Worst improvement based Iterated Local Search

Sara Tari, Matthieu Basseur, Adrien Goëffon

Laboratoire d'Etude et de Recherche en Informatique d'Angers
UFR sciences, 2 boulevard Lavoisier 49045 Angers cedex 01
{sara.tari, matthieu.basseur, adrien.goeffon}@univ-angers.fr

Abstract. To solve combinatorial optimization problems, many metaheuristics use first or best improvement hill-climbing as intensification mechanism in order to find local optima. In particular, first improvement offers a good tradeoff between computation cost and quality of reached local optima. In this paper, we investigate a worst improvement-based moving strategy, never considered in the literature. Such a strategy is able to reach good local optima despite it requires a significant additional computation cost. Here, we investigate if such a pivoting rule can be efficient when considered within metaheuristics, and especially within iterated local searches (ILS). In our experiments, we compete an ILS using a first improvement pivoting rule with an ILS using an approximated version of worst improvement pivoting rule. Both methods are launched with the same number of evaluation on bit-string based fitness landscapes. Results are analyzed thanks to some landscapes' features in order to determine if worst improvement principle should be considered as a moving strategy in some cases.

1 Introduction

Since the last two decades, the number of new metaheuristics in the literature is unreasonably growing. Although the creativity of researchers can be outstanding, it is not conceivable to create a new paradigm for each considered optimization problem. Moreover, it is hard to choose the most adequate metaheuristic to tackle a given optimization problem among the great variety of existing methods. Indeed, many metaheuristics are proposed for a single problem without a further analysis of their behavior. As stated by Sörensen in [10], there is a need to understand metaheuristics components in order to know when and how to use them. Nevertheless, there exists some studies which try to obtain insights about metaheuristics behavior, some of them using the concept of fitness landscapes. Some previous works are particularly dedicated to hill-climbing algorithms (climbers) and will be discussed in section 3.

At first, focusing on climbers may seem obsolete since there exists metaheuristics more efficient to handle optimization problems. Yet, climbers are particularly basic and often used as intensification mechanisms of more sophisticated metaheuristics. We believe there is a need to deconstruct metaheuristics in order to obtain insights about their behavior and one way of doing this is to study some basic components of metaheuristics, including intensification techniques.

In this work, we focus on a climbing technique based on the worst improvement pivoting rule, which is a counter intuitive moving strategy since it selects the least improving neighbor at each step of the search. Such a strategy showed interesting results in a previous work [11], since it generally allows the attainment of better local optima than when using first improvement. Yet, such a method needs a higher computational budget, even if worst improvement approximation variants offer interesting tradeoffs between quality and computation cost. Here, we propose to investigate further the behavior of worst improvement and approximated variants. First, we provide an extended analysis of such a climber, thanks to new experiments also combined with landscape analysis. Then, we compete iterated local searches using first improvement and worst improvement variants climbers, in order to determine their relative efficiency in an iterated context. Indeed, even if first improvement is outperformed by worst improvement in terms of local optima quality, its low computation cost allow to perform more climbing processes in a fixed maximal number of evaluations.

This study is conducted on two different types of bit-string fitness landscapes: NK landscapes and UBQP landscapes (derived from Unconstrained Binary Quadratic Programming problem instances).

In the next section we introduce definitions of concepts and problems used in this study. Section 3 is devoted to motivations of our work and previous results. In section 4 we report and analyze results by the means of some combinatorial landscape properties. Finally, in the last section we discuss our work.

2 Definitions

The concept of fitness landscape was originally introduced by Wright [12] in the field of theoretical biology, to represent an abstract space which links individual genotypes with their reproductive success. In particular, fitness landscapes are useful to simulate mutational paths and to observe the effect of successive mutations. Nowadays, such a concept is used in various fields to study the behavior of complex systems. In evolutionary computation, fitness landscapes can help to understand and thus to predict the behavior of neighborhood-based solving algorithms regardless of the problem considered. Let us introduce some concepts and definition, then we will present fitness landscape instances used in this paper.

2.1 Fitness Landscapes and related concepts

A fitness landscape is formally defined by a triplet $(\mathcal{X}, \mathcal{N}, f)$ where $\mathcal{X}$ denotes the set of feasible solutions also called search space, $\mathcal{N}$ denotes the neighborhood function which assigns a set of neighboring solutions to each solution of the search space and f denotes the fitness function which assigns a value to each solution.

Fitness landscapes are determined by many characteristics relative to their topology, ruggedness, neutrality, etc. In the current work we mainly focus on two classical features: size and ruggedness.

In the following the size of a combinatorial fitness landscape will denote its number of solutions. The ruggedness is directly linked to the number and the repartition of the local optima. A landscape with a few local optima and large basins of attraction can be views as *smooth* whereas a landscape with many local optima and small basins is *rugged*. The basin of attraction of a local optimum [8] refers to the set of solutions from which a basic hill-climbing algorithm has a substantial probability to reach the considered local optimum. The presence of several local optima is directly related to the epistasis phenomenom. Epistasis occurs when the presence or absence of a given gene influences the fitness variation induced by a mutation. Sign epistasis occurs when a specific mutation improves a solution and deteriorates another one, as depicted in fig. 1. If the two solutions where the mutation is applied are neighbors we refer to 1-sign-epistasis, if the distance between these solutions is of k we refer to k-sign-epistasis [2]. In an iterative improvement context, we mostly refer to sign epistasis since such methods mostly focus on the improving and deteriorating aspect of moves.

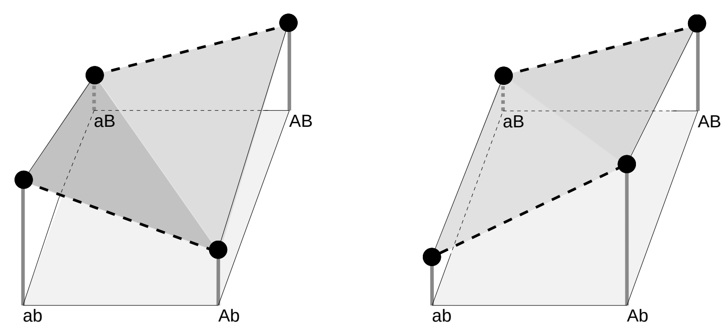

Fig. 1. Left-hand side : the presence of a mutation $a \to A$ affects the effect of the mutation $b \to B$ which becomes beneficial, there is sign-epistasis. Right-hand side: no sign epistasis.

