



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-03384306v2

HAL Id: hal-03384306

<https://hal.science/hal-03384306v2>

Submitted on 17 Jan 2024

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^{a,b}, Matthieu Basseur^{c,*} and Adrien Goëffon^c

^aUniv. Lille, CNRS, Centrale Lille, UMR 9189 - CRISTAL, F-59000 Lille, France

^bLaboratoire d'Informatique Signal et Image de la Côte d'Opale, Univ. Littoral Côte d'Opale, F-62100 Calais, France

^cLaboratoire d'Étude et de Recherche en Informatique d'Angers, Université d'Angers, F-49045 Angers cedex 01, France

E-mail: sara.tari@univ-littoral.fr; matthieu.basseur@univ-angers.fr; adrien.goëffon@univ-angers.fr

Received DD MMMM YYYY; received in revised form DD MMMM YYYY; accepted DD MMMM YYYY

Abstract

In this work, we study partial neighborhood local search (PNLS) techniques, which consist of adaptive walks where moves are chosen in a random subset of the current solution neighborhood. PNLSs balance between intensification and diversification is mainly determined by its single parameter λ designing the subset size. We analyze and discuss three PNLSs variants, using the abstraction of several combinatorial optimization problems into fitness landscapes: NK landscapes, Unconstrained Binary Quadratic Programming, Flow-shop scheduling, and Quadratic Assignment. Our empirical study first analyses the structure of these landscapes through indicators. Then, we perform a parameter study of PNLSs for two computational budgets to study the impact of the sample size on the balance between intensification and diversification on different landscapes. Moreover, these experiments allow us to set an appropriate parameter value to compare the ability of PNLSs to reach good-quality solutions accurately. Finally, we compare PNLS variants with two classical metaheuristics, identifying links between landscape characteristics and algorithms behavior.

Keywords: fitness landscapes, local search, discrete optimization, metaheuristics

1. Introduction

In the field of combinatorial optimization, research can follow two complementary directions: on the one hand, problem-oriented approaches that exploit specificities of the studied problem (Moura, 2019; Guerine et al., 2020), and on the other hand, generic approaches that try to provide guidelines to establish solvers that can handle optimization problems in general (Bianchi et al., 2009; Hao, 2012). This observation can be done for exact algorithms as well as for heuristic search algorithms. When it comes to search algorithms, problem-oriented ones use the structure of a given problem to guide the search toward high-quality solutions (Yafrani et al., 2018). In contrast, generic methods aim at finding guidelines that can be applied to many combinatorial optimization problems (Srinivas, 2019).

* Author to whom all correspondence should be addressed (e-mail: matthieu.basseur@univ-angers.fr).

Heuristics search algorithms include neighborhood-based metaheuristics that consist of navigating through the search space guided by a neighborhood relation and a move strategy. 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 within the search space (Blum and Roli, 2003). The general principle of neighborhood-based metaheuristics is elementary since it mainly consists of constructing search trajectories by applying a move strategy on a neighborhood structure.

Methods proposed in the field of metaheuristics are more sophisticated than ever, especially since the dramatic increase in the number of dispensable bio-inspired methods (Michalewicz, 2012; Sörensen, 2015). While metaheuristics are meant to be generic methods, a significant part of them is dedicated to solving a given problem. They usually exploit relevant problem specificities, which leads to be highly efficient on a given problem but also reduces their generic aspect. Using such problem-oriented approaches aims to achieve the best possible results but also requires designing sophisticated methods that involve many mechanisms and parameters interacting in complex ways. Thus, understanding the behavior of such approaches is a complicated task (Sorensen et al., 2018; Sala and Müller, 2020).

Nevertheless, improving the understanding of metaheuristics can provide insights on when and how to use them. Such knowledge is crucial to determine which approach is the most suited for a given problem, particularly when 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.

Neighborhood-based metaheuristics operate from associations between solutions. More fundamental studies about such search strategies can be realized through the fitness landscapes analysis (Pitzer and Affenzeller, 2012; Malan and Engelbrecht, 2014). 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 connections between solutions. One can establish correlations between the dynamics of methods and the characteristics of landscapes to obtain insights on why the method is efficient or not, and why a problem instance can be considered as more or less difficult to solve. Among the properties that characterize fitness landscapes, some are recognized to make the landscape challenging to tackle. Investigating methods with a landscape-aware approach can help in determining the type of landscapes where some methods are expected to be efficient (Liefoghe et al., 2017; Baumann et al., 2019).

