



HAL
open science

Partial neighborhood local searches

Sara Tari, Matthieu Basseur, Adrien Goëffon

► **To cite this version:**

Sara Tari, Matthieu Basseur, Adrien Goëffon. Partial neighborhood local searches. *International Transactions in Operational Research*, 2022, 29 (5), pp.2761-2788. 10.1111/itor.12983. hal-03384306v1

HAL Id: hal-03384306

<https://hal.science/hal-03384306v1>

Submitted on 9 Feb 2023 (v1), last revised 17 Jan 2024 (v2)

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.

Partial Neighborhood Local Searches

Sara Tari · Matthieu Basseur · Adrien
Goeffon

Received: date / Accepted: date

Abstract PNLS 3 mthodes + 2 state of the art : expe param + problmes
comportement-paysages

1 Introduction

In the field of combinatorial optimization, research can follow two conflictual directions: on the one hand, problem-oriented approaches that exploit specificities of the studied problem, and generic approaches that try to provide guidelines to establish solvers that can handle optimization problems in general.

This observation can be done for exact algorithms as well as for heuristic search algorithms. When it comes to search algorithms, problem-oriented algorithms use the structure of a given problem to guide the search toward high-quality solutions. In contrast, generic methods aim at finding guidelines that can be applied to many combinatorial optimization problems.

Heuristics search algorithms mainly include metaheuristics that often consist of navigating through the search space with a couple (neighborhood relation, move strategy) and return the best-encountered solution at the end of the search. The navigation process creates a sample of the search space where solutions are examined, thus obtaining a good sample is crucial for such methods. Finding an appropriate neighborhood relation and an appropriate move strategy is the key to find good solutions of the search space. The general

Sara Tari
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

Matthieu Basseur
second address

principle of local search based metaheuristics is elementary since it mainly consists of using a neighborhood relation and a move strategy.

As stated by Sörensen in [20], methods proposed in the field of metaheuristics are more sophisticated than ever, especially since the dramatic increase in the number of bio-inspired methods. While metaheuristics are meant to be generic methods, a significant part of them is dedicated to solving a given problem. Most of these methods mainly exploit problem specificities to navigate through the search space and are still called metaheuristics. While such methods can be particularly efficient to tackle problems, authors should not label them as metaheuristics but more as metaheuristic-based methods. The study of sophisticated methods involves the consideration of many parameters, which interact in a complex way. Understanding the behavior of such approaches is a complicated task.

Nevertheless, improving the understanding of metaheuristics can provide insights on when and how to use them. Such knowledge is particularly crucial to determine which approach is the most suited for a given problem. This is particularly the case when the approaches are intended to be generic. Analyzing metaheuristics, instead of only proposing new methods without further knowledge, is the primary key to better use such methods.

Metaheuristics are often based on a neighborhood relation that associates a set of solutions to each solution of the search space. To analyze such methods, one can use the notion of fitness landscapes [25]. The concept of fitness landscapes leads to an abstraction of the problem where the height of solutions corresponds to their fitness, and the neighborhood relation defines the connexions between solutions. This abstraction allows studying optimization methods in a more general way by using landscape properties. One can establish correlations between the dynamics of methods and the properties of landscapes to obtain insights on why the method is efficient or not. The properties that characterize fitness landscapes, among them some are recognized to make the landscape challenging to tackle. Investigating methods with a landscape approach can highlight tendencies that help to determine the type of landscapes where some methods are expected to be efficient.

While fitness landscapes allow the study of any neighborhood-based search method, investigating sophisticated search strategies without preliminary studies is particularly complicated. Indeed, many metaheuristics consist of assembled mechanisms whose individual contribution and parameterization are challenging to measure. However, studying interactions between mechanisms require beforehand to understand precisely their individual behavior. For this reason, there is a need to deconstruct metaheuristics to facilitate the understanding of what makes a search algorithm efficient. Even if advanced metaheuristics alternate different mechanisms to manage the balance between a sufficient intensification of the search on promising areas of the search space and a sufficient exploration of the search space, one can often achieve such a balance with fewer mechanisms.

In a previous study [5], we highlighted the relevance of accepting a fixed part of the less deteriorating neighboring solutions during the search process;

the efficiency of strategies such as simulated annealing on many problems supports this principle.

Here we are interested in more stochastic searches that reduce the exploratory effort of neighborhoods and increase the frequency of steps during the search. We propose to consider Partial Neighborhood Local Search algorithms (PLNS), which are based on random restriction of the neighborhood at each step of the search. In particular, we investigate the Sample Walk algorithm (SW) introduced in [23], which can be viewed as a local search transposition of a $(1, \lambda)$ -ES. We also consider a previously described algorithm that also falls into the scope of PNLs (ID Walk [16]). The two main contributions of this work are the following. First, we perform a parameter sensitivity analysis of PLNS methods, which only consists of determining the appropriate λ values. Then, we study the behavior of partial-neighborhood based methods on different landscapes to highlight some relations between landscape properties and PLNS efficiency.

The outline of the paper is as follows. In the next section, we present fitness landscapes and related concepts. Section 3 presents the partial neighborhood local searches we study in this work. The three last sections are dedicated to empirical analysis of fitness landscapes and PLNS algorithms. Experiments are realized on various bit-string and permutation landscapes derived from various problems: NK landscapes and Unconstrained Binary Quadratic Programming problem (binary string solution representation), as well as Quadratic Assignment Problem and Flowshop Scheduling Problem (permutation solution representation). In section 4, we report experiments that allow to characterize landscapes according to indicators. In section 5, some experiments are conducted in order to determine the adequate parameterization of the considered algorithms. Particular attention is dedicated to PLNS methods and the sensibility of the results to λ values. In section 6, we confront PLNS methods with a tabu search and an iterated local search. The last section provides a discussion on the current work, which leads us to propose future perspectives.

2 Fitness landscapes

Nowadays, fitness landscapes are used in various fields to apprehend the behavior of complex systems better. In evolutionary computation, such a model can help to observe difficulties induced by a given problem when tackled with an optimization method. Indeed, the difficulty of solving a problem is not only determined by the instance of the problem but also depends on the considered method. In particular, as showed for some problems (e.g., flow-shop [15]), choosing a relevant neighborhood operator is essential to ease the navigation of search methods through the search space. Fitness landscapes are based on the chosen neighborhood operator and allow the study of some methods following landscape properties.

2.1 Definitions

A fitness landscape is a triplet $(\mathcal{X}, \mathcal{N}, f)$ where \mathcal{X} denotes the search space, $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ a neighborhood function describing a set of neighbors associated to each solution, and f the fitness function which assigns a score to each solution. Given a combinatorial problem instance (\mathcal{X}, f) , one can derive a fitness landscape by adding a neighborhood structure. Neighborhood-based search algorithms behavior can be analyzed by studying such derived landscapes.

Therefore, several fitness landscapes can be defined for a given instance, depending on the neighborhood definition possibilities. In some cases, one can also use different solutions representations as well as different search spaces (for instance, including non-feasible solutions or excluding some feasible solutions). A primary interest of fitness landscapes in evolutionary computation is the study of neighborhood-based optimization methods depending on landscapes' properties. Most of these properties, such as the ones described in [13], can influence the behavior of search algorithms. In this study, we mainly focus on the dimension, the ruggedness, and the neutrality of landscapes.

The dimension of a landscape is determined by the size of the search space and the connectivity between solutions. The size of the search space refers to the number of candidate solutions and directly affects the difficulty of finding a given solution. The connectivity corresponds to the (usually fixed) number of neighbors of a solution.

The ruggedness of a landscape is a significant property that determines the difficulty of optimizing the underlying problem using the considered neighborhood operator. It mainly refers to the number of local optima, their distribution through the search space, and the size of their basins of attraction. The basin of attraction [17] of a local optimum is the set of solutions that can lead toward the considered local optimum through a basic hill-climbing algorithm. A rugged landscape would have many local optima with small basins of attractions. On the contrary, a smooth landscape would have few local optima with large basins of attraction. A landscape with the maximal smoothness only has one local optimum (which is the global optimum). The ruggedness of a landscape can be evaluated by computing or estimating the number of local optima and the size of their basins of attraction.

Ruggedness is also related to the epistasis phenomenon, that reflects the degree of variable interdependency between genes [6]. When the interdependence between decision variables is high, it is difficult to estimate how the value of a particular variable affects the solution fitness. Such a landscape has high epistasis since the effect of a mutation depends on the presence of other mutations. The ruggedness estimation, thanks to the epistasis phenomenon, is discussed in the next section.

Neutrality occurs in a landscape when two neighboring solutions have the same fitness value.

Some landscapes have a significant neutrality rate, and the presence of such a feature can have a non-negligible effect on the number and distribution of local optima. Indeed, landscapes with high levels of neutrality are generally

hard to solve and induce questions about how to tackle neutral moves within local search algorithms. In particular, many metaheuristics have a move policy to navigate through plateaus (set of connected neutral neighbors) [14].

Neutrality can have a significant impact on the number and distribution of local optima, and therefore on the ability of search algorithms to find good solutions. Indeed, it has a smoothing effect that makes it more challenging to guide the search towards good solutions, since their fitness does not allow comparison. When it is not taken into account, neutrality can thus act as a brake on search methods. The presence of plateaus (definition 1) in the landscape induces an additional difficulty for search algorithms that can wander in these areas, which can lead to a large number of unnecessary evaluations and prevent reaching better solutions. Note that the artificial addition of neutrality in a rough landscape can, however, also help to direct the search towards good solutions, in particular by preventing it from being trapped in some local optima [1].

Definition 1 *A plateau is a set of connected points of the same fitness and corresponds to a related sub-graph induced by $\mathcal{S} \subseteq \mathcal{X}$, such that $\forall(x_i, x_j) \in \mathcal{S}^2, f(x_i) = f(x_j)$.*

Thus, on a landscape with a high level of neutrality, taking this characteristic into account in order to limit the difficulties encountered during search is often crucial. The quantification of this neutrality is an aspect that should not be neglected when studying or proposing resolution methods.

2.2 Landscape features

In this section, we describe the indicators used in this paper to characterize landscapes. The ruggedness level of landscapes is a key aspect determining the problems' hardness. At the same time, ruggedness cannot be clearly described using a single indicator [13]. Here, we will focus on the *autocorrelation function*, the *k-ruggedness*, and the *average distance between local optima*. Moreover, we also consider the *neutral rate* of landscapes since neutrality also induces a significant impact on the behavior of search algorithms.

2.2.1 Autocorrelation function

The *autocorrelation function* introduced by Weinberger in [24] is a widely used ruggedness indicator. It requires to perform several random walks (s_0, s_1, s_2, \dots) through the considered landscape. This indicator calculates the fitness correlation between the solutions encountered during the random walk according to their distances. In classical fitness landscapes derived from combinatorial optimization problems, the correlation between two close solutions tends to be high. In contrast, it tends to be close to zero on two distant solutions. The rate of decay in function of the distance indicates the ruggedness rate of the

considered landscape. More formally, the following autocorrelation function ρ provides an estimation of the influence of the distance on the variation of solution fitness between pairs of solutions within the walk:

$$\rho(l) = \frac{\langle f(x_t)f(x_{t+l}) \rangle_{x_0,t} - \langle f(x_t) \rangle_{x_0,t}^2}{\langle f(x_t)^2 \rangle_{x_0,t} - \langle f(x_t) \rangle_{x_0,t}^2} \quad (1)$$

x_0 and t indicate that the average values are computed from the set of starting solutions x_0 and along the complete random walk.