For most combinatorial fitness landscapes it is not conceivable to enumerate all solutions, thus many landscape properties such as ruggedness cannot be exhaustively calculated. Many indicators can be used in order to estimate landscape properties [7], especially ruggedness. Here, we focus on the k-ruggedness indicator formally introduced in [2] and based upon k-sign epistasis. k-ruggedness refers to the k-epistasis rate within a sampling of pairs of k-distant solutions. 1-ruggedness which refers to the 1-epistasis rate on several pairs of mutation reflects local ruggedness whereas k-ruggedness (with $k > 1$) refers to a more global ruggedness of the landscape.

2.2 Bit-string landscapes instances

The NK landscape model [5] is widely used to generate artificial combinatorial landscapes with tunable size and ruggedness. NK landscapes are regularly used to understand the behavior of evolutionary process in function of ruggedness levels.

Properties of NK landscapes are determined by means of two parameters N and K. N refers to the length of bit-string solutions and $K \in \{0, \ldots, N-1\}$ specifies the variable interdependency hence the ruggedness of the landscape. Setting K to 0 leads to a completely smooth landscape consequently containing a single local (then global) optimum. Increasing K induces to increase the ruggedness of generated landscapes.

The topology of an NK landscape is a N-dimensional hypercube whose vertices are solutions and edges are neighborhood transitions (1-flip moves). The fitness function f_{NK} to be maximized is defined as follows:

$$f(x)_{NK} = \frac{1}{N} \sum_{i=1}^{N} C_i(x_i, \Pi_i(x))$$

x_i is the i-th bit of the solution x. Π_i is a subfunction which defines the dependencies of bit i, with $\Pi_i(x)$ such that $\pi_j(i) \in \{1, \ldots, N\} \setminus \{i\}$ and $| \cup_{j=1}^{N} \pi_j(i)| = K$. Subfunction $C_i : \{0,1\}^{K+1} \to [0,1)$ defines the contribution value of x_i w.r.t. its set of dependencies $\Pi_i(x)$. NK landscapes instances are determined by the $(K+1)$-uples $(x_i, x_{\pi_1(i)}, \ldots, x_{\pi_K(i)})$ and a matrix C of fitness contribution which describes the $2^N \times (K+1)$ possible contribution values.

The Unconstrained Binary Quadratic Programming problem (UBQP) is a NP-hard problem [4] regularly used to reformulate a large scope of real-life problems. An instance of UBQP is composed of a matrix Q filled with $N \times N$ positive or negative integers. A solution of UBQP is a binary vector x of size N where $x_i \in \{0,1\}$ corresponds to the i-th element of x. The objective function f_{UBQP} to be maximized is described as follows:

$$f_{UBQP}(x) = \sum_{i=1}^{n} \sum_{j=1}^{N} q_{ij} x_i x_j$$

The considered UBQP neighborhood is defined by the N solutions being at a Hamming distance of 1 (i.e. thanks to one-flip neighborhood operator). Thus N-dimensional NK and UBQP landscape instances only differ by their fitness function.

In experimental sections, a sample of 28 NK and 24 UBQP landscapes is used. NK landscapes are randomly generated using different size and variable interdependency parameter values: $N \in \{128, 256, 512, 1024\}, K \in \{1, 2, 4, 6, 8, 10, 12\}$. For UBQP landscapes, we use a set of 24 UBQP random instances[1], using different size and density parameter values: $N \in \{2048, 4096\}$, $d \in \{0.10, 0.25, 0.50, 0.75, 1\}$. The density $d \in [0,1]$ represents the expected proportion of non-zero values in the matrix Q.

[1] UBQP instances have been obtained with the instance generator provided at `http://www.personalas.ktu.lt/~ginpalu/ubqop_its.html`.

3 Worst improvement hill-climbing

3.1 Pivoting rule choice

A hill-climbing algorithm (climber) consists of selecting an improving neighbor of the current solution at each step of the search. When the current solution has no improving neighbor, a local optimum is reached and then implies the end of the climbing process. The main interrogation when designing a climber concerns the neighbor selection rule (*pivoting rule*), which directly affects the capacity of the search algorithm to reach good local optima. This is particularly true since the global optimum (or at least a high local optimum) can actually be reached using a single climber from most solutions, assuming the use of a relevant pivoting rule [3].

Several works about climbers [9, 1, 11] investigate the effects of using classical first and best improvement pivoting rules on the capacity of climbers to reach good solutions. These studies highlight that first improvement is often the most efficient rule to reach good local optima on sufficiently large and rugged landscapes. Let us recall that first improvement (F) consists of selecting the first improving neighbor encountered at each step of the search whereas best improvement (B) selects the improving neighbor with the highest fitness value at each step of the search. Since performing smaller steps with first improvement often leads toward better local optima, a study considered the worst improvement (W) pivoting rule [11], which consists of selecting the improving neighbor with the lowest fitness value at each step of the search. This work showed that W is more efficient than F and B on non-smooth NK landscapes (some results are extracted on the left side of table 1).

Worst improvement requires a very high computational budget in terms of number of evaluations since it evaluates the whole neighborhood at each step of the search and generally performs more steps than first and best improvement. To overcome this issue, intermediate rules W_k have been proposed in [11]. W_k approximates W by selecting the solution with the lowest fitness among k improving neighbors at each step of the search. This process avoids the generation of the whole neighborhood at each step of the search and can drastically reduce the required number of evaluation to attain a local optimum. Some results of the aforementioned study are extracted in the right side of table 1. W_k leads toward better solutions than first and best improvement even with $k = 2$ while significantly reducing the number of evaluations during the hill-climbing process. Let us notice that smooth NK landscapes are not considered since worst improvement is not efficient on such landscapes in comparison to a best-improvement hill-climbing

3.2 Additional experiments

In the current study we first perform additional experiments on UBQP landscapes (see section 2.2),following the same protocol than in [11]. For each couple (landscape, method) we perform 100 runs starting from the same set of 100

