Scuba Search: when selection meets innovation
Sébastien Verel, Philippe Collard, Manuel Clergue

To cite this version:

HAL Id: hal-00160035
https://hal.archives-ouvertes.fr/hal-00160035
Submitted on 4 Jul 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—We proposed a new search heuristic using the scuba diving metaphor. This approach is based on the concept of evolvability and tends to exploit neutrality in fitness landscape. Despite the fact that natural evolution does not directly select for evolvability, the basic idea behind the scuba search heuristic is to explicitly push the evolvability to increase. The search process switches between two phases: Conquest-of-the-Waters and Invasion-of-the-Land. A comparative study of the new algorithm and standard local search heuristics on the NKq-landscapes has shown advantage and limit of the scuba search. To enlighten qualitative differences between neutral search processes, the space is changed into a connected graph to visualize the pathways that the search is likely to follow.

I. INTRODUCTION

In this paper we propose a novel heuristic called Scuba Search that allows to exploit the neutrality that is existing in many real-world fitness landscapes.

This section presents the interplay between neutrality in fitness landscapes and metaheuristics. Section II describes the Scuba Search heuristic in detail. In order to illustrate the efficiency of this heuristic, we use the NKq-landscape as a model of neutral fitness landscape. This one is developed in section III. The experiment results are given in section IV where comparisons are made with standard heuristics. Section V analyzes the neutral search process of scuba search. We point out in section VI the shortcoming of the approach and propose a generic scuba search heuristic. Finally, we summarize our contribution and present plans for future work.

A. Neutrality

The metaphor of ‘adaptive landscape’ introduced by S. Wright [1] has dominated the view of adaptive evolution: an uphill walk of a population on a mountainous fitness landscape in which it can get stuck on suboptimal peaks. Results given by molecular evolution has changed this picture: Kimura [2] establishes that the overwhelming majority of mutations are either effectively neutral or lethal and in the latter case purged by negative selection. This theory is called the theory of molecular evolution. This theory can help us to revisit the metaphor of adaptive landscape and define a neutral landscape.

In neutral landscape population can walk on mountain but also on plateaus (neutral networks). The dynamics of population evolution is then a metastable evolution as proposed by Gould and Eldredge [3] characterized by long periods of fitness stasis (population stated on a ‘neutral network’) punctuated by shorter periods of innovation with rapid fitness increase.

In the field of evolutionary computation, neutrality plays an important role. Under the assumption that the neutral networks are nearly isotropic\(^1\), Barnett [4] proposes an heuristic adapted to neutral landscape: the Netcrawler process. It is a random neutral walk with a mutation mode adapted to local neutrality. The per-sequence mutation rate is optimized to jump to one neutral network to another one. When the isotropic assumption is not verified, for a population based algorithm, Nimwegen et al. [5] show the population’s limit distribution on the neutral network is solely determined by the network topology. The population seeks out the most connected areas of the neutral network.

In real-world problems as in design of digital circuits [6][7][8], in evolutionary robotics [9] or in genetic programming [10], neutrality is implicitly embedded in the genotype to phenotype mapping. Another possibility in evolutionary optimization is to introduce artificial redundancy into the encoding [11][12]. This may improve the evolvability of genotype or create neutral paths to escape from suboptimal peaks.

B. Evolvability

Evolvability is defined by Altenberg [13] as ”the ability of random variations to sometimes produce improvement”. This concept refers to the efficiency of evolutionary search; it is based upon the work by Altenberg [14]: ”the ability of an operator/representation scheme to produce offspring that are fitter than their parents”. Smith et al. [15] focus on the ideas of evolvability and neutrality; they plot the average fitness of offspring over fitness of parents (considering 1-bit mutation as an operator). As enlighten by Turney [16] the concept of evolvability is difficult to define. As he puts it: ”if s and s' are equally fit, s is more evolvable than s' if the fittest offspring of s is more likely to be fitter than the fittest offspring of s'”. Following this idea we define evolvability as a function (see section II-B).

II. SCUBA SEARCH

A. The conquest of the waters

Keeping the landscape as a model, fill each area between two peaks (local optima) with water allow lakes to emerge. Thus, the landscape is bathed in an uneven sea; areas under water represent non-viable solutions. So now there are paths from one peak to the other one for a swimmer. The key, of

\(^1\)Same neutral degree and same probability to jump
course, remains to locate an attractor which represents the system’s maximum fitness. In this framework, the problem is how to cross a lake without global information. We use the scuba diving metaphor as a guide to present the principles of the so-called scuba search (SS). This heuristic is a way to deal with the problem of crossing between peaks and so avoid to be trapped in the vicinity of local optima. The problem is what drives the swimmer from one edge to the opposite edge of the lake? The classic view is the one of a swimmer drifting at the surface of a lake. The new metaphor is a scuba diver seeing the world above the water surface. We propose a new heuristic to cross a neutral net getting information above-the-surface (ie. from fitter points of the neighborhood).

