

Multidimensional Knapsack Problem: The Influence of Representation

Jorge Tavares

jast@dei.uc.pt

Centre for Informatics and Systems of the University of Coimbra
Polo II - Pinhal de Marrocos
3030 Coimbra, Portugal

Francisco B. Pereira

xico@dei.uc.pt

Polytechnic Institute of Coimbra
Quinta da Nora
3030 Coimbra, Portugal

Ernesto Costa

ernesto@dei.uc.pt

Centre for Informatics and Systems of the University of Coimbra
Polo II - Pinhal de Marrocos
3030 Coimbra, Portugal

Abstract

Fitness landscape analysis techniques are used to better understand the influence of genetic representations and associated variation operators when solving a combinatorial optimization problem.

Five representations are investigated for the Multidimensional Knapsack problem. Common mutation operators (like bit-flip mutation) and classic 1-point and uniform crossover are employed to generate fitness landscapes. Measures such as fitness distance correlation and autocorrelation are applied to examine the landscapes associated with the tested genetic encodings. Furthermore, additional experiments are made to observe the effects of adding heuristics and local optimization to the representations.

The goal of this work is to study the influence of representations on the design of efficient evolutionary algorithms. A comprehensive set of experiments is performed to accomplish this task. Results show that the selection of a suitable representation is crucial when solving combinatorial optimization problems. Encodings with a strong heuristic bias are more efficient and the addition of local optimization techniques further enhance its performance. Finally, results show that, for the Multidimensional Knapsack problem, the choice of a representation influences the selection of mutation.

Keywords

Representation, fitness landscape analysis, mutation, crossover, heuristic bias, local optimization, multidimensional knapsack problem.

1 Introduction

Evolutionary algorithms are efficient techniques to discover good quality solutions for difficult combinatorial optimization problems. The choice of a suitable representation plays a crucial role in the design of these biological inspired techniques and is a key issue to improve search performance.

Most of the research in this area deals with the application of an evolutionary algorithm to a specific problem. There are usually two lines of approach. The first one is to apply a standard implementation of an evolutionary algorithm to a new problem. The other deals with proposing a novel technique that improves the performance of previous approaches. In most of these investigations, the focus is on showing *how* the evolutionary algorithm was able to solve the problem, i.e., finding new or improved candidate solutions, and not in *why* it was able to solve it. Describing and explaining why the proposed approaches have that behavior, or how they affect its dynamics and how they compare to different methods, is something not often explored.

The main goal of this work is to study the influence of representations on the design of efficient evolutionary algorithms for combinatorial optimization problems. The Multidimensional Knapsack problem (MKP) will be used as a benchmark for our study. We will use fitness landscape analysis to conduct a comprehensive investigation on the properties of different representations commonly adopted when evolutionary algorithms are applied to this class of problems.

Fitness landscapes (Wright, 1932) illustrate the association between the search space and the fitness space. An evolutionary algorithm can be seen of as navigating a landscape in order to find the highest peak. Higher points in the search space correspond to solutions with higher fitness. A representation and associated variation operators influence the search efficiency of an evolutionary algorithm. With this in mind, the choice of a representation and operators can be made based on the study of the difficulties of the corresponding fitness landscapes. Since fitness landscape analysis techniques were introduced they have become a valuable tool to investigate why a given evolutionary algorithm works.

The Multidimensional Knapsack problem is a well-known combinatorial optimization problem for each several successful applications of evolutionary algorithms exist, e.g. (Chu and Beasley, 1998). For this problem there is a large selection of representations, which implies that a careful analysis of representation, variation operators and other techniques, such as the use of heuristics and local optimization, is essential to the design of an effective algorithm for this problem.

In this work we present our investigations towards a better understanding of the role of representation and genetic operators when evolutionary algorithms are applied to the MKP. The study presented in this article continues and expands our initial work on this topic (Tavares et al., 2006). We add significant new results considering the application of fitness landscapes analysis on the MKP. In addition to the research on landscapes produced by mutation operators, the analysis now includes landscapes generated by crossover. To study these kind of landscapes, appropriate measures are presented. To the best of our knowledge, this is the first study regarding a complete fitness landscapes analysis on the MKP.

The paper is structured as follows: a description and formal definition of the problem is presented in section 2; section 3 presents an overview of evolutionary techniques applied to the MKP; in section 4 we explain the concept of fitness landscapes and describe the analysis techniques used for this work; experimental results and discussion are reported in section 5; finally, in section 6 we present some conclusions.

2 The Multidimensional Knapsack Problem

The MKP is a well-known NP-Hard combinatorial optimization problem, with a wide range of applications such as cargo loading, cutting stock problems, resource allocation in computer systems and economics (Martello and Toth, 1990).

This problem is also known in the literature as the *M-Dimensional Knapsack Problem*, the *Multiconstraint Knapsack Problem*, the *Multi-Knapsack Problem* and the *Multiple Knapsack Problem*. Additionally, some authors also include in their name the term *zero-one*, e.g., the *Multidimensional zero-one knapsack problem*. Using alternative names for the same problem is potentially confusing but since historically, this designation *Multidimensional Knapsack Problem* has been the most widely used (Chu and Beasley, 1998), we adopt this same naming.

The problem can be described as follows: given two sets of n items and m knapsack constraints (or resources), for each item j a profit p_j is assigned and for each constraint i a consumption value r_{ij} is designated. The goal is to determine a set of items that maximizes the total profit, not exceeding the given constraint capacities c_i . Formally, it is stated as:

$$\text{maximize} \quad \sum_{j=1}^n p_j x_j \quad (1)$$

$$\text{subject to} \quad \sum_{j=1}^n r_{ij} x_j \leq c_i, \quad i = 1, \dots, m \quad (2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (3)$$

$$\text{with} \quad p_j > 0, \quad r_{ij} \geq 0, \quad c_i \geq 0 \quad (4)$$

The decision variable is the binary vector $x = (x_1, \dots, x_n)$. Each item j is mapped to a bit. When $x_j = 1$, the corresponding item is considered part of the solution. The special case of $m = 1$ is generally known as the *Knapsack Problem*, or the *Unidimensional Knapsack Problem*, and is solvable in pseudo-polynomial time (it is only weakly NP-Hard). Nonetheless, for $m > 1$ the problem is strongly NP-Hard (Garey and Johnson, 1979). This yield that exact techniques and exhaustive search algorithms, e.g., branch-and-bound, are only of practical use to solve MKP instances of small size, since they are in general too much time-consuming.

This problem is included in the general class of covering and packing problems. According to Gottlieb (1999a), these two types of problems are structurally equivalent since we can locate the global optima on the boundaries of the feasible regions. In the particular case of the MKP, the feasible solutions contained on the boundary cannot be improved since the insertion of more items will cause the violation of resource capacities.

The MKP has been widely studied for the past few years and many theoretical and empirical studies exist for a different number of knapsack problem variants. For a comprehensive review of these techniques, including exact methods and heuristics, consult (Martello and Toth, 1990), (Chu and Beasley, 1998) or (Kellerer et al., 2004). In the next section, we will present and discuss the most common evolutionary approaches to the problem.

3 Evolutionary Approaches

Evolutionary algorithms have been widely applied to the MKP and shown to be effective for searching and finding good quality solutions (consult e.g. (Chu and Beasley, 1998)). Regardless of that success, designing an efficient evolutionary algorithm for the MKP is a difficult task where the issue of choosing an appropriate constraint-handling technique is important. As Gottlieb (1999a) showed, the success of an evolutionary algorithm for the MKP is strongly dependent on the ability of the algorithm to restrict, or focus, the search on the boundary of the feasible region.

As such, the most successful evolutionary algorithms are based on repairing and local optimization techniques or heuristic decoders, as for example, the approaches developed by Chu and Beasley (1998) and Raidl (1998). The generality of the proposed algorithms found in the literature, Chu and Beasley (1998), Michalewicz and Arabas (1994), Hinterding (1994), Thiel and Voss (1994) and Gottlieb (2000), can also be separated in two groups according to the adopted representation and the associated variation operators. Evolutionary algorithms may use a direct representation, encoded as a binary string, where each bit is mapped to an item. A bit set to 1 indicates that the corresponding item is packed into the knapsack. Moreover, evolutionary algorithms may adopt indirect encodings, such as permutations or weight-codings. In this case, the algorithm needs a decoder that translates the chromosome into the actual solution.

The choice of the representation and variation operators is a pivotal decision when designing an evolutionary algorithm. The role of heuristics and local optimization is essential for an efficient evolutionary approach for the MKP. Still, the selection of the representation and variation operators can also play a significant part on the outcome of the algorithm (Raidl and Gottlieb, 2005).

Next, we briefly describe the most common representations for this problem and a simple description of some significant works regarding evolutionary algorithms and MKP. The focus is on genetic representations; for each one of them, we present associated variation operators, as well as decoding procedures for the indirect representations.

3.1 Genetic Representations

Binary Representation

With this encoding, a solution is represented by a characteristic bit vector, where each bit is mapped to an item. A bit set to 1 indicates that the corresponding item is packed into the knapsack. There are two options to deal with infeasible solutions: penalty-based fitness functions (consult Gottlieb (2001) for a comprehensive study on fitness functions for the MKP) or repair operators. Chu and Beasley (1998) proposed a repair mechanism, that iteratively removes items until all constraints are satisfied. In addition, each solution is improved by local optimization. Both methods are guided by a heuristic that orders the items according to the ratio of profit and resource consumption. This approach was later improved by Raidl (1998), by using the heuristic associated with the weight-coding representation, described later. For a binary representation, classical crossover and mutation operators can be used, such as n-point crossover and uniform crossover, and bit-flip mutation, which may cause constraint violations.

Ordinal Representation

For the MKP, a chromosome is a vector $v = (v_1, \dots, v_n)$, where each position v_k belongs to the set $\{1, \dots, n - k + 1\}$ for $k \in \{1, \dots, n\}$. The vector is mapped to a permutation π of the items. This permutation is built in the following manner: an ordered list $L =$

(L_1, \dots, L_n) is created, initialized with all the items; next, vector v is traversed in order, from first to last position, and each v_k specifies a position in L ; the referenced element, L_{v_k} , is removed from L and is inserted in permutation π , thus representing element π_k . As an example, assume $v = (1, 2, 3, 2, 1)$ and the initial ordered list $L = (1, 2, 3, 4, 5)$. Vector v is interpreted by successively removing the elements 1, 3, 5, 4, 2 from L . This process generates the permutation $\pi = (1, 3, 5, 4, 2)$.

To decode the permutation into a feasible MKP solution, a *first-fit* heuristic is applied. The heuristic works by building a feasible solution traversing all variables in the order given by the permutation π . An item π_j is inserted into the solution if it does not violate any constraint. To be more precise, we start with a feasible solution, representing an empty solution, $x = (0, \dots, 0)$ and each item in the order given by the permutation π . Then, each corresponding decision variable x_{π_j} , with $j = 1, \dots, n$, is altered from 0 to 1 if the insertion of item π_j does not violate any constraint.

This representation has the advantage to enable the use of standard genetic operators. Classical crossover operators, e.g. uniform crossover, can be used. The mutation operator randomly chooses a position $k \in \{1, \dots, n\}$ and then picks a new value, v_k , using a uniform distribution, from the set $\{1, \dots, n - k + 1\}$.

Evolutionary algorithms for the MKP using this representation can be found in (Michalewicz and Arabas, 1994) and (Gottlieb and Raidl, 2000). These studies report poor results in comparison with other representations.

Permutation Representation

Permutations are typical representations for scheduling and routing problems. Additionally, they have been widely applied to the MKP (Raidl, 1998), (Hinterding, 1994), (Thiel and Voss, 1994) and (Gottlieb, 2000).

The representation consists of a permutation of all items, $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ denoted by $\pi = (\pi_1, \dots, \pi_n)$. Decoding it into a feasible solution is done by a *first-fit* heuristic, in the same manner as in the ordinal representation. The representation needs genetic operators that can preserve the permutations, such as uniform order based crossover (Gottlieb and Raidl, 2000) or partially-match crossover, and swap mutation (Michalewicz, 1992). Permutation-based evolutionary algorithms have achieved good results when applied to the MKP, as reported by the works of Hinterding (1994) and Gottlieb (2000).

Random-Key Representation

Proposed by Bean (1994) the random-key representation is an alternative approach to encode permutations without the need for specific operators. A random-key is a vector of real values $w = (w_1, \dots, w_n)$ where each position represents an item j , assigned by a weight $w_j \in [0, 1]$. The decoder works by sorting the real-valued vector, yielding a permutation π with the associated weights indexes. Once again, the attained permutation is decoded by means of a *first fit* heuristic. As stated before, this representation allows the use of the classical crossover and mutation operators. In the case of mutation, the operator usually performs positional mutation, i.e, it randomly draws values from the interval $[0, 1]$, according to a uniform or normal distribution, replacing a value in a random position in the chromosome.

Evolutionary approaches based on random-keys for MKP can be found in (Hinterding, 1994) and (Hinterding, 1999). The reported results show that evolutionary approaches based on this representation can achieve good results.

Weight-Coding Representation

The weight-coding representation, a general technique successfully applied to a variety of combinatorial optimization problems, is the most successful decoder-based technique for the MKP. It was first used by Cotta and Troya (1998) and later improved by Raidl (1999).