The result is a plot of autocorrelation $\rho(l) \in [-1, 1]$, where $l \in \{0, \dots, t\}$ is the step size of considered walks of length n . $|\rho(l)| = 1$ indicates a maximal correlation whereas $|\rho(l)|$ close to 0 indicates almost no correlation.

2.2.2 k -ruggedness

While the efficiency of metaheuristics is often influenced by deteriorating and improving moves, the autocorrelation measure is based on fitness variation and does not sufficiently consider sign epistasis. Consequently, we also use the k -ruggedness indicator based upon the k -sign-epistasis principle introduced hereafter.

Let s_0 and s_k two solutions distant from k with respect to the considered neighborhood (e.g., the Hamming distance while considering binary strings and the 1-flip operator). Let s'_0 and s'_k neighbors of respectively s_0 and s_k resulting from the same move application (e.g., the same bit is flipped, considering binary strings and 1-flip). Moreover, the solutions s'_0 and s'_k have to respect the constraints $d(s_0, s'_k) = k + 1$ and $d(s'_0, s'_k) = k$ (considering binary strings and 1-flip, the flipped bit cannot differ in s_0 and s_k). The k -sign-epistasis is defined as follow:

$$\phi(s_0, s'_0, s_k, s'_k) = \text{epis}((f(s_0) - f(s'_0)) \times (f(s_k) - f(s'_k))) \quad (2)$$

where $\text{epis}(x)$ equals to 0 if x is positive, and 1 otherwise. The k -sign-epistasis concept is somehow natural in binary string landscapes and is depicted in figure 1. However, applying this concept to other solution representations is more complicated and less intuitive since the landscape is then not necessarily a hypercube.

1-ruggedness is called local ruggedness and refers to the 1-epistasis rate on several pairs of mutations. k -ruggedness reflects a global ruggedness and refers to the k -epistasis rates when the same mutation is applied to solutions of distance k .

2.2.3 Average distance between local optima

The number of local optima is the primary aspect of characterizing the problem's hardness and is strongly related to the two previous measures introduced. These measures are useful for evaluating this aspect, but consider a uniform repartition of ruggedness throughout the landscape. The study local optima

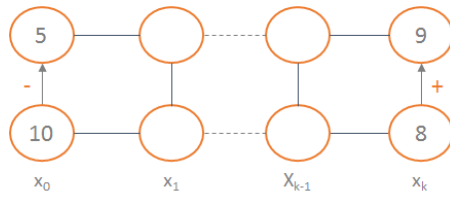


Fig. 1 Illustration of k -sign-epistasis while applying a given mutation on two solutions x_0 and x_k (x_0 and x_k are linked by a minimal set of solutions $\{x_1 \dots x_{k-1}\}$, such that $\forall i \in [1..k], x_i \in \mathcal{N}(x_{i-1})$).

repartition can also provide information on the difficulty of solving a given landscape. Here we consider the *average distance between local optima*, a simple way to estimate when local optima are uniformly distributed or not. Let O be a set of local optima found by executing a first improvement hill-climbing algorithm. D_{LO} is the average distance between two solutions of O .

Let D_{rand} be the average distance between two random solutions of the search space. If D_{LO} and D_{rand} are similar, then local optima are likely to be uniformly distributed in the search space. Oppositely, the more D_{LO} is smaller than D_{rand} , the more clustered are the local optima.

A known hypothesis related to the distribution of local optima in some landscapes is that landscapes derived from specific combinatorial problems have a *central massif* structure. The local optima of fitness landscapes having a central massif (*big valley*) structure are clustered around a central global optimum (low D_{LO} value). In such cases, there exist many local optima that are easy to escape from, making the optimization of such landscapes less difficult.

2.2.4 Neutrality

Neutrality influences the behaviour of search methods, especially since it implies a situation where it is not possible to determine through fitness whether a neighbouring solution is preferable to another. A combinatorial landscape considered neutral is not a flat landscape, but rather a landscape where a significant proportion of transitions between pairs of solutions are neutral.

Thus, on a landscape with strong neutrality, taking this characteristic into account to limit the difficulties encountered during the search is often crucial. The quantification of this neutrality is an aspect that should not be neglected when studying or proposing resolution methods. Neutrality can be quantified using various indicators, for example, through random walks or the identification of neutral networks. In our work, we define the rate of neutrality (definition 2), which corresponds to the average proportion of neutral neighbors in a solution. The simplicity of the measurement makes it possible to easily approximate its value by sampling the search space, even if it does not take into account the distribution of neutrality in the landscapes, as is sometimes the case [3].

Definition 2 *The neutral rate of a fitness landscape $(\mathcal{X}, \mathcal{N}, f)$, is defined as follows:*

$$v(\mathcal{X}, \mathcal{N}, f) = \frac{\#\{(x, x'), x \in \mathcal{N}(x'), f(x) = f(x')\}}{\#\{(x, x'), x \in \mathcal{N}(x')\}}$$

2.3 From combinatorial optimization problems to fitness landscapes

Studying methods behavior is facilitated by the use of models allowing tunable ruggedness, such as NK landscapes presented hereafter. Studies on artificial landscapes with such features are essential to apprehend the link between ruggedness and mechanisms efficiency better. The general aspect of such landscapes is limited since their ruggedness is uniformly distributed, unlike in real problems. In light of this specific fact, conducting studies related to ruggedness on only NK landscapes should not be considered as sufficient. Tackling landscapes derived from real problems can help to obtain better insights into the behavior of the considered methods within such studies.

In the following, we first introduce NK landscapes, then we present three different classes of problems and their associated landscapes: UBQP, FSP, and QAP. We use these different classes of problems in the experiments of the paper.

NK landscapes are a model of binary fitness landscapes introduced by Kaufmann [10]. They are widely used when studying the link between ruggedness and methods behavior since their specificity is to have a tunable ruggedness. These landscapes have two parameters: N , which specifies the number of variables, and K , which determines the level of variable interdependency and directly influences the ruggedness rate. Setting K to zero leads to an entirely smooth landscape with no variable interdependency, whereas setting K to $N - 1$ leads to an entirely rugged (random) landscape.

The fitness function F of NK landscapes to be maximized is defined as follows:

$$f(x) = \frac{1}{N} \sum_{i=1}^N C_i(x_i, \Pi_i(x)) \quad (3)$$

x_i is the i -th bit of the solution x , Π_i a subfunction which defines the dependencies of bit i , with $\Pi_i(x)$ such that $\pi_j(i) \in \{1, \dots, N\} \setminus \{i\}$ and $|\cup_{j=1}^N \pi_j(i)| = K$, subfunction $C_i : \{0, 1\}^{K+1} \rightarrow [0, 1)$ 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 $(x_i, x_{\pi_1(i)}, \dots, x_{\pi_K(i)})$ and a matrix C of fitness contribution which describes the $2^N \times (K + 1)$ possible contribution values.

The neighborhood operator under consideration in the study is the *one-flip*. Two solutions x and x' are neighbors iff their Hamming distance is equal to 1.

Various parameterizations of NK instances ($N \in \{128, 256, 512, 1024\}$, $K \in \{1, 2, 4, 6, 8, 10, 12\}$) will be considered in experiments in order to observe algorithms behavior in various contexts (size and ruggedness).

The Unconstrained Binary Quadratic Programming problem (**UBQP**) is an NP-hard problem [8], which can reformulate a vast scope of real-life problems in various fields. An instance of UBQP is composed of a matrix $n \times n$ of constants which can be positive or negative. A solution 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 to be maximized is described as follows:

$$f(I) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j \quad (4)$$

The neighborhood operator under consideration to define fitness landscapes associated with the UBQP problem is the one-flip operator. Hence, each solution has a neighborhood of size n . We used an instance generator (proposed and provided by Gintaras Palubeckis) to generate several instances of different sizes and density. The density d defines the rate of values equal to zero in the matrix Q .

The Flowshop Scheduling Problem (**FSP**) [21] is a widely studied scheduling problem. Among the numerous FSP variants, we consider the permutation flow-shop with total completion time minimization. An instance of FSP can be composed by n jobs J_1 to J_n to be scheduled, m machines where jobs must be scheduled in a specific order m_1 to m_m , and a set of $n \times m$ tasks t_{ij} where t_{ij} represents the processing time of the job J_i on the machine m_j . Let us notice that two jobs cannot be scheduled simultaneously on a machine. Moreover, on the variant under consideration, all jobs must be scheduled in the same order on each machine. A solution I is then represented by a permutation of jobs. Each task is scheduled to the date s_{ij} .

The objective function C_{max} , also called total completion time or makespan, to be minimized is described as follows:

$$C_{max}(x) = \max_{i \in \{1, \dots, n\}} \{s_{im} + t_{im}\}$$

The search space S_n is the set of permutations of $\{1, \dots, n\}$, and is of size $n!$. The most efficient neighborhood operator to solve this problem consists of moving a job to a different position and is called insert operator N_{ins} . This operator allows the definition of the fitness landscape corresponding to the triplet (S_n, N_{ins}, C_{max}) . We select 6 representative instances with $j \in \{30, 50\}$ jobs and $m \in \{10, 15, 20\}$ machines [22].

The Quadratic Assignment Problem (**QAP**) [11] is a permutation problem considered as one of the most difficult to solve NP-hard problems [19]. QAP consists of affecting n units to n locations in function of a distance matrix D and flow between units matrix F . Let d_{ij} be the distance between locations i and j , and f_{rs} the flow between units r and s . A permutation I describes the affectations of units to locations is a solution to this problem. The objective function to be minimized then corresponds to the sum of distances multiplied by the corresponding flows and is defined as follows:

$$f(\Pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\Pi_i \Pi_j}$$

where Π_i represents the i -th element of the solution $\Pi \in S_n$. FSP and QAP share common search spaces S_n . Here, the considered neighborhood operator to solve QAP is the commonly used swap operator N_{swap} , which consists of exchanging the location of two units. For each solution Π , the neighborhood size remains the same: $|N_{swap}| = \binom{n}{2} \forall \Pi \in S_n$. The associated landscape is then defined by (S_n, N_{swap}, F) . We use 8 instances of size $n \in [42, 90]$ from the QAPLIB [7]. These instances come from two classes: `lipa` and `sko`. `lipa` instances are asymmetric instances with known optimal solutions while `sko` instances have rectangular distances and pseudorandom numbers as entries in flow matrices.

In the following, we use these four problems: two bitstring problems and two permutation problems. Notice that the representations of permutations lead to different landscapes for which some characterization indicators must be redefined. This aspect will be discussed later in section 5.

3 Partial neighborhood local searches

3.1 Local search

Local search algorithms [9] are widely used metaheuristics. In their initial form, such methods are particularly straightforward and therefore allow the addition of a considerable number of mechanisms to enhance the search process and its capacity to reach good solutions.

A local search algorithm explores the search space by navigating through a set of solutions with respect to a neighborhood relation and an evaluation function. Algorithm 1 describes a local search where x is the current solution, $(N)(x)$ the neighborhood of x and x^* an archive of the best-encountered solution.

Such an algorithm is defined by the way to determine an initial solution, a move policy (also called pivoting rule) and a stopping criterion. Many move policies were defined in a general way, leading to well-known metaheuristics such as simulated annealing or tabu search.

