Hill-climbing Strategies on Various Landscapes: An Empirical Comparison

Matthieu Basseur
LERIA, Université d'Angers, France
matthieu.basseur@univ-angers.fr

Adrien Goëffon
LERIA, Université d'Angers, France
adrien.goeffon@univ-angers.fr

ABSTRACT

Climbers constitute a central component of modern heuristics, including metaheuristics, hybrid metaheuristics and hyperheuristics. Several important questions arise while designing a climber, and choices are often arbitrary, intuitive or experimentally decided. The paper provides guidelines to design climbers considering a landscape shape under study. In particular, we aim at competing best improvement and first improvement strategies, as well as evaluating the behavior of different neutral move policies. Some conclusions are assessed by an empirical analysis on a large variety of landscapes. This leads us to use the NK-landscapes family, which allows to define landscapes of different size, rugosity and neutrality levels. Experiments show the ability of first improvement to explore rugged landscapes, as well as the interest of accepting neutral moves at each step of the search. Moreover, we point out that reducing the precision of a fitness function could help to optimize problems.

Keywords

Local search, hill-climbing, fitness landscapes, NK-landscapes, neutrality

1. INTRODUCTION

Neighborhood search techniques are commonly-used algorithms for optimizing hard combinatorial problems. Among them, basic iterative improvement methods like climbers are generally used as components of more sophisticated local search techniques or modern metaheuristics (tabu search, memetic algorithms and so on). A climber consists in reaching a local optimum by iteratively improving a single solution with local modifications. Although most of metaheuristics use climbers or variants as intensification mechanisms, they mainly focus on determining how to escape local optima. Nevertheless, several important questions have to be considered while designing a climber. Usually, the design of a local search mainly focus on defining a neighborhood structure and choosing a solution which initiate the search. However there are several questions which are regularly considered during the conception process, but not really empirically or theoretically investigated. Among them, one can identify two main issues. First, the *pivoting rule* choice. To the best of our knowledge, there is no real consensus on the benefit of using a best-improvement strategy rather than a first-improvement one, or *vice versa*. Secondly, the *neutral moves policy*. The use of neutral moves during the climbing should be experimentally analyzed. In particular, it is contradictory that traditional climbers only allow strictly improving moves, while a derived search strategy, e.g. simulated annealing, systematically accept neutral moves.

Most of local search and evolutionary computation contributions are focusing on the design and evaluation of advanced and original search mechanisms. However, those aforementioned elementary components are rarely discussed in the experimental analysis.

Since the efficiency of advanced search methods is usually dependent to the problem under consideration, it should be interesting to determine if their elementary components are themselves dependent to the considered search space topology. In this study, we aim to evaluate the behavior and efficiency of basic search methods on search spaces of different size, rugosity and neutrality. To this end, we will use NK-landscapes [5] model to simulate problems with different structures and sizes. Since the basic NK model does not induce landscapes with neutrality, we will especially focus on three appropriated variants: NKp [1], NKq [7] and NKr (presented in the paper).

Several studies on local search behaviors has been proposed for specific problems, such as SAT [2] or TSP [3]. More recently, Ochoa *et al.* [8] investigated the behavior of first and best improvement algorithms on NK-landscapes, by exhaustive exploration of small search spaces. In this paper, we extend this last study by evaluating both pivoting rules and neutral moves effects on large size problems, which are those considered while using metaheuristics. To achieve this, we will evaluate empirically the relative efficiency of climber variants. The aim of this study is to compare basic ways to navigate through a search space, rather than to propose an efficient sophisticated algorithm to solve NK instances.

The next section focus on climbers design: after recalling some general definitions, we will discuss on climbers key components. Section 3 first describes the NK-landscapes model and different ways to add neutrality with NKp, NKq and NKr, before providing an empirical analysis of the neutrality degree of such problems. In section 4, climber variants behavior is estimated on a large scale of landscape

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.

shapes. Finally, the last section summarizes the main contributions and provides hints for future investigations.

2. DESIGNING CLIMBERS

2.1 Fitness landscapes, climber algorithms: general background

Even if we assume that the reader is familiar with the main notions of combinatorial optimization and local search algorithms, let us recall briefly the main elements used in the rest of the paper. For more detailed definitions and comprehensive review of local search principles, we refer the reader to [4].

A *combinatorial optimization problem instance* can be defined as a pair (X, f), where X is a discrete set of *feasible solutions*, and $f : X \to \mathbb{R}$ a scalar *objective function* which has to be maximized or minimized. In the remaining, we will consider the maximization context, without loss of generality. Solving a problem (X, f) consists then in finding $x^\star \in \text{argmax}_{x \in X}\, f(x)$.

A *fitness landscape* is a triplet $(\mathcal{X}, \mathcal{N}, \mathfrak{f})$, where $\mathcal{X}$ is a set of *configurations* (or *candidate solutions*) called *search space*, $\mathcal{N} : \mathcal{X} \to 2^{\mathcal{X}}$ is a *neighborhood relation*, and $\mathfrak{f}$ is a *fitness function* (or *evaluation function*). $\mathfrak{f}(x)$ is the *fitness* (or *height*) of x. $x' \in \mathcal{N}(x)$ is a *neighbor* of a configuration $x \in \mathcal{X}$. $\mathcal{N}(x)$ is called the *neighborhood* of x. A *neutral neighbor* of a configuration x is a configuration x' such that $x' \in \mathcal{N}(x)$ and $\mathfrak{f}(x') = \mathfrak{f}(x)$. A *plateau* (or *neutral network*) is a graph (V, E) where $V \in \mathcal{X}$, $E = \{(v_i, v_j) \text{ s.t. } v_j \in \mathcal{N}(v_i) \text{ and } \mathfrak{f}(x_i) = \mathfrak{f}(x_j), \forall x_i, x_j \in V\}$, *i.e.* a set of connected configurations of same fitness. A (non-strict) *local optimum* is a configuration x such that $\forall x' \in \mathcal{N}(x), \mathfrak{f}(x') \leqslant \mathfrak{f}(x)$. A *global optimum* is a configuration $x^\star \in \text{argmax}_{x \in \mathcal{X}}\, \mathfrak{f}(x)$.

A *local search algorithm* (or *neighborhood search algorithm*) aims at finding the best configuration of $\mathcal{X}$ (thanks to $\mathfrak{f}$) while exploring a subset of $\mathcal{X}$ relatively to $(\mathcal{X}, \mathcal{N}, \mathfrak{f})$. Since in many cases, $\mathcal{X} \equiv X$ and $\mathfrak{f} \equiv f$, solving an instance (X, f) with a local search algorithm consists then in defining a neighborhood relation $\mathcal{N}$ and a move strategy for exploring efficiently the landscape $(X, \mathcal{N}, f)$.