In this representation, a chromosome consists of a real-valued vector of weights $w = (w_1 \dots, w_n)$, where each item j of the MKP is associated with a weight $w_j \in [0, 1]$. The decoding process of the genotype to the phenotype is made of two steps. The first step consists of transforming the original problem P into a modified problem P' by multiplying the original items profits with the associated weight, i.e., biasing the original problem. The last step requires the use of a fast heuristic to find a solution to P' and to evaluate it according to the original problem.

Several decoding heuristics and techniques for biasing the original problem have been studied (Raidl, 1999). The heuristics that often work best are based on surrogate and lagrangian relaxation techniques. The decoding heuristic using the surrogate relaxation method is preferred due to its lower computational requirements. The original problem is simplified by transforming all m constraints into a single constraint:

$$\sum_{j=1}^n \left(\sum_{i=1}^m a_i r_{ij} \right) x_j \leq \sum_{i=1}^m a_i c_i \quad (5)$$

where $a_i, i = 1, \dots, m$, is the surrogate multiplier for the i th constraint. To derive the surrogate multipliers, one of the simplest methods is to solve the Linear Programming (LP) relaxation of the original problem (i.e., the decision variables x_i can take any value $\in [0, 1]$) and to use the values of the *dual variables* as the surrogate multipliers.

To obtain a heuristic solution to the MKP, the *profit/pseudo-resource consumption ratios*, u_j , are calculated as:

$$u_j = \frac{p_j}{\sum_{i=1}^m a_i r_{ij}} \quad (6)$$

A lower pseudo-utility ratio heuristically indicates that an item has a lower profit with a higher resource consumption, whilst a high ratio reflects a more efficient item. According to the u_j values, we sort the items in a decreasing order, adding them to the solution one at a time if none of the constraints are violated. This process is similar to the *first-fit* heuristic previously described in the decoding steps for encodings where permutations are involved.

The surrogate multipliers are only calculated once for the original problem at the beginning, as a preprocessing step. This is done to ensure low computation costs and as a result of this step, the decoding process begins by determining the u_j values.

This representation allows the use of standard crossover and mutation variation operators. Like the random-key representation, positional mutation is used. For details of this method, please refer to (Raidl, 1999). This approach has attained good results when applied to the MKP.

Other Representations

The encodings described before represent the majority of evolutionary representation developed and studied for the MKP. Nonetheless, it is possible to find in the literature other works which include different codifications. These are not analyzed in this paper but we will describe them for the sake of completeness.

In addition to permutation-based representations, Hinterding (1994) also used in his work a *variable-length representation*. This representation contains a list of items that fit into the knapsack. The items are inserted according to the ordering that they appear, which means that this encoding is similar to a permutation of variable size. The proposed genetic operators are specific, e.g. the injection crossover (Hinterding, 1994). Results attained are inferior to permutation-based approaches since it is possible to produce feasible individuals to which further items could be added.

Genetic programming over context-free languages with linear constraints was investigated by Bruhn and Geyer-Schulz (2002) who presented several results on variants of the MKP. In this case, a solution is encoded by a *derivation tree* produced by the grammar. An item is selected by a leaf node and each node contains the capacity of the sub-tree it represents (the used and free). This method allows the verification of the capacity constraints for each sub-tree. Standard genetic programming variation operators are used, with an initialization routine which produces a *derivation tree* by applying a randomly chosen rule of the grammar. This routine is computationally expensive since infeasible solutions are rejected until a feasible one is created. Results reported by the authors (Bruhn and Geyer-Schulz, 2002) suggest that this approach is superior to penalty-based evolutionary algorithms but Raidl and Gottlieb (2005) claim otherwise: the inferior performance of the evolutionary algorithm used by Bruhn and Geyer-Schulz (2002) in their study has its source on the chosen suboptimal configuration made by the authors.

Despite the fact of not presenting a complete different representation for the MKP, it is worth mentioning the work presented by Cleary and O'Neill (2005). Here, the authors describe how the standard genotype-phenotype mapping process of grammatical evolution can be enhanced with an attribute grammar to allow it to operate as a decoder-based evolutionary algorithm. The use of an attribute grammar maintains a context-sensitive and semantic information of the capacity constraints of the MKP, in which the attribute grammar specification is used to perform decoding similar to a first-fit heuristic. Presented results are encouraging but not superior to previous evolutionary approaches.

3.2 Analysis of Evolutionary Algorithms for the MKP

Although many different evolutionary algorithms have been applied to the MKP, there are just a few studies that aim to perform a comprehensive analysis of the behavior of such techniques.

The most significant work on this subject was done by Raidl and Gottlieb (Raidl and Gottlieb, 2005). In this study, five representations (four indirect decoder-based representations and one direct representation) are examined according to important aspects in an evolutionary algorithm, such as initialization and variation operators. The study comprises a general comparison performance of the five encodings, as well as a comprehensive empirical investigation of different characteristics of an evolutionary algorithm. In this case, three properties, locality, heritability and heuristic bias, are considered. A set of static measures is used to gain insights about the evolutionary algorithm behavior when using the tested representations. This investigation helped to confirm, extend and unify previous studies about the MKP, (Raidl and Gottlieb, 1999), (Gottlieb, 1999b), (Gottlieb, 2000) and (Gottlieb and Raidl, 2000).

In a research initiated by Branke et al. (2005, 2006), a dynamic version of the MKP is proposed and analyzed according to the changing fitness landscape. This work is a contribution to a better comprehension of evolutionary algorithms when facing chang-

ing environments. To the best of our knowledge, this is the first work regarding this variant of the MKP and as such, results and evolutionary efficiency are not comparable to the algorithms for static versions of the problem.

In a previous paper (Tavares et al., 2006), we used the fitness landscape framework to initiate a study with five representations for the MKP. The analysis also contained some repair operators, local optimization and heuristics. The focus of this investigation was on fitness landscapes generated by mutation operators. The analysis of fitness distance correlation and autocorrelation helped to gain some insights on the performance differences achieved by different representations. The study shows that the choice of an encoding without a strong heuristic bias can create some difficulties for an evolutionary algorithm designed for the MKP. Here we will continue the investigation, extending this research to landscapes produced by crossover operators, and also adding a more clear and deep overall analysis. In the next section we will describe our analysis framework.

4 Fitness Landscapes

The concept of *fitness landscape*, introduced by Wright (1932) to demonstrate the dynamics of biological evolutionary optimization, has been useful for the analysis and understanding of evolutionary algorithm's behavior. In addition, the study of fitness landscapes can be of value in designing an evolutionary algorithm since it can help to predict its performance.

Usually, evolutionary search can be represented by three spaces: the *search space*, the *phenotype space* and the *fitness space*. The fitness space reflects the solution quality whilst the search space is made of the candidate solutions. A candidate solution is represented by its genotype, which can be mapped to its phenotype by a decoder function. This is not always true since in the case of direct representations, a candidate solution is only represented by its genotype. The set of all possible genotypes is denominated *genotype space* and it is equivalent to the search space. This equivalence is possible for the reason that, variation operators work in the genotype space and, an evolutionary algorithm searches for genotypes that decode into phenotypes with high fitness.

Fitness landscapes describe the relation between the *search space* and the *fitness space*. Regarding the *search space* as a landscape, an evolutionary algorithm can be seen of as navigating through it in order to find the highest peak of the landscape. The height of a point in the *search space* (the genotype) reflects the fitness of the decoded solution (the phenotype) associated with that point. The structure of a landscape influences the dynamics of an evolutionary algorithm. Since the representation and operators define the manner an evolutionary algorithm can perform more efficiently its search, the choice of its representation and operators for a given problem can be based on the study of the difficulties of the corresponding landscape.

Fitness landscapes can be defined as follows: a set of points (solutions) X , a fitness function f which assigns a numeric value to each solution, and a distance operator d defining the neighborhood within set X . Formally, a fitness landscape is a tuple

$$\mathcal{L} = (X, f, d) \tag{7}$$

with a fitness function defined as

$$f : X \rightarrow \mathfrak{R} \tag{8}$$

Thus, the fitness landscape can be interpreted as a graph

$$G_{\mathcal{L}} = (X, E) \tag{9}$$

with edge set

$$E = \{(x, y) \in X \times X \mid d(x, y) = d_{min}\} \quad (10)$$

Being d_{min} the minimum distance between two points in the search space. The diameter of a landscape $diamG_{\mathcal{L}}$, is the maximum distance in the search space.

As an example, for binary representations with a bit-flip mutation operator, the solution set is $X = \{0, 1\}^n$, with $G_{\mathcal{L}}$ a hypercube of dimension n and the Hamming distance as the distance operator. This yields $d_{min} = 1$ and $diamG_{\mathcal{L}} = n$.

4.1 Measures for Landscape Analysis

As mentioned before, the structure of landscapes influences the ability of an evolutionary algorithm to perform an efficient search. There are several properties that define the structure of fitness landscapes, such as the distribution of the fitness function, the number and distribution of local optima, the structure of the basins of attraction, the distribution of peaks in the search space, the presence and structure of neutral networks, and landscape ruggedness. These characteristics are well known and have been studied in the evolutionary computation community.

4.1.1 Fitness Distance Correlation

One way to measure problem difficulty is determining how close is the relation between fitness value and distance to the nearest optimum, in the search space. Fitness distance correlation, coefficient ϱ , can be estimated by

$$\varrho(f, d) \approx \frac{1}{\sigma(f)\sigma(d)m} \sum_{i=1}^m (f_i - \bar{f})(d_i - \bar{d}) \quad (11)$$

with a given set of points x_i of size m (the random walk length) and $f_i = f(x_i)$ the fitness value and $d_i = d_{opt}(x_i)$ the minimum distance to a global optimum solution. The \bar{f} and $\sigma(f)$ are the mean and standard deviation. The search should be easy, for selection-based algorithms, when fitness increases as the distance to the optimum decreases. This indicates the existence of a *path* via solutions with increasing fitness values. A value of -1.0 for ϱ shows that fitness and distance to the optimum are perfectly related, thus the search is easy. A value of $\varrho = 1.0$ indicates the opposite.

4.1.2 The Autocorrelation Function

The structure of a fitness landscape can be examined by measuring the degree of correlation between points on the landscape. The degree of correlation depends on the difference between the fitness values of the points. Smoother landscapes are highly correlated, making the search for an evolutionary algorithm easier. This is the result of similar fitness values. If the difference of fitness values is higher, the landscape is less correlated which implies a rugged landscape, thus being harder to search in it. To measure the ruggedness of a fitness landscape we calculate the *autocorrelation function* $\rho(d)$, which measures the correlation of all pairs of points in the search space with distance d . It can be estimated by computing a large sample of solutions:

$$\rho(d) = \frac{1}{\sigma^2(f)|X^2(d)|} \sum_{(x,y) \in X^2(d)} (f(x) - \bar{f})(f(y) - \bar{f}) \quad (12)$$

where $X^2(d)$ is the set of all pairs of points in the search space with distance d and $|X^2|$ the number of pairs in the set. Another possibility to estimate $\rho(d)$ is to perform

a random walk. In this case, the random walk of a time series $\{f(x_t)\}$ defines the correlation of two points s steps away in a m length random walk:

$$\rho_{rw}(s) \approx \frac{1}{\sigma_f^2(m-s)} \sum_{t=1}^{m-s} (f(x_t) - \bar{f})(f(x_{t+s}) - \bar{f}) \quad (13)$$

If the landscape is *statistically isotropic*, the time series forms a stationary random process, therefore a single random walk is sufficient to obtain ρ_{rw} (Merz, 2000).

4.1.3 Correlation Length

This measure directly represents the ruggedness of a fitness landscape: the higher the value of the correlation length l , the smoother the landscape is; the lower the value of l , the more rugged the landscape is. The correlation length is defined as

$$l = -\frac{1}{\ln(|\rho(1)|)} \quad (14)$$

for $\rho(1) \neq 0$. It is useful to normalize it with the diameter of the landscape:

$$\xi = \frac{l}{\text{diam}G_{\mathcal{L}}} \quad (15)$$

The closer the normalized correlation length is to 1, the higher the correlation is. If $\xi = 0$ or close to 0, there is no correlation.

4.2 Limitations of Fitness Landscapes

Although fitness landscape analysis can be useful in performance prediction and analysis for evolutionary algorithms (Merz, 2000), it has some limitations. Jones and Forrest (1995) suggested fitness distance correlation as a measure for search difficulty and it can, for some problems, correctly predict difficult. However, for problems with an unknown global optima, the measure is no longer directly applicable. It can be approximated but it may lead to incorrect predictions as shown by Altenberg (1997).

Moreover, in spite of properties of fitness landscapes (e.g., the number of local optima in the landscape, the structure of the basins of attraction or the landscape ruggedness) provide some understanding of the environment and the search dynamics of evolutionary algorithms, it still can be a difficult task to predict, even for some, easy problems, the performance of an algorithm as demonstrated by Reeves (1999). Reliable and generally applicable measures of difficulty for fitness landscapes are still needed to be developed and studied (Raidl and Gottlieb, 2005).

Regardless of the limitations found, fitness landscape analysis is a valuable tool to the study and design of evolutionary algorithms, as Merz and Freisleben (1999) have illustrated.

5 Landscape Analysis

In this section, the influence of representations is investigated when solving the MKP. This analysis is performed by conducting an investigation on the fitness landscapes defined by different representations.

We consider five representations: binary, ordinal, permutation, random-key and weight-coding. For each representation we will study landscapes generated by both variation operators: mutation and crossover. Standard mutation operators are used

to create landscapes for each one of the representations. These are: bit-flip mutation, integer flip mutation, swap and uniform flip mutation. For the weight-coding representation, we will first use the uniform flip operator and later we will switch to a different one. This is accomplished to examine the influence of the operator heuristic bias. As for landscapes generated by crossover, standard one-point and uniform crossover operators are applied. The only exception is related to the permutation encoding. In this case, both one-point and uniform crossover are altered to allow the manipulation of permutations.