Landscape	F	B	W	W_2	W_4	W_8	W_{16}
$NK_{256,4}$	.7128 2k	.7211 19k	**.7274** 136k	.7233 5k	.7254 10k	.7257 21k	.7262 41k
$NK_{256,8}$	.7179 2k	.7147 13k	**.7267** 284k	.7218 13k	.7243 31k	.7259 73k	.7267 66k
$NK_{256,12}$	.7053 2k	.7015 10k	**.7129** 346k	.7089 5k	.7105 13k	.7124 33k	.7122 74k
$NK_{1024,4}$	.7238 12k	.7232 302k	**.7298** 2336k	.7253 25k	.7270 54k	.7286 122k	.7291 223k
$NK_{1024,8}$	.7215 13k	.7176 214k	**.7330** 5837k	.7251 31k	.7285 74k	.7306 172k	.7316 7330k
$NK_{1024,12}$	.7107 14k	.7064 166k	**.7210** 8544k	.7150 34k	.7178 84k	.7197 206k	.7206 487k

Table 1. Extract of previous results obtained with various hill-climbing on NK landscapes from [11]. For each couple (landscape,climber) we report the average fitness value of local optima as well as the average number of evaluations below.

randomly generated solutions. Results show that on UBQP landscapes (table 2, left side), worst improvement always leads toward better local optima averages than first and best improvement but requires a huge number of fitness evaluations. Approximated versions of worst improvement are also considered (tab. 2 - rigth side) and W_k appears to be particularly efficient while drastically reducing the number of evaluations. These results are similar to those obtained on NK landscapes and confirm the potential interest of worst improvement and its approximed variants.

The worst improvement rule was initially described in order to obtain better insights on what makes a climber efficient. Results showed that, despite the high computation cost, performing small improvements drives toward high local optima. Then, worst improvement approximations were proposed to obtain efficient pivoting rules which are not time consuming and thus potentially usable when tackling an optimization problem.

First improvement is the fastest and most simple way to reach local optima, which is an advantage when considering a fixed number of evaluations since it is able to reach a maximal number of local optima compared to any other strict hill-climbing pivoting rule. Considering more advanced pivoting rules which are more efficient but also more time consuming have to be investigated in a iterated context. Indeed, it is relevant to wonder if reaching less local optima than with first improvement is counterbalanced by their higher quality. In particular, we investigate if iterating W_k can be a better alternative than iterating first improvement in some cases. Let us recall that W_k leads toward better solutions than a climber using a first improvement pivoting rule, but it still requires at least twice more evaluations.

Landscape	F	B	W	W_2	W_4	W_8	W_{16}
$UBQP_{128,25}$	23398 1k	23388 7k	**23519** 61k	23510 2k	23511 5k	23512 12k	23517 26k
$UBQP_{128,50}$	32784 1k	32458 8k	32913 110k	32982 2k	33052 6k	**33063** 16k	33006 38k
$UBQP_{128,75}$	45435 1k	45282 7k	45571 138k	**45610** 2k	45600 7k	45527 16k	45524 41k
$UBQP_{128,100}$	50104 1k	49896 8k	50475 176k	50359 2k	50592 7k	50495 17k	**50608** 47k
$UBQP_{256,25}$	69183 2k	69034 31k	69518 510k	69340 6k	69503 16k	**69574** 37k	69505 85k
$UBQP_{256,50}$	102037 2k	101866 34k	102121 885k	102143 7k	102168 17k	102132 43k	**102180** 106k
$UBQP_{256,75}$	132076 3k	131494 35k	**133027** 1254k	132710 7k	132791 19k	132958 48k	132829 119k
$UBQP_{256,100}$	144186 2k	143724 32k	144817 1507k	144597 7k	144673 19k	144671 51k	**144826** 133k
$UBQP_{512,10}$	132133 6k	131881 136k	131954 1614k	**132232** 15k	132202 34k	132119 77k	131902 170k
$UBQP_{512,50}$	284049 8k	283560 140k	**286723** 8144k	285842 21k	286223 48k	286696 122k	286530 311k
$UBQP_{512,75}$	361205 8k	360518 138k	**362802** 11020k	361983 20k	362528 50k	362678 130k	362713 338k
$UBQP_{512,100}$	419889 7k	418458 137k	421017 12797k	420398 20k	421004 53k	**421391** 133k	421141 352k
$UBQP_{1024,10}$	350574 17k	349469 536k	351940 15087k	350826 39k	351678 93k	351928 220k	**352137** 524k
$UBQP_{1024,50}$	797468 22k	795302 558k	**800153** 72777k	798871 53k	799807 140k	799851 354k	799907 901k
$UBQP_{1024,75}$	998493 22k	995156 564k	1001466 95316k	999585 54k	1000837 141k	**1001775** 366k	1001059 949k
$UBQP_{1024,100}$	1130314 22k	1127151 568k	1134202 114228k	1133432 63k	1134057 147k	1134332 384k	**1134953** 1029k
$UBQP_{2048,10}$	991463 49k	988944 2230k	**994765** 134784k	993698 112k	994032 264k	994163 604k	994586 1473k
$UBQP_{2048,25}$	1626645 55k	1623603 2276k	1630519 325672k	1628570 126k	1629699 302k	1629879 734k	**1630585** 1886k
$UBQP_{2048,50}$	2377355 60k	2373279 2295k	**2382892** 584078k	2380144 139k	2381361 345k	2381433 888k	2381383 2235k
$UBQP_{2048,100}$	3058752 71k	3045996 2291k	**3074422** 951650k	3062042 168k	3065725 420k	3068646 1104k	3068596 2905k
$UBQP_{4096,10}$	2775627 147k	2771590 9333k	2788231 1179075k	2781304 312k	2785430 713k	2787136 1671k	**2788788** 4089k
$UBQP_{4096,25}$	4552524 160k	4545854 9364k	**4568967** 27611448k	4558515 350k	4562055 836k	4567451 2098k	4568374 5078k
$UBQP_{4096,50}$	6470526 178k	6457298 9504k	**6495855** 4847505k	6479307 426k	6483570 1025k	6489013 2580k	6489455 6012k
$UBQP_{4096,100}$	9014549 188k	8997240 9569k	**9053431** 7562717k	9027282 437k	9039464 1140k	9046036 2865k	9047765 7336k

Table 2. Average fitness values (rounded to the nearest integers) of local optima obtained from 100 runs for each couple (climber, landscape) on UBQP landscapes. The average number of evaluations is reported below.