A *hill-climbing* algorithm (or *climber*) is a basic local search strategy which navigates through the search space in allowing only non-deteriorating moves. Given an initial configuration called *starting point*, a traditional climber iteratively moves to better neighbors, until it reaches a local optimum. Such a search mechanism, also known as *iterative improvement*, allows to distinguish several variants which are discussed hereafter.

2.2 Climber components

The design of a climber implies several choices, whose effects are not clearly established. Let us point out four conceptual issues that need to be discussed.

Pivoting rule

The *best improvement* strategy (or *greedy hill-climbing*) consists in selecting, at each iteration, a neighbor which achieves the best fitness. This implies to generate the whole neighborhood at each step of the search, unless an incremental evaluation of all neighbors can be performed. On the contrary, the *first improvement* strategy accepts the first evaluated neighbor which satisfies the moving condition. This avoids the systematic generation of the entire neighborhood and allows more conceptual options.

Neutral move policy

A *basic hill-climbing* algorithm does not allow neutral moves (*i.e.* moves to neutral neighbors) during the search, and only performs improving moves until reaching a local optimum. Question of neutral moves can be considered to escape local optima (*neutral perturbation*, NP) when the fitness landscape contains a substantial proportion of neutral transitions (on smooth landscapes). Another variant, called *stochastic hill-climbing*, can accept indifferently neutral or improving neighbors throughout the search, even before reaching a local optimum. It is not very obvious to determine the influence of the neutral move policy on the quality of the configurations reached. However, it is interesting to note that the more advanced simulated annealing algorithm, which allows some deteriorating moves during the search, systematically accepts neutral moves under consideration.

Neighborhood evaluation

The first condition to assert systematically that a configuration is a local optimum is to use a basic climber which favors improving moves. The second one is to be able to detect when the whole neighborhood has been evaluated. This only can be possible while considering an *exhaustive* neighborhood evaluation, either in evaluating the entire neighborhood incrementally, or in generating neighbors without replacement. Nevertheless, several factors can make difficult or impossible the exhaustive neighborhood evaluation (complex representation of configurations, specific neighborhood operator, very large neighborhood). In such a case, neighbors are usually generated at *random* and with replacement.

Neighborhood exploration

Technically and following the definitions presented in [4], the first improvement strategy consists in exploring the neighborhood with the same *deterministic* rule order during the entire search. The first neighbor which satisfies the move policy is then selected. An alternative way is to explore at each step the neighborhood in a *random* order. This exploration method reduces cycling risks which can occur when combining stochastic hill-climbing with a deterministic exploration of the neighborhood. While combined with a basic move policy, it should be interesting to see if there exists any efficiency difference between both exploration methods.

Figure 1 summarizes the different ways to design a climber according to the issues discussed hereabove. Clearly, specific representation of configurations, neighborhood structures or evaluation functions, lead to favor one or several choices. However, in many cases, every option can be considered. In future sections, we will evaluate the influence of such climber parametrization in function of the landscape structure.

3. NK-LANDSCAPES AND NEUTRALITY

3.1 NK-landscapes

The NK family of landscapes [5] is a problem-independent model for constructing multimodal landscapes. NK-landscapes use a basic search space, with binary strings as configurations and bit-flip as neighborhood (two configurations are neighbors iff their Hamming distance is 1). Characteristics of an NK-landscape are determined by two parameters

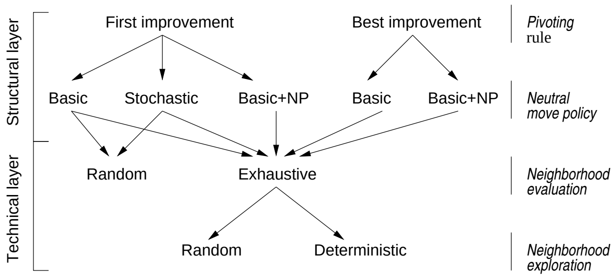

Figure 1: Climber strategies.

N and K. N refers to the size of binary string configurations, which defines the search space size ($|\mathcal{X}| = 2^N$). K specifies the rugosity level of the landscape; indeed, the fitness value of a configuration is given by the sum of N terms, each one depending on $K+1$ bits of the configuration. Thus, by increasing the value of K from 0 to $N-1$, NK-landscapes can be tuned from smooth to rugged. In particular, if $K=0$, then the landscape contains only one local (global) optimum; on the contrary, setting K to $N-1$ leads to a random fitness assignment.

In NK-landscapes, the fitness function $f : \{0,1\}^N \to [0,1)$ to be maximized is defined as follows.

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} c_i(x_i, x_{i_1}, \ldots, x_{i_K}) \qquad (1)$$

where $c_i : \{0,1\}^{K+1} \to [0,1)$ defines the component function associated with each variable x_i, $i \in \{1,\ldots,N\}$, and where $K < N$.

NK-landscapes instances are both determined by the $(K+1)$-uples $(x_i, x_{i_1}, \ldots, x_{i_K})$ and the $2^N.(K+1)$ c_i result values corresponding to a fitness contribution matrix C whose values are randomly generated in $[0,1)$. The usual precision of random values imply that plateaus are almost absent on NK-landscapes. Nevertheless, there are several ways to add neutrality to NK-landscapes. In 1998, two distinct models of neutrality were proposed simultaneously by Barnett (NKp [1]), and Newman *et al.* (NKq [7]), which are described hereafter. Moreover, we add a third way to obtain neutral landscapes by rounding fitnesses of traditional NK-landscapes.

3.2 NKp-landscapes

Probabilistic NK-landscapes [1] are particular NK instances in which the fitness contribution matrix contains many zeros. While generating an NKp instance, values of C are set to 0 with a probability of p, the others being randomly generated in $[0,1)$. An NKp-landscape class is then determined by three parameters: N, K and p. In extreme cases, classic NK-landscapes are obtained for $p=0$, while landscapes are entirely flat when $p=1$.

3.3 NKq-landscapes

To add neutrality to NK-landscapes, Newman *et al.* introduced quantised NK-landscapes [7], by fluctuating the discretization level of c_i result values. Indeed, limiting their possible values increase the number of neutral neighbors. Thus, NKq implies a third parameter $q \geqslant 2$ which specifies the c_i. functions codomain size. The maximal degree of neutrality is reached when $q=2$ (C is then a binary matrix), and decreases while q increases.

