Characterizing Locality in Decoder-Based EAs for the Multidimensional Knapsack Problem

Jens Gottlieb[1] and Günther R. Raidl[2]

[1] Department of Computer Science, Technical University of Clausthal,
Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany
gottlieb@informatik.tu-clausthal.de

[2] Institute for Computer Graphics, Vienna University of Technology,
Karlsplatz 13/1861, 1040 Vienna, Austria
raidl@apm.tuwien.ac.at

Abstract. The performance of decoder-based evolutionary algorithms (EAs) strongly depends on the locality of the used decoder and operators. While many approaches to characterize locality are based on the fitness landscape, we emphasize the explicit relation between genotypes and phenotypes. Statistical measures are demonstrated to reliably predict locality properties of selected decoder-based EAs for the multidimensional knapsack problem. Empirical results indicate that (i) strong locality is a necessary condition for high performance, (ii) the concept of heuristic bias also strongly affects solution quality, and (iii) it is important to maintain population diversity, e.g. by phenotypic duplicate elimination.

1 Introduction

Locality is known as an important factor for well-working evolutionary algorithms (EAs) [3, 11, 14, 22]. Although locality can be interpreted in several ways, all interpretations are motivated by the same basic idea: Small changes in genotype performed by evolutionary operators such as mutation and crossover should result in small changes of phenotype, where phenotype is identified by the represented solution or its fitness. EAs which do not fulfill this condition at least partly act like pure random search, hence are not efficient. Thus, the design process of EAs for any problem should be guided by the locality principle [22], which is sometimes also termed principle of strong causality [21].

Many EAs are based on decoders, which map the genotype of a solution onto its phenotype. Obviously, such approaches can only be successful if the employed decoder supports locality. The goal of this paper is to present a new technique for characterizing locality of decoder-based EAs with particular emphasis on the *multidimensional knapsack problem* (MKP), which is stated as

$$\text{maximize} \quad \sum_{j \in J} p_j x_j \tag{1}$$

$$\text{subject to} \quad \sum_{j \in J} r_{ij} x_j \leq c_i, \quad i \in I \tag{2}$$

$$x_j \in \{0,1\}, \qquad j \in J \tag{3}$$

with $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$ denoting the sets of resources and items, respectively. The MKP is a prominent example for an NP-complete combinatorial optimization problem with a wide range of applications [4]. Therefore, many exact and heuristic algorithms have been developed for the MKP and diverse variants of it [2, 15], and in particular several EAs were proposed [7].

Since it is difficult to analyze even simple EAs using direct encoding and primitive operators, most approaches to characterize locality are based on empirical rather than theoretical investigations. Often, locality is characterized by correlation measures based on fitness landscapes [3, 11, 14], however approaches based on that idea do not directly consider the effects of the used decoder and operators. Therefore we propose a locality concept which is independent of actual fitness values and explicitly examines the structural effects of decoder and employed mutation and crossover operators. Our approach involves measurements that can be applied without performing time-consuming EA runs. In this way it becomes possible in advance to identify and discard EAs which do not provide strong locality. Since the basic principle of our locality characterization is presented in a general fashion, it can be adapted to other decoder-based EAs.

This paper is organized as follows. Section 2 provides an overview of decoder-based EAs of varying complexity for the MKP together with an empirical comparison of the achieved performance on a standard MKP test suite. Our approach to characterize locality is introduced in Sect. 3. Section 4 presents empirical measurements performed for four selected decoder-based EAs. Our results point at important properties and differences of these EAs, which help to explain the achieved performance. Conclusions are given in Sect. 5.

2 Decoder-Based EAs for the MKP

Many EAs with different constraint handling strategies have been proposed for the MKP, see [7] for a survey. On the one hand, there are approaches directly working in the *phenotyic search space* $P = \{0, 1\}^n$. They rely on penalizing or repairing infeasible solutions. The currently best EAs for the MKP we are aware of [2, 7, 17] are based on heuristic repair and local improvement methods to focus search on the boundary of the feasible region, which is known to contain the optimum [6]. On the other hand, some EAs proceed by exploring an arbitrary *genotypic search space* G, which is mapped into P by a *decoder*. Such a decoder usually employs problem-specific knowledge ensuring to generate only feasible solutions, hence no penalties or repair methods are necessary to deal with infeasible solutions, and usually simple operators can be used. These *decoder-based EAs* perform an indirect search for solutions, and therefore their success strongly depends on the employed decoder. Obviously, such EAs can only be successful if the fittest parts of P are covered, i.e. there exist genotypes which decode into these phenotypes. Furthermore, a decoder is required to be computationally fast, since otherwise the evaluation of many solution candidates would be too time-consuming. Another important factor is how well a decoder supports locality. Since our goal is to characterize several decoder-based EAs concerning this last aspect, we proceed by a brief introduction and empirical comparison of them.