Partial neighborhood local searches (PNLS) constitute a simple instantiation of local search algorithms. In the following, we first discuss about our motivations then we describe PNLS with more details.

3.2 Motivations

Let us recall the aim of this study is to obtain insights to conceive local search algorithms. More precisely, we focus on establishing links between optimization problem structure and efficiency of local searches, not to tackle a particular problem to beat records. It is useful to focus on basic methods to better isolate and study some mechanisms used among search algorithms. In the case of local

Algorithm 1 Local search algorithm

```

1: Choose  $x_0 \in \mathcal{X}$  (initialization)
2:  $x \leftarrow x_0$ 
3:  $x^* \leftarrow x$ 
4: while stop criterion not reached do
5:   Select a neighbor  $x' \in \mathcal{N}(x)$ 
6:    $x \leftarrow x'$ 
7:   if  $f(x) > f(x^*)$  then
8:      $x^* \leftarrow x$ 
9:   end if
10: end while
11: return  $x^*$ 

```

search algorithms, many studies focus on climbers since they do not require many mechanisms and are widely used as a component within metaheuristics. Moreover, a study [4] showed that despite their simplicity, climbers have a high potential. In fact, such algorithms can often reach the global optimum regardless the initial solution.

Some studies focusing on investigating climbers pivoting rules in accordance to ruggedness showed that among widely used first and best improvement, the first improvement is, in general, the most likely to reach high local optima. For instance, Ochoa et Al. [18] showed, by exhaustive analysis on small NK landscapes, that first improvement is efficient to climb significantly rugged landscapes while best improvement leads to better local optima on both highly smooth and highly rugged landscapes. Let us notice that difficult landscapes derived hard optimization problems are in general significantly rugged. Furthermore, in [1], an empirical study of such pivoting rules on landscapes derived from various optimization problems highlighted the interest of the first improvement on significantly rugged large-sized landscapes. Another study [2] investigated the worst improvement and showed it was more likely than first and best improvement to lead toward good local optima on significantly rugged landscapes.

The acceptance of neutral solutions can potentially avoid the termination of the climbing process and drive toward higher local optima. Since improving the current solution seems more natural, the selection of a neutral neighbor is often only considered once a strict local optimum is reached. In [1] a stochastic hill-climber which indifferently selects the first encountered improving or neutral neighbor was investigated. Experiments on various landscapes show that such a pivoting rule clearly outperforms climbers selecting improving neighbors as a priority.

Since accepting neutral neighbors as well as improving ones during the search process helps to reach higher pikes, the effect of adding artificial neutrality within climbers was studied in [5]. In the proposed method, artificial neutrality is obtained by discretizing an adaptative fitness function within a climber. The resulting climber outperforms classic hill-climbing process since the search is less likely to be trapped in local optima. Intuitively, this performance is linked to the addition of neutrality which tends to decrease the

ruggedness rate within the landscape. However, adding too much neutrality can lead to more difficult landscapes by creating plateaus and inducing an increased number of moves. A too high value of artificial neutrality induces too much diversification within the process by allowing the selection of weak solutions, while an adequate rate of artificial neutrality helps to diversify the search within climbers sufficiently.

Classic climbers are usually used as intensification mechanisms within sophisticated metaheuristics since they almost only perform pure intensification during a search process. However, sophisticated metaheuristics using climbers also own a diversification mechanism to achieve a right balance between intensification and diversification during the search. In the climbing process mentioned above, the diversification mechanism is directly included. Although such a method leads toward higher local optima compared to traditional climbers, it remains a hill-climbing process. A climber is a restrictive local search which necessarily induces a high amount of intensification and in this case, the systematic evaluation of the whole neighborhood at each step of the search. Besides, the discretized evaluation function used in this method induces a modification of the landscape to perform the search; not altering the landscape to study methods in function of ruggedness rates is more appropriate.

Here, we wonder how to simulate the mechanism induced by artificial neutrality in a less restrictive local search, without the systematic evaluation of the whole neighborhood and with no landscape modification. In general, a non-strict local search offers more possibilities to improve solutions than a climber, and a local search performing fewer evaluations at each step induces more steps and intuitively more diversification for the same cost credit. Indeed, evaluating only a few neighbors promotes diversification by reducing the possibility to encounter a particularly good solution at the next step of the search. Using this principle within a non-strict local search can help to manage the balance between intensification and diversification with a single mechanism, as opposed to local search algorithms requiring two separate mechanisms to achieve this balance (for example, an Iterated Local Search).

3.3 Methods

Partial Neighborhood Local Search (PNLS) selects at each step of the search a solution from a sample of neighbors whose (maximum) size, denoted as λ , is configurable. The PLNS process is detailed in algorithm 2, and is a possible instantiation of local search (see algorithm 1). Let us precise that the λ neighboring solutions are not necessarily generated and evaluated. As proposed later in the paper, variants can generate and evaluate a *maximum* of λ neighbors at each step of the search.

PNLSs mainly manage the balance between intensification and diversification through the value of λ . For instance, when $\lambda = 1$, the PNLSS correspond to random walks and are then exclusively diversifying, whereas when $\lambda = |\mathcal{N}|$ the search is particularly intensifying (particularly when the selection pro-

Algorithm 2 Partial Neighborhood Local Search algorithm

```

1: Choose  $x_0 \in \mathcal{X}$  (initialization)
2:  $x \leftarrow x_0$ 
3:  $x^* \leftarrow x$ 
4: while stop criterion not reached do
5:    $\mathcal{N}_\lambda \leftarrow$  subset of  $\lambda$  random solutions of  $\mathcal{N}(x)$ 
6:   Select a neighbor  $x' \in \mathcal{N}_\lambda$ 
7:    $x \leftarrow x'$ 
8:   if  $f(x) > f(x^*)$  then
9:      $x^* \leftarrow x$ 
10:  end if
11: end while
12: return  $x^*$ 

```

cess is based upon fitness values). The strategies used to select neighbors also influence the balance between intensification and diversification.

In this paper, we study the impact of different selection strategies for PLNSs. In particular, we propose the sampled walk algorithm (**SW**). **SW** is particularly simple to implement, in particular since it applies the same selection strategy to improving and deteriorating neighbors. We confront **SW** with ID_{walk} or the *intensification/diversification walk* proposed in [16], which is declined in two variants : ID_{best} and ID_{any} . These two ID_{walk} variants are interesting since they can be classified as PNLs algorithms.

Then, we study three PLNSs algorithms that operate as follows:

- **SW** selects the solution with the best fitness from a sample of λ neighboring solutions, whether it is improving or deteriorating;
- ID_{best} selects the first improving solution encountered. If no improving solution is evaluated, then the best deteriorating neighbor among the sample of neighbors λ is selected;
- ID_{any} selects the first improving solution encountered. If no improving solution is evaluated, then a neighbor is randomly selected from the sample of λ neighbors.

Thus, the only difference between **SW** and ID_{best} belongs in the pivoting rule applied to improving neighbors. **SW** selects the best improving neighbor among $\kappa \leq \lambda$, while ID_{best} selects the first encountered improving neighbour. Considering a fixed value of λ , the intensification rate is then naturally higher for **SW** than for ID_{best} .

The difference between ID_{best} and ID_{any} consists of the rule applied when only deteriorating neighbors are evaluated. When the sample contains no improving neighbor, ID_{best} selects the best solution the *lambda* neighbors (like in **SW** process), whereas ID_{any} selects it randomly, which induces a higher rate of diversification intuitively when no improving solution is found.

The table 1 illustrates a step of the three PNLs in an example. This example shows the selection probabilities of each of the ten neighbors of a solution for **SW**, ID_{best} and ID_{any} . The fitness of the current solution is 100, and the neighborhood contains four improving neighbors (in a maximization

Solution	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Fitness	120	112	108	103	98	95	91	88	84	80
SW ($\lambda = 2$)	9/45	8/45	7/45	6/45	5/45	4/45	3/45	2/45	1/45	0
ID _B ($\lambda = 2$)	1/6	1/6	1/6	1/6	5/45	4/45	3/45	2/45	1/45	0
ID _A ($\lambda = 2$)	1/6	1/6	1/6	1/6	1/18	1/18	1/18	1/18	1/18	1/18
SW ($\lambda = 4$)	84/210	56/210	35/210	20/210	10/210	4/210	1/210	0	0	0
ID _B ($\lambda = 4$)	13/56	13/56	13/56	13/56	10/210	4/210	1/210	0	0	0
ID _A ($\lambda = 4$)	13/56	13/56	13/56	13/56	1/84	1/84	1/84	1/84	1/84	1/84

Table 1 Illustration of the selection process of SW, ID_{best} and ID_{any}, on a example of a neighborhood of 10 solutions $\{S_1, \dots, S_{10}\}$. The fitness of the current solution being 100, S_1, \dots, S_4 are improving and S_5, \dots, S_{10} are deteriorating. Values indicate the selection probability of each couple {selection method, solution}.

context). The probabilities given corresponds to the exact values. For instance, considering SW, S_2 is selected when it is appears in the sample, but not S_1 . The selection probabilities of each solution make it possible to identify the differences between the PNLs. First, SW promotes more intensification than ID_{best} and also ID_{any}, which is the least intensive strategy. Moreover, the selection probabilities induced by SW perfectly respect the relation of order of neighbors' fitnesses.

In the next section, we perform an empirical comparison of the different algorithms to observe the consequences, in terms of efficiency, induced by the selection strategy and the sample size, that influences balances between diversification and intensification.

We compare the PNLs with two classic local searches : an iterated local search (ILS) and a taboo search (TS) to observe whether considering partial neighborhoods is effective in achieving good solutions.

4 Landscape analysis through indicators

In this section we report the analysis of the landscapes derived from the problems introduced in section 2.3. The indicator used for the analysis are those presented in section 2.2.

The results are reported in the tables 2, 3, ??, and ??. In these tables, the two first columns report $\rho(1)$ and 1-ruggedness estimated with a sample of 100,000 solutions.

Starting from these solutions, 100,000 random walks have been performed to estimate the autocorrelation function. Then we report the length of the steps (in %) when thresholds are reached. Here these thresholds are setted to 0.8 and 0.5. When the threshold are quickly reached, it corresponds to a low correlation between fitness correlation and distance between solution, and then a landscape more difficult to explore efficiently using local searches. Similarly, thresholds are used to report δ -epistasis (values 0.1 and 0,25). δ -epistasis values have been estimated only for binary landscapes, since this indicator is defined only in this context and can not be easily adapted in a permutation representation context.