3.4 NKr-landscapes

Generating a high neutrality landscape using NKp imply a high value of p (near to 1 – see section 3.5). However, the resulting optimization problem intuitively tends to be simplified since it mainly consists in finding the non-null values of C. According the following experiments, the NKq model fails to generate high neutrality landscapes. We propose here to artificially create neutrality by rounding the fitness function of a classic NK model. Then, in this *rounded* NK (NKr), the original NK-landscape is divided in a predefined number of levels.

Let us notice that, contrary to NKp and NKq, NKr is not a generalization of the NK model where the fitness contribution matrix is depending of a neutrality parameter. Here, r affects the final fitness values without modifying the contribution matrix generation policy:

$$f(x,r) = \frac{\lfloor r.f(x) \rfloor}{r} \qquad (2)$$

A significant consequence is that an NKr landscape is a smoothed version of a corresponding NK-landscape. In particular, $f(x_1) < f(x_2) \Rightarrow f(x_1, r) \leqslant f(x_2, r)$, which implies that every global optima of an NK instance is also a global optima of the corresponding NKr instance. NKr will allow us to determine if smoothing the landscape could help to solve optimization problems.

3.5 Landscapes neutrality

To evaluate the neutrality level of NK{pqr}-landscapes, we propose to determine empirically the proportion of neutral neighbors according to the landscapes characteristics (N, K, p, q, r).

Landscapes panel

We first focus of four sizes $N = \{128, 256, 512, 1024\}$ and four rugosity levels $K = \{1,2,4,8\}$ which correspond to commonly-used NK-landscapes parametrizations for testing metaheuristics. For each couple (N,K), we have randomly generated 10 NK instances, *i.e.* 10 landscapes. Secondly, we used these 10 instances to generate 10 corresponding NKp, NKq and NKr instances, for 4 p, 4 q and 4 r parametrizations. In NKp, each value of the original fitness contribution matrices are replaced by 0 with a probability of p. NKq instances are obtained by discretizing fitness contribution values among q using a rounding function $\rho(c) = \frac{\lfloor qc \rfloor}{q-1}$. In NKr, original fitness contribution matrices remained unchanged, while the fitness function is rounded as described in equation 2. In this section, we will use 1920 distinct landscapes (192 parametrizations, 10 instances per parametrization) which allow us to determine the parameters sensitivity for obtaining neutral landscape.

Neutrality estimation

The degree of neutrality of a landscape is evaluated by measuring the proportion of neutral neighbors within a sample of configurations. More precisely, for each parametrization, 10 instances (landscapes) and 10,000 random configurations per instance are considered. In table 1, we first evaluate the whole neighborhood on each configuration and its associated rate of neutral neighbors (*random* column). Secondly, the LO column corresponds to the average rates of neutral neighbors on the local optima obtained after applying a basic first-improvement climber from the 10,000 random configurations.

Empirical analysis

Table 1 provides the average neutrality rate of the 192 landscape classes considered, while figure 2 outputs the distribution of neutrality rates. This leads us to several comments:

- Only NKp and NKr can produce landscapes with high neutrality. Indeed, even with its most neutral parametrization ($q = 2$), a random NKq landscape is unlikely to provide a neutral neighbor rate over 0.5. In particular, when K is higher, there are very few neutral neighbors, especially around local optima.

- NKp allow to generate high neutrality landscapes, provided that p is close to 1. Indeed, when p is high, c_i values are mainly zeros, which implies that bit-flip neighbors fitnesses often remains unchanged, especially if K is low. Unfortunately, such a landscape is easier to climb since the fitness is only affected by a few non-null values. Moreover, even a high level of neutrality does not imply the existence of totally flat areas in the search space unless $p = 1$ (in figure 2, even with $p = 0.99$, there is almost no configuration for which all neighbors are neutral).

- NKr allow to generate landscapes with a high level of neutrality as well as totally flat areas. Indeed, as shown in figure 2, with a low enough value of r, a significant part of the search space is composed by configurations with only neutral neighbors.

- Unsurprisingly, neutrality of NKr depends on N, while NKp and NKq-landscapes characteristics are similar when N is fluctuating. Indeed, increasing N proportionnaly reduces the average fitness variation between two neighbors (that is implied by equation 1).

4. COMPARING CLIMBERS

In this section, we aim at evaluating the ability of climbers to explore various landscapes. Thus, different climbing strategies introduced in section 2 will be applied on landscapes defined in section 3.

In figure 1, we divided the components in two layers. The *structural layer* identifies the fundamental choices which will mainly determine the search strategy, while the *technical layer* is mostly dependent of the problem itself. In particular, an exhaustive neighborhood evaluation is clearly preferable to a random one, provided its implementation does not imply any additional complexity cost. Moreover, a random neighborhood exploration order seems more natural to provide a stochastic search, even if usual local search definitions follow a deterministic order.

In the following, we will focus on the ability of a climber to reach good configurations depending on structural options (pivoting rule, neutral move policy). Different technical choices have been tested, but are not reported here since experiments have not shown any significant efficiency difference between all variants.

Then, the experimental analysis will compare five climbers variants combining pivoting rule (PR) alternatives and the neutral move policy (NMP). All climbers start from a random configuration, and stop after $10.N^2$ configuration evaluations (unless for basic move policy which stops when a local optimum is reached). Let us notice that this maximal number of evaluations has been set to allow a convergence of the search, after observing no significant improvements for longer searches.

Each climber will be executed 10,000 times on a benchmark set of 208 instances: 16 basic NK-landscapes parametrizations, as well as 192 instances which corresponds to the 192 NK{pqr} landscapes parametrizations selected in the previous section. For each instance, the five climbers start their searches from a single set of 10,000 starting points, in order to cancel the initialization bias.

Empirical analysis

Experiment results are given on tables 2 and 3 which focus respectively on the NK, NKp, NKq and NKr instances. For each couple climber/instance, we report the average fitness of the 10,000 resulting configurations. For each instance, the best average value appears in bold. Moreover, we indicate in grey methods which are not statistically outperformed by any other method (w.r.t. the Mann-Whitney test with a 5% significance level).

Results obtained on the basic NK-landscapes are given in table 2. In this table, results include only two variants which correspond to the pivoting rule alternatives. Indeed, the basic NK-landscapes do not contain a significant number of neutral neighbors. Then, experiments show equivalent results whatever the neutral move policy being adopted. Anyway, this table provides us a significant piece of information while comparing the best improvement and the first improvement pivoting rules. Best improvement statistically outperforms first improvement when $K \in \{1, 2\}$, and first improvement appears more efficient while K increases. In other words, best improvement is well-suited to explore smooth landscapes, whereas first improvement seems more adapted to explore a rugged one.