All representations cover the feasible region of the search space with the exception of the direct encoding. As such, we use a penalty function as recommended by Gottlieb (2001) since it can better guide the search to the feasible region:

$$penalty(x) = \frac{p_{max} + 1}{r_{min}} \times \max\{CV(x, i) \mid i \in I\} \quad (16)$$

Where p_{max} is the maximum profit p_i and r_{min} is the minimum resource consumption $r_{i,j}$, with $I = \{1, \dots, m\}$ and $J = \{1, \dots, n\}$. CV is calculated by:

$$CV(x, i) = \max(0, \sum_{j \in J} r_{ij}x_j - c_i) \quad (17)$$

In this study we consider the binary representation in different forms. First, we will be concerned with the simple binary encoding with penalty-based fitness function, then we will add the *profit/pseudo-resource consumption ratios* to obtain a better solution, and finally, we will add repair and local optimization. This is accomplished to better examine the effect of heuristics and local search on the encoding. It is important to observe the effect of adding these methods to a representation. We begin our analysis on standard and simple encodings that do not use heuristics particularly biased towards fitter phenotypes.

5.1 Experimental Setup

For our analysis, we selected problem instances from two different MKP test suites, available from the *OR-Library*¹, as well as another benchmark suite provided by Glover and Kochenber². We performed a comprehensive set of experiments with available instances from both suites. Due to space limitations, in this paper we present results from some selected instances. Results obtained with other examples follow the same trend.

From the first suite, we will analyze results from instances containing 28 and 50 items, with 10 and 5 constraints respectively (named P01 and P02). From the second dataset, we selected instances with 100 items and 5 constraints and 25% of tightness (designated CB01 and CB02). From the last benchmark suite, we chose the instance with 100 items and 15 constraints (labeled GK01).

A fitness distance analysis requires that the global optima are known (or a very near-global optimum solution). For these instances we solved them to optimality by running a MIP solver from the GNU Linear Programming Kit³. It is important to notice that for our analysis we only consider the existence of one global optimum. Additionally, the distance between the known optimal solution and the candidate solutions in our random walks is calculated at the phenotype level. This simply means that, when

¹<http://mscmga.ms.ic.ac.uk/info.html>

²<http://hces.bus.olemiss.edu/tools.html>

³<http://www.gnu.org/software/glpk/glpk.html>

using indirect encodings, all solutions contained in a random walk are converted to its phenotype. The distance is given by the Hamming distance between the solution and the known optimum solution. For each one of the random walks, we performed 50 runs with 100000 steps.

5.2 Mutation Fitness Landscapes

In table 1 we present, for all five MKP instances, the results of the applied measures, regarding fitness distance correlation and autocorrelation for mutation landscapes. The first column indicates the representation used (BR - binary representation, OR - ordinal representation, PR - permutation representation, RK - random-key representation, WC - weight-coding representation), the column with n indicates the number of items and m the number of resources. The average of the minimum distance between the optimum solution and the best found individual in the random walk can be observed on the next column ($\overline{d_{opt}}$) with the standard deviation in brackets. The next column presents values for the fitness distance correlation (ρ), followed by the correlation (l) and correlation length (ξ) columns. The values in bold give the best representation for a given problem instance.

Table 1: Summary results for mutation landscapes.

Representation	Instance			Measures			
	Name	n	m	$\overline{d_{opt}}$	ρ	l	ξ
BR	P01	28	10	14.03 (0.36)	0.07	7.12	0.25
OR	P01	28	10	12.02 (0.35)	-0.51	4.61	0.16
PR	P01	28	10	12.01 (0.35)	-0.51	7.64	0.27
RK	P01	28	10	12.03 (0.36)	-0.53	10.97	0.39
WC	P01	28	10	7.53 (0.34)	-0.58	14.51	0.52
BR	P02	50	5	25.01 (0.49)	0.09	16.18	0.32
OR	P02	50	5	20.67 (0.57)	-0.45	6.23	0.12
PR	P02	50	5	20.64 (0.53)	-0.41	11.72	0.23
RK	P02	50	5	20.72 (0.54)	-0.28	17.05	0.34
WC	P02	50	5	17.48 (0.46)	-0.37	19.43	0.39
BR	CB01	100	5	49.95 (0.76)	-0.05	35.74	0.36
OR	CB01	100	5	49.70 (0.71)	-0.57	4.09	0.04
PR	CB01	100	5	49.64 (0.71)	-0.53	7.04	0.07
RK	CB01	100	5	49.63 (0.72)	-0.56	8.12	0.08
WC	CB01	100	5	40.85 (0.69)	-0.62	10.85	0.11
BR	CB02	100	5	50.00 (0.64)	-0.02	36.65	0.37
OR	CB02	100	5	49.78 (0.71)	-0.60	4.78	0.05
PR	CB02	100	5	49.81 (0.70)	-0.54	6.89	0.07
RK	CB02	100	5	49.64 (0.67)	-0.53	9.22	0.09
WC	CB02	100	5	40.68 (0.74)	-0.59	10.78	0.11
BR	GK01	100	15	49.95 (0.75)	0.06	27.32	0.27
OR	GK01	100	15	49.99 (0.70)	-0.15	3.95	0.04
PR	GK01	100	15	49.99 (0.77)	-0.12	8.37	0.08
RK	GK01	100	15	49.91 (0.77)	-0.19	8.98	0.09
WC	GK01	100	15	49.41 (0.74)	-0.08	9.33	0.09

The data presented in table 1 shows that the average distances between the best found individual and the optimal solution, for the first two instances (P01 and P02), are similar for the ordinal, permutation and random-key encodings. Binary representation has the worst performance while weight-coding has the best results. In terms of percentage, the binary encoding is always around a value of 50% from the optimum, whilst the ordinal, permutation and random-key representations are around 42%. To complete, weight-coding is on the interval of 26% to 35%. This pattern is not found on the larger instances. For all the instances with $n = 100$, the average distance to the optimum solution is very similar for all tested representations, around the value of 50%. The exception is weight-coding which performs slightly better on instances CB01 and CB02 but on the same level as the other encodings on instance GK01. It is interesting to notice that, as n increases, the gap between the different representations decreases. This effect might be explained, to a certain degree, to the tightness of the different instances. The smaller instances have a higher tightness value than the larger ones. This could be an indication that the heuristic bias, not present in the binary encoding, is important for tighter instances. On the other hand, it can also be owing to the fact of the increasing size of the search space of the larger instances.

When looking at the fitness distance correlation coefficient, the observed pattern is different. One might expect that the weight-coding representation with the uniform flip operator would perform better in comparison to the others encodings, simply because it has a stronger heuristic bias on its decoder. From column ρ , it is clear that this does not happen. The weight-coding scheme only has the best coefficient value for instances PC01 and CB01, and a close second best for instance CB02. For the other two instances, PC02 and GK01, the fitness distance correlation for weight-coding is worse than ordinal and random-key encodings, respectively. The only clear pattern presented for this column is the poor performance of the binary encoding. In fact, contrary to the previous measure, the fitness distance correlation does not improve as n increases. With values near 0.0, ranging from 0.09 to -0.05 , the order of magnitude cannot be compared with the other representations. In a general way, the other four representations have similar values around -0.50 . The clear exception is instance GK01. For the instance with more constraints, the ρ values are all above -0.20 , indicating that for this particular instance, all representations had several difficulties to reach better solutions.

From this table, it is possible to distinguish differences between the decoder-based encodings and the direct encoding, which is not an unexpected result. Differences between the three decoder-based encodings with the first-fit heuristic, and the decoder with the profit/pseudo-resource consumption ratios, are less visible. There is an indication that the weight-coding representation performs slightly better than the others but that might be also dependent on the problem instance.

Moreover, it is also important to observe the fitness distance plots of the several representations. This type of plot presents information about the distribution of the random walk points, in which the fitness is plotted against their minimum distance to an optimum. The fitness distance plot provides additional insight since it contains more information.

Figure 1 contains the fitness distance plots for binary, ordinal, permutation, random-key and weight-coding representations, on the CB02 instance (which is representative for the analyzed MKP instances). From the five plots, it is possible to observe the distribution of the candidate solutions induced by the encodings. The first conclusion that can be drawn from these plots is that none of the representations produce highly correlated landscapes. The landscapes do not have an ideal distribution of

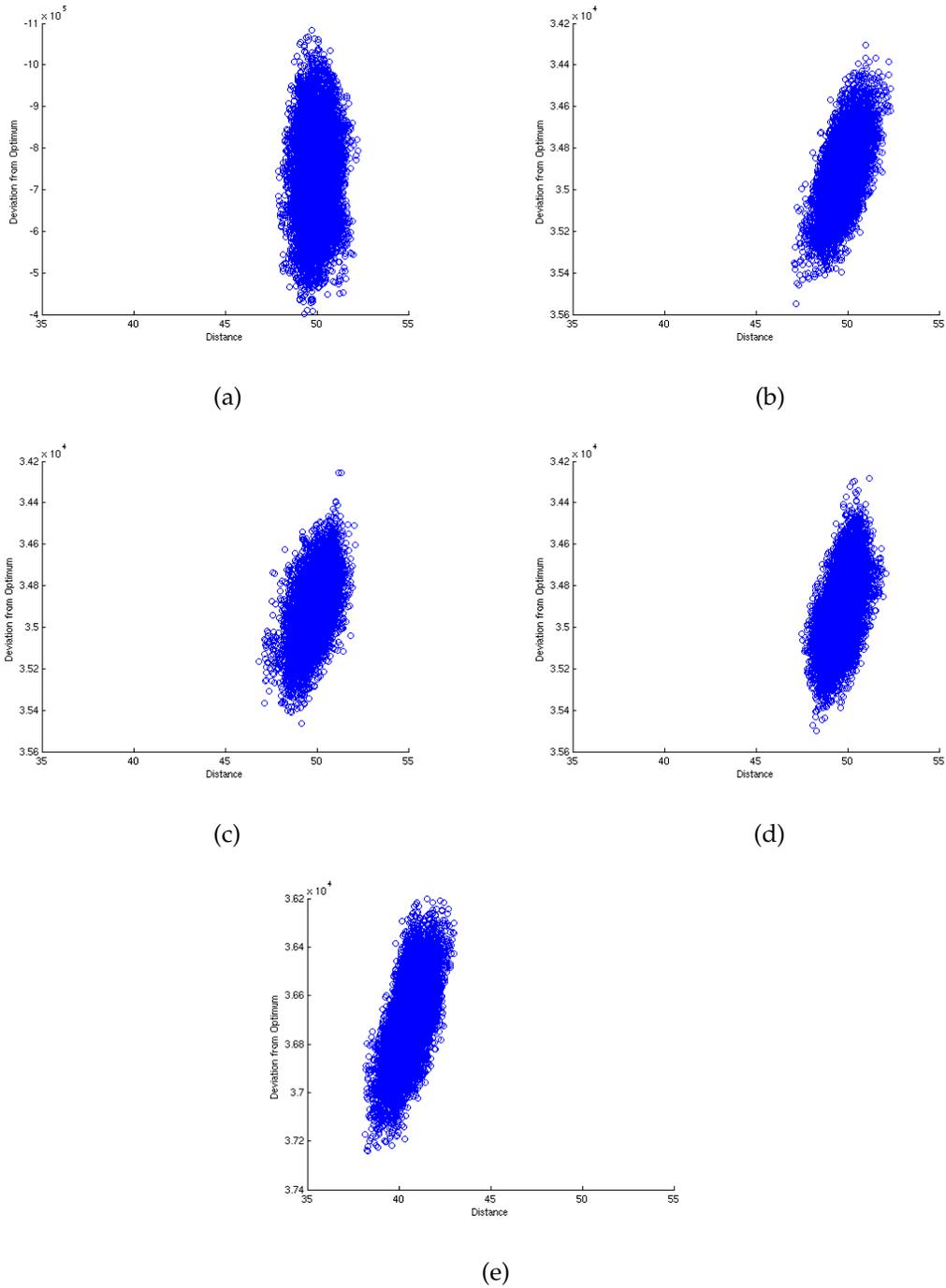


Figure 1: Fitness distance plots for Binary (a), Ordinal (b), Permutation (c), Random-Key (d) and Weight-Coding (e) representations, on the CB02 instance.

points since the optimum solution cannot be found by *jumping* from one solution to a better one with successively decreasing the jump distance.

Although the plots show that fitness and distance are correlated with similar distribution shapes, not all representations present the same configuration. Binary representation does not show a concentration closer to the optimum as the others, which is in accordance with the fitness distance correlation coefficient. We can observe that concentration on the remaining encodings but the weight-coding representation is closer to the optimum. In fact, the fitness distance plot reveals that local minimums are found only in a fraction of the search space with smaller distance to the optimum. All other encodings have a larger distance to the optimum. It is possible to observe that ordinal, permutation and random-key encodings show identical plots.

In order to gain more discernment, fitness distance correlation should be examined in association with the autocorrelation measures. To determine the correlation length we performed a series of random walks to calculate the autocorrelation function, with a phenotypic distance of 1. The neighborhood operator used was based on the Hamming distance between two bit strings (for all representations, after decoding them into a bit string). The results of the autocorrelation analysis are also presented in table 1.