B. Scuba Search Algorithm

Despite the fact that natural evolution does not directly select for evolvability, there is a dynamic pushing evolvability to increase [16]. As Dawkins [17] states, “This is not ordinary Darwinian selection but it is a kind of high-level analogy of Darwinian selection”. The basic idea behind the SS heuristic is to explicitly push evolvability to increase. Before presenting this search algorithm, we need to introduce a new type of local optima, the local-neutral optima. Indeed with SS heuristic, local-neutral optima will allow transition from neutral to adaptive evolution. So evolvability will be locally optimized.

Given a search space $S$ and a fitness function $f$ defined on $S$, some more precise definitions follow.

**Definition:** A neighborhood structure is a function $V : S \rightarrow 2^S$ that assigns to every $s \in S$ a set of neighbors $V(s)$ such that $s \in V(s)$

**Definition:** The evolvability of a solution $s$ is the function $evol$ that assigns to every $s \in S$ the maximum fitness from the neighborhood $V(s)$: $\forall s \in S$, $evol(s) = \max\{f(s') | s' \in V(s)\}$

**Definition:** For every fitness function $g$, neighborhood structure $W$ and genotype $s$, the predicate $isLocal$ is defined as: $isLocal(s, g, W) = (\forall s' \in W(s), g(s') \leq g(s))$

**Definition:** For every $s \in S$, the neutral set of $s$ is the set $N(s) = \{s' \in S | f(s') = f(s)\}$, and the neutral neighborhood of $s$ is the set $\mathcal{V}n(s) = V(s) \cap N(s)$

**Definition:** For every $s \in S$, the neutral degree of $s$, noted $Degn(s)$, is the number of neutral neighbors of $s$, $Degn(s) = |\mathcal{V}n(s)| - 1$

**Definition:** A solution $s$ is a local maximum iff $isLocal(s, f, V)$

**Definition:** A solution $s$ is a local-neutral maximum iff $isLocal(s, evol, \mathcal{V}n)$

There are two overlapping dynamics during the Scuba Search process. The first one is identified as a neutral path. At each step the scuba diving remains under the water surface driven by the hands-down fitnesses; that is fitter fitness values reachable from one neutral neighbor. At that time the flatCount counter is incremented. When the diving reaches a local-neutral optimum, that is if all the fitnesses reachable from one neutral neighbor are selectively neutral or disadvantageous, the neutral path stops and the diving starts up the Invasion-of-the-Land. Then the gateCount counter increases. This process goes along, switching between Conquest-of-the-Waters and Invasion-of-the-Land, until a local optimum is reached.

**Algorithm 1 Scuba Search**

1. flatCount ← 0, gateCount ← 0
2. Choose initial solution $s \in S$
3. repeat
   1. while not $isLocal(s, evol, \mathcal{V}n)$ do
      1. $M = \max\{evol(s') | s' \in \mathcal{V}n(s) \{-s\}\}$
      2. if $evol(s) < M$ then
         1. choose $s' \in \mathcal{V}n(s)$ such that $evol(s') = M$
         2. $s \leftarrow s'$, flatCount ← flatCount +1
      end if
   end while
   1. choose $s' \in \mathcal{V}(s) - \mathcal{V}n(s)$ such that $f(s') = evol(s)$
   2. $s \leftarrow s'$, gateCount ← gateCount +1
4. until $isLocal(s, f, \mathcal{V})$

III. Model of Neutral Landscape

In order to study the Scuba Search heuristic we have to use landscapes with a tunable degree of neutrality. The $N K q$ fitness landscapes family proposed by Newman et al. [18] has properties of systems undergoing neutral selection such as RNA sequence-structure maps. It is a generalization of the $N K$-landscapes proposed by Kauffman [19] where parameter $K$ can tune the ruggedness and parameter $q$ the degree of neutrality of the landscape.