Table 3.a summarizes the results obtained on the NKp landscapes. As expected, better results are achieved by climbers which allow neutral move, either all along the search (stochastic), or after detecting a local optimum (basic+NP). The few cases where a basic climber is not statistically outperformed by other methods corresponds to landscapes with a very small neutrality level (less than 6% of neutral neighbors, see table 1). When $p = 0.99$, basic climbers are clearly inefficient, while the three other variants almost systematically obtain configurations of equal fitness; this can be explained by the relative simplicity of these instances.

NKq instances experiments lead to more relevant outcomes. One can see in table 3.b that neutral moves are necessary to climb landscapes containing even a small level of neutrality. Indeed, basic climbers are always statistically outperformed by others. Moreover, this table emphasizes significant differences between the three strategies allowing neutral moves. First, stochastic climbers reach best results on most instances, especially on more rugged and/or neutral landscapes (high K, low q). This is particularly interesting since, to our knowledge, basic policies – with or without neutral perturbations – are more traditionally used while designing metaheuristics. However, a best improvement strategy combined with neutral perturbations remains suitable in smooth landscapes, especially with lowest levels

NKp

Landscape			Neutral neighbors	
N	K	p	Random	LO
128	1	.50	11.23%	6.80%
		.80	44.55%	35.10%
		.95	81.75%	75.22%
		.99	94.92%	92.64%
	2	.50	5.57%	2.42%
		.80	30.85%	16.84%
		.95	73.62%	56.84%
		.99	94.40%	89.23%
	4	.50	1.17%	0.29%
		.80	14.52%	3.44%
		.95	60.50%	29.13%
		.99	89.66%	73.45%
	8	.50	0.05%	0.00%
		.80	3.42%	0.25%
		.95	41.32%	6.68%
		.99	83.39%	44.32%

Landscape			Neutral neighbors	
N	K	p	Random	LO
256	1	.50	11.91%	7.38%
		.80	45.09%	34.58%
		.95	81.46%	74.15%
		.99	96.11%	94.21%
	2	.50	5.40%	2.29%
		.80	30.82%	16.57%
		.95	74.86%	58.79%
		.99	94.22%	88.94%
	4	.50	1.18%	0.29%
		.80	15.19%	3.66%
		.95	60.85%	29.21%
		.99	90.77%	75.65%
	8	.50	0.05%	0.01%
		.80	3.55%	0.24%
		.95	41.25%	6.65%
		.99	83.70%	44.72%

Landscape			Neutral neighbors	
N	K	p	Random	LO
512	1	.50	12.17%	7.57%
		.80	45.27%	34.92%
		.95	81.69%	74.80%
		.99	96.29%	94.55%
	2	.50	5.39%	2.18%
		.80	31.18%	16.74%
		.95	74.31%	57.97%
		.99	94.57%	89.62%
	4	.50	1.23%	0.29%
		.80	15.24%	3.76%
		.95	61.17%	29.52%
		.99	90.19%	74.54%
	8	.50	0.05%	0.00%
		.80	3.51%	0.23%
		.95	41.40%	6.56%
		.99	83.64%	44.55%

Landscape			Neutral neighbors	
N	K	p	Random	LO
1024	1	.50	12.00%	7.79%
		.80	44.70%	34.15%
		.95	81.67%	74.51%
		.99	96.10%	94.23%
	2	.50	5.62%	2.31%
		.80	31.46%	16.80%
		.95	74.31%	58.05%
		.99	94.15%	88.75%
	4	.50	1.23%	0.27%
		.80	15.11%	3.65%
		.95	61.24%	29.67%
		.99	90.72%	75.62%
	8	.50	0.07%	0.01%
		.80	3.56%	0.26%
		.95	41.27%	6.46%
		.99	83.62%	44.38%

NKq

Landscape			Neutral neighbors	
N	K	q	Random	LO
128	1	10	6.35%	5.65%
		5	15.21%	12.83%
		3	24.38%	21.89%
		2	41.12%	37.23%
	2	10	4.76%	3.08%
		5	12.42%	8.60%
		3	21.32%	16.05%
		2	33.33%	25.50%
	4	10	2.91%	1.09%
		5	9.37%	4.33%
		3	16.21%	8.03%
		2	26.16%	14.21%
	8	10	1.75%	0.36%
		5	6.85%	1.65%
		3	11.80%	3.14%
		2	19.23%	5.83%

Landscape			Neutral neighbors	
N	K	q	Random	LO
256	1	10	6.62%	5.13%
		5	15.26%	13.41%
		3	25.03%	22.26%
		2	39.88%	35.29%
	2	10	4.73%	2.93%
		5	12.21%	8.45%
		3	21.29%	15.84%
		2	33.26%	25.34%
	4	10	2.99%	1.14%
		5	9.44%	4.29%
		3	16.18%	7.88%
		2	26.15%	14.03%
	8	10	1.78%	0.32%
		5	6.85%	1.55%
		3	11.84%	2.96%
		2	19.26%	5.60%

Landscape			Neutral neighbors	
N	K	q	Random	LO
512	1	10	6.70%	5.41%
		5	15.19%	13.36%
		3	26.07%	22.87%
		2	40.16%	36.06%
	2	10	4.65%	2.76%
		5	12.35%	8.73%
		3	21.03%	15.36%
		2	33.47%	25.43%
	4	10	3.02%	1.00%
		5	9.36%	4.15%
		3	16.14%	7.73%
		2	26.15%	13.92%
	8	10	1.80%	0.27%
		5	6.84%	1.46%
		3	11.84%	2.82%
		2	19.25%	5.41%

Landscape			Neutral neighbors	
N	K	q	Random	LO
1024	1	10	6.55%	5.16%
		5	15.23%	13.38%
		3	25.87%	22.53%
		2	40.51%	36.16%
	2	10	4.66%	2.68%
		5	12.33%	8.83%
		3	21.22%	15.57%
		2	33.65%	25.50%
	4	10	3.02%	0.97%
		5	9.40%	4.21%
		3	16.18%	7.79%
		2	26.18%	13.97%
	8	10	1.78%	0.24%
		5	6.85%	1.42%
		3	11.85%	2.77%
		2	19.27%	5.34%

NKr

Landscape			Neutral neighbors	
N	K	r	Random	LO
128	1	10^4	0.80%	0.69%
		10^3	9.38%	5.50%
		10^2	65.57%	50.56%
		10^1	93.68%	90.17%
	2	10^4	0.66%	0.42%
		10^3	7.72%	3.74%
		10^2	58.47%	40.47%
		10^1	93.07%	89.25%
	4	10^4	0.59%	0.22%
		10^3	5.85%	2.03%
		10^2	49.50%	26.38%
		10^1	91.13%	86.27%
	8	10^4	0.43%	0.09%
		10^3	4.27%	0.91%
		10^2	39.01%	13.19%
		10^1	88.16%	82.35%