A brief overview of the results reveals a very interesting pattern: for instances P01 and P02, weight-coding representation achieved the highest correlation value, for instances CB01, CB02 and GK01, binary representation attained the highest values. On the other end, ordinal encoding has the lowest autocorrelation values for all problem instances. Another interesting fact is the correlation for all decoder-based representations on instances with 100 items: the values are similar and very low, when comparing to the direct encoding. In fact, binary representation shows consistent values across the instances (between 0.25 and 0.37), and this does not happen for the indirect encodings.

How can these differences be explained? Part of the answer relies on the the type of mutation operator. The lower correlation for the ordinal representation is in compliance with previous studies (Raidl and Gottlieb, 2005). Locality indicates that small variations in the genotype space, usually originated by mutation, imply small variations in the phenotype space. Strong locality allows a search algorithm to efficiently explore the neighborhood of the current solutions, whilst a weak locality prevents evolutionary search from a meaningful exploration of the phenotype space because small variations often cause strong phenotypic changes. The weak locality for this representation means that a single change in one of its genes can cause a major effect on the decoded solution, thus the lower correlation which implies a more difficult search.

In contrast, this does not happen, to such a degree, on the others representations. Here, the flip mutation operator also changes a single gene but the effect on the phenotype is not as dramatic as in the ordinal representation (with the exception of permutations). This effect can explain why ordinal representation has the worst correlation. Furthermore, binary representation achieves higher correlations since the occurrence of mutation can change its genotype without causing a major disruption on its phenotype, i.e., this representation has a higher locality. This effect is more evident on the larger instances.

In this case, weight-coding and binary representations are shown to have better correlated landscapes than the other representations. Taking into consideration the fitness distance correlation analysis, we can conclude that weight-coding representation should provide a good option for solving the MKP, at least, for mutation-based algorithms.

5.3 Crossover Fitness Landscapes

Studying the behavior of mutation-based algorithms using the above mentioned methodology is simple and well suited. The reason is that mutation operators are good to generate random walks for correlation analysis. These operators can be seen as producing a time series of fitness values for a succession of solutions (Merz, 2000). Normally, a mutation operator is more appropriate to a certain landscape if the correlation is higher. With regard to crossover generated landscapes, it is a complete different issue.

According to Merz (2000), various attempts have been made to generalize fitness landscape analysis to crossover operators. The first difficult, when it comes to use crossover as the variation operator, is the fact that it works with solutions, the parents, and creates two new solutions (or one, depending on the operator), the offspring. It is then necessary to define how to apply this operator to generate a time series of values. In this work we followed an approach similar to the one defined by Horjik and Manderick (1995). A time series is simply produced by repeatedly applying crossover to two candidate solutions. One of the solutions is always randomly generated whilst the other one, starts as a random initial solution but later is replaced from the best of the two offspring. Given the parents and the offspring we measure the fitness and distance to produce the landscape.

With this information, the previous fitness distance correlation measure and correlation are updated for crossover analysis. For both measures, we take into consideration the average fitness of both parents, f_p , the average fitness of both offspring, f_o , as well as the average distance of the children to the known optimum solution, d_o . The measures for analysis are redefined in the following way. Fitness distance correlation for crossover, ϱ_{cx} , can be estimated by

$$\varrho_{cx}(f_o, d_o) \approx \frac{1}{\sigma(f_o)\sigma(d_o)m} \sum_{i=1}^m (f_{o_i} - \bar{f}_o)(d_{o_i} - \bar{d}_o) \quad (18)$$

with a given pair of points x_i of size m (the random walk length). As previously defined for mutation-landscapes, a value of -1.0 for ϱ_{cx} shows that fitness and distance to the optimum are perfectly related, indicating an easy search, whilst a value of $\varrho_{cx} = 1.0$ signals the opposite.

As in (Manderick et al., 1991), the parent-offspring correlation is given by:

$$\rho_{cx} = \frac{cov(f_p, f_o)}{\sigma(f_p)\sigma(f_o)} \quad (19)$$

where $cov(f_p, f_o)$ is the covariance of two variables, in this particular case, the average fitness of the parents and the offspring.

In table 2 we present the results regarding crossover analysis. The table gathers data from 1-point crossover and uniform crossover. Uniform crossover uses probability $p = 0.5$, for directly transferring the position of the parent to the offspring (which is commonly used). As before, the first column indicates the representation used, the second column the tested instance, while the last two columns provide the data for the crossover operators. For each operator, we show the average of the minimum distance between the optimum solution and the average offspring found in the random walk ($\overline{d_{cxopt}}$) with the standard deviation in brackets. Next, we present the values for the fitness distance correlation (ϱ_{cx}), followed by the parent offspring correlation (ρ_{cx}).

Table 2: Summary results for crossover landscapes.

Representation	Instance			1-Point Crossover			Uniform Crossover		
	Name	n	m	$\overline{d_{cx,opt}}$	ϱ_{cx}	ρ_{cx}	$\overline{d_{cx,opt}}$	ϱ_{cx}	ρ_{cx}
BR	P01	28	10	13.23 (0.36)	-0.04	0.72	13.46 (0.37)	-0.03	0.54
OR	P01	28	10	10.46 (0.35)	-0.47	0.83	10.87 (0.35)	-0.48	0.68
PR	P01	28	10	12.12 (0.36)	-0.51	0.90	10.96 (0.35)	-0.50	0.81
RK	P01	28	10	10.55 (0.35)	-0.51	0.87	10.99 (0.35)	-0.49	0.82
WC	P01	28	10	5.66 (0.30)	-0.57	0.90	6.05 (0.31)	-0.55	0.86
BR	P02	50	5	24.53 (0.51)	0.01	0.78	24.65 (0.50)	0.01	0.65
OR	P02	50	5	19.20 (0.51)	-0.26	0.81	19.51 (0.52)	-0.29	0.62
PR	P02	50	5	20.87 (0.53)	-0.37	0.90	19.58 (0.53)	-0.32	0.78
RK	P02	50	5	19.19 (0.51)	-0.28	0.79	19.58 (0.53)	-0.34	0.78
WC	P02	50	5	16.31 (0.45)	-0.40	0.82	16.48 (0.44)	-0.39	0.80
BR	CB01	100	5	49.64 (0.72)	-0.02	0.80	49.74 (0.71)	-0.01	0.75
OR	CB01	100	5	46.37 (0.66)	-0.58	0.79	47.66 (0.68)	-0.59	0.55
PR	CB01	100	5	49.59 (0.71)	-0.56	0.91	47.58 (0.69)	-0.61	0.72
RK	CB01	100	5	46.09 (0.66)	-0.58	0.74	47.55 (0.69)	-0.61	0.72
WC	CB01	100	5	37.52 (0.64)	-0.56	0.75	37.87 (0.64)	-0.55	0.71
BR	CB02	100	5	49.62 (0.71)	-0.03	0.82	49.72 (0.71)	-0.03	0.75
OR	CB02	100	5	46.47 (0.66)	-0.58	0.81	47.73 (0.67)	-0.57	0.57
PR	CB02	100	5	49.87 (0.71)	-0.56	0.91	47.76 (0.69)	-0.62	0.73
RK	CB02	100	5	46.25 (0.67)	-0.59	0.76	47.74 (0.69)	-0.61	0.73
WC	CB02	100	5	36.96 (0.64)	-0.57	0.76	37.37 (0.66)	-0.57	0.73
BR	GK01	100	15	49.89 (0.71)	-0.02	0.78	49.91 (0.71)	0.01	0.72
OR	GK01	100	15	49.33 (0.73)	-0.09	0.78	49.48 (0.72)	-0.08	0.52
PR	GK01	100	15	49.86 (0.71)	-0.09	0.90	49.53 (0.72)	-0.10	0.67
RK	GK01	100	15	49.35 (0.72)	-0.09	0.71	49.39 (0.71)	-0.08	0.65
WC	GK01	100	15	48.92 (0.71)	-0.10	0.71	48.95 (0.71)	-0.08	0.65

A first inspection of the results allows us to observe some important aspects. By looking at the best values, marked in bold, it is clear that there is not a representation that outclasses the other. In fact, this only happens when we look at the distance between offspring and the optimum solution. For every single instance, weight-coding representation is better than all the other encodings, whilst between these representations, the differences are small. Furthermore, it seems the type of instance has an effect on the representations performance. Considering the fitness distance correlation, the instances from the first dataset, $P01$ and $P02$, are also dominated by the weight-coding representation. When looking into the remaining instances, weight-coding is not able to perform better than other encodings. In fact, encodings which make use of a *first-fit* heuristic in their decoding process from genotype to phenotype seem to be more suitable. This is especially visible in the random-key representation, as it has the best fitness distance correlation values for instances $CB01$ and $CB02$ on both crossover operators.

In addition, the results found in table 2 reveal that binary representation has inferior values when comparing to all other representations, regardless of the problem instance. Above all, the fitness distance correlation values for binary representation are

considerable lower. They are near to 0.0, while for other representations the values are, in average, closer to -0.5 .

The fitness distance plots for crossover landscapes provide some auxiliary information. Figure 2 presents the plots for 1-point crossover and figure 3 for uniform crossover, on all the tested representations. The first observable fact is that for both operators the shapes of the distribution of the landscapes points are very similar. This is a first indication that there are no significant differences between choosing one of the crossover operators. This is not an unexpected result since the ordering of the items in the chromosome is irrelevant. Apart from this, we can find different fitness distance plots for the crossover landscapes on the tested encodings.

Looking at the plots for the binary representation, we can observe an uncorrelated landscape. The distance to the optimum is large as well as between the landscape elements. If the shape of the distribution was more round and closer to a circle, the landscape would be totally uncorrelated. This effect is not seen on the other representations. For the remaining representations, the landscapes show that fitness and distance are correlated with similar distribution. In contrast with the fitness distance correlation coefficient, the most noticeable fact is that weight-coding representation is closer to the optimum. We conclude that this difference was significant to be reflected on the coefficient values. Looking at these fitness distance plots, we can conclude that they are similar to the plots of the mutation landscapes in terms of distribution shape. In spite of this, it is possible to distinguish a minor difference between the plots of weight-coding: the crossover plots are closer to the optimum. This effect can be explained by the stronger heuristic bias in this encoding.

As for the parent-offspring correlation, the results are somewhat inconclusive. The correlation between parents and their offspring is similar for all representations, attaining some high values. Even for the binary encoding the correlation values are good, a fact already observed in the mutation analysis. Only ordinal encoding presents slighter inferior values when comparing to the other representations. As a final observation, neither crossover operator demonstrate a superior performance.

How can these observations be explained? The first reason is related to the specific properties of the operator; the dynamics of crossover are more difficult to analyze as previous studies have shown (Merz, 2000). Building a precise time series from a random walk generated is a hard task, when compared to mutation operators. The second reason is the dependency on parental distance. Larger parental distance usually induces larger crossover innovations (Raidl and Gottlieb, 2005). Here, the crossover landscapes are generated from pairs of solutions where one of them is always created independently at random. This fact may ensure a larger parental distance but at the same time, it might create a weak dependency on the distance between the parents thus allowing a certain loss of homogeneity required to produce fitter offspring. Instead of creating one of the parents independently at random, another possibility would be to derive it from the first parent solution via n consecutively applied mutations. Further investigation on this issue is required.

Representation must allow for crossover operators the capacity of efficient exchange of genetic material. For the MKP, we can observe that all studied representations clearly perform that task since they cover the feasible region of the search space. The notable exception is binary encoding, as confirmed by the results of fitness distance correlation.