Let us notice that we do not consider best improvement in this study since it requires the evaluation of the whole neighborhood at each step of the search and is regularly less efficient than other rules on considered landscapes.

Here, we investigate the efficiency of Iterated Local Searches (ILS) [6] using F and W_k as pivoting rules dedicated to intensification phases. The considered ILS consists of iterating climbing processes until a maximal number of evaluations is reached. The first climbing process starts from a randomly generated solution, whereas the following ones start from a solution obtained by applying P random moves from the last obtained local optimum The next section presents empirical results of ILS that only differs with the hill-climbing pivoting rule on several landscapes. We also analyze their behavior thanks to landscapes features.

4 Experimental analysis

4.1 Experimental protocol

This experimentation is conducted on the set of 52 binary-string based fitness landscapes described in section 2.2, using different fitness functions (NK, UBQP), sizes and ruggedness parameterization. It consists to compare the performances of iterated local searches (which alternate hill-climbing and random perturbations) which only differ by the pivoting rule used during the climbing phases.

To obtain a fair comparison, all ILS runs are performed using different values of parameter P on each landscape: $P \in \{5, 10, 15, 20\}$ on NK landscapes (higher values of P are useless here), and $P \in \{5, 10, 20, 30, 40, 50, 60, 70, 80\}$ on UBQP landscapes. Compared climbers are F, W_2, W_4, W_8, W_{16}. These different values of k allow various tradeoffs between quality of approximations and computation cost of climbers W_k. Recall that F is equivalent to W_1. Using $k = 2$ leads to the least precise approximation of worst improvement but is sufficient to induce a behavior quite different from first improvement F. Using $k = 16$ allow a very precise approximation of worst improvement and very similar results despite a reduced cost.

In the following we note $\mathrm{ILS_F}$ the iterated local search which uses a first improvement climbing rule and $\mathrm{ILS_{W_k}}$ iterated local searches which use W_k. For each triplet (landscape, method, P) 100 runs starting from the same set of 100 randomly generated solutions are performed. The set of methods is composed of $\mathrm{ILS_F}$, $\mathrm{ILS_{W_2}}$, $\mathrm{ILS_{W_4}}$, $\mathrm{ILS_{W_8}}$, $\mathrm{ILS_{W_{16}}}$.Each run stops after 100 million of evaluation and returns the fitness value of the best solution encountered during the search. For each triplet we record the average of the 100 resulting fitness values. In the next section we mainly report the best average fitness values obtained for each couple (landscape, method) as well as the value of P leading to this average.

4.2 Results

On considered NK landscapes (tab. 3), we note that a first improvement based ILS always lead toward better local optima averages than using a worst improvement approximation W_k for climbing processes. The better quality in average of local optima obtained through climbers using W_k is not sufficient to counterbalance the higher number of local optima reached with climbers F, since an ILS process only returns the best local optimum encountered. Note that the number of perturbations required to reach the best averages is always higher on smooth NK landscapes ($K = 1$ or $K = 2$). It could probably come from the fact that on smooth landscapes local optima are less numerous with larger basins of attraction. Succesfully escaping from these local optima requires more random moves.

Landscape	ILS_F	ILS_{W_2}	ILS_{W_4}	ILS_{W_8}	$ILS_{W_{16}}$
$NK_{128,1}$	**.7245**[20]	**.7245**[20]	**.7245**[20]	.7166[20]	.7155[20]
$NK_{128,2}$	**.7423**[15]	.7415[15]	.7403[15]	.7394[15]	.7387[15]
$NK_{128,4}$	**.7958**[5]	.7956[5]	.7946[5]	.7941[5]	.7938[5]
$NK_{128,6}$	**.7995**[5]	.7955[5]	.7916[5]	.7892[5]	.7877[5]
$NK_{128,8}$	**.7949**[5]	.7896[5]	.7842[5]	.7818[5]	.7795[5]
$NK_{128,10}$	**.7847**[5]	.7784[5]	.7742[5]	.7717[5]	.7700[5]
$NK_{128,12}$	**.7724**[5]	.7676[5]	.7631[5]	.7601[5]	.7588[5]
$NK_{256,1}$	**.7200**[20]	.7179[20]	.7148[20]	.7124[20]	.7111[20]
$NK_{256,2}$	**.7425**[5]	.7396[10]	.7373[10]	.7351[10]	.7339[15]
$NK_{256,4}$	**.7917**[5]	.7899[5]	.7883[5]	.7876[5]	.7874[5]
$NK_{256,6}$	**.8007**[5]	.7957[5]	.7922[5]	.7907[5]	.7899[5]
$NK_{256,8}$	**.7892**[5]	.7828[5]	.7785[5]	.7751[5]	.7742[5]
$NK_{256,10}$	**.7782**[5]	.7721[5]	.7666[5]	.7642[5]	.7629[5]
$NK_{256,12}$	**.7663**[5]	.7602[5]	.7556[5]	.7527[5]	.7520[5]
$NK_{512,1}$	**.7040**[20]	.6984[20]	.6926[20]	.6894[20]	.6882[20]
$NK_{512,2}$	**.7453**[5]	.7419[10]	.7393[10]	.7381[10]	.7371[5]
$NK_{512,4}$	**.7806**[5]	.7770[5]	.7750[5]	.7742[5]	.7736[5]
$NK_{512,6}$	**.7940**[5]	.7899[5]	.7872[5]	.7858[5]	.7850[5]
$NK_{512,8}$	**.7886**[5]	.7842[5]	.7801[5]	.7783[5]	.7773[5]
$NK_{512,10}$	**.7781**[5]	.7731[5]	.7676[5]	.7651[5]	.7641[5]
$NK_{512,12}$	**.7671**[5]	.7612[5]	.7552[5]	.7524[5]	.7512[5]
$NK_{1024,1}$	**.7087**[15]	.7012[20]	.6942[20]	.6907[20]	.6898[20]
$NK_{1024,2}$	**.7428**[20]	.7385[20]	.7353[20]	.7332[15]	.7321[15]
$NK_{1024,4}$	**.7797**[5]	.7759[5]	.7736[5]	.7726[5]	.7718[5]
$NK_{1024,6}$	**.7890**[5]	.7857[5]	.7835[5]	.7817[5]	.7813[5]
$NK_{1024,8}$	**.7850**[5]	.7826[5]	.7801[5]	.7787[5]	.7787[5]
$NK_{1024,10}$	**.7753**[10]	.7735[5]	.7710[5]	.7694[5]	.7690[5]
$NK_{1024,12}$	**.7656**[5]	.7640[5]	.7609[5]	.7583[5]	.7582[5]

Table 3. Best averages fitnesses obtained from 100 ILS runs on NK landscapes. Values between brackets correspond to the number of perturbations P driving toward the reported best averages.