N K	$\rho(1)$	1-rug.	$\rho(l)$		δ -rug.		$d_{LO}(d_{Rd})$	Neut.
			≤ 0.8	≤ 0.5	≥ 0.1	≥ 0.25		
128 1	0.9793	0.5%	8.0%	28.3%	15.7%	43.3%	24.5 (64)	0.00%
128 2	0.9745	1.2%	7.0%	22.0%	7.9%	22.9%	44.9 (64)	0.00%
128 4	0.9584	3.1%	4.7%	14.2%	3.1%	10.2%	60.9 (64)	0.00%
128 6	0.9449	5.2%	3.1%	10.2%	2.4%	7.1%	63.5 (64)	0.00%
128 8	0.9290	7.4%	3.1%	7.9%	1.6%	4.7%	63.9 (64)	0.00%
128 10	0.9140	9.6%	2.3%	7.0%	1.6%	3.9%	64.0 (64)	0.00%
128 12	0.8983	11.8%	2.3%	5.5%	0.8%	3.1%	64.0 (64)	0.00%
256 1	0.9899	0.3%	8.6%	27.4%	16.1%	43.9%	50.2 (128)	0.00%
256 2	0.9870	0.6%	6.7%	20.8%	7.5%	22.4%	93.7 (128)	0.00%
256 4	0.9808	1.5%	4.7%	14.5%	3.1%	10.2%	121.4 (128)	0.00%
256 6	0.9718	2.6%	3.5%	10.6%	1.9%	6.7%	126.7 (128)	0.00%
256 8	0.9659	4.1%	2.7%	8.2%	1.2%	4.7%	127.7 (128)	0.00%
256 10	0.9571	5.3%	2.3%	6.7%	1.2%	3.9%	127.0 (128)	0.00%
256 12	0.9488	6.5%	2.0%	5.5%	0.8%	3.1%	128.0 (128)	0.00%
512 1	0.9945	0.1%	8.4%	27.2%	16.7%	45.0%	101.7 (256)	0.00%
512 2	0.9937	0.3%	6.8%	22.1 %	7.0%	22.3%	187.4 (256)	0.00%
512 4	0.9894	0.8%	4.5%	14.2%	2.9%	10.4%	243.3 (256)	0.00%
512 6	0.9853	1.4%	3.3%	10.1%	1.8%	6.4%	253.3 (256)	0.00%
512 8	0.9823	2.1%	2.5%	8.2%	1.2%	4.7%	255.4 (256)	0.00%
512 10	0.9777	2.7%	2.2%	6.5%	0.9%	3.7%	255.8 (256)	0.00%
512 12	0.9747	3.6%	1.8%	5.7%	0.7%	3.1%	260.0 (256)	0.00%
1024 1	0.9977	0.1%	8.5%	27.0%	15.7%	42.7%	200.0 (512)	0.00%
1024 2	0.9970	0.2%	6.8%	22.1%	7.6%	23.6%	366.9 (512)	0.00%
1024 4	0.9951	0.4%	4.4%	14.3%	2.9%	10.1%	486.7 (512)	0.00%
1024 6	0.9929	0.7%	3.1%	10.2%	1.8%	6.5%	507.2 (512)	0.00%
1024 8	0.9912	1.1%	2.5%	7.8%	1.2%	4.8%	510.6 (512)	0.00%
1024 10	0.9897	1.4%	2.2%	6.5%	0.9%	3.7%	511.8 (512)	0.00%
1024 12	0.9870	1.8%	1.8%	5.5%	0.7%	3.1%	511.9 (512)	0.00%

Table 2 NK landscapes indicator values.

The three last columns of the tables report respectively the average distance between local optima and the neutrality rate.

The average distance between local optima is computed from 1000 local optima obtained with a first improvement hill-climber, and the value between parenthesis correspond to the average distance between random solutions.

The neutrality rate is estimated using a sample of 100,000 random pairs of neighboring solutions.

REFAIRE ET COMPLETER TOUS LES TABLEAUX ! — i + Neut + dist moyenne

Analyse:

The indicator values computed for each class of landscape brings us some pieces of information. Some general observations are given below, and these results are also used in the next sections to identify relations between algorithm efficiency and landscapes properties.

- **NK landscapes:** The autocorrelation function and δ -ruggedness are strongly correlated with the values of K. On the less rugged landscapes, the average distance between local optima is low. This is particularly visible when we compare the values with the average distance between random solutions.

UBQP n d	$\rho(1)$	1-rug.	$\rho(l)$		δ -rug.		$d_{LO}(d_{Rd})$	neut.
			≤ 0.8	≤ 0.5	≥ 0.1	≥ 0.25		
128 25	0.9798	2.3%	7.8%	27.3%	5.5%	27.6%	0.098%	26.3 (64)
128 50	0.9782	3.1%	8.6%	25.8%	4.7%	27.6%	0.060%	24.2 (64)
128 75	0.9791	3.6%	8.6%	25.8%	5.5%	30.7%	0.046%	21.5 (64)
128 100	0.9779	4.3%	8.6%	26.1%	5.5%	29.9%	0.041%	23.6 (64)
256 25	0.9894	1.5%	8.6%	26.6%	5.1%	28.6%	0.060%	52.2 (128)
256 50	0.9899	2.1%	8.6%	27.7%	5.5%	29.8%	0.037%	28.5 (128)
256 75	0.9895	2.7%	9.0%	27.3%	5.5%	31.8%	0.032%	30.8 (128)
256 100	0.9880	2.8%	8.6%	26.6%	5.1%	29.0%	0.027%	42.0 (128)
512 10	0.9946	0.6%	8.4%	27.2%	5.5%	32.5%	0.064%	59.4 (256)
512 50	0.9942	1.5%	8.2%	27.5%	5.1%	29.5%	0.028%	63.5 (256)
512 75	0.9940	1.8%	8.6%	26.6%	4.9%	29.3%	0.022%	59.3 (256)
512 100	0.9949	2.3%	8.6%	29.3%	5.5%	32.5%	0.019%	59.9 (256)
1024 10	0.9990	0.6%	8.2%	26.0%	5.2%	30.5%	0.046%	156.1 (512)
1024 50	0.9973	1.0%	8.4%	27.0%	4.9%	27.9%	0.019%	157.6 (512)
1024 75	0.9969	1.1%	8.3%	27.8%	4.9%	28.1%	0.019%	130.1 (512)
1024 100	0.9973	1.5%	8.0%	25.9%	5.2%	28.0%	0.015%	133.2 (512)
2048 10	0.9984	0.3%	8.5%	26.4%	5.2%	29.5%	0.038%	269.7 (1024)
2048 25	0.9985	0.7%	8.0%	27.6%	5.0%	27.6%	0.022%	230.7 (1024)
2048 50	0.9987	0.9%	8.5%	26.6%	5.0%	29.2%	0.013%	238.1 (1024)
2048 100	0.9990	1.0%	8.8%	27.7%	4.8%	27.7%	0.011%	307.2 (1024)
4096 10	0.9994	0.2%	8.8%	28.2%	4.7%	28.8%	0.027%	536.1 (2048)
4096 25	0.9992	0.5%	8.5%	26.3%	5.0%	28.9%	0.013%	512.0 (2048)
4096 50	0.9989	0.5%	8.5%	27.2%	4.7%	28.5%	0.010%	483.5 (2048)
4096 100	0.9983	0.7%	8.2%	26.7%	5.7%	30.0%	0.007%	486.6 (2048)

Table 3 UBQP landscapes indicator values. INVERSER LES VALEURS DES DEUX DERNIERES COLONNES.

Inst.		$\rho(1)$	1-rug.	neut.	moy(dist) Rd	moy(dist) LO
FSP	30 10	0.9343	9.99%	12.1%	21.6	20.7
	30 15	0.9181	10.59%	9.1%	21.6	19.0
	30 20	0.9152	11.47%	5.7%	21.6	19.5
	50 10	0.9526	7.63%	13.5%	38.7	37.3
	50 15	0.9504	8.37%	9.0%	38.7	35.8
	50 20	0.9478	9.06%	7.2%	38.7	35.7

Table 4 FSP landscapes indicator values. FUSIONNER 2 DER COL et INVERSER AVEC LA NEUT.

Inst.		$\rho(1)$	1-rug.	neut.	moy(dist) Rd	moy(dist) LO
QAP	lipa70a	0.9430	5.16%	0.7%	65.2	65.0
	lipa80a	0.9508	4.54%	0.6%	75.0	74.9
	lipa90a	0.9549	4.09%	0.6%	84.9	84.8
	sko42	0.9136	9.61%	1.0%	37.7	36.6
	sko49	0.9245	8.78%	0.9%	44.5	43.5
	sko56	0.9339	8.04%	0.7%	51.4	50.1
	sko64	0.9412	7.50%	0.7%	59.2	58.0
	sko72	0.9522	6.90%	0.6%	67.1	65.7

Table 5 QAP landscapes indicator values.FUSIONNER 2 DER COL et INVERSER AVEC LA NEUT.

NK landscapes contains no significant neutrality (note that some modified NK landscapes with neutrality are proposed in the literature).

- **UBQP landscapes:** A COMPLETER 1rug/drug != auto , dist OL
- **FSP landscapes:** A COMPLETER rug, neut_i=*l*-M
- **QAP landscapes:** A COMPLETER QAP: rug, neut

5 Evaluating PNLs parameterization

The experiments proposed seek to compare five algorithms: three PNLs (SW, ID_{best}, and ID_{any} and two state-of-the-art algorithms (ILS and TS).

The local searches considered require a single parameter (excluding the stop criterion), and their mode of operation is reduced to elementary pivot rules. Their simplicity makes it possible to understand better the differences in behavior induced through experimentation. The small number of parameters of each variant makes it possible to observe better the differences caused by their variation. The main parameter of each variant mostly influences the balance between intensification and diversification during the search.

5.1 Experimental protocol

In this section, we first describe which value has been used for parameterization analysis. Then, we describe how the experimental study will be performed in the rest of the paper.

5.1.1 Parameter values tested

ILS_F uses the first improvement pivoting rule and performs \mathcal{M} random moves when a local optimum is reached. The first improvement pivoting rule is widely used and leads toward good quality local optima in a significantly reduced number of evaluations compared to other pivoting rules such as the best improvement pivoting rule.

For each problem, we performed a set of experiments, using different \mathcal{M} values. The most effective parameterization has been used for comparison between algorithms. The values tested are reported in table 6.

Since ILS_F uses the first improvement rule, the entire neighborhood is not systematically evaluated. Although ILS_F partially evaluate the neighbourhood at each step of the search, it can not be included in PNLs. Indeed, SW and ID_{walk} systematically reduce the maximum size of the neighbourhood via the λ parameter (unless $\lambda = |\mathcal{N}|$), which is not the case for ILS_F. During the descent phases, ILS_F can evaluate the entire neighborhood, and the size of the sample to be evaluated cannot be determined *a priori*.

TS uses a list of bits of size \mathcal{L} , which is sufficient to ensure a minimal distance between solutions along the walk. Indeed, we use a landscape-oriented

Problem	\mathcal{L} values tested	\mathcal{M} values tested
NK	{5, 10, 15, 20}	{5, 10, 15, 20}
UBQP	{5, 10, 15, 20, 30, 40, 50, 60}	{5, 10, 15, 20, 30, 40, 50, 60, 70, 80}
QAP	{8, 12, 16, 20, 24, 28, 32}	{5, 10, 15, 20, 30, 40, 50, 60, 70, 80}
FSP	{4, 6, 8, 10, 12, 14, 16, 18}	{5, 10, 15, 20, 30, 40, 50, 60, 70, 80}

Table 6 Values tested for \mathcal{L} (tabu list size in TS) and \mathcal{M} (number of perturbations in ILS).

approach, and it implies to use a distance-based list rather than a solution-based list. The pivoting rule used is based upon the best improvement pivoting rule, which is the most commonly used within tabu searches.

TS does not evaluate the solutions (or moves) in the tabu list, therefore, selects a solution in a reduced neighborhood of the current solution. Even if TS evaluates a fixed number of neighbors at each step (except when the tabu list does not yet contain \mathcal{L} moves), the solutions not to be evaluated are determined by the last movements made, not by a stochastic process.