It is possible to conclude for this problem, that the choice of crossover operators is not very influential on the representation used. In fact, what can be stated is that

Multidimensional Knapsack Problem: The Influence of Representation

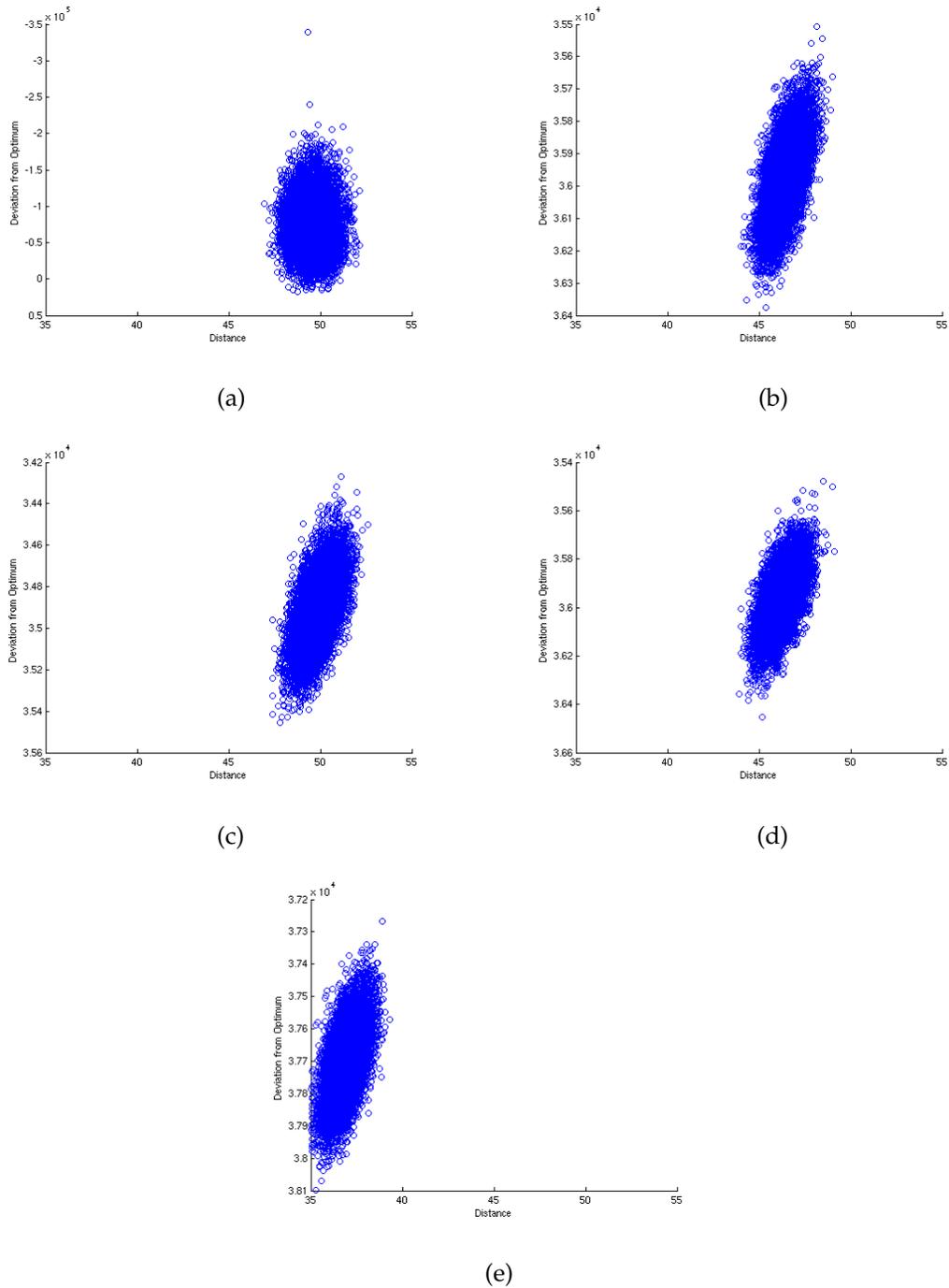


Figure 2: Fitness distance correlation plots for Binary (a), Ordinal (b), Permutation (c), Random-Key (d) and Weight-Coding (e) representations, on the CB02 instance, with 1-point crossover.

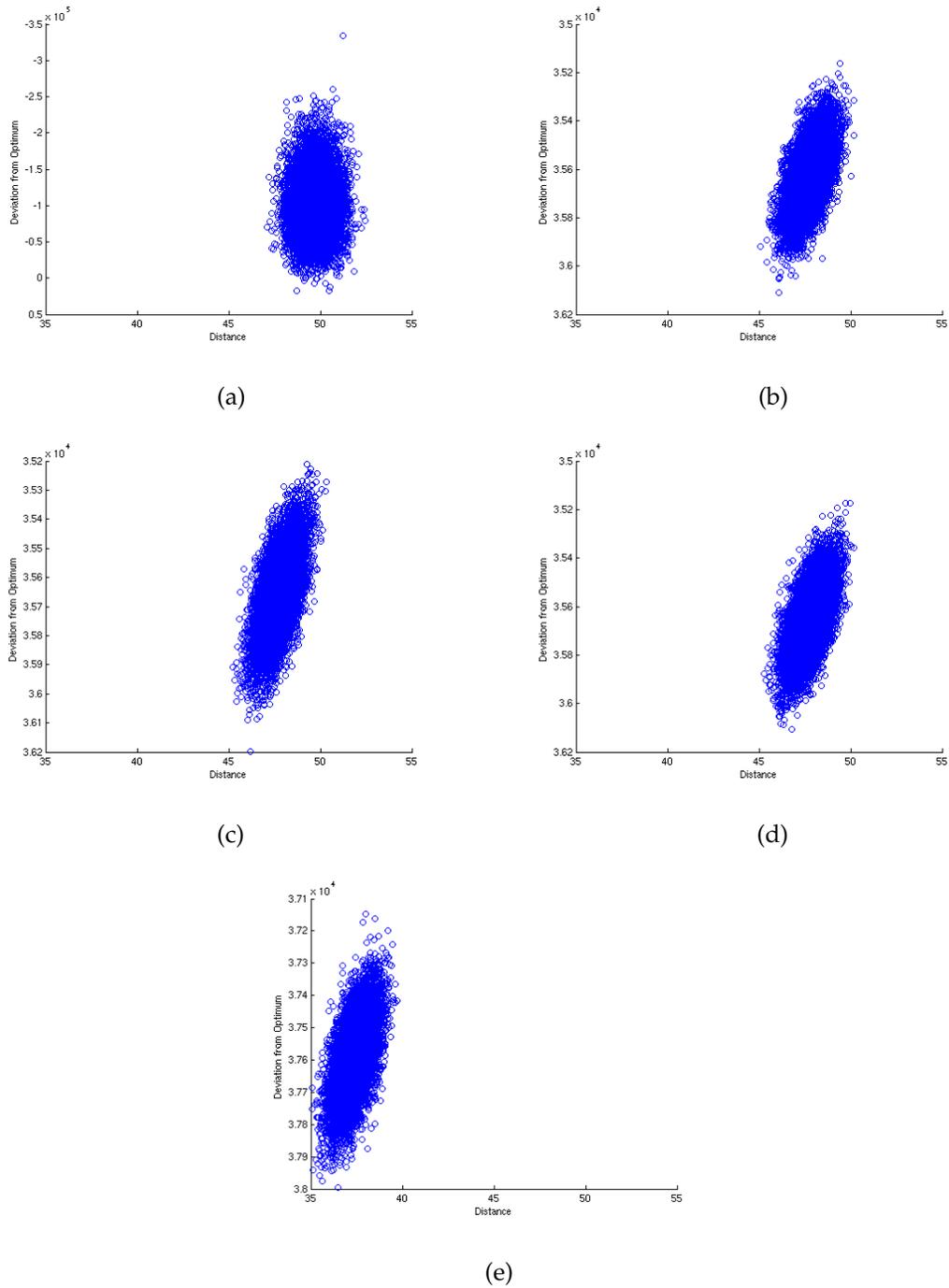


Figure 3: Fitness distance correlation plots for Binary (a), Ordinal (b), Permutation (c), Random-Key (d) and Weight-Coding (e) representations, on the CB02 instance with uniform crossover.

these two operators in conjunction with these representations have the same behaviour. There is no clear advantage of a single representation which indicates that other factors must be taken into more consideration.

5.4 The effect of heuristics and local improvement

From previous sections, it is possible to infer the importance of heuristics and other techniques, such as repair operators, in the design of efficient evolutionary algorithms for the MKP.

All decoder-based representations include some simple heuristic mechanisms, essential to its interpretation. We will now complement our study by investigating a few other techniques that can be added to some of the encodings addressed at this research. Our goal is to perform an all inclusive study. Many additional techniques could be developed and used but we restrain ourselves to the most common used in evolutionary algorithms for the MKP (Raidl and Gottlieb, 2005).

Some additional tests were made regarding the binary and the weight-coding representations. Our experimentation covers mutations and crossover generated landscapes for these two representations. In our tests, we will consider the following improvements:

- *Binary representation with local repair* (BR with R): the standard encoding will be subject to a local method that repairs infeasible solutions by randomly removing items until all constraints are satisfied;
- *Binary representation with profit/presudo-resource consumption ratios heuristic repair* (BR with HR): an infeasible solution is repaired by removing items according to the ordering of the items given by the profit/presudo-resource consumption ratios;
- *Binary representation with heuristic repair and local improvement* (BR with HR + LI): the union of the previous steps with the insertion of items in a feasible solution, as long as it does not violates any constraint;
- *Weight-coding representation with log-normal* (WC with log-normal): the heuristic is encoded in the mutation variation operator. The operator replaces a randomly selected position on the chromosome by sampling a random number which follows a log-normal distribution:

$$w_j = (1 + \gamma)^{N(0,1)}, j = 1, \dots, n. \quad (20)$$

where $\gamma > 0$ is a strategy parameter that controls the average intensity of biasing and $N(0,1)$ denotes a normally distributed random number with mean 0 and standard deviation 1. We set $\gamma = 0.05$ as recommended by (Raidl, 1999).

In table 3 we present, for all tested instances, the results of the applied measures for the four types of binary encodings and two types of weight-coding (we include the results for simple BR and WC for comparison) using mutation as the variation operator. Table 4 contains the results for the crossover operators.

A brief perusal of the results generated by mutation operators, as shown in table 3 reveals one important aspect: the weight-coding representation with the log-normal operator achieved the highest correlation value for all instances and the highest fitness distance correlation value for all instances with the exception of instance GK01. At a

Table 3: Summary results for the effect of adding heuristics and local optimization for mutation landscapes.

Representation	Instance			Measures			
	Name	n	m	$\overline{d_{opt}}$	ϱ	l	ξ
BR	P01	28	10	14.03 (0.36)	0.07	7.12	0.25
BR with R	P01	28	10	14.33 (0.38)	-0.39	9.93	0.35
BR with HR	P01	28	10	12.99 (0.43)	-0.58	12.46	0.44
BR with HR + LI	P01	28	10	1.78 (0.27)	-0.90	10.11	0.36
WC	P01	28	10	7.53 (0.34)	-0.58	14.51	0.52
WC with log-normal	P01	28	10	0.19 (0.42)	-0.98	45.36	1.62
BR	P02	50	5	25.01 (0.49)	0.09	16.18	0.32
BR with R	P02	50	5	26.13 (0.53)	-0.21	15.07	0.30
BR with HR	P02	50	5	23.93 (0.52)	-0.42	19.78	0.40
BR with HR + LI	P02	50	5	13.88 (0.17)	-0.84	18.28	0.37
WC	P02	50	5	17.48 (0.46)	-0.37	19.43	0.39
WC with log-normal	P02	50	5	12.78 (0.31)	-0.86	75.17	1.50
BR	CB01	100	5	49.95 (0.76)	-0.05	35.74	0.36
BR with R	CB01	100	5	50.37 (0.68)	-0.25	15.50	0.15
BR with HR	CB01	100	5	43.67 (0.87)	-0.43	21.83	0.22
BR with HR + LI	CB01	100	5	6.58 (1.55)	-0.96	23.79	0.24
WC	CB01	100	5	40.85 (0.69)	-0.62	10.85	0.11
WC with log-normal	CB01	100	5	9.62 (2.86)	-0.99	138.31	1.38
BR	CB02	100	5	50.00 (0.64)	-0.02	36.65	0.37
BR with R	CB02	100	5	49.94 (0.71)	-0.34	15.79	0.16
BR with HR	CB02	100	5	43.25 (0.83)	-0.47	19.73	0.20
BR with HR + LI	CB02	100	5	12.08 (1.11)	-0.91	12.77	0.13
WC	CB02	100	5	40.68 (0.74)	-0.59	10.78	0.11
WC with log-normal	CB02	100	5	11.41 (2.60)	-0.98	133.35	1.33
BR	GK01	100	15	49.95 (0.75)	0.06	27.32	0.27
BR with R	GK01	100	15	50.33 (0.70)	-0.06	15.64	0.16
BR with HR	GK01	100	15	49.80 (0.69)	-0.03	14.86	0.15
BR with HR + LI	GK01	100	15	40.04 (0.44)	-0.98	29.84	0.30
WC	GK01	100	15	49.41 (0.74)	-0.08	9.33	0.09
WC with log-normal	GK01	100	15	44.30 (0.80)	-0.61	51.66	0.52

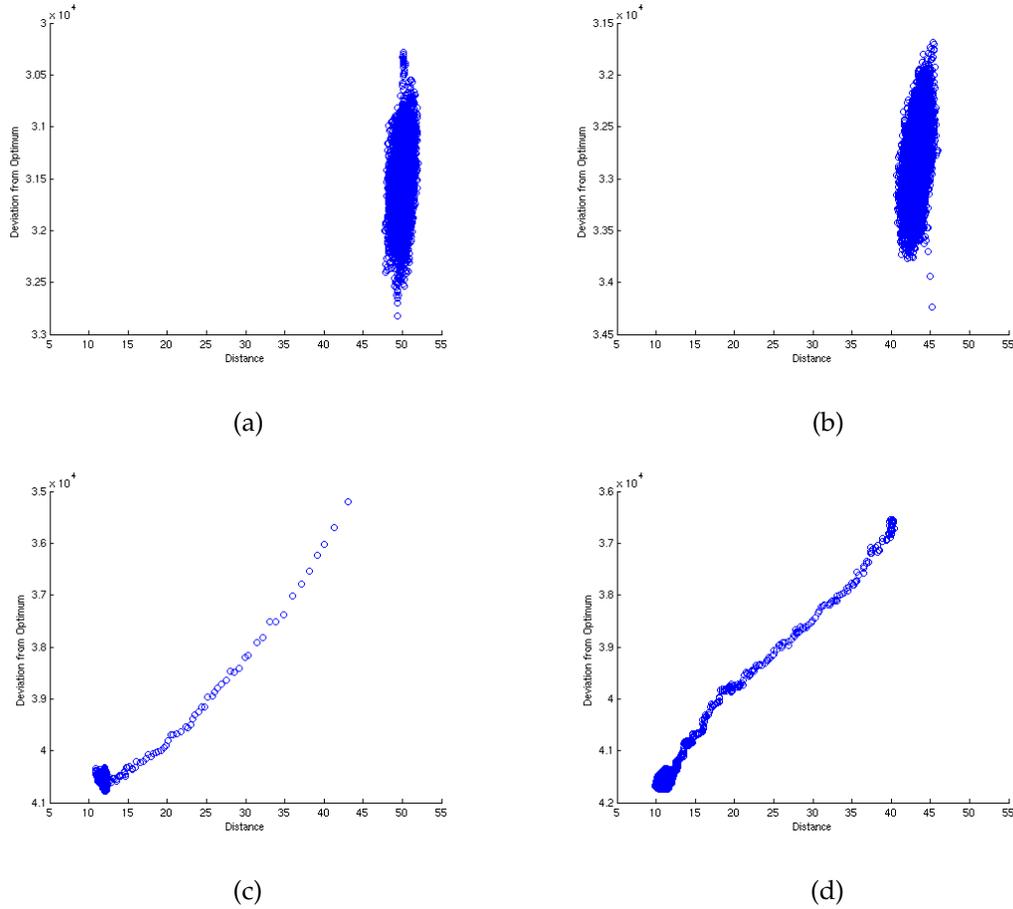


Figure 4: Fitness distance correlation plots for Binary with repair (a), Binary with heuristic repair (b), Binary with heuristic repair and local improvement (c) and Weight-coding with log-normal operator (d) on the CB02 instance.

closer examination, we can verify that the attained values are very good. For the fitness distance correlation, the encoding has values near -1.0 (instances P01, CB01 and CB02). When comparing these values to the average minimal distance from the best individual to the optimum, we can observe that they are also the lowest. Only for instances CB01 and GK01 this effect is not visible. Close to the weight-coding representation with log-normal operator is binary representation with consumption ratios heuristic repair and local improvement. This encoding is able to achieve very good values for the fitness distance correlation for all instances (in instance GK01 it has the best value), as well as the average distance from the best individual found to the optimum (instances CB01 and GK01). Regarding autocorrelation, this encoding does not attain the same degree of success like the weight-coding representation (the values are similar to those attained by the simple direct encoding).