On UBQP landscapes (see table 4), we observe that on least large landscapes ($N \leq 1024$), almost all considered methods always reach the same local optimum, which we expect to be the global optimum[2]. Interestingly, 1024-dimensional random NK landscapes are conversely considered as large landscapes since even efficient evolutionary techniques fail to easily detect a global optimum. We then perform an additional analysis in order to obtain the number of evaluations required by each ILS to reach an expected global optimum (on these landscapes derived from *easy* UBQP instances). Results are reported in figure 3 which compare, for each landscapes, the computational budget used by F and $\mathrm{W_2}$ for reaching an expected global optimum: $\mathrm{ILS_F}$ is faster than $\mathrm{ILS_{W_2}}$, but the difference is relatively low.

On all larger UBQP landscapes $N \geq 2048$, $\mathrm{ILS_{W_2}}$ leads in average toward better local optima than $\mathrm{ILS_F}$. Moreover, on 4096-dimensional UBQP landscapes, generating up to 4 and sometimes 8 improving neighbors before selecting the one with the lowest fitness can be more relevant than using first improvement.

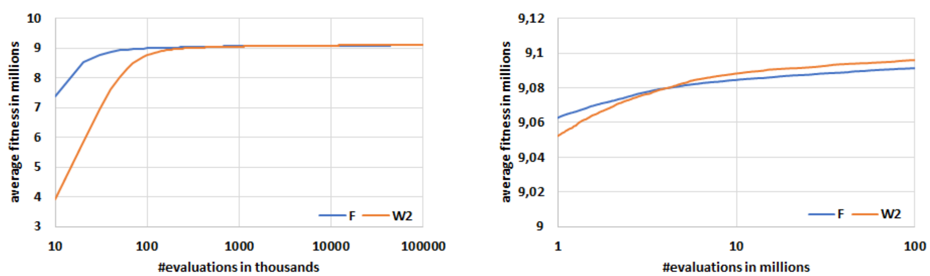

Fig. 2. Evolution of the best average fitness among all considered $\mathrm{ILS_F}$ and all considered $\mathrm{ILS_{W_2}}$ on UBQP landscape where $N = 4096$ and $d = 100$.

Figure 2 depicts the evolution of the best average fitness during runs on an UBQP landscape instance ($N = 4096$, $d = 100$), for ILS_F and ILS_{W_2}. On this landscape, ILS_{W_2} starts to outperform ILS_F around 10 million evaluations. Let us notice that ILS_F and ILS_{W_2} have a similar compared evolution during the search on all considered large landscapes. Let us recall that single climbing processes using $\mathrm{W_2}$ require more than twice evaluations than a climbing process using first (see table 2). In particular, on considered UBQP landscapes with $N \geq 2048$ a single first improvement climber requires on average 49k to 188k evaluations before terminate, whereas a $\mathrm{W_2}$ climber requires from 112k to 437k evaluations.

[2] In the following we use the term *expected global optimum* when a same local optimum is always reached by a set of methods. Of course, constantly obtaining the same final solution (or fitness) does not guarantee its optimality, which could only be proved using complete methods.

.

Landscape	best avg	ILS$_{\mathrm{F}}$	ILS$_{\mathrm{W}_2}$	ILS$_{\mathrm{W}_4}$	ILS$_{\mathrm{W}_8}$	ILS$_{\mathrm{W}_{16}}$
UBQP$_{128,25}$	24087.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{128,50}$	33440.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{128,75}$	46180.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{128,100}$	51130.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{256,25}$	70861.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-30.50]}$	**0.0**$_{[5-20]}$
UBQP$_{256,50}$	102914.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-30]}$
UBQP$_{256,75}$	133641.0	**0.0**$_{[10-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{256,100}$	146377.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{512,10}$	134112.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-50]}$	**0.0**$_{[5-30]}$	**0.0**$_{[5-20]}$
UBQP$_{512,50}$	102914.0	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$
UBQP$_{512,75}$	133641.0	**0.0**$_{[10-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-20]}$
UBQP$_{512,100}$	424728.0	**0.0**$_{[10-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-80]}$	**0.0**$_{[5-60]}$
UBQP$_{1024,10}$	356679.0	**0.0**$_{[60-80]}$	**0.0**$_{[40-50]}$	-0.7$_{[30]}$	-0.7$_{[20]}$	-21.8$_{[20]}$
UBQP$_{1024,50}$	808138.0	**0.0**$_{[50-80]}$	**0.0**$_{[40]}$	**0.0**$_{[20]}$	-0.2$_{[10]}$	-16.6$_{[10]}$
UBQP$_{1024,75}$	1007737.0	**0.0**$_{[30-80]}$	**0.0**$_{[20-40]}$	**0.0**$_{[10]}$	-0.5$_{[5]}$	-8.3$_{[10]}$
UBQP$_{1024,100}$	1145975.0	**0.0**$_{[20-80]}$	**0.0**$_{[20-80]}$	**0.0**$_{[20-80]}$	**0.0**$_{[10-40]}$	**0.0**$_{[10-20]}$
UBQP$_{2048,10}$	1004593.1	-34.5$_{[80]}$	**0.0**$_{[60]}$	-64.2$_{[40]}$	-154.8$_{[30]}$	-390.0$_{[30]}$
UBQP$_{2048,25}$	1641377.7	-75.8$_{[70]}$	**0.0**$_{[50]}$	-117.6$_{[20]}$	-209.5$_{[10]}$	-197.5$_{[10]}$
UBQP$_{2048,50}$	2398651.0	-49.4$_{[80]}$	**0.0**$_{[50]}$	-51.7$_{[30]}$	-145.5$_{[20]}$	-597.8$_{[10]}$
UBQP$_{2048,100}$	3100346.3	-832.8$_{[80]}$	**0.0**$_{[80]}$	-382.5$_{[40]}$	-813.3$_{[30]}$	-1499.8$_{[30]}$
UBQP$_{4096,10}$	2809642.0	-953.7$_{[80]}$	**0.0**$_{[80]}$	-277.5 $_{[50]}$	-818.1$_{[30]}$	-1033.7$_{[30]}$
UBQP$_{4096,25}$	4597565.2	-1536.5$_{[80]}$	**0.0**$_{[80]}$	-155.1$_{[60]}$	-731.0$_{[30]}$	-1640.4$_{[20]}$
UBQP$_{4096,50}$	6530759.4	-2914.8$_{[80]}$	**0.0**$_{[80]}$	-216.3$_{[70]}$	-1634.4$_{[40]}$	-3725.8$_{[30]}$
UBQP$_{4096,100}$	9095931.9	-4640.3$_{[80]}$	**0.0**$_{[80]}$	-49.3$_{[60]}$	-1940.0$_{[30]}$	-5023.4$_{[30]}$