As realized for ILS_F , the most effective parameterization has been used for comparison between algorithms (see table 6).

Each PNLs tested here requires only one parameter (λ). For each method, we consider several λ values to study their influence on their dynamics. Ideally, all possible values should be examined, but this would lead to a particularly high number of executions. Although automatic parameterization methods exist (for example *i-race* [12]), we have chosen not to use them in this study: they are often effective in leading the search for good solutions without considering the constraints of parameter setting, but determining an appropriate parameter value in this way would limit our observations.

Indeed, even if a given parameter value allows the search to reach good solutions quickly, it can prevent a longer process from finding better solutions (often due to an inadequate balance between intensification and diversification). Also, this automatic setting could be seen as an additional mechanism for local searches.

It should be remembered that we are seeking to obtain a better understanding of the impact of different components of PNLs and that the more numerous they are, the more complicated their analysis is.

The first part of experiments reported in the next section is then dedicated to analyse the impact of λ values on PNLs efficiency.

5.1.2 Stop criterion

The criteria is determined by a maximum number of solution evaluations. This number is set arbitrary at 100 million in our experiments, a deliberately high budget so that a sufficient convergence can be observed on the different landscapes. The best fitness achieved for a reduced budget is also extracted from the information collected during the execution of the methods. In this work, we observe the behavior of the five algorithms after 1 million and 100 million evaluations.

5.1.3 Experimental setup

For each triplet (landscape, method, λ), 100 runs were performed from the same initial set of 100 randomly generated solutions to reduce the stochastic bias. We recall that several parameter values are tested for each method, inducing a too high number of runs to report all results. We thus only report the best average obtained from 100 runs for each couple (landscape, method) for 1 million and 100 million of evaluations.

Doing so allow us to perform a statistical analysis to compare methods. In the following, we consider that a method A statistically dominates a method B on a set of 100 runs when $\frac{1}{2^{100}} \sum_{i=0}^{\mathcal{S}} \binom{100}{\mathcal{S}} \geq 0.95$, ie. $\mathcal{S} \geq 58$ w.r.t. a binomial test (with a p -value greater than 0.95). For each instance, we also report the difference (in %) between each method and the best average fitness obtained.

5.2 PNLs Parameter sensitivity analysis

We test several λ parameter values for each PNLs (i.e. SW, ID_{best}, and ID_{any}). We first describe how we determine the values tested. Then we report visualization of the parameter sensitivity through figures reporting tradeoffs between algorithm efficiency and λ values. Last, we summarize and discuss the best λ values determined by our experiments.

5.2.1 Panel of λ values

Binary landscapes: we consider all powers of 2 between 1 and the size of the neighborhood ($\lambda = 2^k, k \in \{1, 2, \dots, \log_2(|\mathcal{N}|)\}$). On NK landscapes, we additionally consider values of λ in steps of 4 between the two tested values of λ that frame the value leading to the best average fitness, to more accurately determine the relevant parameter values according to roughness. For example, if $\lambda = 32$ leads to the best fitness average, we consider all intermediate values in steps of 4 between 16 and 64 (excluded). We consider the intermediate values in steps of 8 on landscapes where $N = 1024$.

Permutation landscapes: the values of λ used correspond to different ratios of the size of the neighborhood, as follow : $\lambda \in \left\{ \frac{|\mathcal{N}|16}{|\mathcal{N}|}, \frac{|\mathcal{N}|15}{|\mathcal{N}|}, \dots, \frac{|\mathcal{N}|2}{|\mathcal{N}|} \right\}$.

5.2.2 Parameter sensitivity analysis

Considering several parameter values makes it possible to compare local searches more fairly, limiting the risk that one of them is more efficient because it is better configured. It also allows to observe the evolution of the capacity of a given local search to reach good solutions according to the setting. This analysis could help us to determine the degree of sensitivity of methods to parameterization. It can also provide guidelines to design useful automatic parameterization based on landscapes.

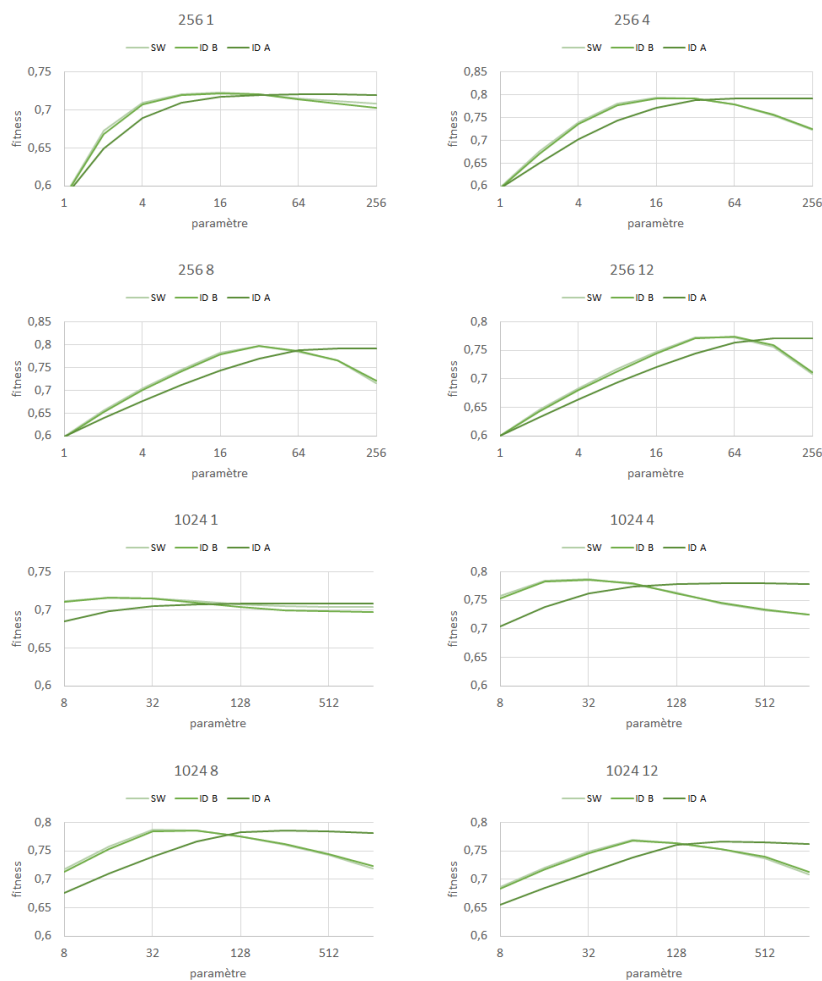


Fig. 2 Fitness variation during executions with different λ values: NK landscapes with $N = 1024$ and $N = 1024$.

For the algorithms studied here, observing this evolution provides information on how to set an appropriate balance between intensification and diversification on a given landscape. This section presents and discusses the evolution of the average fitness of methods according to their parameter setting on NK landscapes.

The figure 2 illustrates average fitness obtained with respect to the settings of SW , ID_{best} and ID_{any} on several NK landscapes during executions of 100 million evaluations. The values of λ reported here are powers of two, which excludes the values tested in a second step, although they are considered in the rest of the section. The fitness averages obtained by SW and ID_{best} evolve

similarly: their efficiency is very close, at least when the values of the (maximum) size of the neighborhood to be evaluated are powers of two. These two PNLs differ only in the pivotal rule applied when at least one improving neighbor is encountered. This similarity could explain that these two variants do not present significantly different dynamics on NK landscapes. Compared to SW and ID_{best} , ID_{any} behavior evolves differently depending on the value of λ . When this value is high, ID_{any} is more effective in finding good solutions. Remember that when ID_{any} does not find any improving solutions, it randomly selects one of the λ solutions evaluated, which leads to a greater diversification than the other two PNLs, and influences the most appropriate number of neighbors to evaluate. Indeed, selecting a deteriorating neighbor without maximizing its fitness implies a significant degradation of the quality of the current solution. After such a movement, the research process requires a higher rate of intensification and thus a low rate of diversification. Considering more neighbors increases the likelihood of meeting an improved neighbor, especially when the search reaches higher solutions in the fitness landscape, and then limits diversification rates, which encourages more search in good areas of the landscape. When epistasis is low, the range of λ values that lead to the best solutions on average is larger. For the three PNLs, a positive correlation exists between the evolution of the value zones leading to the best average fitness and the value of K (and therefore the roughness rates).

5.2.3 Summary of best sample sizes for PLNS

The parameter values that lead local search to the best average fitness on each landscape for 1 and 100 million evaluations are reported and discussed in the following. We note the best parameter value differently for each method : respectively λ_{SW} , λ_{IDb} , λ_{IDa} , \mathcal{M} , \mathcal{L} for SW, ID_{best} , ID_{any} , ID_{any} , ILS and tt TS.

NK landscapes: The table 7 presents the most appropriate λ values found on NK landscapes and 1 million evaluations. The statistical analysis of the results is not always significant for this experiment. Then, when several λ values lead to results that are not comparable statistically, the least value is reported.

In general, the most effective setting for SW and ID_{best} is similar on the considered landscape. For these two PNLs, the most appropriate number of neighbors to evaluate (at most) at each step increases with the roughness levels (K). This result suggests that SW and ID_{best} require a higher level of intensification on rougher landscapes.

The ideal number of neighbours to evaluate (λ_{IDa}) for ID_{any} is higher than for the other two PNLs, which is consistent with the higher diversification rates it induces.

The values of λ_{IDa} are more stable for $N \in \{512, 1024\}$ and generally increase with the epistasis factor (and therefore roughness), indicating a higher need for intensification on such landscapes.

N K	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	N K	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}
128 1	8	8	40	15	15	256 1	16	16	88	15	20
128 2	12	16	72	15	15	256 2	16	16	88	10	15
128 4	16	16	56	5	5	256 4	16	16	72	5	5
128 6	16	20	80	5	5	256 6	24	24	64	5	5
128 8	24	24	64	5	5	256 8	32	32	88	5	5
128 10	32	28	112	5	5	256 10	36	40	112	5	5
128 12	32	36	64	5	5	256 12	48	48	152	5	5
512 1	24	24	128	20	50	1024 1	24	24	128	15	50
512 2	24	24	128	15	50	1024 2	24	24	128	20	50
512 4	24	24	128	15	50	1024 4	24	24	128	20	50
512 6	24	32	128	10	40	1024 6	24	32	128	20	50
512 8	32	40	128	10	20	1024 8	40	40	128	15	40
512 10	48	48	128	5	15	1024 10	48	56	128	15	30
512 12	64	64	256	5	10	1024 12	56	64	256	10	20

Table 7 Parameter values that leads to bests average fitness. 100 executions of 1 million evaluations are done for each couple (landscape, method) on NK landscapes.

The most appropriate number of perturbations \mathcal{M} applied between the intensifying phases (descents) of ILS is higher on smooth landscapes. A greater number of random moves leads to a higher level of diversification in the search process. Performing more random moves allows the search to move away from the last local optima encountered and to be more likely to escape from their basin of attraction, allowing the search to explore new areas of the landscape. Like \mathcal{M} , the length of the tabu list that leads to the best averages is lower on rough landscapes. On smooth landscapes, \mathcal{L} increases with N : the most appropriate length of the tabu list increases with the size of NK landscapes. For these landscapes, the list contains tabu bits whose value cannot change. This ensures a distance of \mathcal{L} between two solutions separated by \mathcal{L} steps. A higher value implies a greater distance between the solutions encountered during the search, and thus a greater diversification.