From the examination of the fitness distance plots in figure 4, it is possible to distin-

Table 4: Summary results for the effect of adding heuristics and local optimization for crossover landscapes.

Representation	Instance			1-Point Crossover			Uniform Crossover		
	Name	n	m	$\overline{d_{opt}}$	ϱ_{cx}	ρ_{cx}	$\overline{d_{opt}}$	ϱ_{cx}	ρ_{cx}
BR	P01	28	10	13.23 (0.36)	-0.04	0.72	13.46 (0.37)	-0.03	0.54
BR with R	P01	28	10	13.08 (0.36)	-0.40	0.92	13.19 (0.36)	-0.40	0.85
BR with HR	P01	28	10	10.04 (0.39)	-0.64	0.96	11.04 (0.40)	-0.60	0.90
BR with HR + LI	P01	28	10	3.63 (0.33)	-0.68	0.54	7.00 (0.38)	-0.67	0.72
BR	P02	50	5	24.53 (0.51)	0.01	0.78	24.65 (0.53)	0.01	0.65
BR with R	P02	50	5	25.14 (0.52)	-0.29	0.82	25.25 (0.49)	-0.28	0.81
BR with HR	P02	50	5	22.02 (0.50)	-0.35	0.90	22.28 (0.45)	-0.35	0.88
BR with HR + LI	P02	50	5	15.04 (0.40)	-0.56	0.65	16.08 (0.47)	-0.51	0.65
BR	CB01	100	5	49.64 (0.72)	-0.02	0.80	49.74 (0.71)	-0.01	0.75
BR with R	CB01	100	5	48.19 (0.69)	-0.41	0.75	48.36 (0.70)	-0.42	0.71
BR with HR	CB01	100	5	39.90 (0.83)	-0.53	0.84	40.04 (0.81)	-0.52	0.81
BR with HR + LI	CB01	100	5	33.81 (0.84)	-0.62	0.77	36.34 (0.78)	-0.58	0.72
BR	CB02	100	5	49.62 (0.71)	-0.03	0.82	49.72 (0.71)	-0.03	0.75
BR with R	CB02	100	5	47.85 (0.68)	-0.41	0.76	48.01 (0.69)	-0.42	0.73
BR with HR	CB02	100	5	39.26 (0.84)	-0.54	0.86	39.83 (0.83)	-0.53	0.82
BR with HR + LI	CB02	100	5	34.26 (0.86)	-0.64	0.78	36.30 (0.82)	-0.61	0.72
BR	GK01	100	15	49.89 (0.71)	-0.02	0.78	49.91 (0.71)	0.01	0.72
BR with R	GK01	100	15	50.38 (0.71)	-0.05	0.77	50.35 (0.72)	-0.03	0.72
BR with HR	GK01	100	15	49.66 (0.67)	-0.06	0.81	49.50 (0.66)	-0.06	0.75
BR with HR + LI	GK01	100	15	48.40 (0.71)	-0.10	0.72	48.73 (0.69)	-0.12	0.66

guish two simple and clear patterns: low correlated landscapes and highly correlated landscapes. Binary representation with repair and binary representation with heuristic repair do not have a highly correlated landscape, whilst binary representation with heuristic repair and local improvement, and weight-coding representation with the log-normal operator have highly correlated landscapes. These representations have near-ideal distributions since the solutions very near to the optimum can be reached by *jumping* from one solution to another by successively reducing the *jump* distance. For these two encodings, the addition of heuristics and/or local improvement was influential in reshaping the distributions shapes (for binary representation in one configuration) in comparison to the distributions of the representations without heuristics and/or local improvement. From ellipsoidal shapes, the distributions are now closer to linear lines in direction to the optimum with a strong concentration on that point. Fitness and distance are clearly correlated.

This analysis reveals that adding heuristics and/or local improvement methods to a representation is crucial to obtain better results when solving the MKP. Examining the weight-coding representation, using a mutation variation operator that is more sensitive to the problem domain helps to improve performance. Since the float values represent surrogate multipliers, it is clear that the mutation provided by a uniform distribution will cause strong phenotypic changes, whilst a log-normal distribution will handle the genotype in a more subtle way, since it gives the advantage that small changes of weights are made with higher probabilities and allowing at the same large changes, but

with less probability. The log-normal operator ensures a higher locality for the weight-coding representation. In this particular case, the addition of a heuristic – by means of a more controlled probability distribution – helped the encoding to improve its fitness distance correlation and autocorrelation values, for all tested problem instances.

The binary representation needs a more careful investigation. The introduction of a local search method was sufficient to improve the performance when comparing to the simple direct encoding (as the works of Chu and Beasley (1998) and Raidl (1999) show) but unable to achieve the values of the ordinal, permutation and random-key representations. These three previous encodings still perform better than a direct encoding even when we compare them to the binary representation with the ratios heuristic (BR with HR). The exception is instance P01, although the differences are small. It seems these two improvements, when examined separate from each other, can only introduce a slightly weaker heuristic bias than the first-fit heuristic used for ordinal, permutation and random-key encodings. When heuristic repair and local improvement are combined, the result is very different and the performance is comparable to the weight-coding representation with the log-normal mutation operator. In terms of autocorrelation, the effects are not the same. For all variants of the direct encoding, the autocorrelation values are similar, only with marginal differences.

We follow now to the crossover generated landscapes analysis. In this case, only the modifications made to the simple binary encoding are eligible. For weight-coding representation the heuristic was introduced on the mutation operator and the same heuristic is not directly applicable to the crossover operator.

Focusing our attention on the results from table 4, we can see that for crossover generated landscapes the fitness distance correlation and minimum average distance to the optimum solution, the direct encoding with the heuristic repair plus the local improvement method attains the best results. Immediately after is the weight-coding representation. These results are not surprising and confirm the previous observations. Since crossover has a role of exploration, the addition of mechanisms that enable some exploitation when evaluating the individuals permits better solutions to be found. In spite of this fact, the degree of improvement obtained by introducing the combined use of these two extra mechanisms (heuristics and local improvement) is not so wide in comparison with mutation. The interesting fact is, the sole use of these mechanisms might be more sensitive to the instance's structure. For the instances of the first benchmark, the heuristic repair performs better than simple repair, while for the second benchmark instances, simple repair gives better results. The type of crossover does not affect the representation's performance.

In terms of parent-offspring correlation, results are more consistent when comparing with mutation. Tested encodings reveal higher values of correlation with some minor exceptions: for the first instance, P01, binary representation with uniform crossover (0.54); and binary representation with heuristic repair and local improvement, on both operators (0.65). For all other experiments correlation values are high, in the range of [0.7, 0.9]. This fact can be explained by the strong heuristic bias that is introduced and by binary encodings. As already seen with mutation, the binary encoding permits good correlation values since alterations to a candidate solution generate neighbor solutions which are not only close but, with the addition of these mechanisms, changes on the fitness value are not deeply affected. As before, in terms of correlation the type of crossover is not influential on the encodings behavior.

Figure 5 presents the fitness distance plots for the crossover landscapes generated by the encodings with heuristics and/or local improvement. On the left column we

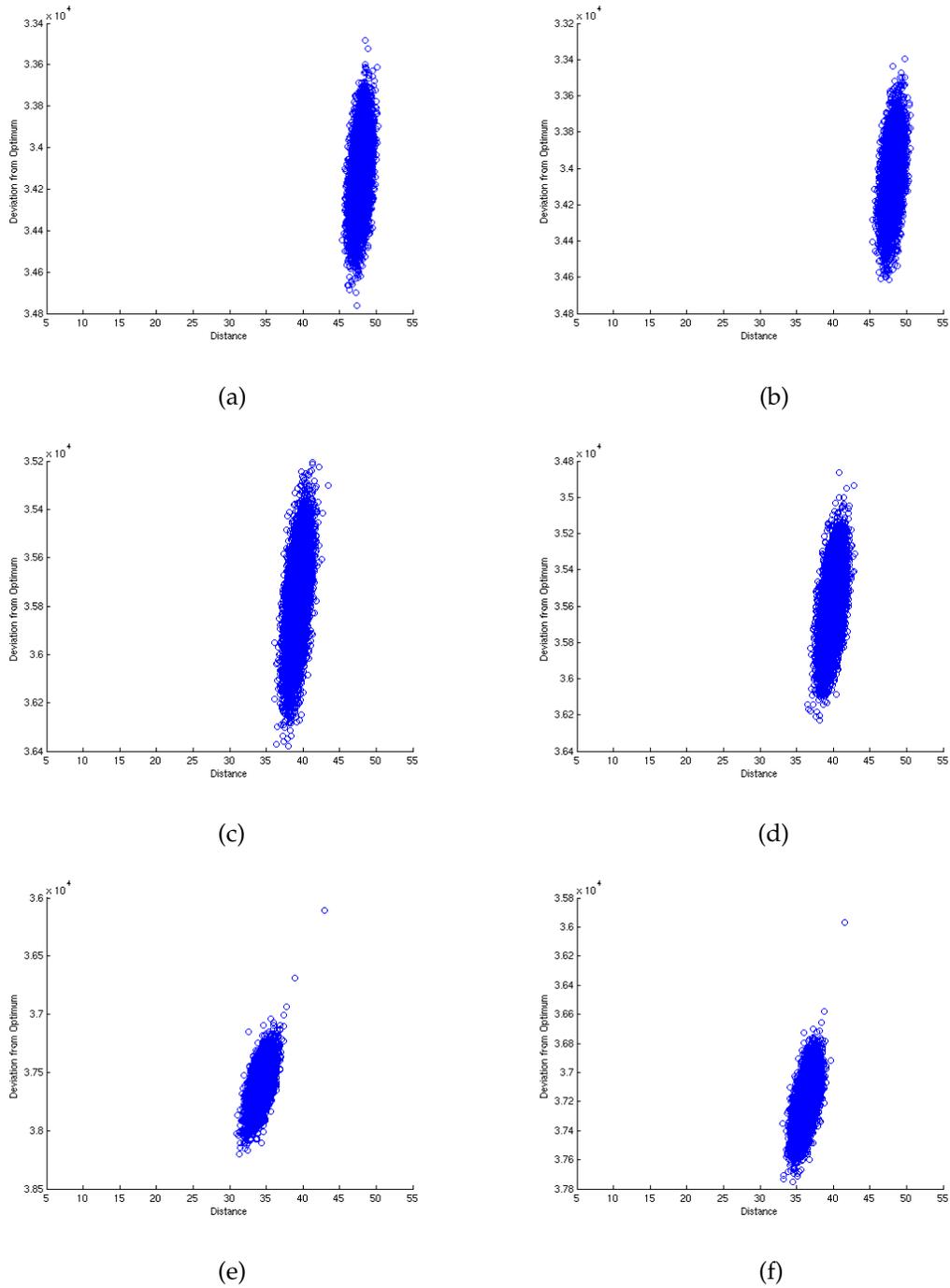


Figure 5: Fitness distance correlation plots for Binary with repair (a), Binary with heuristic repair (c), Binary with heuristic repair and local improvement (e) with 1-point crossover and and uniform crossover (b,d,f) on the CB02 instance.

can see the landscapes produced by 1-point crossover operator and on the right, the landscapes created by uniform crossover.

The first conclusion is that there are no differences between the operators. The shapes of the landscapes elements distributions are identical with no significant differences, even in terms of the distance to the optimum. The second observable fact is that the increasing of complexity on the decoding process leads to a minimal distribution concentration closer to the optimum. Even binary representation with the consumption heuristic repair and local improvement, provides significant advantage over the other encodings. Although the fitness distance correlation coefficient shows differences between the different types of modifications made to binary representation, the plots do not reflect the same order of improvement. This is an indication that crossover operators are not very sensitive to the effect of these methods. The enhancements affect more the landscapes created by mutation operators. Partially this may be due to the manner crossover landscapes are produced.