Table 4. Comparative ILS results on UBQP landscapes (fitness gaps from best average).

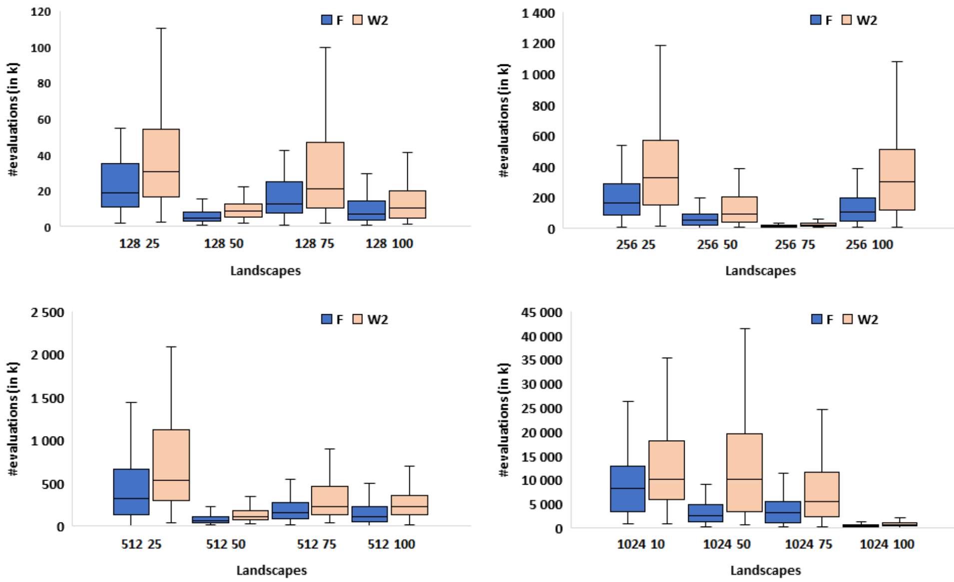

Fig. 3. Number of evaluations required to reach an expected global optimum from 100 runs of $\mathrm{ILS_F}$ and $\mathrm{ILS_{W_2}}$ on UBQP landscapes.

Perturbations used by these iterated local searches to escape from local optima (strategy: P random moves from the last local optimum reached) generally lead toward solutions slightly better than while restarting the search from a new random solution, especially since climbings following perturbations are faster than the first one. For instance, with the computational budget of the initial climbing process of ILS_{W_2}, ILS_F can proceed to many hill-climbings (as F is quicker than W_2, and following ILS_F climbings are also quicker than F). Thus the fact that it requires almost 10 million of evaluations for ILS_{W_2} to outperform ILS_F makes sense.

4.3 ILS performances and landscape features

Experiments show a difference of overall efficiency between $\mathrm{ILS_{W_k}}$ and $\mathrm{ILS_F}$ on NK and UBQP landscapes. This subsection is devoted to obtain insights on when using W_k within ILS leads toward better local optima. To this aim we measure for all considered landscapes the k-ruggedness evolution as well as the repartition of local optima.

k-ruggedness values are computed from sets of 100,000 pairs of solutions which are distant by k bits. In tables 5 and 6 , we report percentages corresponding to the ratios between the numbers of distinct 1-flips required to reach a given k-ruggedness value, and the maximal value $N-1$. We also estimate the

1-ruggedness of landscapes (ie. the proportion of 1-sign-epistasis) since local sign epistasis is an obstacle for climbers and directly affects their performance.

Results show that on NK landscapes (table 5), the k-ruggedness grows faster on more rugged landscapes (when K increases) and is stable when N increases. For a given value of N, the 1-ruggedness is higher on more rugged landscapes and for a given value of K, the 1-ruggedness decreases on larger landscapes. This last observation is coherent since a single mutation has more impact on smaller NK landscapes than on larger ones. In random NK landscapes, the ruggedness is directly correlated with the parameters ratio K/N. So we use these observations as references to understand the structure of UBQP landscapes.