For a budget of 100 million evaluations, the settings leading each method to the best average fitness are given in the table 8. The evolution of these settings according to ruggedness for SW and ID_{best} varies slightly between 1 million evaluations (table 7) and 100 million evaluations. Globally, the most appropriate values of λ remain almost unchanged when we consider 1 million or 100 million evaluations. The value slightly increases with the number of evaluations allowed to the search on large NK landscapes. This result is understandable since when more solution evaluation can be done, the diversifying effort can be increased. Some similar observations are usually reported in evolutionary algorithms while they consider the tradeoff between computation time allowed and selection pressure.

The most appropriate number of perturbations \mathcal{M} to be applied between descent phases for ILS decreases on most landscapes when the number of solution evaluations increases, except on very smooth landscapes (when $K = 1$). In general, it would seem that for this type of landscape, a high rate of diversification should be maintained. Perhaps such a diversification allows to

N K	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	N K	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}
128 1	8	8	16	5	20	256 1	16	16	96	15	20
128 2	12	16	40	10	20	256 2	24	24	96	5	20
128 4	16	16	40	5	20	256 4	20	20	192	5	20
128 6	16	16	56	5	15	256 6	24	24	184	5	20
128 8	20	20	72	5	15	256 8	32	32	112	5	15
128 10	24	32	120	5	10	256 10	36	40	184	5	15
128 12	28	36	96	5	10	256 12	44	52	184	5	15
512 1	16	16	256	20	50	1024 1	16	16	256	15	50
512 2	16	24	128	5	50	1024 2	24	24	256	5	50
512 4	24	24	128	5	50	1024 4	24	24	256	5	50
512 6	24	32	256	5	30	1024 6	32	32	256	5	50
512 8	40	40	256	5	30	1024 8	40	48	256	5	40
512 10	56	48	256	5	20	1024 10	56	64	256	10	30
512 12	64	64	256	5	15	1024 12	72	80	256	5	20

Table 8 Parameter values leading to the best average fitness over 100 executions for each pair (landscape, method) on NK landscapes and 100 million evaluations.

transit between huge basins of attraction. For rougher landscapes, the best value tested is generally the smallest considered ($\mathcal{M} = 5$): few disturbances are sufficient to increase the possibility of achieving new good optima (the number of local optima being higher, and their basin of attraction smaller). For a budget of 100 million evaluations, the settings leading each method to the best average fitness are given in the table 8. The evolution of these settings according to ruggedness for SW and ID_{best} varies slightly between 1 million evaluations (table 7) and 100 million evaluations. Globally, the most appropriate values of λ remain almost unchanged when we consider 1 million or 100 million evaluations. The value slightly increases with the number of evaluations allowed to the search on large NK landscapes. This result is understandable since when more solution evaluation can be performed, the diversifying effort can be increased. Some similar observations are usually reported in evolutionary algorithms while they consider the tradeoff between computation time allowed and selection pressure.

The value of the tabu list size \mathcal{L} that leads to the best results is generally higher for a budget of 100 million evaluations. Also, a higher value also seems more appropriate on smooth landscapes. Tabu search requires more diversification on smooth landscapes to achieve good results. This result is probably due to the larger size of the attraction areas. More diversification is also needed when the search is longer because intensifying deeply in a restricted area limits the exploration of the landscape and then does not likely lead to better solutions.

UBQP landscapes: The best settings for each local search are presented in the table 9. With a budget of 1 million evaluations, average values are generally stable for most landscapes. Only λ_{SW} fluctuates over three landscapes. On other landscapes, SW and ID_{best} systematically maximize their performance on similar settings.

		UBQP		Parameter							Parameter				
		Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}		
1 m.	2048 10	128	128	256	50	60	4096 10	64	128	256	50	60			
	2048 25	128	128	256	50	60	4096 25	128	128	256	50	60			
	2048 50	128	128	256	50	60	4096 50	128	128	256	50	60			
	2048 100	64	128	256	50	60	4096 100	64	128	256	50	60			
100 m.	2048 10	128	128	128	50	40	4096 10	128	128	256	50	60			
	2048 25	128	128	128	50	30	4096 25	128	128	256	50	50			
	2048 50	128	128	128	50	30	4096 50	128	128	256	50	50			
	2048 100	128	128	128	50	40	4096 100	128	128	256	50	50			

Table 9 Parameter values leading to the best average fitness over 100 executions for each pair (landscape, method) on UBQP landscapes for a budget of 1 million (top) and 100 million (bottom) evaluations.

For ILS, the most appropriate number of perturbations to be applied after each descent is systematically high. This result is partly related to the large size of the landscapes and their smooth structure, where applying a large number of perturbations seems to be a good strategy to escape from the (probably large) basins of attraction of the local optima encountered. This analysis also applies to the length of the tabu list \mathcal{L} , which need to be large enough to ensure a sufficient diversification during the search.

When we consider a budget of 100 million evaluations, the best settings remain stable for each method on every UBQP landscape. On the contrary, the most appropriate parameter values of TS vary significantly between 1 million and 100 million evaluation. The \mathcal{L} length of the list seems to depend partially on the size of the landscape and is smaller for 100 million evaluations, leading to a high level of intensification during the search.

QAP landscapes: The results are presented in table 10. For a budget of 1 million evaluations on `lipa` landscapes, the sample size of neighbors that allow SW and ID_{best} to obtain their best average fitness are approximatively equivalent. For ID_{best} , this size is generally larger, which can be explained by the selection of the first improved neighbor encountered, which leads to a less intensifying search than SW which select the best-encountered neighbor (considering a fixed sample size value). On landscapes `sko`, the sample of neighbours that leads to best results of SW and ID_{best} is generally smaller. The value of λ that maximizes their performance is not only related to the size of the landscapes.

As with the previous classes of landscapes tested, ID_{any} usually requires a very large sample of neighbors to obtain its best average fitness. On some landscapes, mainly `lipa`, the sample corresponds to the entire neighborhood of the current solution, indicating that ID_{any} requires a strong intensification effort to be effective. In such cases, this PNLs corresponds to ILS, which applies a random move to escape local optima. The pivoting rule applied to deteriorating neighbors affects the balance between intensification and diversification, so that the sample size needs to be very large for ID_{any} to be effective.

		QAP		Parameter					QAP		Parameter				
		Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}		
1 m.	lipa70a		345	345	1207	5	8	sko49	106	106	1176	5	16		
	lipa80a		451	526	3106	5	8	sko56	8	118	770	5	24		
	lipa90a		445	572	4005	5	8	sko64	8	134	672	5	24		
	ska42		78	86	861	5	20	ska72	2	170	852	5	32		
		Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}		
100 m.	lipa70a		8	185	1207	5	24	ska49	130	147	1176	5	16		
	lipa80a		8	243	3160	5	12	ska56	32	128	1540	5	24		
	lipa90a		16	250	4005	5	24	ska64	8	224	2016	5	24		
	ska42		78	128	861	5	16	ska72	2	170	2556	5	24		
		Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}		

Table 10 Parameter values leading to the best average fitness over 100 executions for each pair (landscape, method) on QAP landscapes (1 and 100 million evaluations).

The ideal number of perturbations among those considered for ILS is systematically $\mathcal{M} = 5$, which indicates that the diversification provided by this mechanism needs to be limited. Lower values should be considered and may be more appropriate to maximize the effectiveness of this method. The most suitable length of the tabu list \mathcal{L} is smaller on **lipa** landscapes, which leads to a higher intensification than on **ska** landscapes. The length of the list increases accordingly with the size of the landscapes.

For a budget of 100 million evaluations, SW and ID_{best} require a smaller λ value than for 1 million evaluations. This observation is particularly visible on **lipa** landscapes. The value of λ for ID_{best} does not change for landscapes **lipa** but increases for **ska**, indicating a stronger need for intensification to achieve good solutions on these landscapes. The most appropriate sample size value for ID_{any} does not evolve among **lipa** landscapes with this larger number of evaluations. This size increases on larger **ska** landscapes, indicating a limited need for diversification to conduct the search towards good solutions.

The most appropriate number of perturbations among values tested for ILS is stable ($\mathcal{M} = 5$). Compared to previous values, the length of the tabu list leading to the best solutions on average increases for **lipa** landscapes and remains relatively stable on **ska** landscapes. Increasing the diversification of tabu search leads to better solutions when the number of evaluations is higher on **lipa** landscapes.

FSP landscapes: The best settings obtained are reported in table 11. With a budget of 1 million evaluations, the values of λ_{SW} and λ_{IDb} leading to the best average fitness are close, with λ_{IDb} consistently higher. The maximum sample size to be evaluated is positively correlated with the number of machines, and thus negatively correlated with the neutrality rates. In landscapes with a significant level of neutrality, a small λ value is sufficient to allow quick navigation in plateaus, each step requiring only a few solution evaluations.

As already observed on other landscapes, ID_{any} requires a larger neighborhood sample than the two other PNLs. Likely to ID_{any} and SW, this value is negatively correlated with the neutrality rate. The iterated local search systematically requires the lowest number of perturbations tested in the experi-

	FSP		Parameter					FSP		Parameter				
	Inst.		λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.		λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}
1 m.	30	10 01	58	54	217	5	10	50	10 01	153	153	408	5	18
	30	15 01	96	108	435	5	6	50	15 01	153	163	612	5	16
	30	20 01	87	108	435	5	6	50	20 01	188	175	1225	5	14
100 m.	Inst.		λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.		λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}
	30	10 01	145	145	435	5	12	50	10 01	188	153	350	5	16
	30	15 01	145	145	435	5	12	50	15 01	272	245	2450	5	12
	30	20 01	174	145	435	5	12	50	20 01	408	350	2450	5	12

Table 11 Parameter values leading to the best average fitness over 100 executions for each pair (landscape, method) on FSP landscapes. Results for 1 million (top) and 100 million (bottom) evaluations).

ments. The most appropriate length of the tabu list decreases with the number of machines, meaning that TS requires more diversification when landscape neutrality is high. Forcing a greater distance between solutions by increasing the length of the \mathcal{L} tabu list limits the risk of cycling during the search, but increases the probability that the search will get stuck on plateaus.

The values of the parameter λ of the methods ID_{Sw} and ID_{best} leading to the best average fitness are generally higher considering 100 million evaluations than while considering 1 million. With this large budget, the value of λ is globally stable according to the different numbers of machines m for instances of size $j = 30$. When $j = 50$, the value λ increases with the number of machines. This different evolution is probably due to the larger size of the landscapes, which makes the search convergence slower. To achieve better solutions, on average, with 100 million evaluations, ID_{any} requires a sample size greater or equal to that used when the budget equals to 1 million evaluations. As with the other two PNLs, the sample size is stable when $j = 30$, and also when $j = 50$.

The results of this section show that among the PNLs, SW and ID_{best} often require similar settings to maximize their effectiveness. The ideal size of the neighborhood sample seems to be partly correlated with ruggedness. ID_{any} requires a larger sample size than the other two PNLs, which is related to the higher diversification induced by the random selection of deteriorating neighbors. In general, there are slight variations in the ideal settings for the two budgets studied. The convergence levels at which the results are examined mostly vary according to the landscape properties and especially its size and ruggedness.

6 Empirical comparison of PNLs

This section is dedicated to compare PNL algorithms with ILS_F and TS. Each method is executed using the parameter value found in the experiments reported in the previous section. We also use the experimental protocol described in the previous section.