As we can see from these results, a pattern arises in terms of the representations performance according to their heuristic bias. This is essential true when looking at the fitness distance correlation coefficient. The representations with stronger heuristic bias (weight-coding with log-normal operator and binary representation with heuristic repair and local improvement) achieve the best results, very close to -1.0 . The encodings with a weak heuristic bias do perform well but clearly below the two previous encodings, while the simple binary representation, as expected, is the one with the poorer performance. The fitness landscape analysis is compliant with previous studies on representations for the MKP, like Raidl and Gottlieb (2005).

5.5 Statistical Analysis

The analysis described on the previous sections was able to provide some insights about the role of representation on the MKP. Nevertheless, part of the presented results raised some concerns, if under certain conditions, the representations have a significant performance difference, or, if they simply can be considered equivalent. To better understand these situations, a statistical analysis is presented.

We start our analysis by testing the differences found in the coefficient values contained in all the previous tables. Specifically, it is important to establish what statistically significant differences exist between the distance values, fitness distance correlation values, for mutation and crossover landscapes, as well as the parent-offspring correlation, 1-point crossover and uniform crossover operators, for all tested representations.

The function used for the hypothesis testing is the *Wilcoxon rank sum* test, which is equivalent to the *Mann-Whitney U* test. This test performs a two-sided rank sum test of the hypothesis that two independent samples come from distributions with equal medians. If the null hypothesis is true, the medians are equal. The sets of data are assumed to come from continuous distributions that are identical except possibly for a location shift, but are otherwise arbitrary. The reason to use this test instead of *t-test* is that the former assumes the data to come from a normal distribution with unknown variance. The difference between hypothesis test procedures often arises from differences in the assumptions that the researcher is willing to make about the data sample. In our case, the fitness landscapes generated by the different representations follow different distributions. Although most of them follow normal distributions, e.g., ordinal, permutation and random-key encodings, others simply do not follow it, as confirmed by the *Lilliefors* test. For this reason, it is preferable to use the *Wilcoxon rank sum* test

instead of the *t-test*. On all performed tests the significance level is 5% ($\alpha = 0.05$). We present results for the CB02 instance since the other problem instances follow the same trend.

Table 5: Wilcoxon rank sum test results for mutation landscapes; "+" indicates null hypothesis is true whilst "-" indicates null hypothesis can be rejected at 5% significance level. Lower triangle reports to fitness distance correlation coefficients; upper triangle to distance values.

	BR	OR	PR	RK	WC	BR+R	BR+HR	BR+HR+LI	WC+log
BR		-	+	+	-	+	-	-	-
OR	-		+	-	-	+	-	-	-
PR	-	+		+	-	+	-	-	-
RK	-	+	+		-	+	-	-	-
WC	-	+	+	+		-	-	-	-
BR+R	-	-	-	-	-		-	-	-
BR+HR	-	-	-	-	-	-		-	-
BR+HR+LI	-	-	-	-	-	+	-		-
WC+log	-	-	-	-	-	-	-	-	

Table 6: Wilcoxon rank sum test results for crossover operators; "+" indicates null hypothesis is true whilst "-" indicates null hypothesis can be rejected at 5% significance level.

	BR	OR	PR	RK	WC	BR+R	BR+HR	BR+HR+LI
ρ_{cx}	-	+	+	+	-	+	+	+
Q_{cx}	+	+	+	+	+	+	+	+

Table 5 shows the significant differences found for the distance and fitness distance coefficient values, on mutation landscapes. A brief overview of the data presented in the table allow us to verify that for most of the representations, the coefficient values found are statistically significant different. For all cases marked with a "-" sign, the null hypothesis can be rejected, i.e., the medians of the tested data are not equal. This means that when testing the landscapes generated by two different representations, they cannot be considered equivalent. In fact, on a closer look to the table, the lower triangle, which reports to the fitness distance correlation coefficient values, shows us that the group with no significant differences are the ordinal, permutation and random-key representations. In general terms, this fact is also true when considering the distance values (reported in the upper triangle). The exception is binary representation with repair. When comparing this encoding with the base representations, the differences are not statistically different. The pattern previously observed in mutation landscapes is confirmed by this test.

Regarding crossover landscapes, from the fitness landscape analysis was not possible to clearly draw conclusions. In fact, differences between representations were not so strong as the ones found in mutation landscapes. When performing this test to crossover coefficient values, results are not different. For all representations it is not possible to reject the hypothesis in a consistent way. With some exceptions, e.g., per-

Table 7: Wilcoxon rank sum test results for mutation landscapes; "+" indicates null hypothesis is true whilst "-" indicates null hypothesis can be rejected at 5% significance level.

	BR	OR	PR	RK	WC	BR+R	BR+HR	BR+HR+LI
OR	-							
PR	-	+						
RK	-	+	+					
WC	-	-	-	-				
BR+R	-	-	-	-	-			
BR+HR	-	-	-	-	-	-		
BR+HR+LI	-	-	-	-	-	-	-	
WC+log	-	-	-	-	-	-	-	-

Table 8: Wilcoxon rank sum test results for crossover landscapes; "+" indicates null hypothesis is true whilst "-" indicates null hypothesis can be rejected at 5% significance level. Lower triangle reports to 1-point crossover; upper triangle to uniform crossover.

	BR	OR	PR	RK	WC	BR+R	BR+HR	BR+HR+LI
BR		-	-	-	-	-	-	-
OR	-		+	-	-	-	+	-
PR	-	+		+	-	-	+	-
RK	-	+	-		-	-	+	-
WC	-	-	-	-		-	-	-
BR+R	-	-	-	-	-		-	-
BR+HR	-	-	-	-	-	-		-
BR+HR+LI	-	-	-	-	-	-	-	

mutation and random-key encodings, no statistical significant differences are found between different representations. Another important question raised from the previous analysis is: what is the difference between the two types of crossover operators; can we find significant differences between 1-point crossover and uniform crossover when applied to the MKP? By looking at table 6 it is possible to observe that for most of the representations, there is not an advantage of a single operator. The only visible statistically differences between the use of the two operators are found in binary and weight-coding representations, when considering the fitness distance correlation coefficient. For the parent-offspring correlation, no differences can be found.

In spite of being important to analyze the differences of the coefficient values, it is even more to test the fitness landscapes. Statistically testing the landscapes will provide us with a more clear picture of the influence of the representation on the MKP. In this case, we will analyze the differences between the landscapes by analyzing the time series.

Tables 7 and 8 present the results of the *Wilcoxon rank sum* test to the fitness landscapes, mutation and crossover landscapes respectively. A brief overview of both tables reveals a similar pattern to the one presented by the coefficient tables. We can find significant differences for all the representations with the exception of the group ordi-

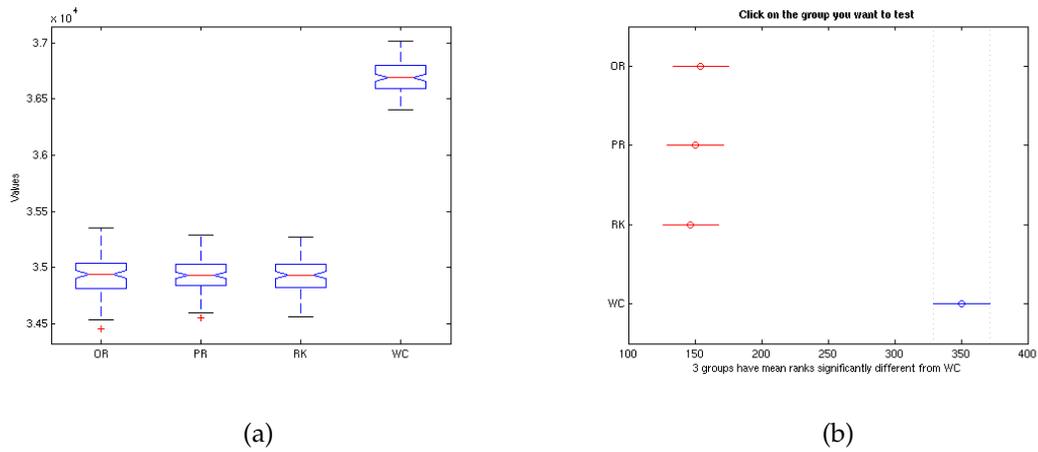


Figure 6: Box-plot (a) and Rank graph (b) for mutation landscapes on the Binary, Ordinal, Permutation, Random-Key and Weight-Coding representations.

nal, permutation and random-key representations when considering fitness landscapes produced by mutation operators. With crossover the results are slightly different. For 1-point crossover, the test results in the acceptance of the null hypothesis only for the pair ordinal and random-key representation, whilst for uniform crossover is with the pairs ordinal and permutation representation, permutation and random-key representation, as well as all these representations with binary with consumption ratios heuristic. These results are interesting since the previous analysis provided more uncertain differences between representation. This is an indication that the fitness landscapes contain slightly differences which were not detected by the measures. Despite that, representation plays a lesser role in crossover landscapes when comparing to mutation ones.

The application of this test can be useful to establish and clarify the existing differences between two sets of data, in this case, two representations. However, it is also important to analyze and confirm differences in groups of data. A good example is, for example, to compare differences between the representations with the *first-fit* heuristic bias or the group of binary encodings, and see if the observable differences between the elements are not only statistically different but also how much. In this case we need to use another function for the hypothesis testing: *Kruskal-Wallis* test. This test is a non-parametric version of the classical one-way *ANOVA*, and an extension of the *Wilcoxon rank sum* test to more than two groups. It compares samples from two or more groups of data and returns the p-value for the null hypothesis that all samples are drawn from the same population, or equivalently, from different populations with the same distribution. We will examine some of the previous situations to better clarify existing groups of representations. The significance level is 5% ($\alpha = 0.05$). The results of the *Kruskal-Wallis* test are presented in graphical form. This allows a quicker and better understanding of the achieved results.

Figure 6 contains the box-plot and rank graph for mutation landscapes, for the set of ordinal, permutation, random-key and weight-coding representations. In this test binary representation is not included since it is already establish there are significant

Multidimensional Knapsack Problem: The Influence of Representation

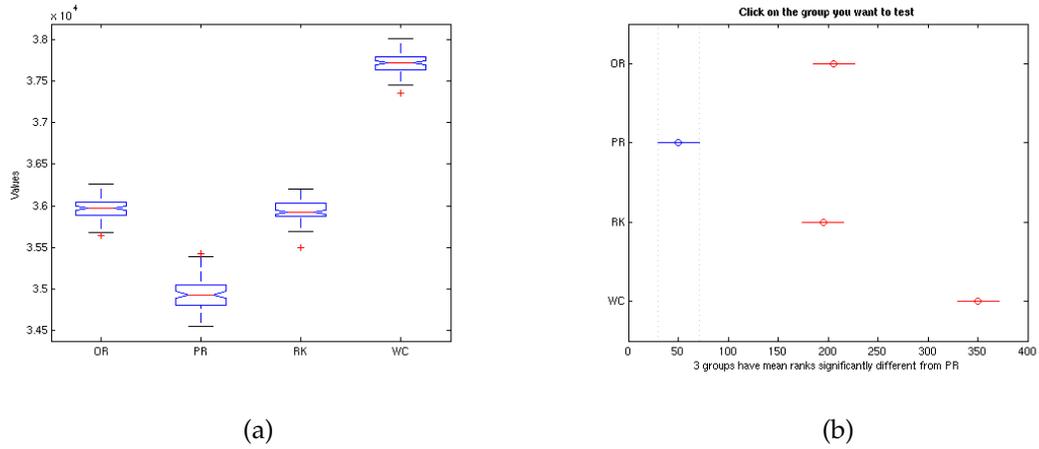


Figure 7: Box-plot (a) and Rank graph (b) on the Binary, Ordinal, Permutation, Random-Key and Weight-Coding representations for 1-point crossover.

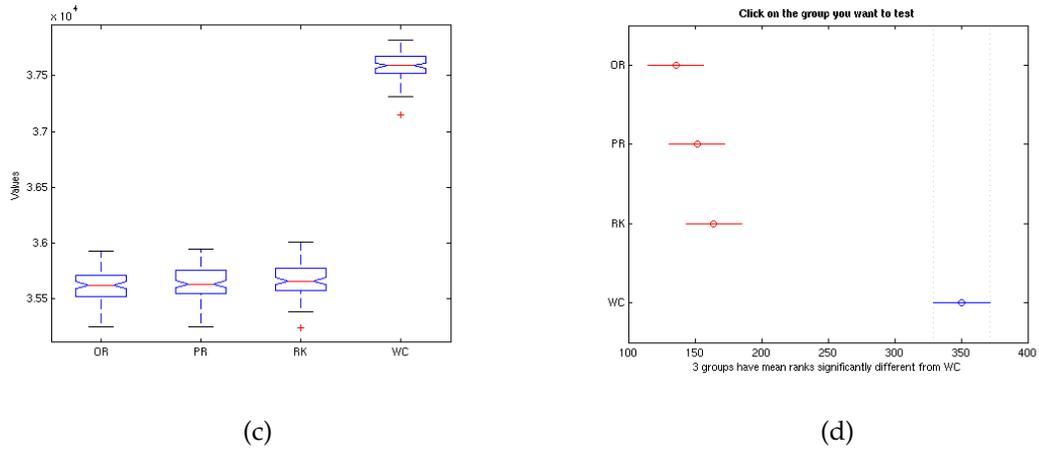


Figure 8: Box-plot (a) and Rank graph (b) on the Binary, Ordinal, Permutation, Random-Key and Weight-Coding representations for uniform crossover.