2.1 Permutation Based EA

The *permutation based EA* (PBEA) has been proposed by Hinterding for the (unidimensional) knapsack problem [9] and can easily be adapted to the MKP [17, 24]. A solution candidate is represented by a permutation $\pi : J \rightarrow J$ of the items. The decoder starts with the feasible solution $x = (0, \dots , 0)$ and traverses all variables x_j in the order determined by π, increasing the corresponding variable from 0 to 1 if this does not violate any resource constraint. Hinterding employs standard permutation operators, namely uniform order based crossover and swap mutation which randomly exchanges two different positions.

2.2 Ordinal Representation Based EA

The *ordinal representation based EA* (OREA) was originally considered for the traveling salesperson problem (TSP) [8], however its application to MKP is straightforward. Solution candidates are represented by a vector v with $v_a \in \{1, \dots , n - a + 1\}$ for $a \in J = \{1, \dots , n\}$. The decoder is based on a list initially containing all items in a predefined order and starts with the MKP solution $x = (0, \dots , 0)$. Items are iteratively removed from the list and checked for inclusion in the solution. In detail, v is scanned from the first to its last position, interpreting each entry v_a as a position in the current list. Such a position identifies the next item j, for which x_j is increased to 1 if the resource constraints remain satisfied. Since each checked item is removed from the list, its size decreases by 1 during each step and reaches length 1 when the last item is to be selected. This representation has the interesting property that classical one-point crossover is applicable because resulting offsprings always represent legal solutions. Moreover, a simple mutation operator can be used which randomly chooses a position a and then draws v_a from $\{1, \dots , n - a + 1\}$. However, a closer look at the decoding procedure reveals that a change in a single position of v might have a major impact on the decoded solution since each item selection modifies the list, thus, influences all following item selections. According to our locality conception we expect OREA to yield bad results due to its weak locality.

2.3 Surrogate Relaxation Based EA

Raidl proposed the *surrogate relaxation based EA* (SREA) which represents solution candidates by real-valued weights for the items [18]. These weights are used to temporarily modify the profits p_j in the objective function (1) yielding a similar but slightly different MKP instance. This biased problem is then solved by a surrogate duality based heuristic. The solution obtained in this way is also feasible for the original, unbiased problem since the resource constraints (2) remain unchanged. The heuristic, which has originally been proposed by Pirkul [16], starts with the solution $x = (0, \dots , 0)$ and traverses all items according to decreasing profit/pseudo-resource consumption ratio. Variables x_j are set to 1 if no resource constraint is violated. Pseudo-resource consumptions are determined via reasonable surrogate multipliers which are obtained from the linear programming (LP) relaxed MKP. Since this process would result in solving the LP relaxation for each solution candidate, Raidl suggests to determine the surrogate multipliers only once for the original problem in a preprocessing step to decrease the computational effort [18]. SREA uses uniform crossover and a mutation operator which is applied 3 times to each new genotype, modifying a randomly chosen weight by resetting it to a new random value. The results of SREA are the best among all decoder-based EAs for the MKP we are aware of.

2.4 Lagrangian Relaxation Based EA

The *Lagrangian relaxation based EA* (LREA) was also proposed by Raidl and is basically equivalent to SREA, except for the heuristic used to generate a solution for the biased problem [18]. LREA employs the procedure introduced by Magazine and Oguz [13] to obtain a solution via Lagrangian relaxation. Since the determination of exact Lagrange multipliers is too time-consuming, some reasonable (but usually suboptimal) multipliers are calculated by a simpler heuristic. Each obtained solution is then locally improved by traversing the variables according to decreasing profit and increasing them if feasibility can be maintained.

2.5 Comparison of Decoder-Based EAs

All considered EAs are based on decoders which use resource information to produce only solutions on the boundary of the feasible region, hence the search concentrates on the most promising parts of the phenotype space. Nevertheless the EAs significantly differ in the employed problem specific knowledge. While the decoders of PBEA and OREA ignore profit information, the heuristics employed by LREA and SREA strongly depend on it. Thus, SREA and LREA exploit more knowledge about the problem structure. We remark another interesting relation between the decoders of PBEA, OREA, and SREA: Both OREA and SREA internally produce a permutation of the items which is then interpreted in exactly the same fashion as PBEA decodes solution candidates.

We compare the described decoder-based EAs on a standard test suite of MKP benchmark problems introduced by Chu [2] and available from the OR-Library[3]. The test suite contains 10 instances for each combination of $m \in \{5, 10, 30\}$, $n \in \{100, 250, 500\}$, and tightness ratio $\alpha \in \{0.25, 0.5, 0.75\}$ (each problem has been generated randomly such that $c_i = \alpha \sum_{j \in J} r_{ij}$ holds for all $i \in I$). We selected the first problem of each category yielding a total of 27 problems and performed three runs for each instance. A similar general setup as in [2, 7, 17, 18] was chosen for all EAs, namely population size 100, parent selection via tournaments of size 2, steady-state replacement (replacing the worst individual), crossover probability 1.0, duplicate elimination (a newly