Landscape			Neutral neighbors	
N	K	r	Random	LO
256	1	10^4	1.82%	1.14%
		10^3	17.86%	8.62%
		10^2	82.23%	68.08%
		10^1	95.85%	93.01%
	2	10^4	1.59%	0.91%
		10^3	15.09%	6.29%
		10^2	78.39%	61.56%
		10^1	95.20%	91.25%
	4	10^4	1.18%	0.42%
		10^3	11.73%	3.49%
		10^2	72.10%	52.48%
		10^1	93.68%	89.41%
	8	10^4	0.86%	0.16%
		10^3	8.57%	1.56%
		10^2	62.97%	36.42%
		10^1	91.56%	86.38%

Landscape			Neutral neighbors	
N	K	r	Random	LO
512	1	10^4	3.97%	2.18%
		10^3	34.48%	19.36%
		10^2	91.18%	83.38%
		10^1	97.09%	94.53%
	2	10^4	3.08%	1.42%
		10^3	28.95%	13.11%
		10^2	89.20%	79.45%
		10^1	96.49%	93.41%
	4	10^4	2.35%	0.72%
		10^3	22.73%	6.98%
		10^2	86.00%	73.59%
		10^1	95.59%	92.01%
	8	10^4	1.72%	0.27%
		10^3	16.90%	2.95%
		10^2	81.05%	64.00%
		10^1	94.02%	89.50%

Landscape			Neutral neighbors	
N	K	r	Random	LO
1024	1	10^4	7.53%	3.21%
		10^3	58.10%	40.82%
		10^2	95.57%	91.04%
		10^1	97.88%	95.83%
	2	10^4	6.17%	2.19%
		10^3	50.82%	30.03%
		10^2	94.65%	89.25%
		10^1	97.54%	95.18%
	4	10^4	4.71%	1.14%
		10^3	41.83%	17.26%
		10^2	92.95%	86.13%
		10^1	96.93%	94.04%
	8	10^4	3.44%	0.46%
		10^3	32.29%	7.50%
		10^2	90.48%	81.29%
		10^1	95.80%	91.97%

Table 1: Estimated rate of neutrality for NKp (top), NKq (middle) and NKr (bottom) landscapes.

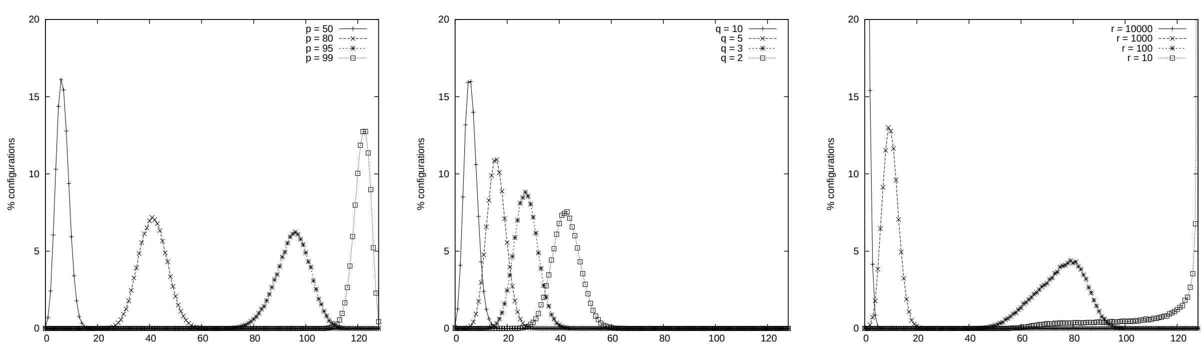

Figure 2: Distribution of 10,000 random configurations by their numbers of neutral neighbors for NKp, NKq and NKr: example with $N = 128$ and $K = 2$.

PR	First	Best	PR	First	Best	PR	First	Best	PR	First	Best
128_1	.7021	**.7090**	256_1	.7021	**.7079**	512_1	.6897	**.6953**	1024_1	.6969	**.7039**
128_2	.7021	**.7082**	256_2	.7066	**.7094**	512_2	.7135	**.7174**	1024_2	.7146	**.7197**
128_4	.7254	**.7260**	256_4	**.7235**	.7204	512_4	**.7200**	.7193	1024_4	**.7246**	.7242
128_8	**.7142**	.7108	256_8	**.7166**	.7122	512_8	**.7206**	.7155	1024_8	**.7216**	.7174

Table 2: Climbers results on NK-landscapes. Only two variants, with no neutral moves, are outputed.