*****AJOUTER LEGENDE DOM STAT**** (cf these)

N K	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	<i>moy_{max}</i>	SW	ID _B	ID _A	TS	ILS
128 1	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7245	0.04	0.06	0.01	1.35	0.00
128 2	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7415	0.00	0.05	0.38	1.20	0.33
128 4	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7922	0.03	0.10	0.00	4.94	0.13
128 6	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7915	0.00	0.12	0.36	4.97	0.62
128 8	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7883	0.00	0.08	0.58	4.05	1.19
128 10	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7796	0.00	0.16	0.56	2.75	1.36
128 12	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7689	0.00	0.27	1.03	2.13	1.63
256 1	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7210	0.00	0.04	0.55	1.28	0.49
256 2	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7430	0.00	0.08	0.68	3.00	0.68
256 4	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7877	0.00	0.14	0.40	4.91	0.49
256 6	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7921	0.05	0.00	0.68	2.33	0.58
256 8	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7819	0.00	0.23	0.47	1.22	0.75
256 10	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7741	0.00	0.33	0.63	0.65	1.07
256 12	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7633	0.00	0.10	0.56	0.57	1.11
512 1	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7044	0.00	0.08	0.59	0.52	0.51
512 2	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7448	0.00	0.05	0.71	1.92	0.64
512 4	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7751	0.00	0.12	0.84	1.31	0.72
512 6	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7825	0.00	0.22	0.75	1.37	0.80
512 8	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7781	0.00	0.36	0.76	1.18	0.86
512 10	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7684	0.00	0.24	0.55	0.82	0.62
512 12	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7581	0.00	0.05	2.89	0.59	0.38
1024 1	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7127	0.00	0.12	1.10	1.07	1.09
1024 2	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7474	0.00	0.13	1.29	2.70	1.19
1024 4	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7749	0.00	0.24	1.28	3.22	1.33
1024 6	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7762	0.00	0.23	0.90	2.57	0.04
1024 8	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7711	0.00	0.22	0.75	2.15	1.39
1024 10	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7625	0.00	0.26	0.55	1.73	0.67
1024 12	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	0.7535	0.00	0.15	0.64	1.41	0.93

Table 12 Comparison of three PNLs variants, TS and ILS with a budget of 1 million evaluations on NK landscapes. Left: statistical dominances between methods; right: best average fitness achieved and deviations to this fitness for each method (in %).

6.0.1 Binary landscapes

We present the statistical dominance relations between the local search variants and their average fitness for 1 million evaluations on NK landscapes in the table 12. SW almost systematically obtains the best average fitness on these landscapes and dominates other local searches in most cases. Although ID_{best} is to SW in its behavior and ability to find good solutions, it is regularly dominated by the latter. ID_{any} is particularly efficient on two landscapes of small size ($N = 128$). On other landscapes, it is systematically dominated by the other PNLs. Nevertheless, ID_{any} regularly dominates at least TS or ILS_F. TS is the least effective local search most of the time, even if it dominates ILS on a few landscapes. ILS often statistically dominates TS but rarely the PNLs variants. For instance, ILS_F is the most efficient local search on the easiest NK landscape in our sample ($N = 128$, $K = 1$), where it systematically achieves the best local optimum encountered, which is not the case for PNLs variants.

The results for a budget of 100 million evaluations are presented in table 13. The ability of SW and ID_{best} to achieve good solutions on average is more similar than after 1 million evaluations. In some cases, SW is dominated by ID_{best} or at least ID_{best} leads to better solutions on average, especially on the smallest landscapes. As landscape size increases, SW is less often dominated by ID_{best}. Interestingly, on the largest landscapes, ID_{best} is systematically domi-

nated by *SW*. This result is probably due to the fixed budget for each method. Indeed, considering the results for a lower budget (where *SW* dominates ID_{best}) in most cases, it seems that despite a similar behavior of the two PNLSSs, *SW* reaches good solutions more quickly (on average). This method converges towards good solutions more quickly on NK landscapes. With a large number of evaluations with respect to the size of the treated landscape, *SW* and ID_{best} seem to converge towards solutions of close quality. Remember that these two methods mostly vary in their move policy, only when selecting improving moves: *SW* select the best-improving neighbor from the sample, while ID_{best} select the first encountered improving neighbor of the sample. Although the effectiveness of these rules is correlated with the roughness on NK landscapes for descents, there does not seem to be a correlation between the comparative efficiency of these two partial neighborhood-based methods and the roughness levels of landscapes. It is possible that the often slightly larger most appropriate sample size of the neighborhood for ID_{best} counterbalance the differences induced by its selection strategy, which is naturally less intensifying than *SW*. For *SW* and ID_{best} , the strategy for selecting deteriorating neighbors is to choose the best from the sample, which results in a reduced rate of diversification compared to other methods. The possibility of selecting deteriorating neighbours at each step of the search could partially explain the change in the relative effectiveness of these strategies for PNLSSs compared to other local search based on descents, such as *TS* and *ILS*. For *SW* and ID_{best} , the diversification rate depends mainly on the sample size, which tends to increase with K . We present the statistical dominance relations between the local search variants and their average fitness for 1 million evaluations on NK landscapes in the table 12. *SW* almost systematically obtains the best average fitness on these landscapes and dominates other local searches in most cases. Although ID_{best} is to *SW* in its behavior and ability to find good solutions, it is regularly dominated by the latter. ID_{any} is particularly efficient on two landscapes of small size ($N = 128$). On other landscapes, it is systematically dominated by the other PNLSSs. Nevertheless, ID_{any} regularly dominates at least *TS* or *ILS*. *TS* is the least effective local search most of the time, even if it dominates *ILS* on a few landscapes. *ILS* often statistically dominates *TS* but rarely the PNLSS variants. For instance, *ILS* is the most efficient local search on the easiest NK landscape in our sample ($N = 128$, $K = 1$), where it systematically achieves the best local optimum encountered, which is not the case for PNLSS variants.

A higher budget does not significantly increase the effectiveness of ID_{any} compared to other local searches. It is the least efficient variant among the PNLSSs, which must be due to the excessive diversification induced by the strategy while selecting deteriorating neighbors (i.e., randomly). On NK landscapes, convergence seem particularly slow and could be a consequence of a too low intensification around areas of interest of the landscapes. Indeed, in such areas, only a few improving movements are generally possible, and ID_{any} would tend to make too many random movements, moving away from these

N K	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	<i>moy_{max}</i>	SW	ID _B	ID _A	TS	ILS
128 1											0.7245	0.00	0.00	0.00	1.10	0.00
128 2											0.7424	0.00	0.00	0.04	0.74	0.02
128 4											0.7959	0.00	0.00	0.00	0.08	0.00
128 6											0.8004	0.00	0.01	0.11	0.67	0.12
128 8											0.8021	0.00	0.06	0.45	1.22	0.89
128 10											0.7936	0.00	0.16	0.49	1.36	1.11
128 12											0.7819	0.00	0.03	0.44	1.15	1.22
256 1											0.7220	0.01	0.00	0.29	1.41	0.27
256 2											0.7444	0.04	0.00	0.24	2.61	0.25
256 4											0.7933	0.00	0.00	0.92	1.39	0.20
256 6											0.8045	0.14	0.00	1.52	0.31	0.47
256 8											0.7961	0.12	0.00	0.57	0.83	0.86
256 10											0.7862	0.00	0.02	2.49	1.05	1.01
256 12											0.7754	0.02	0.00	2.56	1.25	1.17
512 1											0.7078	0.00	0.03	0.57	1.00	0.54
512 2											0.7509	0.00	0.00	0.77	2.57	0.74
512 4											0.7860	0.00	0.04	0.75	0.19	0.69
512 6											0.7989	0.00	0.06	0.57	0.30	0.61
512 8											0.7939	0.00	0.05	0.56	1.13	0.67
512 10											0.7829	0.00	0.05	0.49	0.86	0.62
512 12											0.7720	0.00	0.01	0.49	0.80	0.64
1024 1											0.7163	0.00	0.03	1.11	1.56	1.05
1024 2											0.7522	0.00	0.01	1.26	3.30	1.26
1024 4											0.7878	0.00	0.07	0.99	2.84	1.03
1024 6											0.7949	0.00	0.07	0.70	0.63	0.74
1024 8											0.7901	0.00	0.17	0.54	0.65	0.65
1024 10											0.7793	0.00	0.09	0.46	0.68	0.52
1024 12											0.7694	0.00	0.06	0.39	0.53	0.49

Table 13 Comparison of three PNLs variants, TS and ILS_F with a budget of 100 million evaluations on NK landscapes.

promising areas. Despite this, this characteristic allows ID_{any} to be effective in so-called easier instances.

TS is often statistically dominated, and leads to the lowest average fitness values. It is also the only local search tested that does not systematically lead to the best local optimum encountered for the easiest landscape. This local search is the least stochastic, which naturally leads to less diversification, even with a longer tabu list. It may become trapped more often in areas of the landscape of low interest. ILS_F tends to be more efficient on smooth landscapes than on rough landscapes where it is dominated many times. This result could be due partially to its parameter configuration: on rougher NK landscapes, the value chosen corresponds to the lowest tested. Since the parameter directly controls the number of random moves to be applied to escape a local optimum, this value could be too high for rough landscapes. It would drive too far away from previously discovered promising areas.

We provide the statistical dominances between the methods as well as the averages fitness obtained on UBQP landscapes in the table 14. For a budget of 1 million evaluations, ILS_F leads globally to the best solutions on landscapes of size $N = 2048$. On larger landscapes ($n = 4096$), the most effective method is ID_{best}. At least one of these two aforementioned local searches systematically dominates SW. On these landscapes, when the density parameter is minimal or maximal, the ability to achieve good solutions of SW is equivalent to that

UBQP		SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	<i>moy_{max}</i>	SW	ID _B	ID _A	TS	ILS
1 m.	2048 10											1002458.6	0.06	0.03	0.21	24.30	0.00
	2048 25											1638960.1	0.05	0.00	0.14	24.59	0.00
	2048 50											2394696.1	0.05	0.03	0.18	24.37	0.00
	2048 100											3090605.8	0.07	0.00	0.21	24.11	0.02
	4096 10											2799531.5	0.05	0.00	0.15	72.81	0.07
	4096 25											4580315.7	0.24	0.00	0.12	71.54	0.03
	4096 50											6510686.5	0.21	0.00	0.11	71.39	0.08
4096 100											9064548.5	0.05	0.00	0.13	71.76	0.06	
100 m.	2048 10											1004293.5	0.03	0.02	0.05	0.00	0.00
	2048 25											1641192.6	0.02	0.02	0.05	0.00	0.00
	2048 50											2398443.3	0.03	0.03	0.05	0.00	0.01
	2048 100											3099318.7	0.04	0.03	0.07	0.00	0.02
	4096 10											2808263.7	0.07	0.03	0.05	0.00	0.02
	4096 25											4595741.7	0.06	0.06	0.05	0.00	0.05
	4096 50											6527995.0	0.06	0.05	0.06	0.00	0.04
4096 100											9093039.3	0.07	0.07	0.07	0.00	0.06	

Table 14 Comparison of three PNLs variants, TS and ILS_F with a budget of 1 and 100 million evaluations on UBQP landscapes.

of ILS. On the other two landscapes of this size, SW is less effective and is statistically dominated by ID_{any}.