[3] http://mscmga.ms.ic.ac.uk/info.html

Table 1. Average results of the EAs

m	n	gap [%]				duplicate ratio [%]			
		PBEA	OREA	SREA	LREA	PBEA	OREA	SREA	LREA
5	100	0.53	1.19	0.53	0.52	6.40	34.92	12.11	5.89
5	250	0.25	2.05	0.17	0.16	4.33	33.34	4.28	2.97
5	500	0.21	3.18	0.07	0.10	4.62	34.00	3.62	2.56
10	100	1.00	1.59	1.00	1.00	5.12	36.14	6.84	3.81
10	250	0.57	2.33	0.34	0.35	4.88	36.14	5.11	2.39
10	500	0.54	3.29	0.19	0.25	4.93	36.62	4.58	2.24
30	100	1.78	2.76	1.67	1.70	7.72	38.76	8.73	3.62
30	250	0.97	3.46	0.75	0.87	6.36	38.27	7.52	2.94
30	500	0.85	4.00	0.47	0.59	7.30	37.98	6.49	2.87
total		0.74	2.65	0.58	0.62	5.74	36.24	6.65	3.27

generated individual is only accepted if it is not already contained in the population), and an evaluation limit of 200 000 non-duplicate solutions. We observed that in particular duplicate avoidance is essential to prevent an overcrowding of the population by many duplicates of only few different solutions (premature convergence) [19]. Duplicates should be identified on phenotypic rather than genotypic level, i.e. an individual is rejected if its decoded solution is already represented in the current population. The solution quality is measured by the *gap* of the objective value w.r.t. the optimal value of the LP-relaxed problem, i.e. $1 - max^{\mathrm{EA}}/opt^{\mathrm{LP}}$ with max^{EA} and opt^{LP} denoting the best objective value found by the EA and the optimal value of the LP relaxation of MKP, respectively.

Table 1 shows average results determined from the 9 runs per m, n-combination and EA. The *duplicate ratio* (DR) represents the ratio of rejected duplicates among all generated solutions. As expected, OREA yields the worst gap. Furthermore, the high DR indicates that the used operators tend to produce many duplicates. The other EAs perform quite well compared to OREA, so they should also provide locality. The best quality is obtained by SREA, probably due to the employed heuristic. LREA achieves the lowest DR, however PBEA and SREA also yield an acceptable DR which is an order of magnitude smaller than that of OREA. We conclude that the operators of PBEA, SREA, and LREA mostly generate new solutions. In general, we consider PBEA, SREA, and LREA to be well adapted to the MKP, in contrast to OREA which is viewed as an example for a badly designed decoder-based EA. The rest of this paper examines the effects of locality, which helps to explain the results from Table 1.

3 Measures for Locality Characterization

Many approaches were proposed to predict an EA's performance for a given problem. Since several techniques are based on different interpretations of locality, we briefly review them to enable a clear distinction from our new approach to measure the locality of decoder-based EAs. Many proposals from literature are based on the fitness landscape which enables an examination of the relation between solution candidates and their fitness values. Beside some theoretical proofs of convergence rates for EAs applied to relatively simple test functions (e.g. evolution strategies applied to the corridor or the sphere model [20]), Manderick et al. proposed to use correlation measures to examine the effects of operators [14]. They randomly generated parents, applied crossover to produce offsprings, and then calculated the correlation coefficient for the average fitness of parents and offsprings. Their approach predicted the performance of several operators in case of *NK*-landscapes and the TSP and has also been used for several other problems, e.g. minimum span frequency assignment [25]. Fogel and Ghozeil suggested to focus on the operators' abilities to produce offsprings with higher fitness than the parents [3]. Their model also considers the parent selection strategy to reflect actual EA dynamics and has been used for real-valued problems and the TSP. A different approach termed fitness distance correlation (FDC) was investigated by Jones and Forrest for classical genetic algorithms using binary encoding [11]. FDC is based on the intuition that fitness values should reflect the distance to an optimal solution. They proposed to randomly generate solution candidates and calculate the correlation of their fitness values to the distances to the optimum. For this purpose the Hamming distance is used, but a distance metric relying on the operators which actually define the edges of the landscape graph would be more appropriate. Jones and Forrest reported their approach being a reliable predictor of performance on the examined problems [11], however Altenberg provided a counterexample to show that such approaches might be misleading if actual EA dynamics are not considered [1]. In general, all formerly discussed approaches have some drawbacks limiting their ability to predict performance [1, 12], hence alternative approaches should be devised.

While most fitness based techniques only implicitly consider the encoding and used operators, in particular for decoder-based EAs the locality characteristics of these parts seem to be most important, hence should rather be explicitly examined. Locality should alternatively be interpreted in terms of the explicit relation between genotypes and phenotypes, which emphasizes the structural effects of operators and the employed decoder. Sendhoff et al. derived such locality conditions, motivated by the claim that small genotypic changes should imply small phenotypic changes [23]. Their probabilistic measures, which are solely focused on the mutation operator, were successfully applied to continuous parameter optimization and structure optimization. Based on Sendhoff et al.'s approach, Igel examined the probabilistic measures for *NK*-landscapes and problems in the genetic programming context [10]. He concluded that the proposed conditions are helpful to compare several codings and operators for a given problem.

We propose a statistical locality concept which considers crossover and mutation operators, enabling a separate analysis of each.