NMP	Basic		Stoch.	Basic+NP	
PR	First	Best	First	First	Best
128_1_p50	.4211	.4283	.4320	.4317	**.4366**
128_1_p80	.2004	.2043	.2241	.2239	**.2253**
128_1_p95	.0523	.0528	.0651	.0651	**.0653**
128_1_p99	.0115	.0116	**.0146**	**.0146**	**.0146**
128_2_p50	.4708	.4761	.4754	.4763	**.4806**
128_2_p80	.2617	.2659	**.2876**	.2840	.2868
128_2_p95	.0787	.0814	.1130	.1131	**.1138**
128_2_p99	.0176	.0174	.0305	.0305	**.0306**
128_4_p50	.5247	.5230	**.5257**	.5250	.5233
128_4_p80	.3258	.3293	**.3380**	.3367	.3363
128_4_p95	.1275	.1262	**.1780**	.1774	.1769
128_4_p99	.0286	.0292	.0788	.0785	**.0789**
128_8_p50	.5223	.5160	**.5233**	.5223	.5161
128_8_p80	.3332	.3301	**.3367**	.3346	.3317
128_8_p95	.1654	.1637	**.1922**	.1901	.1896
128_8_p99	.0489	.0506	**.1138**	.1123	.1129
256_1_p50	.4131	.4197	.4237	.4235	**.4283**
256_1_p80	.2000	.2025	.2208	.2210	**.2222**
256_1_p95	.0527	.0529	**.0655**	.0654	**.0655**
256_1_p99	.0106	.0107	**.0139**	**.0139**	**.0139**
256_2_p50	.4738	.4809	.4795	.4788	**.4850**
256_2_p80	.2642	.2693	.2888	.2893	**.2915**
256_2_p95	.0802	.0818	.1169	.1170	**.1176**
256_2_p99	.0167	.0168	.0303	.0303	**.0304**
256_4_p50	.5257	.5223	.5241	**.5268**	.5230
256_4_p80	.3285	.3242	**.3386**	.3381	.3364
256_4_p95	.1257	.1292	.1800	.1775	**.1801**
256_4_p99	.0297	.0303	**.0791**	.0790	.0788
256_8_p50	**.5261**	.5208	.5251	**.5261**	.5208
256_8_p80	.3350	.3331	**.3409**	.3364	.3342
256_8_p95	.1688	.1710	**.1931**	.1901	.1918
256_8_p99	.0497	.0515	**.1155**	.1150	.1144
512_1_p50	.4092	.4155	.4195	.4195	**.4238**
512_1_p80	.1981	.2011	.2183	.2184	**.2199**
512_1_p95	.0543	.0547	.0674	.0674	**.0676**
512_1_p99	.0108	.0108	**.0142**	**.0142**	**.0142**
512_2_p50	.4804	.4857	.4853	.4855	**.4901**
512_2_p80	.2673	.2715	.2912	.2907	**.2934**
512_2_p95	.0825	.0841	.1184	.1187	**.1192**
512_2_p99	.0172	.0173	**.0317**	**.0317**	**.0317**
512_4_p50	.5226	.5209	.5228	**.5231**	.5216
512_4_p80	.3249	.3227	.3346	**.3357**	.3341
512_4_p95	.1259	.1267	.1782	.1767	**.1783**
512_4_p99	.0294	.0302	**.0785**	.0780	.0781
512_8_p50	.5326	.5258	.5325	**.5327**	.5260
512_8_p80	.3415	.3371	.3419	**.3432**	.3383
512_8_p95	.1690	.1700	**.1936**	.1929	.1935
512_8_p99	.0494	.0507	.1158	**.1162**	.1147
1024_1_p50	.4177	.4233	.4270	.4272	**.4311**
1024_1_p80	.2014	.2045	.2231	.2231	**.2246**
1024_1_p95	.0547	.0550	.0683	.0683	**.0685**
1024_1_p99	.0111	.0111	**.0147**	**.0147**	**.0147**
1024_2_p50	.4809	.4858	.4861	.4864	**.4904**
1024_2_p80	.2663	.2710	.2915	.2916	**.2937**
1024_2_p95	.0833	.0852	.1189	.1188	**.1194**
1024_2_p99	.0174	.0175	.0319	.0319	**.0320**
1024_4_p50	.5245	.5248	**.5269**	.5252	.5258
1024_4_p80	.3246	.3261	**.3378**	.3357	.3361
1024_4_p95	.1268	.1304	.1787	.1779	**.1796**
1024_4_p99	.0285	.0292	**.0784**	.0783	.0783
1024_8_p50	**.5323**	.5269	.5317	**.5323**	.5270
1024_8_p80	.3429	.3382	**.3449**	.3444	.3397
1024_8_p95	.1713	.1688	**.1959**	.1943	.1943
1024_8_p99	.0498	.0514	**.1163**	**.1163**	.1160

(a)

NMP	Basic		Stoch.	Basic+NP	
PR	First	Best	First	First	Best
128_1_q10	.7219	.7266	.7294	.7283	**.7304**
128_1_q5	.7312	.7402	.7484	.7478	**.7507**
128_1_q3	.7581	.7663	.7902	.7909	**.7950**
128_1_q2	.8056	.8062	**.8594**	**.8594**	**.8594**
128_2_q10	.7228	.7269	.7297	.7300	**.7345**
128_2_q5	.7442	.7451	**.7687**	.7664	.7683
128_2_q3	.7688	.7738	**.8186**	.8175	.8176
128_2_q2	.8177	.8212	**.8914**	.8898	.8904
128_4_q10	.7504	.7470	.7557	.7553	**.7567**
128_4_q5	.7657	.7684	**.7922**	.7893	.7912
128_4_q3	.7891	.7891	**.8396**	.8376	.8372
128_4_q2	.8387	.8333	**.9309**	.9273	.9265
128_8_q10	.7341	.7311	**.7417**	.7383	.7351
128_8_q5	.7531	.7534	**.7734**	.7693	.7652
128_8_q3	.7838	.7755	**.8199**	.8145	.8126
128_8_q2	.8264	.8220	**.9020**	.8941	.8965
256_1_q10	.7206	.7275	.7271	.7267	**.7323**
256_1_q5	.7377	.7434	.7518	.7516	**.7557**
256_1_q3	.7626	.7682	.7938	.7938	**.7948**
256_1_q2	.8095	.8143	**.8672**	.8670	.8668
256_2_q10	.7245	.7278	.7318	.7307	**.7340**
256_2_q5	.7390	.7463	.7610	.7608	**.7640**
256_2_q3	.7713	.7784	.8183	.8159	**.8193**
256_2_q2	.8095	.8144	**.8907**	.8879	.8886
256_4_q10	.7441	.7438	**.7503**	.7498	.7490
256_4_q5	.7582	.7588	**.7865**	.7838	.7819
256_4_q3	.7894	.7879	**.8436**	.8407	.8429
256_4_q2	.8323	.8343	**.9363**	.9297	.9288
256_8_q10	.7368	.7343	**.7425**	.7402	.7369
256_8_q5	.7575	.7527	**.7762**	.7741	.7710
256_8_q3	.7871	.7835	**.8255**	.8196	.8200
256_8_q2	.8285	.8277	**.9098**	.9039	.9018
512_1_q10	.7073	.7123	.7126	.7122	**.7170**
512_1_q5	.7233	.7299	.7408	.7402	**.7430**
512_1_q3	.7518	.7564	.7823	.7824	**.7835**
512_1_q2	.7786	.7814	**.8368**	.8354	.8355
512_2_q10	.7327	.7355	.7399	.7385	**.7422**
512_2_q5	.7480	.7532	.7722	.7704	**.7724**
512_2_q3	.7709	.7768	.8187	.8175	**.8196**
512_2_q2	.8227	.8287	**.9060**	.9023	.9021
512_4_q10	.7415	.7380	**.7494**	.7461	.7468
512_4_q5	.7599	.7570	**.7862**	.7824	.7812
512_4_q3	.7863	.7867	**.8398**	.8356	.8351
512_4_q2	.8322	.8315	**.9348**	.9291	.9286
512_8_q10	.7417	.7368	**.7457**	.7454	.7413
512_8_q5	.7597	.7542	**.7804**	.7760	.7765
512_8_q3	.7909	.7843	**.8320**	.8260	.8248
512_8_q2	.8366	.8324	**.9132**	.9074	.9087
1024_1_q10	.7144	.7209	.7203	.7195	**.7256**
1024_1_q5	.7336	.7418	.7514	.7509	**.7550**
1024_1_q3	.7611	.7691	.7955	.7953	**.7971**
1024_1_q2	.7883	.7920	**.8416**	.8411	.8414
1024_2_q10	.7360	.7388	.7434	.7422	**.7462**
1024_2_q5	.7532	.7581	.7768	.7763	**.7783**
1024_2_q3	.7788	.7836	.8259	.8253	**.8260**
1024_2_q2	.8216	.8268	**.9067**	.9036	.9045
1024_4_q10	.7464	.7447	**.7540**	.7511	.7501
1024_4_q5	.7619	.7637	**.7897**	.7850	.7858
1024_4_q3	.7924	.7928	**.8469**	.8424	.8432
1024_4_q2	.8334	.8348	**.9348**	.9286	.9294
1024_8_q10	.7440	.7372	**.7479**	.7467	.7428
1024_8_q5	.7606	.7580	**.7830**	.7789	.7772
1024_8_q3	.7918	.7879	**.8350**	.8299	.8292
1024_8_q2	.8380	.8340	**.9191**	.9125	.9121