While fitness landscapes can be used to study neighborhood-based search algorithms, investigating sophisticated exploration 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. Advanced metaheuristics often alternate different mechanisms to manage the balance between sufficiently intensifying the search on promising areas of the search space and sufficiently exploring the search space (Xu and Zhang, 2014). Achieving such a balance is often possible with fewer mechanisms.

Here, we are interested in more stochastic search algorithms that reduce the effort dedicated to neighborhood exploration and increase the frequency of steps during the search. We propose to consider Partial Neighborhood Local Search algorithms (PNLS), which randomly restrict the neighborhood at each search step. In particular, we investigate the Sample Walk algorithm (SW) introduced in Tari et al. (2017), which, in a certain way, can be viewed as a local search transposition of a $(1, \lambda)$ evolution strat-

egy (Rechenberg, 1984; Beyer, 1994; Engelbrecht, 2007), with mutation possibilities restricted by the neighborhood function. We also consider a previously described algorithm that also falls into the scope of PNLS (ID_{walk} in Neveu et al. (2004)). The two main contributions of this work are the following. First, we perform a parameter sensitivity analysis of PNLS 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 PNLS efficiency. The experiments include the consideration of four classical optimization problems and two different computational budgets.

The outline of the paper is as follows. In the next section, we recall the main concepts relating to fitness landscapes and local search and discuss some observations that motivate the study of partial neighborhood local searches. The three last sections are dedicated to the empirical analysis of fitness landscapes and PNLS algorithms. We realize experiments on various bit-string and permutation landscapes derived from various problems and described in section 3: NK landscapes and unconstrained binary quadratic programming problem (binary string solution representation), as well as quadratic assignment problem and flow-shop scheduling problem (permutation solution representation). In section 4, we report experiments that allow us to characterize landscapes according to indicators. In section 5, we report the experimental results which aim to determine the adequate parameterization of considered algorithms. Particular attention is devoted to PNLS methods and the sensibility of the results to λ values. In section 6, we confront PNLS methods with a tabu search and an iterated local search. The last section provides some conclusions and points out some possible ways forward.

2. Background

2.1. Fitness Landscapes

The concept of fitness landscapes takes its origin in Wright (1932), and are nowadays used in various fields to better apprehend the behavior of complex systems. 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 problem instance but also depends on the considered method. In particular, as showed for some problems (e.g., flow-shop scheduling (Murata et al., 1996)), choosing a relevant neighborhood operator is essential to ease the navigation of search methods through the search space. Fitness landscapes describe the search space structured through a neighborhood relation and a fitness function. Landscapes properties can give insights about the adequacy and complementarity of search algorithm components.

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 derivated 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 various search spaces (for example, including non-feasible solutions or excluding some feasible solutions). A primary interest of fitness landscapes is the study of neighborhood-based optimization methods

depending on landscape properties. Most of these properties, such as the ones described in Pitzer and Affenzeller (2012); Malan and Engelbrecht (2013), can influence the behavior of search algorithms. In particular, the characteristics relating to the ruggedness and the neutrality of the landscape strongly influence the difficulty of optimizing the underlying problem using the considered neighborhood operator: number and distribution of local optima, size of their basins of attraction (Ochoa et al., 2008), presence of plateaus (Marmion et al., 2011).

The ruggedness level of landscapes is a key aspect determining the hardness of the problem. At the same time, ruggedness cannot be clearly described using a single indicator (Malan and Engelbrecht, 2013). In our study, we 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 has a significant impact on the behavior of search algorithms.

The *autocorrelation function* (Weinberger, 1990) ρ reports the correlation between the fitness of solutions and their distances. One can estimate this dependence from a random walk in the landscape. While the efficiency of metaheuristics is often influenced by deteriorating and improving moves, the autocorrelation measure is based on fitness variation. It does not sufficiently consider sign epistasis, which leads to the appearance of local optima (Kvitek and Sherlock, 2011) and tends to influence local search algorithms performance. Consequently, we also use the *k-ruggedness* indicator based upon the *k-sign-epistasis* principle introduced hereafter.