3.1 Distance Metrics in Genotype and Phenotype Spaces

To characterize the locality of the operators within the genotype space G and the phenotype space P, we quantify the distance of two arbitrary solutions in both G and P. Therefore we introduce distance metrics to measure how many different *properties* (either genotypic or phenotypic) are present in two solutions.

For the MKP, the definition of a *phenotypic distance metric*

$$d_P(x, y) := \sum_{j \in J} |x_j - y_j| \quad \text{for } x, y \in P \tag{4}$$

is straightforward, since the Hamming distance counts the number of variables with different values (different phenotypic properties) in the two solutions. For other combinatorial optimization problems, the phenotypic distance usually needs to be defined in some different, meaningful way, which might not always be as obvious as for the MKP. E.g. in case of the TSP, the total number of different edges might be an appropriate measure since edges can be seen as the most important phenotypic properties of TSP solutions [8].

The definition of a *genotypic distance metric* $d_G(X, Y)$ for $X, Y \in G$ is not that straightforward as it depends on the specific encoding and the operators. To remain general, we implicitly define $d_G(X, Y)$ via the mutation operator:

1. Two identical genotypes have distance 0, i.e. $d_G(X, X) = 0$ for $X \in G$.
2. Two distinct genotypes X, Y have distance $d_G(X, Y) = 1$ and are called *adjacent* if the probability to produce Y from X by a single mutation is greater than 0.
3. In general, a genotypic distance $d_G(X, Y) = k$ means that at least k mutations are necessary to transform X into Y.

For the considered EAs, the proposed definitions of $d_G(X, Y)$ and $d_P(x, y)$ satisfy the metric conditions, namely identity, symmetry, and the triangular inequality.

3.2 Mutation Innovation *MI*

Usually, the mutation operator is not applied exactly once but either with a given probability or a certain number of times. Let $X^m \in G$ represent the solution obtained from $X \in G$ by applying mutation with exactly this probability or rate. Note that according to the previous definitions, k consecutive mutations will produce an offspring with $d_G(X, X^m) \leq k$.

Let x and x^m be the phenotypes corresponding to X and X^m. Assuming X to be a random variable with uniform distribution within G, X^m, x, and x^m are dependent random variables. We define the *mutation innovation* as

$$MI := d_P(x, x^m)\,, \tag{5}$$

which describes how much phenotypic "innovation" is introduced by the mutation. *MI* is a random variable, whose distribution immediately reflects several important aspects concerning locality of mutation.

We have to consider the case $MI = 0$, occurring with probability $P(MI = 0)$ and meaning that the mutation has not affected the phenotypic properties. Large values for $P(MI = 0)$ indicate that either mutation often does not change any genotypic properties or many different genotypes are mapped to the same phenotype, which reflects a high degree of redundancy in G. There are two possible reasons for such high redundancy.

Firstly, $|G|$ might be significantly larger than $|P|$. Often such a representational redundancy decreases performance, but sometimes it may also be beneficial and lead to better final results [22].

A second reason may be that the decoder contains local improvement techniques or heuristics which always or mostly lead to preferred phenotypes in a restricted subset $P' \subset P$. We call this effect *heuristic bias*. Therefore, solutions $x \in P \setminus P'$ cannot be represented or have substantially smaller probabilities to be generated. While such a restriction of P might sometimes be advantageous, it must be ensured that promising areas and particularly the global optima are covered [18, 22]. As already mentioned, the four considered EAs for the MKP restrict the search space to the boundary of the feasible region, therefore, they work with heuristic bias.

$P(MI = 0)$ can principally be controlled by tuning the mutation probability or rate. Obviously, a higher mutation rate would decrease $P(MI = 0)$, but the usually resulting larger changes in genotype may also affect the EA to behave more like inefficient random search. Now, consider only the cases in which mutation produces an offspring x^m which actually differs from x. Then the expected value for MI under this restriction, called $E(MI|MI > 0)$, and the standard deviation $\sigma(MI|MI > 0)$ are good indicators for the locality of mutation. Only if both $E(MI|MI > 0)$ and $\sigma(MI|MI > 0)$ are reasonably small, successful mutations lead in general to similar phenotypes. Large values signalize that very different solutions are frequently generated, only negligible or none locality is given, and hence the search of the EA tends to be a random search.

3.3 Crossover Innovation CI_k

When using binary crossover, a new genotype $X^c \in G$ is generated from two parental solutions $X^{p_1}, X^{p_2} \in G$. Let $x^{p_1}, x^{p_2}, x^c \in P$ be the phenotypes corresponding to X^{p_1}, X^{p_2}, X^c. Usually, the result of the crossover operation is strongly influenced by the similarity of the parents. In early stages of an EA run the population has high diversity, hence most selected parents differ significantly, while in later stages the population is likely to be converged, i.e. similar parents are frequently involved in crossover applications. To reflect population diversity in an approximate way using our notion of genotypic distance, let us assume that X^{p_2} is produced by applying $k \geq 1$ consecutive mutations to X^{p_1}, i.e. the genotypic distance between the parents is $d_G(X^{p_1}, X^{p_2}) \leq k$. Additionally, we take the duplicate elimination of the EAs into account by considering only phenotypicly non-identical parents, therefore we presume $d_P(x^{p_1}, x^{p_2}) > 0$. We then define the *crossover innovation*

$$CI_k := \min(d_P(x^c, x^{p_1}), d_P(x^c, x^{p_2})) \tag{6}$$

as the phenotypic distance of x^c to its closer parent. If we interpret the genotypes X^{p_1}, X^{p_2} as random variables with the restriction $d_G(X^{p_1}, X^{p_2}) \leq k$, then $X^c, x^{p_1}, x^{p_2}, x^c$, and in particular CI_k are dependent random variables.