Landscape	k-ruggedness		1-rug.	Landscape	k-ruggedness		1-rug.
	≥ 0.1	≥ 0.25			≥ 0.1	≥ 0.25	
$NK_{128,1}$	15.7%	43.3%	0.5%	$NK_{256,1}$	16.1%	43.9%	0.3%
$NK_{128,2}$	7.9%	22.9%	1.2%	$NK_{256,2}$	7.5%	22.4%	0.6%
$NK_{128,4}$	3.1%	10.2%	3.1%	$NK_{256,4}$	3.1%	10.2%	1.5%
$NK_{128,6}$	2.4%	7.1%	5.2%	$NK_{256,6}$	1.9%	6.7%	2.6%
$NK_{128,8}$	1.6%	4.7%	7.4%	$NK_{256,8}$	1.2%	4.7%	4.1%
$NK_{128,10}$	1.6%	3.9%	9.6%	$NK_{256,10}$	1.2%	3.9%	5.3%
$NK_{128,12}$	0.8%	3.1%	11.8%	$NK_{256,12}$	0.8%	3.1%	6.5%
$NK_{512,1}$	16.7%	45.0%	0.1%	$NK_{1024,1}$	15.7%	42.7%	0.1%
$NK_{512,2}$	7.0%	22.3%	0.3%	$NK_{1024,2}$	7.6%	23.6%	0.2%
$NK_{512,4}$	2.9%	10.4%	0.8%	$NK_{1024,4}$	2.9%	10.1%	0.4%
$NK_{512,6}$	1.8%	6.4%	1.4%	$NK_{1024,6}$	1.8%	6.5%	0.7%
$NK_{512,8}$	1.2%	4.7%	2.1%	$NK_{1024,8}$	1.2%	4.8%	1.1%
$NK_{512,10}$	0.9%	3.7%	2.7%	$NK_{1024,10}$	0.9%	3.7%	1.4%
$NK_{512,12}$	0.7%	3.1%	3.6%	$NK_{1024,12}$	0.7%	3.1%	1.8%

Table 5. Ruggedness information of NK landscapes.

1-ruggedness values of UBQP landscapes (see table 6) seem to increase according to their density parameter. This value is the only one, among considered values, differing on UBQP landscapes and corresponds to the 1-ruggedness of non-smooth NK landscapes of same size. Moreover, on larger landscapes, the 1-ruggedness decreases less than on NK landscapes of same size. The k-ruggedness evolution is similar for all UBQP landscapes, but does not correspond to the k-ruggedness evolution of any NK landscape . A k-ruggedness value superior to 0.1 is quickly reached (with few mutations), such as on medium-rugged NK landscapes. Yet, a k-ruggedness value superior to 0.25 is lately reached, such as on smooth NK landscapes.

It means that the sign epistasis repartition, which directly affects the ruggedness, is different than the one of NK landscapes: the evolution of 1-ruggedness shows that the local sign epistasis is high on UBQP landscapes whereas the k-ruggedness global evolution leads to think that they globally have a reduced sign epistasis like smooth landscapes. Consequently, we assume that considered UBQP landscapes are locally rugged but globally smooth. Indeed, the 1-ruggedness values, which refer to the sign epistasis between neighboring solutions, reveal locally rugged landscapes. On the contrary, the k-epistasis with high values of k, which refers to the sign epistasis between distant solutions, reveals globally smooth landscapes.

Landscape	k-ruggedness		1-rug.	Landscape	k-ruggedness		1-rug.
	≥ 0.1	≥ 0.25			≥ 0.1	≥ 0.25	
$UBQP_{128,25}$	5.5%	27.6%	2.3%	$UBQP_{256,25}$	5.1%	28.6%	1.5%
$UBQP_{128,50}$	4.7%	27.6%	3.1%	$UBQP_{256,50}$	5.5%	29.8%	2.1%
$UBQP_{128,75}$	5.5%	31.5%	3.6%	$UBQP_{256,75}$	5.5%	31.8%	2.7%
$UBQP_{128,100}$	5.5%	30.7%	4.3%	$UBQP_{256,100}$	5.1%	29.0%	2.8%
$UBQP_{512,10}$	5.9%	32.5%	0.6%	$UBQP_{1024,10}$	5.2%	30.5%	0.5%
$UBQP_{512,50}$	5.1%	32.3%	1.5%	$UBQP_{1024,25}$	4.9%	28.0%	1.0%
$UBQP_{512,75}$	4.9%	29.3%	1.8%	$UBQP_{1024,50}$	5.0%	28.6%	1.1%
$UBQP_{512,100}$	5.5%	32.5%	2.3%	$UBQP_{1024,100}$	5.2%	28.0%	1.5%
$UBQP_{2048,10}$	5.2 %	29.5%	0.3%	$UBQP_{4096,10}$	4.7%	28.8%	0.2%
$UBQP_{2048,25}$	5.0%	27.6%	0.7%	$UBQP_{4096,25}$	5.0%	28.9%	0.5%
$UBQP_{2048,50}$	5.0%	29.2%	0.9%	$UBQP_{4096,50}$	4.7%	28.5%	0.3%
$UBQP_{2048,100}$	4.8%	27.7%	1.0%	$UBQP_{4096,100}$	5.7%	30.0%	0.7%

Table 6. Ruggedness information of UBQP landscapes.

In addition we perform a sampling of local optima in considered landscapes. Local optima are collected from 1000 hill-climbing algorithms using a first improvement hill-climbing. In tables 7 and 8 we report the average hamming distance between all distinct local optima found on each landscapes as well as the number their numbers. Note that median distances are similar to average distances and are not reported. Results on NK landscapes (table 7) show that on sufficiently rugged landscapes, the average hamming distance is close to $N/2$. It means that local optima are uniformly distributed in the landscape, since $N/2$ corresponds to the average hamming distance between two random solutions. On smooth NK landscapes the average hamming distance between considered local optima is lower, meaning that local optima are packed in a relatively reduced area of the landscape. On UBQP landscapes (table 8), the average distances of local optima are largely lower than $N/2$ and even often lower than distances observed on very smooth NK landscapes. Moreover, the number of distinct local optima found on smaller considered landscapes are way lower than on same size NK landscapes and could explain why associated problem instances are easier to solve. The observed number and repartition of local optima on UBQP landscapes tend to confirm the assumed structure of such landscapes, that is locally rugged but globally very smooth.