Let x_0 and x_k two k -distant solutions with respect to the considered neighborhood (e.g., the Hamming distance while considering binary strings and the 1-flip operator). Let x'_0 and x'_k neighbors of respectively x_0 and x_k resulting from the same move application (e.g., the same bit is flipped, considering binary strings and 1-flip). Moreover, the solutions x'_0 and x'_k have to respect the constraints $d(x_0, x'_k) = k + 1$ and $d(x'_0, x'_k) = k$ (considering binary strings and 1-flip, the flipped bit cannot differ in x_0 and x_k). The *k-sign-epistasis* is defined as $k\text{-epis}(x_0, x'_0, x_k, x'_k) = 0$, if $(f(x_0) - f(x'_0)) \cdot (f(x_k) - f(x'_k))$ is positive, and 0 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 can be complicated, if not impossible, in particular when the landscape is not 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 . The result is a plot of *k-ruggedness* $\in [0, 1]$ (0 indicates the absence of sign epistasis in the landscape, whereas a landscape with random fitness values tends to have a *k-ruggedness* value close to 0.5).

Finally, the number of local optima is the primary aspect of characterizing a problem hardness and is strongly related to the two previous measures previously introduced. These measures are useful for evaluating this aspect but assume a uniform distribution of ruggedness throughout the landscape. The study of local optima distribution can also provide information on the difficulty of solving a given landscape. Here we consider the *average distance between local optima*—more accurately, *given the distribution induced by a search strategy*.

Let O be a set of distinct 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

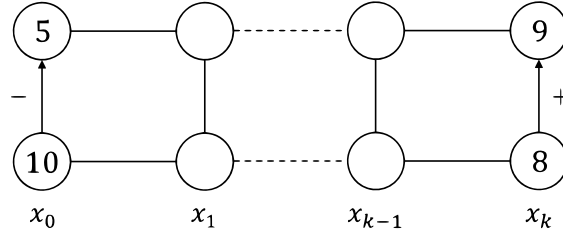


Fig. 1. Illustration of k -sign-epistasis while applying a given mutation on two solutions x_0 and x_k (x_0 and x_k are linked by a minimal set of solutions $\{x_1, \dots, x_{k-1}\}$, such that $\forall i \in \llbracket 1, k \rrbracket, x_i \in \mathcal{N}(x_{i-1})$).

clustered are the local optima. Of course, this indicator is estimated using a local search strategy so that the set of local optima is conditioned by the effect of the basins of attraction. Nevertheless, the indicator remains worthy of interest in characterizing the landscapes, particularly concerning our study, the object of which is local search algorithms.

Many landscapes derived from specific combinatorial problems tend to have a *central massif* (or *big valley*) structure, where the local optima 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. The difficulty of landscapes can be characterized by their balance between a random distribution of local optima and a *big valley* structure.

Studying methods behavior is facilitated by the use of models allowing tunable ruggedness, such as NK landscapes (Kauffman and Weinberger, 1989). Studies on artificial landscapes with common features are essential to comprehend 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 the ruggedness only on NK landscapes should not be considered sufficient. Tackling landscapes derived from classical discrete optimization problems can help to obtain better insights into the behavior of the considered methods within such studies. The studies presented in this paper are based on both NK landscapes and three classes of academic problems: UBQP, FSP, QAP (see section 5).

2.2. Local Search, Diversification Issue

In their general description, local search algorithms (Hoos and Stützle, 2015) are particularly straightforward and therefore enable 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, $\mathcal{N}(x)$ is the neighborhood of x , and x^* the best-encountered solution.