Obviously, CI_k is 0 if either $x^c = x^{p_1}$ or $x^c = x^{p_2}$. Analogously to $P(MI = 0)$, the probability $P(CI_k = 0)$ indicates the likelihood for crossover generating an offspring which is phenotypicly identical to one of its parents. Usually, $P(CI_k = 0)$ is higher for parents with very similar or equal genotypes, i.e. for small k. A high $P(CI_k = 0)$ for large k indicates that crossover does not mix genotypes well or the degree of redundancy in the mapping $G \to P$ is high. Especially when both probabilities $P(MI = 0)$ and $P(CI_k = 0)$ are high, the EA cannot work efficiently since many duplicate solutions are generated.

Analogously to the mutation, we restrict our considerations in the following to the case of crossover actually producing new solutions with $CI_k > 0$. The expectations $E(CI_k|CI_k > 0)$ and standard deviations $\sigma(CI_k|CI_k > 0)$ for different maximum parent distances k are meaningful indicators for the existence or absence of locality during crossover: For strong locality, $E(CI_k|CI_k > 0)$ should be small for small k and become increasingly larger for larger k. In particular, large values for both, $E(CI_k|CI_k > 0)$ and $\sigma(CI_k|CI_k > 0)$, for small k are strong indicators for weak locality.

3.4 Crossover Loss CL_k

Besides the ability to generate new solutions with adequate distances to their parents, an important aspect of crossover is that a generated solution mainly consists of properties inherited from its parents; only few new properties should be introduced. The importance of such behavior is e.g. described by the building-block hypothesis [5]. To consider this aspect, too, we define the *crossover loss* CL_k as the number of phenotypic properties of x^c which are not inherited from either x^{p_1} or x^{p_2} but are newly introduced. For the MKP this means

$$CL_k := \sum_{j \in J} \delta(x_j^c, x_j^{p_1}, x_j^{p_2}) \tag{7}$$

$$\text{with} \quad \delta(x_j^c, x_j^{p_1}, x_j^{p_2}) = \begin{cases} 0 & \text{if } x_j^c = x_j^{p_1} \text{ or } x_j^c = x_j^{p_2} \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

Considering also the proposed phenotypic distance metric, the crossover loss can alternatively be written as

$$CL_k := \frac{1}{2}(d_P(x^c, x^{p_1}) + d_P(x^c, x^{p_2}) - d_P(x^{p_1}, x^{p_2}))\,. \tag{9}$$

Clearly, $CI_k = 0$ implies $CL_k = 0$. To prevent a bias by the case where crossover does not produce a new solution, we preclude that case and consider the expected values $E(CL_k|CI_k > 0)$ for different maximum parent distances k. Large values for any k immediately signalize weak locality.

4 Empirical Measurements

Determining the different measures introduced in Sect. 3 for specific encoding techniques and crossover and mutation operators in a theoretical way is in general