A. Definition

The fitness function of a $N K q$-landscape is a function $f : \{0, 1\}^N \rightarrow [0, 1]$ defined on binary strings with $N$ loci. Each locus $i$ represents a gene with two possible alleles, 0 or 1. An 'atom' with fixed epistasis level is represented by a fitness components $f_i : \{0, 1\}^{K+1} \rightarrow [0, q − 1]$ associated to each locus $i$. It depends on the allele at locus $i$ and also on the alleles at $K$ other epistatic loci ($K$ must fall between 0 and $N − 1$). The fitness $f(x)$ of $x \in \{0, 1\}^N$ is the average of the values of the $N$ fitness components $f_i$:

$$f(x) = \frac{1}{N(q-1)} \sum_{i=1}^{N} f_i(x_i; x_{i_1}, \ldots, x_{i_K})$$

where $\{i_1, \ldots, i_K\} \subset \{1, \ldots, i−1, i+1, \ldots, N\}$. Many ways have been proposed to choose the $K$ other loci from $N$ loci in
the genotype. Two possibilities are mainly used: adjacent and random neighborhoods. With an adjacent neighborhood, the \( K \) genes nearest to the locus \( i \) are chosen (the genotype is taken to have periodic boundaries). With a random neighborhood, the \( K \) genes are chosen randomly on the genotype. Each fitness component \( f_i \) is specified by extension, i.e., an integer number \( y_i(x_1; x_{i1}, \ldots, x_{ik}) \) from \([0, q - 1]\) is associated with each element \((x_1; x_{i1}, \ldots, x_{ik})\) from \([0, 1]\)^{K+1}. Those numbers are uniformly distributed in the interval \([0, q - 1]\).

### B. Properties

The parameters of \( N K q \)-landscape tune ruggedness and neutrality of the landscape [18][20]. The number of local optima is linked to parameter \( K \). The largest number is obtained when \( K \) takes its maximum value \( N - 1 \). The neutral degree (see tab. 1) decreases when \( q \) increases and when \( K \) increases. The maximal degree of neutrality appears when \( q \) takes the value 2.

### IV. Experiment Results

#### A. Algorithm of Comparison

Three algorithms of comparison are used: two kinds of Hill Climbing and one heuristic adapted to neutral landscape, the Netcrawler Process.

1) Hill Climbing: The simplest type of local search is known as Hill Climbing (HC) when trying to maximize a solution. HC is very good at exploiting the neighborhood; it always takes what looks best at that time. But this approach has some problems. The solution found depends from the initial solution. Most of the time, the found solution is only a local optima. We start off with a probably suboptimal solution. Then we look in the neighborhood of that solution to see if there is something better. If so, we adopt this improved solution as our current best choice and repeat. If not, we stop assuming the current solution is good enough (local optimum).

**Algorithm 2 Hill Climbing**

\[
\text{step} \leftarrow 0 \\
\text{Choose initial solution } s \in S \\
\text{repeat} \\
\quad \text{choose } s' \in V(s) \text{ such that } f(s') = \text{evol}(s) \\
\quad s \leftarrow s', \text{ step} \leftarrow \text{step} + 1 \\
\text{until isLocal}(s, f, V) \\
\]

2) Netcrawler Process: We also compare SS to Netcrawler Process (NC) proposed by Barnett [4]. This is a local search adapted to fitness landscapes featuring neutral networks. Netcrawler uses a mutation per-sequence mutation rate which is calculated from the neutral degree of neutral networks [4].

In the case of \( N K q \)-landscapes, experimentations and estimations of neutral degree given by [20] yield as mutation flips only one bit per genotype. The algorithm [3] displays the Netcrawler process.

**Algorithm 3 Netcrawler Process**

**Require:** stepMax > 0

\[
\text{step} \leftarrow 0 \\
\text{Choose initial solution } s \in S \\
\text{repeat} \\
\quad \text{choose } s' \in V(s) \text{ randomly} \\
\quad \text{if } f(s) \leq f(s') \text{ then} \\
\quad \quad s \leftarrow s' \\
\quad \text{end if} \\
\quad \text{step} \leftarrow \text{step} + 1 \\
\text{until stepMax} \leq \text{step} \\
\]

3) Hill Climbing Two Steps: Hill Climber can be extended in many ways. Hill Climber Two Step (HC2) exploits a larger neighborhood of stage 2. The algorithm is nearly the same as HC. HC2 looks in the extended neighborhood of stage two of the current solution to see if there is something better. If so, HC2 adopts the solution in the neighborhood of stage one which can reach a best solution in the extended neighborhood. If not, HC2 stop assuming the current solution is good enough. So, HC2 can avoid more local optimum than HC. Before presenting the algorithm [4] we must introduce the following definitions:

**Definition:** The extended neighborhood structure\(^3\) from \( V \) is the function \( V^2(s) = \bigcup_{s \in V(s)} V(s) \)

**Definition:** \( \text{evol}^2 \) is the function that assigns to every \( s \in S \) the maximum fitness from the extended neighborhood \( V^2(s) \). \( \forall s \in S, \text{evol}^2(s) = \max\{f(s')|s' \in V^2(s)\} \)

**Algorithm 4 Hill Climbing (Two Steps)**

\[
\text{step} \leftarrow 0 \\
\text{Choose initial solution } s \in S \\
\text{repeat} \\
\quad \text{if } \text{evol}(s) = \text{evol}^2(s) \text{ then} \\
\quad \quad \text{choose } s' \in V(s) \text{ such that } f(s') = \text{evol}^2(s) \\
\quad \text{else} \\
\quad \quad \text{choose } s' \in V(s) \text{ such that } \text{evol}(s') = \text{evol}^2(s) \\
\quad \text{end if} \\
\quad s \leftarrow s', \text{ step} \leftarrow \text{step} + 1 \\
\text{until } \text{isLocal}(s, f, V^2) \\
\]

\(^2\)More exactly to \( \epsilon \)-correlated fitness landscapes

\(^3\)Let’s note that \( V(s) \subset V^2(s) \)

<table>
<thead>
<tr>
<th>q</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>2</td>
<td>15.00</td>
<td>21.33</td>
<td>16.56</td>
<td>12.39</td>
<td>10.69</td>
</tr>
<tr>
<td>3</td>
<td>12.00</td>
<td>13.29</td>
<td>10.43</td>
<td>7.65</td>
<td>6.21</td>
<td>5.43</td>
</tr>
<tr>
<td>4</td>
<td>12.00</td>
<td>6.71</td>
<td>4.30</td>
<td>2.45</td>
<td>1.66</td>
<td>1.24</td>
</tr>
<tr>
<td>100</td>
<td>1.00</td>
<td>0.32</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
B. Parameters setting

All the four heuristics are applied to a same instance of N K q fitness landscape. The search space S is {0, 1}^N, that is bit strings of length N. In this paper all experiments are led with N = 64. The selected neighborhood is the classical one-bit mutation neighborhood: \( \mathcal{V}(s) = \{ s' | \text{Hamming}(s, s) \leq 1 \} \). For each triplet of parameters N, K and q, 10^3 runs were performed. For netcrawler process, stepMax is set to 300^5.

C. Average performances

Figure 2 shows the average fitness found respectively by each of the four heuristics as a function of the epistatic parameter K for different values of the neutral parameter q. In the presence of neutrality, according to the average fitness, Scuba Search outperforms Hill Climbing, Hill Climbing two steps and Netcrawler. Let us note that with high neutrality (q = 2 and q = 3), difference is still more significant. Without neutrality (q = 100) all the heuristics are nearly equivalent, except Netcrawler.

The two heuristics adapted to neutral landscape, Scuba Search and Netcrawler, have on average better fitness value for q = 2 and q = 3 than hill climbing heuristics. These heuristics benefit in NKq-landscapes from the neutral paths to reach the highest peaks.

D. Evaluation cost

Table I shows the number of evaluations for the different heuristics except for Netcrawler. For this last heuristic the number of evaluations is constant whatever K and q values are; and we get the smallest evaluation cost from all the heuristics (maxStep = 300 evaluations). For all the other heuristics, the number of evaluations decreases with K. The evaluation cost decreases as ruggedness increases. For HC and HC2, the evaluation cost increases with q. For HC and HC2, more neutral the landscape is, smaller the evaluation cost. Conversely, for SS the cost decreases with q. At each step the number of evaluations is N for HC and \( \frac{N(N-1)}{2} \) for HC2. So, the cost depends on the length of adaptive walk of HC and HC2 only. The evaluation cost of HC and HC2 is low when local optima are nearby (i.e. in rugged landscapes). For SS, at each step, the number of evaluations is \((1 + \text{Degn}(s))N\) which decreases with neutrality. So, the number of evaluations depends both on the number of steps in SS and on the neutral degree. The evaluation cost of SS is high in neutral landscape.

V. ANALYSIS

According to the average fitness found, Scuba Search outperforms the others heuristics on the NKq fitness landscapes. However, it should be wondered whether efficiency of Scuba Search does have with the greatest number of evaluations. The neutral search process of Netcrawler and Scuba marked by dissimilarity. We will analyze their own strategy.