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

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  $x^*$ 
```

Studying local search move policies leads to insights for establishing links between the structure of optimization problems and local search efficiency. We focus here on basic methods to better isolate and study some mechanisms used among search algorithms. In the case of local search algorithms, hill-climbers do not require many mechanisms and are widely used as metaheuristic components. We previously showed that despite their simplicity, climbers have a high potential since a (random) first improvement hill-climbing often has the *possibility* to reach the global optimum from most initial solutions (Basseur et al., 2014). This means that there often exists a monotonous trajectory leading from a medium-quality solution to the global optimum of the problem. Of course, the issue comes from the exponential number of possible trajectories, and therefore from the difficulty of defining a pivoting rule allowing to make good choices throughout the search. In other words, to exhibit a more restrictive rule that selects an improving neighbor without reducing the possibility of reaching the optimum.

Some studies investigating climbers pivoting rules showed that among widely used first and best improvement, the first improvement is, in general, the most likely to reach high local optima (Hansen and Mladenović, 2006), especially when used in the early phase of the search (Whitley et al., 2013). In general, the first improvement is particularly efficient to climb significantly rugged landscapes while the best improvement leads to better local optima on highly smooth and extremely rugged landscapes (Ochoa et al., 2010; Basseur and Goëffon, 2013). Let us notice that difficult landscapes derived from hard optimization problems are often significantly rugged. Furthermore, on such rugged landscapes, the worst improvement leads more likely toward good local optima than the first and best improvement rules (Basseur and Goëffon, 2014).

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. As previously shown on probabilistic and quantized NK landscapes (which hold neutrality), a stochastic hill-climber which indifferently selects the first encountered improving or neutral neighbor outperforms climbers selecting improving neighbors as a priority (Basseur and Goëffon, 2013). Since accepting indifferently neutral and improving ones during the search process helps to reach higher pikes, we measured the effect of adding artificial neutrality within climbers (Basseur et al., 2015). In the proposed method, artificial neutrality is obtained by discretizing an adaptative fitness function within a climber. The resulting

climber outperforms classic hill-climbing processes 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. In contrast, an adequate rate of artificial neutrality helps to diversify the search within climbers sufficiently.

Classic climbers are usually used as intensification mechanisms within sophisticated single-solution or population-based metaheuristics (Lourenço et al., 2019; Sánchez-Oro et al., 2014; Ersoy et al., 2007) since they almost only perform pure intensification during a search process. However, sophisticated metaheuristics using climbers also own a diversification mechanism to achieve an appropriate 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 to higher local optima than 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.

In Tari et al. (2017), we proposed the SW algorithm, which simulates 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 computational budget. Indeed, evaluating only a few neighbors promotes diversification by reducing the possibility of encountering a particularly good solution at the next step of the search. Using this principle within a non-strict local search can help manage the balance between intensification and diversification with a single mechanism instead of an iterated local search that requires two separate mechanisms.

2.3. Partial Neighborhood Local Searches

We call partial neighborhood local search (PLNS) a search strategy which selects at each step a solution from a *random* sample of neighbors whose (maximum) size, denoted as λ , is configurable. The PLNS process is detailed in algorithm 2, and is a specific local search (see algorithm 1). Let us precise that the λ neighboring solutions are not necessarily evaluated. As considered 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 searches correspond to random walks and are then exclusively diversifying, whereas when λ is set to the size of \mathcal{N} , searches are intensifying (best improvement, if the selection process is based upon fitness values). The neighbor selection strategy also influences the balance between intensification and diversification.

In the remaining of the paper, we study the impact of different selection strategies for PNLSS. In particular, we propose the sampled walk algorithm (SW), a simple way to implement PLNS, which uses the same selection strategy on improving and deteriorating neighbors. We confront SW with ID_{walk} or

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  $x^*$ 

```

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)

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

the *intensification/diversification walk* proposed in Neveu et al. (2004), which is declined in two variants: ID_{best} and ID_{any} . ID_{walk} variants can be classified as PNLS algorithms. These three algorithms operate as follows:

- SW selects the solution with the best fitness from a random sample of λ neighboring solutions, whether it leads to an improving or deteriorating move.
- ID_{best} selects the first improving solution encountered. If λ deteriorating neighbors are evaluated consecutively, then the best one is selected.
- ID_{any} selects the first improving solution encountered. If λ deteriorating neighbors are evaluated consecutively, then a random neighbor is 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 λ , while ID_{best} selects the first encountered improving neighbor. Considering a fixed value of λ , the intensification rate is then higher for SW than for ID_{best} . Let us notice that this study does not focus on specific ways to handle neutrality in landscapes. Thus, in the move strategies described above, neutral neighbors are assimilated as deteriorating ones.