(b)

NMP	Basic		Stoch.	Basic+NP	
PR	First	Best	First	First	Best
128_1_r10000	.7017	**.7087**	.7023	.7017	**.7087**
128_1_r1000	.7005	.7036	.7082	.7057	**.7103**
128_1_r100	.5369	.5310	.7127	**.7131**	.7125
128_1_r10	.4700	.4700	**.5250**	.5000	.5000
128_2_r10000	.7024	.7076	.7061	.7036	**.7088**
128_2_r1000	.7000	.6992	**.7155**	.7108	.7132
128_2_r100	.5692	.5646	**.7313**	.7299	.7306
128_2_r10	.4710	.4710	**.5310**	.5020	.5020
128_4_r10000	.7249	.7254	**.7279**	.7256	.7264
128_4_r1000	.7210	.7179	**.7362**	.7304	.7341
128_4_r100	.6278	.6213	**.7750**	.7608	.7625
128_4_r10	.4870	.4870	**.6000**	.5340	.5330
128_8_r10000	.7147	.7098	.7145	**.7156**	.7120
128_8_r1000	.7112	.7050	**.7216**	.7176	.7138
128_8_r100	.6428	.6393	**.7510**	.7421	.7353
128_8_r10	.4770	.4770	**.6000**	.5560	.5560
256_1_r10000	.7024	.7067	.7046	.7039	**.7091**
256_1_r1000	.6915	.6869	**.7134**	.7095	.7117
256_1_r100	.5037	.5039	**.6874**	.6848	.6842
256_1_r10	.4710	.4710	**.5000**	**.5000**	**.5000**
256_2_r10000	.7052	.7087	.7075	.7062	**.7104**
256_2_r1000	.6953	.6914	**.7211**	.7157	.7168
256_2_r100	.5000	.4998	**.7150**	.7124	.7117
256_2_r10	.4560	.4560	**.5000**	**.5000**	**.5000**
256_4_r10000	.7209	.7179	**.7243**	.7232	.7222
256_4_r1000	.7088	.7029	**.7410**	.7312	.7311
256_4_r100	.5220	.5224	**.7720**	.7575	.7599
256_4_r10	.4700	.4700	**.5000**	**.5000**	**.5000**
256_8_r10000	.7169	.7103	**.7216**	.7177	.7135
256_8_r1000	.7071	.7016	**.7282**	.7224	.7227
256_8_r100	.5633	.5591	**.7714**	.7575	.7583
256_8_r10	.4770	.4770	**.5010**	.5000	.5000
512_1_r10000	.6887	.6922	.6932	.6923	**.6966**
512_1_r1000	.6522	.6389	**.7054**	.7016	.7023
512_1_r100	.4958	.4958	**.6296**	.6021	.6017
512_1_r10	.4550	.4550	**.5000**	**.5000**	**.5000**
512_2_r10000	.7117	.7140	.7170	.7149	**.7189**
512_2_r1000	.6780	.6710	**.7398**	.7298	.7301
512_2_r100	.4985	.4985	**.6667**	.6443	.6430
512_2_r10	.4640	.4640	**.5000**	**.5000**	**.5000**
512_4_r10000	.7193	.7173	**.7240**	.7228	.7222
512_4_r1000	.6906	.6810	**.7543**	.7400	.7383
512_4_r100	.4999	.4999	**.6926**	.6784	.6794
512_4_r10	.4620	.4620	**.5000**	**.5000**	**.5000**
512_8_r10000	.7187	.7138	**.7228**	.7220	.7192
512_8_r1000	.6994	.6908	**.7429**	.7310	.7304
512_8_r100	.5059	.5056	.7066	.7115	**.7129**
512_8_r10	.4750	.4750	**.5000**	**.5000**	**.5000**
1024_1_r10000	.6943	.6974	.7031	.6999	**.7059**
1024_1_r1000	.5750	.5675	**.7135**	.7115	.7114
1024_1_r100	.4979	.4979	**.5857**	.5498	.5492
1024_1_r10	.4640	.4640	**.5000**	**.5000**	**.5000**
1024_2_r10000	.7122	.7130	.7228	.7189	**.7233**
1024_2_r1000	.6133	.6033	**.7498**	.7418	.7418
1024_2_r100	.4999	.4999	**.6043**	.5657	.5652
1024_2_r10	.4690	.4690	**.5000**	**.5000**	**.5000**
1024_4_r10000	.7219	.7180	**.7328**	.7280	.7276
1024_4_r1000	.6486	.6397	**.7748**	.7592	.7589
1024_4_r100	.4979	.4979	**.6158**	.5832	.5851
1024_4_r10	.4670	.4670	**.5000**	**.5000**	**.5000**
1024_8_r10000	.7187	.7133	**.7277**	.7233	.7229
1024_8_r1000	.6656	.6586	**.7617**	.7469	.7481
1024_8_r100	.4977	.4977	**.6208**	.6014	.6016
1024_8_r10	.4600	.4600	**.5000**	**.5000**	**.5000**

(c)

Table 3: Climbers results on NKp (a), NKq (b) and NKr (c) landscapes.

of neutrality. Globally, one observe that the search space size given by parameter N does not influence the overall tendency of the results; although efficiency differences between policies tends to be more significant for larger search spaces.

Major conclusions highlighted with NKq remains valid for NKr as shown in table 3.c In particular, basic climbers are clearly outperformed, and stochastic climbing statistically dominates other variants on most instances. Let us notice that $r = 10$ leads to excessively flat landscapes, then *_*_r10 results are not really significant for comparing climbers. Since these landscapes contains huge plateaus, escaping from them require the use of more specific methods.

