some methods to exactly compute the autocorrelation measures for some specific problems. They are QAP, TSP, UQO and SS. In all the cases the methods take an instance of the problem and compute the autocorrelation coefficient, the autocorrelation length and the Weinberger autocorrelation function [22].

5 Conclusions

The goal of this document is twofold. First, it provides a brief introduction to the landscape theory and presents the main findings of the last years. Second, a new software tool is presented to ease the landscape analysis.

The researcher interested in landscape theory can use the software tool to study the optimization problems and find properties of them. They can also empirically check some theoretical results. The researchers interested in a given problem (landscape) can use the tool to find the elementary landscape decomposition of the problem and apply practical results that can be derived from such decomposition.

At this moment, *Landscape Explorer* is a prototype that can grow very fast with the help of other researchers interested in it. We plan to include in the tool all the new analysis procedures that will appear as the theory of landscape advances.

References

[1] E. Angel and V. Zissimopoulos. On the classiffication of NP-complete problems in terms of their correlation coefficient. *Discr. App. Maths.*, 99:261–277, 2000.

[2] Türker Biyikoglu, Josef Leyold, and Peter F. Stadler. *Laplacian Eigenvectors of Graphs*. Lecture Notes in Mathematics. Springer-Verlag, 2007.

[3] R.E. Burkard, S.E. Karisch, and F. Rendl. QAPLIB - a quadratic assignment problem library. *Journal of Global Optimization*, 10:391–403, 1997.

[4] Francisco Chicano and Enrique Alba. Elementary landscape decomposition of the 0-1 unconstrained quadratic optimization. *Journal of Heuristics*, 2011. in press (doi: 10.1007/s10732-011-9170-6).

[5] Francisco Chicano and Enrique Alba. Exact computation of the expectation curves of the bit-flip mutation using landscapes theory. In *Proceedings of GECCO*, pages 2027–2034, 2011.

[6] Francisco Chicano, Javier Ferrer, and Enrique Alba. Elementary landscape decomposition of the test suite minimization problem. In *Proccedings of SSBSE*, volume 6956 of *LNCS*, pages 48–63, 2011.

[7] Francisco Chicano, Gabriel Luque, and Enrique Alba. Elementary landscape decomposition of the quadratic assignment problem. In *Proceedings of GECCO*, pages 1425–1432, New York, NY, USA, 2010. ACM.

[8] Francisco Chicano, Gabriel J. Luque, and Enrique Alba. Autocorrelation measures for the quadratic assignment problem. *Applied Mathematics Letters*, 2011. in press (doi :10.1016/j.aml.2011.09.053).

[9] Francisco Chicano, L. Darrell Whitley, and Enrique Alba. A methodology to find the elementary landscape decomposition of combinatorial optimization problems. *Evolutionary Computation*, 19(4):597–637, 2011.

[10] Francisco Chicano, L. Darrell Whitley, Enrique Alba, and Francisco Luna. Elementary landscape decomposition of the frequency assignment problem. *Theoretical Computer Science*, 412(43):6002–6019, 2011.

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

[12] Fred Glover, Bahram Alidaee, César Rego, and Gary Kochenberger. One-pass heuristics for large-scale unconstrained binary quadratic problems. *European Journal of Operational Research*, 137(2):272–287, Mar 2002.

[13] Guanzhou Lu, Rami Bahsoon, and Xin Yao. Applying elementary landscape analysis to search-based software engineering. In *Proceedings of SSBSE*, pages 3–8, 2010.

[14] Christian M. Reidys and Peter F. Stadler. Combinatorial landscapes. *SIAM Review*, 44(1):3–54, 2002.

[15] 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, 1995.

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

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

[18] Andrew M. Sutton, Darrell Whitley, and Adele E. Howe. Approximating the distribution of fitness over hamming regions. In *Proceedings of the 11th workshop proceedings on Foundations of genetic algorithms*, FOGA '11, pages 93–104, New York, NY, USA, 2011. ACM.

[19] Andrew M. Sutton, Darrell Whitley, and Adele E. Howe. Mutation rates of the (1+1)-EA on pseudo-boolean functions of bounded epistasis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 973–980, 2011.

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

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

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

[23] Darrell Whitley and Gabriela Ochoa. Partial neighborhoods of the traveling salesman problem. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 529–536, New York, NY, USA, 2011. ACM.

[24] Darrell Whitley, Andrew M. Sutton, and Adele E. Howe. Understanding elementary landscapes. In *Proceedings of GECCO*, pages 585–592, New York, NY, USA, 2008. ACM.

[25] L. Darrell Whitley, Francisco Chicano, Enrique Alba, and Francisco Luna. Elementary landscapes of frequency assignment problems. In *In Proceedings of GECCO*, pages 1409–1416, 2010.

[26] L. Darrell Whitley and Andrew M. Sutton. Partial neighborhoods of elementary landscapes. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 381–388, New York, NY, USA, 2009. ACM.