A. Exploration neighborhood size

The number of evaluations for Scuba Search is greater than the one for HC or for Netcrawler. But it lesser than the one for HC2. This last heuristic realizes a larger exploration of the neighborhood than SS: it pays attention to neighbors with same fitness and all the neighbors of the neighborhood too. However the average fitness found is less good than the one found by SS. So, the number of evaluations is not sufficient to explain good performance of SS. Whereas there is premature convergence towards local optima with HC2. SS realizes a better compromise between exploration and exploitation by examining neutral neighbors.

B. Neutral search process

Difference between the two neutral heuristics can be explained analyzing the respective neutral search process.

For Netcrawler, as mutation is uniform on the neighborhood, the probability of neutral mutation is proportional to the neutral degree: \( P_{\text{neutral Mut}}(s) = \frac{\text{Degn}(s)}{N} \). Neutral mutation depends on the neutral degree only. If a mutation is neutral, no way is preferred, all the neutral paths have the same probability to be chosen. The Netcrawler promotes neutral drift.

For Scuba Search, the probability of neutral mutation increases with the neutral degree (see fig. 3); without however be proportional. This probability is larger for SS than for Netcrawler and depends on parameters q and K also. Contrary to our intuition about neutral mutation, the probability of neutral move decreases as neutrality increases. The probability of neutral mutation for one given neutral degree increases with q and K as well. SS performs a neutral move from genotype s to \( s' \) if \( s' \) have a strictly greater evolvability than \( s \). When neutrality is large, all the neutral neighbors have a strong probability to have the same evolvability. In other words, neutral networks are few connected to neutral sets and then neutral moves are not frequent. When neutrality is a little more important, evolvability of neutral neighbors become quite different and then SS have a larger probability to move while maintaining the same fitness.

Figure 5 shows the average number of steps (gateCount + flatCount) and the average number of neutral mutations.
Fig. 1. Average fitness found on $NKq$-landscapes as function of $K$, for $N = 64$ and $q = 2$ (a), $q = 3$ (b), $q = 4$ (c), $q = 100$ (d)

Fig. 2. Probability of neutral mutation for Scuba Search and Netcrawler as a function of neutral degree for $N = 64$, $q = 3$ (a) and $K = 4$ (b)
two types of change from \( H C \) is specified by the scuba algorithm (see algorithm 1). There are neighborhoods. The choice between neutral path and one individual to one of the fittest genotypes in its neutral landscape. According to the fitness function \( f \), each arrow connects one individual to the fittest genotype in its neighborhood. According to the evolvability function \( evol \), each dotted arrow connects one individual to the fittest genotype in its neutral neighborhood.

Figure 4 shows all the \( H C \) paths. NKq landscape represented as a connected graph (\( N=5, K=2, q=2 \)). According to the fitness function \( f \), each arrow connects one individual to the fittest genotype in its neighborhood. According to the evolvability function \( evol \), each dotted arrow connects one individual to the fittest genotype in its neutral neighborhood.

24). Second, some \( HC \) paths can be replaced by a neutral path (see for instance the new neutral path from node 11 to node 10). Globally the number of local optima and basins of attraction tends to reduce (equal to five here).

According to the fitness function \( f \), each solid arrow in Figure 5 connects one individual to a strict fitter genotype in its neighborhood and dotted arrows connect two neighbors with the same fitness. So, a Netcrawler Search path is a subgraph of this graph.

Figure 5 shows all the \( Scuba \) paths. According to the fitness function \( f \), each arrow connects one individual to one of the fittest genotypes in its neighborhood. We can see eight local optima as well the corresponding basins of attraction for this landscape.

Figure 7 shows all the \( Hill \ Climbing \) paths through the landscape. According to the fitness function \( f \), each arrow connects one individual to one of the fittest genotypes in its neighborhood. We can see eight local optima as well the corresponding basins of attraction for this landscape.

To enlighten differences between the search processes, we map the landscape onto a 2-dimensional space. This technique inspired by Layzeel [21] allows to visualize the pathways within a search space that a heuristic is likely to follow. The space is transformed into a connected graph. Vertices are connected if their Hamming distance from each other is one. According to the heuristic used, \( HC, SS \) or \( NC \), salient edges are picked out, and the rest discarded. Each genotype (vertex) is shaded according to its fitness. This allows to “see” all the neutral sets. To illustrate \( HC, SS, NC \), and \( HC2 \) potential dynamics, a small NKq-landscape, with intermediate epistasis and high neutrality, is considered (\( N = 5, K = 2, q = 2 \)).