Let us recall that the only difference between original NK instances (used table 2) and the ones used here results of the precision of the configuration fitnesses. Consequently, it makes sense to compare directly the NK and NKr results for each (N, K) parametrization. While comparing results from tables 2 and 3.c, one observe that several r parametrizations can reach configurations of much higher fitness values. This is illustrated in figure 3, which compare the efficiency of first and best improvement climbers on original fitnesses (NK-landscape), with rounded ones (NKr-landscape, with $r = 100$ and $r = 1000$). One observe that, for each (N, K) parametrization, an appropriate rounding of the fitness function can lead to more efficient climbings, provided that the bringing of neutrality is exploited. Setting r consists then to find a compromise between more neutrality and less precision. The figure also indicate that appropriate r values increase with N and decrease with K, which is coherent with the fitness function (see equation 1). More generally, an appropriate discretization of evaluation functions should help to solve optimization problems with neighborhood search techniques.

5. CONCLUSION

Climbers are often considered as basic components of advanced search methods. However, influence of their conception choices are rarely discussed through advanced studies. In this paper we have focused on the capacity of different hill-climbing versions to reach good configurations in various landscapes. In particular, we compared the first and best improvement strategies as well as three different neutral move policies. In order to provide an empirical analysis on a large panel of representative instances, we used NK-landscapes with different sizes, rugosity levels and shapes of neutrality. On landscapes with no neutrality, we show that best improvement performs better on smooth landscapes, while first improvement is well-suited on more rugged ones. To evaluate the impact of neutral move policies, we use three models of neutrality: existing NKp and NKq, as well as NKr, which simply consists in rounding fitnesses. First, one observes that stochastic hill-climbings globally reach better configurations than other variants. In other words, at each step of the search, it makes sense to perform the first non-deteriorating move instead of extending the neighborhood evaluation. Moreover, experiments on NKr sheds light on the interest of considering rounded evaluation function during a climbing procedure. Combined with an appropriate neutral move policy, this should avoid to be trapped prematurely in local optima. Indeed, rounding fitnesses intuitively lead to consider also some deteriorating moves during the search. Robustness and efficiency benefits of such

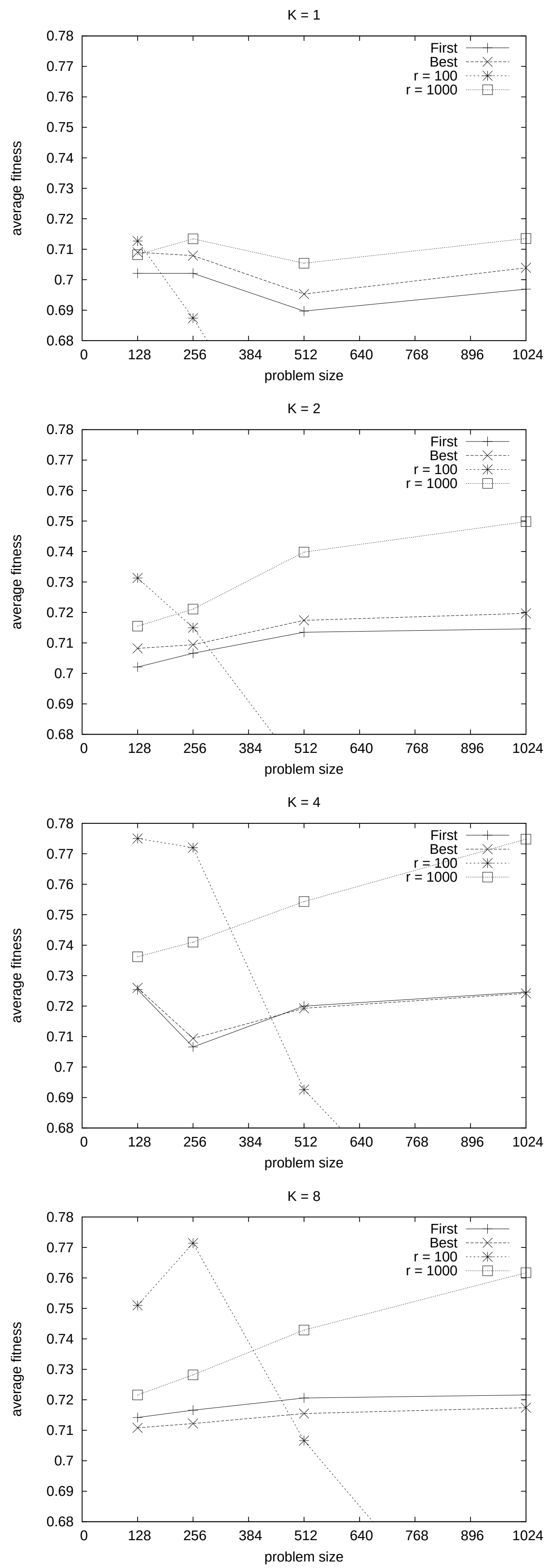

Figure 3: Results obtained by rounding the fitness function.

a move policy have often been verified by advanced search techniques like simulated annealing.

Perspectives of this work mainly includes the extension of this analysis to Iterative Local Search methods [6]. Indeed, several questions arise while considering iterated versions. First, we have to determine to what extent efficient climbers can improve iterated searches. Secondly, a similiar study performed in an iterated context will determine if the overall influence of structural choices remain unchanged. Last, although this study focused on a large scale of landscapes, a next step could be to deal with uncertain and/or multiobjective problems, whose climber components depend on many other factors.

Acknowledgements

This work is partially supported by the Pays de la Loire Region (France) within the LigeRO project.

6. REFERENCES

[1] Lionel Barnett. Ruggedness and neutrality - the NKp family of fitness landscapes. In *Alive VI: Sixth International Conference on Articial Life*, pages 18–27. MIT Press, 1998.

[2] Jeremy Frank, Peter Cheeseman, and John Stutz. When gravity fails: Local search topology. *J. Artif. Intell. Res. (JAIR)*, 7:249–281, 1997.

[3] Pierre Hansen and Nenad Mladenovic. First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5):802–817, 2006.

[4] Holger Hoos and Thomas Stützle. *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[5] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, USA, 1 edition, 06 1993.

[6] Helena R. Lourenço, Olivier Martin, and Thomas Stützle. Iterated local search. In Fred Glover and Gary Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 321–353. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 2002.

[7] M. E. J. Newman and Robin Engelhardt. Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society B*, 265(1403):1333–1338, 07 1998.

[8] Gabriela Ochoa, Sébastien Verel, and Marco Tomassini. First-improvement vs. Best-improvement Local Optima Networks of NK Landscapes. In *Proceedings of the 11th International Conference on Parallel Problem Solving From Nature*, pages 104–113, Krakow Pologne, 09 2010.