The fact that ILS_{W_k} is more efficient on UBQP landscapes than on NK landscapes can be explained by the structure of these landscapes. UBQP landscapes are globally smooth, which at first glance is not an advantage for climbing

Landscape	d_{avg}	#LO	Landscape	d_{avg}	#LO
$NK_{128,1}$	24.48	991	$NK_{256,1}$	50.22	1000
$NK_{128,2}$	44.94	1000	$NK_{256,2}$	93.73	1000
$NK_{128,4}$	60.99	1000	$NK_{256,4}$	121.43	1000
$NK_{128,6}$	63.55	1000	$NK_{256,6}$	126.66	1000
$NK_{128,8}$	63.86	1000	$NK_{256,8}$	127.69	1000
$NK_{128,10}$	63.97	1000	$NK_{256,10}$	127.92	1000
$NK_{128,12}$	63.97	1000	$NK_{256,12}$	127.99	1000
$NK_{512,1}$	101.74	1000	$NK_{1024,1}$	200.02	1000
$NK_{512,2}$	187.37	1000	$NK_{1024,2}$	366.87	1000
$NK_{512,4}$	243.29	1000	$NK_{1024,4}$	486.74	1000
$NK_{512,6}$	253.28	1000	$NK_{1024,6}$	507.17	1000
$NK_{512,8}$	255.40	1000	$NK_{1024,8}$	510.56	1000
$NK_{512,10}$	255.85	1000	$NK_{1024,10}$	511.75	1000
$NK_{512,12}$	255.96	1000	$NK_{1024,12}$	511.95	1000

Table 7. Average distance (d_{avg}) of distinct local optima on NK landscapes (obtained from 1000 first improvement hill-climbings). #LO denotes the number of distinct local optima.

strategies based on worst improvement, yet such landscapes are locally rugged. Pivoting rules based on worst improvement help to bypass low quality local optima and avoid to be trapped by them. Such strategies seem to lead toward higher local optima on UBQP landscapes than on NK landscapes compared to a basic first improvement.

On NK landscapes, the quality of local optima obtained during ILS_{W_k} runs is not sufficiently improved compare to those obtained during an ILS_F, which is able to perform more climbing processes with the same number of evaluations. On UBQP landscapes, ILS_{W_k} achieves a better balance between local optima quality and number of evaluations to reach them than ILS_F.

5 Conclusion

Most climbing processes of metaheuristics are based on first or best improvement pivoting rules. As stated by a previous study [11], worst improvement is able to reach good local optima despite its counter-intuitive nature. In this work, we integrated worst improvement in an Iterative local search context in order to obtain insights on the potential interest of using such a pivoting rule within advanced metaheuristics. We performed experiments on binary-string problems (NK landscapes and UBQP) and combined results observations with landscape analysis. In these experiments we confronted worst improvement based ILS with first improvement ILS. In order to obtain tradeoffs between local optima quality and number of evaluations, proposed ILS use different levels of approximation of worst improvement. Experiments show that on NK landscapes, first improvement remains the most adequate pivoting rule in an ILS context, thanks to its small

Landscape	d_{avg}	#LO	Landscape	d_{avg}	#LO
$UBQP_{128,25}$	26.26	815	$UBQP_{256,25}$	52.18	990
$UBQP_{128,50}$	24.20	378	$UBQP_{256,50}$	26.45	640
$UBQP_{128,75}$	21.5	488	$UBQP_{256,75}$	30.84	566
$UBQP_{128,100}$	23.58	468	$UBQP_{256,100}$	41.98	908
$UBQP_{512,10}$	59.40	999	$UBQP_{1024,10}$	156.11	1000
$UBQP_{512,50}$	63.52	976	$UBQP_{1024,50}$	157.62	1000
$UBQP_{512,75}$	59.27	980	$UBQP_{1024,75}$	130.15	1000
$UBQP_{512,100}$	59.94	969	$UBQP_{1024,100}$	133.19	1000
$UBQP_{2048,10}$	269.67	1000	$UBQP_{4096,10}$	536.15	1000
$UBQP_{2048,25}$	230.67	1000	$UBQP_{4096,25}$	512.03	1000
$UBQP_{2048,50}$	238.1	1000	$UBQP_{4096,50}$	483.50	1000
$UBQP_{2048,100}$	307.22	1000	$UBQP_{4096,100}$	486.59	1000

Table 8. Average distance of distinct local optima on UBQP landscapes.

number of evaluations that allow to reach more local optima. However, on large UBQP landscapes, worst improvement approximations are able to outperform first improvement in an ILS context, despite its heavier computation cost of each climbing process.

We observed the ruggedness and repartition of local optima within all considered landscapes. We deduced that the structure of UBQP landscapes seem to be globally smooth but locally rugged and that local optima are more packed within the search space. These facts induce an advantage for the k worst improvement which has a tendency to avoid to be prematurely trapped in low quality local optima. On such landscapes this advantage has more impact than the small number of evaluations induced by first improvement climbers and allow the ILS versions to be efficient.

It should be interesting to extend this type of work to other problems and/or other metaheuristics in order to determine if alternative pivoting rules could be successfully used in other contexts.

References

1. Matthieu Basseur and Adrien Goëffon. Hill-climbing strategies on various landscapes: an empirical comparison. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 479–486. ACM, 2013.
2. Matthieu Basseur and Adrien Goëffon. Climbing combinatorial fitness landscapes. *Applied Soft Computing*, 30:688–704, 2015.
3. Matthieu Basseur, Adrien Goëffon, Frédéric Lardeux, Frédéric Saubion, and Vincent Vigneron. On the attainability of NK landscapes global optima. In *Seventh Annual Symposium on Combinatorial Search*, 2014.
4. Michael R Gary and David S Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.
5. Stuart A. Kauffman and Edward D. Weinberger. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211 – 245, 1989.

6. Helena R Lourenço, Olivier C Martin, and Thomas Stutzle. Iterated local search. *International series in operations research and management science*, pages 321–354, 2003.
7. Katherine M Malan and Andries P Engelbrecht. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences*, 241:148–163, 2013.
8. Gabriela Ochoa, Marco Tomassini, Sébastien Vérel, and Christian Darabos. A study of NK landscapes' basins and local optima networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 555–562. ACM, 2008.
9. Gabriela Ochoa, Sébastien Verel, and Marco Tomassini. First-improvement vs. best-improvement local optima networks of NK landscapes. *Parallel Problem Solving from Nature, PPSN XI*, pages 104–113, 2010.
10. Kenneth Sörensen. Metaheuristics — the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.
11. Darrell Whitley, Adele E Howe, and Doug Hains. Greedy or not? best improving versus first improving stochastic local search for maxsat. In *AAAI*, 2013.
12. Sewall Wright. *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*, volume 1. 1932.