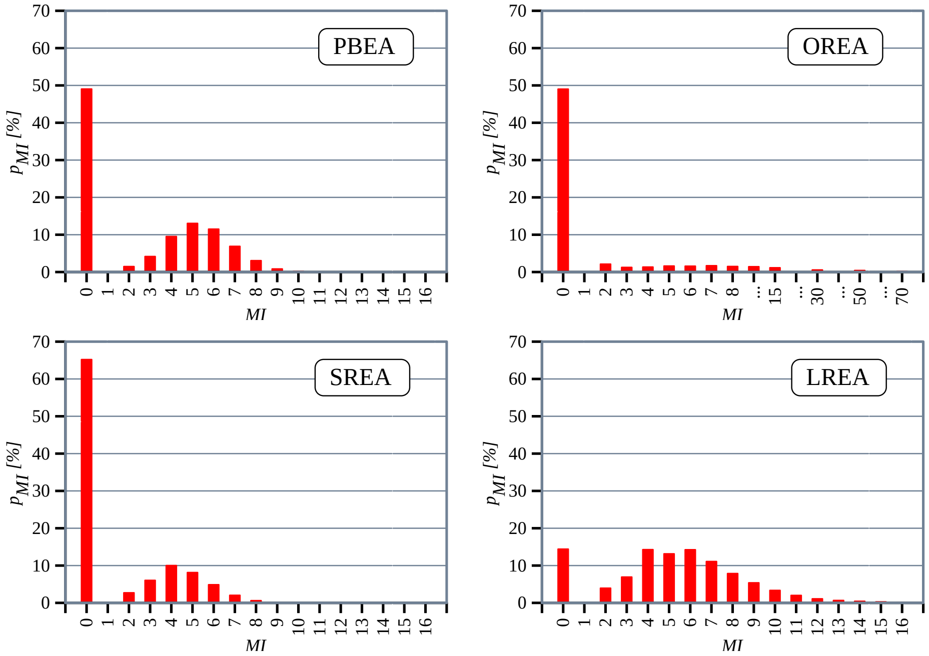

Fig. 1. Empirically determined distributions of MI for the different EAs

Table 2. Empirically estimated characteristics for MI

measure	PBEA	OREA	SREA	LREA
$P(MI = 0)$ [%]:	49.06	49.04	65.22	14.43
$E(MI \mid MI > 0)$:	5.34	27.87	4.47	6.16
$\sigma(MI \mid MI > 0)$:	1.55	22.93	1.44	2.72

a very hard task and especially for more complex EAs like SREA or LREA practically nearly impossible. Therefore, we use an efficient empirical approach to obtain good estimations for specific problem instances. Since these measurements can be performed a priori to any EA run, inadequate encoding techniques or operators providing weak locality can be early recognized.

The empirical results presented in this section are based on an average sized MKP instance with $m = 10$, $n = 250$, and $\alpha = 0.5$, namely the first problem of Chu's benchmarks with these parameters [2]. Although different absolute values have been obtained as results for other instances, this specific problem is representative in the sense that the same basic tendencies have been observed for other instances, too.

MI has been empirically estimated by randomly drawing 20 000 genotypes $X \in G$ and applying mutation to each with the same probability or rate as in the EA. Figure 1 shows histograms for the distributions of MI gained in this way, while Table 2 subsumes the estimations for the measures $P(MI = 0)$, $E(MI|MI > 0)$, and $\sigma(MI|MI > 0)$ introduced in Sect. 3.2.

Most noticeable are the generally large values for $P(MI = 0)$. One reason for this effect is that all four EAs generate only phenotypes on the boundary of the feasible region, therefore in a substantially restricted phenotype subset $P' \subset P$. Furthermore, there exists an encoding redundancy in all EAs due to the different sizes of G and P. While $|P| = 2^n$, the genotype space has size $|G| = n!$ in PBEA and OREA and is even larger for SREA and LREA since real values are allowed for each weight. Although the used mutation rate for both SREA and LREA was 3 (instead of only 1 as in PBEA and OREA), SREA has the highest probability $P(MI = 0)$. The reason is the small biasing factor[4] γ of SREA, which means that the original problem is in general significantly less biased than in LREA [18]. Therefore, the heuristic bias of SREA is clearly stronger than that of LREA. Note that $P(MI = 1) = 0$ for all EAs since $d_P(x, y) \geq 2$ for two distinct solutions $x, y \in P$ lying on the boundary of the feasible region. Regarding $E(MI|MI > 0)$ and $\sigma(MI|MI > 0)$, OREA yields substantially higher values than the other EAs. This result is an immediate indicator for the weak locality of the mutation operator and encoding in OREA and can be explained by the strong dependency of each gene's interpretation from all its predecessors in the genotype. According to the other values for $E(MI|MI > 0)$ and $\sigma(MI|MI > 0)$, SREA provides the highest locality followed by PBEA and LREA.

CI_k and CL_k were empirically estimated for $k \in \{2^i \mid i = 0, \ldots, 9\}$ by randomly generating 20 000 parents X^{p_1} for each k, applying k mutations to each X^{p_1} to obtain the associated second parent X^{p_2}, and then producing offsprings X^c via crossover. According to Sect. 3.3, parents representing identical phenotypes ($d_P(x^{p_1}, x^{p_2}) = 0$) were discarded. Obtained estimations for $P(CI_k|CI_k = 0)$, $E(CI_k|CI_k > 0)$, $\sigma(CI_k|CI_k > 0)$, and $E(CL_k|CI_k > 0)$ are shown in Fig. 2.

The curves for $P(CI_k = 0)$ indicate that PBEA has in general the lowest probability for generating a child identical to one of its parents. Especially for $k = 1$, no other EA is able to produce a solution different from its parents. While $P(CI_k = 0)$ decreases with increasing k down to 0 for PBEA, SREA, and LREA, $P(CI_k = 0)$ remains above 45% for OREA. The reason is that OREA uses one-point crossover, which might frequently exchange genes having no effect on the decoded phenotype, because the phenotypic properties are mainly determined by the first genes. Note that the large duplicate ratio of OREA during actual EA runs (see Sect. 2.5) could have been predicted by the high probabilities $P(CI_k = 0)$ and $P(MI = 0)$. Especially for SREA but also for LREA, k must be relatively high to mostly obtain new solutions that are different from their parents. This observation emphasizes the importance of taking care of the population diversity in the EA by discarding generated phenotypic duplicates in order to enable crossover to work efficiently (and hence prevent premature convergence).

For small k, the expected values $E(CI_k|CI_k > 0)$ are reasonably small and nearly equal for PBEA, LREA, and SREA, but relatively high for OREA. This

[4] Biasing factors were set as proposed in [18]: LREA: $\gamma = 0.2$, SREA: $\gamma = 0.05$.