Considering a budget of 1 million evaluations, TS is particularly ineffective in achieving good solutions for UBQP. Let us notice that in our study, UBQP landscapes instances are significantly larger than NK landscapes instances. The low quality of the solutions returned by TS is probably due to the budget limited to 1 million evaluations, certainly too small compared to the size of the landscapes. TS is the local search that evaluates the largest number of neighbors at each step, as it never randomly determines the sample of the neighborhood to be evaluated. As a result, TS perform fewer steps than other local searches for the same budget and converges more slowly towards good solutions. Indeed, when the budget is set at 100 million evaluations, TS is the search that achieves the best solutions on the largest landscapes.

Considering PNLs variants, ID_{best} tends to outperforms SW which tends to outperforms ID_{any}, for 1 million evaluations as well as for 100 million evaluations.

***** Parler rugosite locale/globale UBQP *****
 ***** ajouter paragraphe + tableaux/figures *****

6.1 Permutation landscapes

We present the statistical dominances between the proposed local searches and their average fitness on FSP landscapes in the table 15. For a budget of 1 million evaluations, SW and ID_{best} lead to solutions of equivalent quality, with a slight advantage for ID_{best}. ID_{any} is generally the least effective PNLs variant, except on the largest landscape with the highest rate of neutrality ($j = 50$, $m = 10$). Also, when $j = 50$, ID_{any} leads to statistically better solutions than ILS and TS. TS regularly dominates ILS, which corresponds to

		FSP	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	<i>moy_{min}</i>	SW	ID _B	ID _A	TS	ILS
1 m.	30 10												1994.6	0.00	0.01	0.14	0.28	0.64
	30 15												2419.6	0.04	0.00	0.34	0.43	0.50
	30 20												2750.3	0.05	0.00	0.35	0.50	0.79
	50 10												3033.7	0.29	0.24	0.00	0.96	0.68
	50 15												3400.1	0.02	0.00	0.16	0.84	1.00
	50 20												3903.9	0.01	0.00	0.26	0.68	1.00
100 m.	30 10												1985.4	0.00	0.01	0.05	0.28	0.34
	30 15												2400.1	0.00	0.00	0.21	0.67	0.47
	30 20												2742.9	0.00	0.00	0.15	0.39	0.34
	50 10												3025.0	0.19	0.18	0.00	0.19	0.34
	50 15												3357.2	0.00	0.03	0.57	0.53	1.52
	50 20												3879.5	0.11	0.00	0.20	0.30	0.95

Table 15 Comparison of three PNLS variants, TS and ILS with a budget of 1 and 100 million evaluations on FSP landscapes.

local research leading to the worst solutions on these landscapes. Anyway, the main information is the global superiority of SW and ID_{best} variants over the other competing local searches.

The relative effectiveness of methods does not change significantly after 100 million evaluations. SW and ID_{best} are never dominated and are therefore not statistically comparable with each other. ID_{any} is the worst PNLS variant, but it regularly dominates SW and ILS, especially over the smallest landscapes considered. When $j = 50$, the ability of TS to achieve good solutions is close to that of ID_{any}. The three PNLS variants systematically dominate ILS. This result could be resulting from the significant rate of neutrality that prevents strict descents of ILS from reaching good solutions. On these landscapes containing plateaus, PNLSs are probably less constrained by neutrality, which would partially explain their greater effectiveness. Unlike ILS, if a non-strict local optimum is met, a neutral neighbor can be selected with PNLS variants. This case probably occurs more regularly for SW and ID_{best} than for ID_{any} due to their deteriorating neighbor selection strategy. This observation would contribute in part to the improved capacity of SW and ID_{best} to achieve good solutions.

The table 16 shows the statistical dominance between the local searches proposed and their average fitness attained over QAP landscapes. Besides, we give the global optima of the *lipa* instances, as well as the best known local optima for *sko* the instances in the table 17. This table also indicates, for each couple (landscape, method), the number of times the best fitness is met. On *lipa* landscapes and 1 million evaluations, SW and ID_{best} statistically dominate the other methods. TS dominates ID_{any}, which sometimes dominates ILS. On *sko* landscapes, ID_{any} allows reaching better solutions than the other local search variants. On the smallest landscapes, the efficiency of ID_{best} and ID_{any} is similar. On the largest landscapes, ILS is the second most effective local search (outperformed by ID_{any}). Finally, TS is the least effective on landscapes *sko*.

QAP		SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A	TS	ILS	<i>moy_{min}</i>	SW	ID _B	ID _A	TS	ILS
1 m.	lipa70a	■	■	■	■	■	■	■	■	■	■	170906.5	0.00	0.00	0.08	0.04	0.09
	lipa80a	■	■	■	■	■	■	■	■	■	■	254767.7	0.00	0.02	0.05	0.03	0.07
	lipa90a	■	■	■	■	■	■	■	■	■	■	362688.1	0.02	0.00	0.06	0.05	0.07
	sko42	■	■	■	■	■	■	■	■	■	■	15856.6	0.08	0.00	0.00	0.33	0.18
	sko49	■	■	■	■	■	■	■	■	■	■	23459.5	0.07	0.00	0.05	0.36	0.16
	sko56	■	■	■	■	■	■	■	■	■	■	34616.6	0.17	0.13	0.00	0.51	0.07
	sko64	■	■	■	■	■	■	■	■	■	■	48730.8	0.15	0.08	0.00	0.61	0.01
sko72	■	■	■	■	■	■	■	■	■	■	66614.0	0.14	0.10	0.00	0.77	0.02	
100 m.	lipa70a	■	■	■	■	■	■	■	■	■	■	169755.0	0.00	0.00	0.12	0.15	0.73
	lipa80a	■	■	■	■	■	■	■	■	■	■	253429.6	0.02	0.00	0.50	0.33	0.57
	lipa90a	■	■	■	■	■	■	■	■	■	■	361253.4	0.06	0.00	0.38	0.28	0.43
	sko42	■	■	■	■	■	■	■	■	■	■	15812.1	0.06	0.02	0.00	0.04	0.45
	sko49	■	■	■	■	■	■	■	■	■	■	23391.5	0.01	0.00	0.07	0.03	0.42
	sko56	■	■	■	■	■	■	■	■	■	■	34474.2	0.09	0.08	0.00	0.13	0.36
	sko64	■	■	■	■	■	■	■	■	■	■	48516.9	0.05	0.05	0.00	0.08	0.28
sko72	■	■	■	■	■	■	■	■	■	■	66337.5	0.06	0.04	0.00	0.10	0.24	

Table 16 Comparison of three PNLs variants, TS and ILS with a budget of 1 and 100 million evaluations on QAP landscapes.

The relative effectiveness of local search on the instances `lipa` slightly changes for a budget of 100 million evaluations in comparison to the results observed for 1 million evaluations. In general, ID_{best} is the local search which reaches a global optimum in most cases, closely followed by SW. ID_{any} and ID_{best} are generally the most effective local searches on landscapes `sko`, followed respectively by SW, TS, ILS.

`lipa` landscapes are known to be easier to solve than `sko` landscapes. Indeed, the global optimum is known for `lipa` instances but not for `sko` instances. Also, the number of times the methods reach the best known optimum is much lower on landscapes `sko`. However, it should be noted that the methods examined correspond to the variant (in terms of parameter setting) that leads to the best average fitness. Thus, for other parameter values that lead to lower average fitness, it remains possible that the best solutions are reached more often.

PNLSs are consistently more effective than ILS and TS on NK, QAP, and FSP landscapes for the two budgets tested (1 million and 100 million evaluations). On UBQP landscapes, PNLs are effective for a low budget because the partial neighborhood allows them to reach good solutions quickly. With a larger budget, TS performs better, partially because of its higher level of intensification. Among the proposed PNLs, SW and ID_{best} often have equivalent effectiveness. The strategy of selecting deteriorating neighbors of ID_{any} induces a strong diversification that regularly prevents it from converging sufficiently towards good solutions. Nevertheless, this strategy is suitable for some landscapes.

7 Discussion

A DEVELOPPER

lipa	Optimum	SW	ID _B	ID _A	ILS	TS	sko	Optimum	SW	ID _B	ID _A	ILS	TS
70a	169755	100	100	81	2	12	42	15812	60	53	99	2	51
80a	253195	80	84	1	0	4	49	23386	22	42	1	0	12
90a	360630	54	65	1	0	0	56	34458	3	0	4	0	1
							64	48498	0	32	10	0	2
							72	66256	0	0	0	0	1

Table 17 Fitness of the global optima on `lipa` instances, Fitness of the best known on `sko` instances, and number of executions for which these fitness are obtained with each local search variant.

SW paysages Param (facile?) rug/taille/(eval)

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. On the efficiency of worst improvement for climbing nk-landscapes. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO '14*, pages 413–420, New York, NY, USA, 2014. ACM.
3. Matthieu Basseur and Adrien Goëffon. Climbing combinatorial fitness landscapes. *Applied Soft Computing*, 30:688–704, 2015.
4. 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.
5. Matthieu Basseur, Adrien Goëffon, and Hugo Traverson. Exploring non-neutral landscapes with neutrality-based local search. In *International Conference on Learning and Intelligent Optimization*, pages 165–169. Springer, 2015.
6. William Bateson, ER Waunders, and Reginald Crundall Punnett. Experimental studies in the physiology of heredity. *Molecular and General Genetics MGG*, 2(1):17–19, 1909.
7. Rainer E Burkard, Stefan E Karisch, and Franz Rendl. Qaplib—a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403, 1997.
8. Michael R Gary and David S Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.
9. Holger H Hoos and Thomas Stützle. Stochastic local search algorithms: An overview. In *Springer Handbook of Computational Intelligence*, pages 1085–1105. Springer, 2015.
10. 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.
11. Tjalling Koopmans and Martin J. Beckmann. Assignment problems and the location of economic activities. Cowles Foundation Discussion Papers 4, Cowles Foundation for Research in Economics, Yale University, 1955.
12. Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The irace package, iterated race for automatic algorithm configuration. Technical report, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
13. 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.
14. Marie-Eléonore Marmion, Clarisse Dhaenens, Laetitia Jourdan, Arnaud Liefvooghe, and Sébastien Verel. Nils: a neutrality-based iterated local search and its application to flowshop scheduling. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 191–202. Springer, 2011.

15. Tadahiko Murata, Hisao Ishibuchi, and Hideo Tanaka. Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering*, 30(4):1061–1071, 1996.
16. Bertrand Neveu, Gilles Trombettoni, and Fred Glover. Id walk: A candidate list strategy with a simple diversification device. In *International conference on principles and practice of constraint programming*, pages 423–437. Springer, 2004.
17. Gabriela Ochoa, Marco Tomassini, Sebastien 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.
18. 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.
19. Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.
20. Kenneth Sörensen. Metaheuristics — the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.
21. Éric Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285, 1993.
22. Eric Taillard et al. Some efficient heuristic methods for the flow shop sequencing problem. *European journal of Operational research*, 47(1):65–74, 1990.
23. Sara Tari, Matthieu Basseur, and Adrien Goëffon. Sampled walk and binary fitness landscapes exploration. In *International Conference on Artificial Evolution (Evolution Artificielle)*, pages 47–57. Springer, 2017.
24. Edward Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological cybernetics*, 63(5):325–336, 1990.
25. Sewall Wright. *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*, volume 1. 1932.