According to the fitness function \( f \), each solid arrow in Figure 5 connects one individual to a strict fitter genotype in its neighborhood and dotted arrows connect two neighbors with the same fitness. So, a Netcrawler Search path is a subgraph of this graph.

Figure 7 shows all the \( Hill \ Climbing \) two steps paths. According to the fitness function \( f \), each arrow connects one individual to one of the fittest genotypes in its neighborhood. The choice between neutral path and \( HC \) path is specified by the scuba algorithm (see algorithm 1). There are two types of change from \( HC \) to \( SS \). First, some new neutral paths appear (see for example the path from node 8 to node 7).

\( In \) our example, low-fitness genotypes are black.
B. Generic Scuba Search

sequence mutation rate suggested by Barnett [4]. For example we could use the per-

degree of neutrality leads evolvability to be constant on each

Road landscape, proposed by Mitchell et al. [22], a high

on a neutral network. For example in the well-known Royal-

explicitly optimize evolvability on a neutral network before

Fig. 7. Hill Climbing two steps paths. NKq landscape represented as a
connected graph (N=5, K=2, q=2). According to the fitness function f, each arrow
connects one individual to a fitter genotype in its extended neighborhood (size 2).

VI. DISCUSSION

A. Limit of the Scuba search

The main idea behind Scuba Search heuristic is to try to explicitly optimize evolvability on a neutral network before performing a qualitative step using a local search heuristic. Optimized evolvability needs evolvability not to be constant on a neutral network. For example in the well-known Royal-Road landscape, proposed by Mitchell et al. [22], a high degree of neutrality leads evolvability to be constant on each neutral network. A way to reduce this drawback is to modify the neighborhood structure \( V \) induced by the choice of sequence mutation rate; for example we could use the per-

sequence mutation rate for example we could use the per-

allows jumping to a fitter solution. For instance, we could use Simulated Annealing [23] or Tabu search [24] to optimize neutral network then jump to the first improvement meets in the neighborhood.

More generally, the Scuba Search heuristic could be extended in three ways:

• evolvability definition,
• local search heuristic to optimize neutral network,
• local search heuristic to jump toward fitter solution.

From these remarks we propose to defined the Generic Scuba Search (alg. [3]).

Algorithm 5 Generic Scuba Search

\[
\begin{align*}
&\text{flatCount} \leftarrow 0, \text{gateCount} \leftarrow 0 \\
&\text{Choose initial solution } s \in S \\
&\text{repeat} \\
&\quad \text{while terminal condition}_1 \text{ not met do} \\
&\quad \quad s \leftarrow \text{Improve}_1(s, \text{eval}, Vn(s)) \\
&\quad \quad \text{flatCount} \leftarrow \text{flatCount} + 1 \\
&\quad \text{end while} \\
&\quad s \leftarrow \text{Improve}_2(s, f, V(s)) \\
&\quad \text{gateCount} \leftarrow \text{gateCount}+1 \\
&\text{until terminal condition}_2 \text{ met}
\end{align*}
\]

VII. CONCLUSION AND PERSPECTIVES

This paper represents a first step demonstrating the potential interest in using scuba search heuristic. According to the average fitness found, SS outperforms hill climbing heuristics and netcrawler on the NKq fitness landscapes. Comparison with HC2 algorithm has shown that SS efficiency does not have with the number of evaluations only. Mapping the landscape onto 2-dimensional space allows qualitative difference between neutral search processes to emerge.

When neutrality is too high the scuba search stops in the ‘middle’ of neutral networks. In future work, we would like to replace Improve1 heuristic by a tabu search allowing to drift on neutral networks.

As our implementation uses hill climbing as Improve1 and Improve2 heuristics, SS is more to be compare to a local search heuristic without deleterious mutation. Thus using Tabu Search on neutral landscape it could be useful to replace HC by SS. At each iteration of tabu search we choose a new better solution which is not in the tabu list. This better solution is a neutral neighbor if its evolvability is greater than evolvability of current solution. If not, it is the fittest neighbor.

Performance of heuristics adapted to neutral landscape depends on the difficulty to optimize neutral networks and the connectivity between networks. It would be interesting to have a measure of neutral networks optimization difficulty. Moreover if we are able to measure the difficulty to jump from one neutral network to another one then we can compare the efficiency of neutral exploration.

Last but not least we obviously have to study scuba search on other problems than NKq-landscapes, in particular on...
real-world fitness landscapes where neutrality already naturally exists.

References