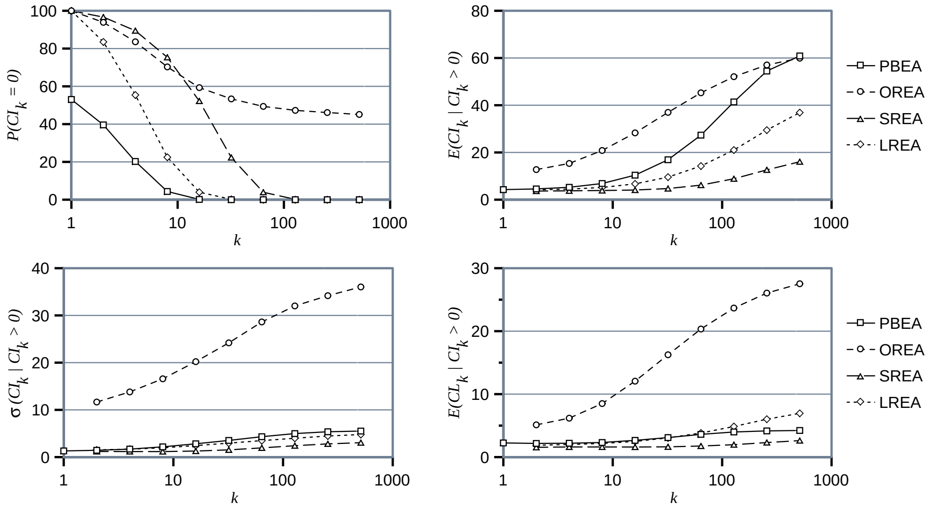

Fig. 2. Empirically determined estimations for $P(CI_k = 0)$, $E(CI_k|CI_k > 0)$, $\sigma(CI_k|CI_k > 0)$, and $E(CL_k|CI_k > 0)$

immediately indicates relatively strong locality for crossover of PBEA, LREA, and SREA, but weak locality for OREA.[5] With increasing k, $E(CI_k|CI_k > 0)$ becomes larger for all EAs. The final values which are quite different for the four EAs indicate that PBEA and OREA are in general capable of generating more innovative solutions than LREA and especially SREA. Since the uniform crossover operator of SREA and LREA mixes genotypes at least as well as the crossover operators in the other EAs, the reason for the smaller CI_k for larger k is again the higher heuristic bias towards a smaller phenotype subset P'. The weak locality of OREA is also clearly indicated by the large standard deviations $\sigma(CI_k|CI_k > 0)$ compared to the other EAs' corresponding values.

The smallest expected crossover loss $E(CL_k|CI_k > 0)$ is achieved by SREA (always less than 3) which indicates that x^c inherits nearly all phenotypic properties from its parents. The corresponding values for PBEA and LREA are also reasonably low but slightly higher than for SREA (for increasing k), while for OREA $E(CL_k|CI_k > 0)$ is substantially larger for all values of k. This reflects the poor capabilities of crossover in OREA to build offsprings by inheriting most phenotypic properties, which once again implies weak locality.

5 Conclusions

We investigated four decoder-based EAs for the multidimensional knapsack problem (MKP), focusing on the locality achieved by the employed decoders and

[5] Note that $E(CI_1|CI_1 > 0)$, $\sigma(CI_1|CI_1 > 0)$, and $E(CL_1|CI_1 > 0)$ are not defined for OREA, LREA, and SREA since CI_1 is always 0.

operators (mutation and crossover). The performed experiments demonstrate the ability of the proposed locality measures to predict poor performance due to weak locality. In case of the MKP, SREA, which is the best performing EA according to the results in Sect. 2.5, offers also the strongest locality regarding mutation and crossover. This is clearly indicated by the small estimated values for $E(MI|MI = 0)$, $E(CI_k|CI_k > 0)$ (for small k), associated standard deviations, and $E(CL_k|CI_k > 0)$. The weak locality of OREA, which is proved by the substantially larger estimations for these measures, is one reason why this EA performs significantly worse than the other considered EAs.

Another important aspect of all four EAs is signalized by the generally high probabilities $P(MI = 0)$ and $P(CI_k = 0)$ for small k: Due to heuristic bias and/or redundancy in the coding, the proportion of operator applications which actually lead to new, different phenotypes may be considerably small. It is therefore important to actively maintain enough diversity in the population, e.g. by discarding generated phenotypic duplicates.

The prediction capabilities of the a priori measurements have also been verified by determining the proposed measures online during actual EA runs (details were not presented here). The online results differed only slightly from the a priori results, provided that a proper k, which can be derived from the population diversity during the EA run, is used for the comparisons regarding CI_k and CL_k.

Generally, our results confirmed locality to be a necessary condition for decoder-based EAs to work well for MKP. However, we are aware that locality is not sufficient for good performance. Also the concept of heuristic bias strongly affects the achieved performance, thus should be examined more detailed. Although the presented locality measures can be used as indicators, it is interesting to check whether additional measures could also reliably predict heuristic bias and hence the total performance. The proposed ideas should also be validated on different problems to verify whether our results can be generalized to hold for other problems, too.