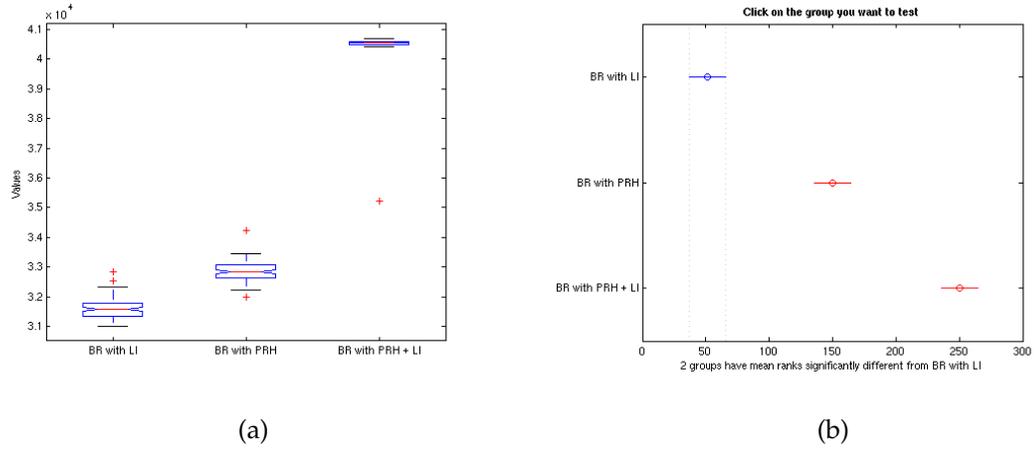


Figure 9: Box-plot (a) and Rank graph (b) for mutation landscapes on Binary representations with heuristics and local improvement.

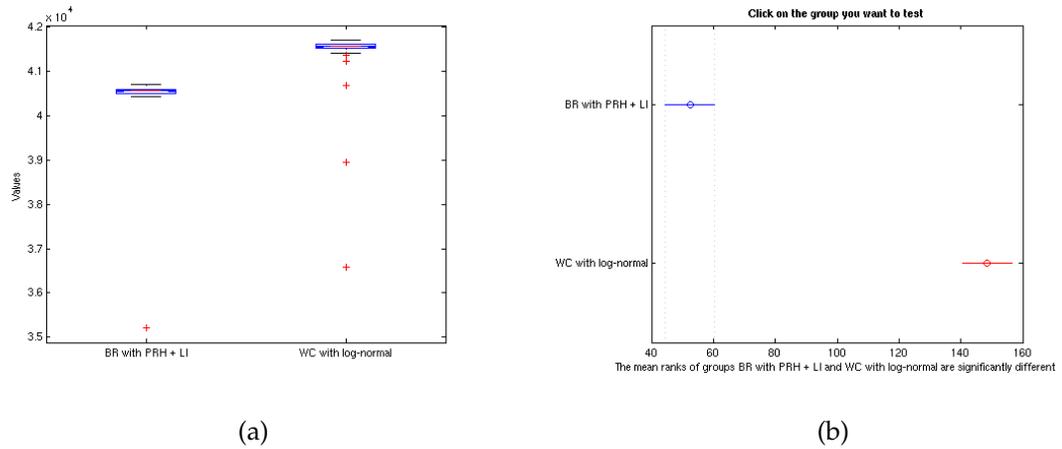


Figure 10: Box-plot (a) and Rank graph (b) for mutation landscapes on Binary with heuristic and local improvement, and Weight-Coding with log-normal operator.

differences between the binary encoding and all the other representations. Figures 7 and 8 show the same data but for the crossover landscapes. The plots are from instance CB02 but other instances follow the same trendline.

From the plots is clearly visible the differences and respective order of magnitude of the different representations. As in the previous analysis, it is possible to conclude that there are no statistically significant differences between the first three representations, but all of these are considerably different from weight-coding. In fact, this behavior is stronger on mutation landscapes while on crossover landscapes, the pattern is only equal with uniform crossover. For 1-point crossover, permutation encoding is statistically different from ordinal and random-key representations, although the differences are larger in relation to weight-coding.

Looking at the rank graphs, the confidence intervals exhibit the same information since the intervals for ordinal, permutation and random-key representations overlap. The overlap is more evident on fitness landscapes generated by mutation operators than crossover operators. This supports previous evidence that weight-coding is a more suitable representation for this problem, considering both variation operators.

Moreover, figure 9 displays the results of the test for the different types of binary representation with the addition of heuristics and/or local improvement. The figure presents the graphs for mutation landscapes but the crossover graphs are equivalent. From these, we can observe that the three types of modifications made to binary representation are statistically different. As expected, the combination of heuristics and local improvement is superior in comparison to repair only. Nevertheless, the box-plot shows that the consumption heuristic might be more suitable to use with binary encodings for the MKP than simple repairing techniques.

Finally, figure 10 presents plots that compares binary representation with heuristic repair and local improvement to weight-coding representation with the log-normal operator. Both encodings achieve the best distribution of local optima as well as the best values for the different coefficients measures. What is not clear is the difference between the two encodings. From the observation of the graphs, weight-coding representation is superior to the direct representation. Although not very distant as the box-plot confirms, the confidence intervals are clearly separated. This confirms that both representations are statistically significantly different.

6 Conclusions

In this article we presented an analysis of fitness landscapes for the Multidimensional Knapsack problem. The goal of this work was to study how the interplay between representations and genetic operators affects the search performance of evolutionary algorithms for the MKP. Although it is a relevant topic, few studies exist that analyze this issue, e.g. (Raidl and Gottlieb, 2005). The results of these investigations may provide useful insights in how to design more efficient algorithms for this class of problems.

Standard tools of fitness landscapes analysis, such as fitness distance correlation and autocorrelation, help to explain differences in performance achieved by different representations. Within a mutation-based evolutionary algorithm, weight-coding with a log-normal mutation operator, and binary representation with a heuristic repair and local improvement, appear to be the most suitable combinations. Considering crossover-based algorithms, the binary representation with the heuristic repair and local improvement is the suitable choice.

Also, the study described in this paper presents a valuable contribution to analyze how heuristics and local search techniques can improve the performance of evolution-

ary algorithms. The heuristic bias effect is studied in depth. In general, heuristics with a strong bias help to achieve better results.

Representation plays an important role when solving the MKP, since choosing an encoding without a strong heuristic bias can create some difficulties for the evolutionary algorithm. As such, the use of heuristics on the decoding process or on a variation operator, as well as the use of local improvement methods can significantly alter the performance for a given representation. A binary string is the most natural encoding, although, it generates infeasible solutions making the task for an algorithm harder. Instead, indirect encodings focus the search on the boundary of the feasible region (Gottlieb, 1999a), suggesting that they might be the most appropriate representations. In any event, this is not a straightforward decision, as confirmed by previous studies, mainly by Raidl and Gottlieb (2005) and Tavares et al. (2006).

Even though our analysis focused on the MKP, the aim of our research is to investigate the influence of genetic representations when solving combinatorial optimization problems. Results presented in this paper add a contribution for this goal and should be considered as a starting point for future studies. This research will now be extended to different combinatorial optimization problems to see if the findings encountered here can be generalized. The outcomes of the global analysis may be important for future applications of evolutionary algorithms to problems with similar properties.

Acknowledgments

The first author would like to acknowledge grant SFRH/BD/12615/2003 from *Fundação para a Ciência e a Tecnologia* (FCT), Portugal.

References

- Altenberg, L. (1997). Fitness distance correlation analysis: An instructive counterexample. In *Seventh International Conference on Genetic Algorithms (ICGA97)*, pages 57–64.
- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160.
- Branke, J., Orbayı, M., and Şima Uyar (2006). The role of representations in dynamic knapsack problems. In Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J. H., Romero, J., Smith, G. D., Squillero, G., and Takagi, H., editors, *Applications of Evolutionary Computing, EvoWorkshops2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoInteraction, EvoMUSART, EvoSTOC*, volume 3907 of *LNCS*, pages 764–775, Budapest. Springer Verlag.
- Branke, J., Salihoglu, E., and Uyar, S. (2005). Towards an analysis of dynamic environments. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1433–1440, New York, NY, USA. ACM Press.
- Bruhn, P. and Geyer-Schulz, A. (2002). Genetic programming over context-free languages with linear constraints for the knapsack problem: first results. *Evol. Comput.*, 10(1):51–74.
- Chu, P. C. and Beasley, J. E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4(1):63–86.
- Cleary, R. and O’Neill, M. (2005). An attribute grammar decoder for the 01 multi-constrained knapsack problem. In *Lecture Notes in Computer Science - Evolutionary Computation in Combinatorial Optimization*, pages 34–45. Springer-Verlag.

- Cotta, C. and Troya, J. (1998). A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In Smith, G., Steele, N., and Albrecht, R., editors, *Artificial Neural Nets and Genetic Algorithms 3*, pages 251–255, Wien New York. Springer-Verlag.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, USA.
- Gottlieb, J. (1999a). *Evolutionary Algorithms for Constrained Optimization Problems*. PhD thesis, Technical University of Clausthal, Clausthal, Germany.
- Gottlieb, J. (1999b). Evolutionary Algorithms for Multidimensional Knapsack Problems: the Relevance of the Boundary of the Feasible Region. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honovar, V., Jakiela, M., and Smith, R. E., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, volume 1, page 787, San Francisco, California. Morgan Kaufmann.
- Gottlieb, J. (2000). Permutation-based evolutionary algorithms for multidimensional knapsack problems. In *SAC '00: Proceedings of the 2000 ACM symposium on Applied computing*, pages 408–414, New York, NY, USA. ACM Press.
- Gottlieb, J. (2001). On the feasibility problem of penalty-based evolutionary algorithms for knapsack problems. In *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing*, pages 50–59, London, UK. Springer-Verlag.
- Gottlieb, J. and Raidl, G. R. (2000). Characterizing locality in decoder-based eas for the multidimensional knapsack problem. In *AE '99: Selected Papers from the 4th European Conference on Artificial Evolution*, pages 38–52, London, UK. Springer-Verlag.
- Hinterding, R. (1994). Mapping order-independent genes and the knapsack problem. In *Proc. of the First IEEE Int. Conf. on Evolutionary Computation*, pages 13–17. IEEE Press.
- Hinterding, R. (1999). Representation, constraint satisfaction and the knapsack problem. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzal, A., editors, *Proceedings of the Congress of Evolutionary Computation*, volume 2, pages 1286–1292, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- Horjik, W. and Manderick, B. (1995). The usefulness of recombination. In *European Conference on Artificial Life*. Springer.
- Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA. Morgan Kaufmann.
- Kellerer, H., Pferschy, U., and Pisinger, D. (2004). *Knapsack Problems*. Springer-Verlag.
- Manderick, B., de Weger, M., and Spiessens, P. (1991). The genetic algorithm and the structure of the fitness landscape. In *4th International Conference on Genetic Algorithms*, pages 143–150. Morgan Kaufmann.
- Martello, S. and Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., New York, NY, USA.

- Merz, P. (2000). *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Eective Search Strategies*. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany.
- Merz, P. and Freisleben, B. (1999). Fitness landscapes and memetic algorithm design. In Corne, D., Dorigo, M., and Glover, F., editors, *New Ideas in Optimization*, pages 245–260. McGraw-Hill, London.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, third edition.
- Michalewicz, Z. and Arabas, J. (1994). Genetic algorithms for the 0/1 knapsack problem. In *ISMIS '94: Proceedings of the 8th International Symposium on Methodologies for Intelligent Systems*, pages 134–143, London, UK. Springer-Verlag.
- Raidl, G. (1999). Weight-codings in a genetic algorithm for the multiconstraint knapsack problem. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A., editors, *Proceedings of the Congress of Evolutionary Computation*, volume 1, pages 596–603, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- Raidl, G. R. (1998). An improved genetic algorithm for the multiconstrained 0-1 knapsack problem. In *Proc. of the 5th IEEE Conference on Evolutionary Computation, 1998 IEEE World Congress on Computational Intelligence*, pages 207–211, Anchorage, Alaska.
- Raidl, G. R. and Gottlieb, J. (1999). On the importance of phenotypic duplicate elimination in decoder-based evolutionary algorithms. In Brave, S. and Wu, A. S., editors, *Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference*, pages 204–211, Orlando, Florida, USA.
- Raidl, G. R. and Gottlieb, J. (2005). Empirical analysis of locality heriability and heuristic bias in evolutionary algorithms: A case study for the multidimensional knapsack problem. *Evolutionary Computation Journal*, 13(4):441–475.
- Reeves, C. R. (1999). Fitness landscapes and evolutionary algorithms. In *AE '99: Selected Papers from the 4th European Conference on Artificial Evolution*, pages 3–20, London, UK. Springer-Verlag.
- Tavares, J., Pereira, F. B., and Costa, E. (2006). The role of representation on the multidimensional knapsack problem by means of fitness landscape analysis. In *Proceedings of the Congress of Evolutionary Computation*, Vancouver, Canada. IEEE Press.
- Thiel, J. and Voss, S. (1994). Some experiences on solving multiconstraint zero-one knapsack problems with genetic algorithms. *INFOR*, 32(4):226–242.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the VI International Conference on Genetics, Vol. 1*, pages 356–366.