References

1. L. Altenberg: *Fitness Distance Correlation Analysis: An Instructive Counterexample*, in Proc. of the 7th Int. Conf. on Genetic Algorithms, East Lansing, MI, pp. 57 – 64, 1997
2. P. C. Chu, J. E. Beasley: *A Genetic Algorithm for the Multidimensional Knapsack Problem*, Journal of Heuristics 4, pp. 63 – 86, 1998
3. D. B. Fogel, A. Ghozeil: *Using Fitness Distributions to Design More Efficient Evolutionary Algorithms*, in Proc. of the 3rd IEEE Int. Conf. on Evolutionary Computation, Nagoya, Japan, pp. 11 – 19, 1996
4. M. D. Garey, D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979
5. D. E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989
6. J. Gottlieb: *Evolutionary Algorithms for Multidimensional Knapsack Problems: The Relevance of the Boundary of the Feasible Region*, in Proc. of the Genetic and Evolutionary Computation Conf., Orlando, FL, p. 787, 1999

7. J. Gottlieb: *On the Effectivity of Evolutionary Algorithms for Multidimensional Knapsack Problems*, in Proc. of Artificial Evolution, Dunkerque, France, 1999
8. J. J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht: *Genetic Algorithms for the Traveling Salesman Problem*, in Proc. of the 1st Int. Conf. on Genetic Algorithms, Hillsdale, NJ, pp. 160 – 168, 1985
9. R. Hinterding: *Mapping, Order-independent Genes and the Knapsack Problem*, in Proc. of the 1st IEEE Int. Conference on Evolutionary Computation, Orlando, FL, pp. 13 – 17, 1994
10. C. Igel: *Causality of Hierarchical Variable Length Representations*, in Proc. of the 5th IEEE Int. Conf. on Evolutionary Computation, Anchorage, AL, pp. 324 – 329, 1998
11. T. Jones, S. Forrest: *Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms*, in Proc. of the 6th Int. Conf. on Genetic Algorithms, Pittsburgh, PA, pp. 184 – 192, 1995
12. L. Kallel, M. Schoenauer: *A Priori Comparison of Binary Crossover Operators: No Universal Statistical Measure, but a Set of Hints*, in Proc. of 3rd European Conf. on Artificial Evolution, Nîmes, France, pp. 287 – 299, 1997
13. M. J. Magazine, O. Oguz: *A Heuristic Algorithm for the Multidimensional Zero–One Knapsack Problem*, European Journal of Op. Res., 16, pp. 319 – 326, 1984
14. B. Manderick, M. de Weger, P. Spiessens: *The Genetic Algorithm and the Structure of the Fitness Landscape*, in Proc. of the 4th Int. Conf. on Genetic Algorithms, pp. 143 – 150, 1991
15. S. Martello, P. Toth: *Knapsack Problems: Algorithms and Computer Implementations*, J. Wiley & Sons, 1990
16. H. Pirkul: *A Heuristic Solution Procedure for the Multiconstrained Zero-One Knapsack Problem*, Naval Research Logistics 34, pp. 161 – 172, 1987
17. G. R. Raidl: *An Improved Genetic Algorithm for the Multiconstrained 0–1 Knapsack Problem*, in Proc. of the IEEE Int. Conf. on Evolutionary Computation, Anchorage, AL, pp. 207 – 211, 1998
18. G. R. Raidl: *Weight-Codings in a Genetic Algorithm for the Multiconstraint Knapsack Problem*, in Proc. of the Congress on Evolutionary Computation, Washington DC, pp. 596 – 603, 1999.
19. G. R. Raidl, J. Gottlieb: *On the Importance of Phenotypic Duplicate Elimination in Decoder-Based Evolutionary Algorithms*, in Late-Breaking Papers Proc. of the Genetic and Evolutionary Computation Conf., Orlando, FL, pp. 204 – 211, 1999.
20. I. Rechenberg: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973
21. I. Rechenberg: *Evolutionsstrategie '94*, Frommann-Holzboog, 1994
22. S. Ronald: *Robust Encodings in Genetic Algorithms*, in D. Dasgupta, Z. Michalewicz (eds.), Evolutionary Algorithms in Engineering Applications, pp. 29 – 44, Springer, 1997
23. B. Sendhoff, M. Kreutz, W. von Seelen: *A condition for the genotype-phenotype mapping: Causality*, in Proc. of the 7th Int. Conf. on Genetic Algorithms, East Lansing, MI, pp. 73 – 80, 1997
24. J. Thiel, S. Voss: *Some Experiences on Solving Multiconstraint Zero-One Knapsack Problems with Genetic Algorithms*, INFOR 32, pp. 226 – 242, 1994
25. C. Valenzuela, S. Hurley, D. Smith: *A Permutation Based Genetic Algorithm for Minimum Span Frequency Assignment*, in Proc. of the 5th Int. Conf. on Parallel Problem Solving from Nature, Amsterdam, The Netherlands, pp. 907 – 916, 1998