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

among the λ neighbors (like in the SW process), whereas ID_{any} randomly selects it, which intuitively induces a higher rate of diversification when no improving solution is found.

Table 1 illustrates an example of a step of the three PNLs. 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 4 improving neighbors (in a maximization context). For instance, considering SW, S_2 is selected when it 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 fitness ordering of neighbors.

The remaining part of this paper is dedicated to analyze the behavior of such local searches and to observe whether considering partial neighborhoods is effective in achieving good-quality solutions.

3. Benchmark Instances, Preliminary Landscapes Analysis

In this study, we used 4 classes of fitness landscapes described by specific fitness function models: NK, UBQP, FSP, and QAP functions. UBQP, FSP, and QAP are classic discrete optimization problems whose definitions are recalled below. This section also reports a feature analysis of all landscapes.

3.1. Models

NK Landscapes. NK functions (Kauffman and Weinberger, 1989) are widely used to describe tunable fitness landscapes to study the influence of ruggedness on the behavior of evolutionary algorithms, especially local searches. NK 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:

$$F(x) = \frac{1}{N} \sum_{i=1}^N C_i(x_i, x_1(i), \dots, x_K(i)) \quad (1)$$

x_i is the i -th bit of the solution x . Let $l_j(i)$ ($j \in \llbracket 1, K \rrbracket$) be the number of the j -th bit linked with bit i : $l_j(i) \in \llbracket 1, N \rrbracket \setminus (\{i\} \cup \bigcup_{k=1}^{j-1} l_k(i))$. Then $x_j(i)$ is the corresponding bit value in solution x . $C_i : \{0, 1\}^{K+1} \rightarrow [0, 1)$ are pseudo-boolean subfunctions which define the contribution value of each x_i and its linked values $x_1(i), \dots, x_K(i)$. NK landscapes instances are determined by the K -uples $(l_1(i), \dots, l_K(i))$ and a matrix C of fitness contribution which describes the $2^{K+1} \times N$ possible contribution values.

The neighborhood relation used in the NK model is the 1-Hamming distance neighborhood, described by the move operator *one-flip*. Various parameterizations of NK instances ($N \in \{128, 256, 512, 1024\}$, $K \in \{1, 2, 4, 6, 8, 10, 12\}$) are considered in the experiments in order to observe algorithms behavior in various contexts (size and ruggedness).

UBQP Landscapes. The unconstrained binary quadratic programming problem (UBQP) is an NP-hard problem (Gary and Johnson, 1979), which can reformulate a vast scope of real-life problems in various fields. A UBQP instance is composed of a matrix $(q_{ij})_{1 \leq i, j \leq n}$ containing both positive and negative values. We note x_i the i -th element of a solution $x \in \{0, 1\}^n$. The objective function f to be maximized is described as follows:

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j \quad (2)$$

To define UBQP landscapes, we equip the search space with the 1-Hamming distance neighborhood. Hence, each solution has a neighborhood of size n . Like NK landscapes, UBQP landscapes then have a hypercube structure. We used the generator proposed in Palubeckis (2004) to generate several instances of different sizes and density. The density d refers to the rate of non-null values in the matrix (q_{ij}) .

FSP Landscapes. The flow-shop scheduling problem (FSP) (Taillard, 1993) is a widely studied scheduling problem. Among the numerous FSP variants, we consider the permutation flow-shop, which considers the minimization of the total completion time. An instance of FSP is 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 job J_i on 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 $x \in S_n$ is then represented by a permutation of jobs. Each task is scheduled to the date s_{ij} .

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

$$C_{max}(x) = \max_{i \in [1, \dots, n]} \{s_{im} + t_{im}\} \quad (3)$$

The search space S_n , of size $n!$, is the set of permutations of $[[1, n]]$. The most efficient neighborhood operator to tackle this problem by a local search algorithm consists of moving a job to a different position and is called insert operator \mathcal{N}_{ins} . Thus a FSP fitness landscape is defined by the triplet $(S_n, \mathcal{N}_{ins}, C_{max})$. We select 6 representative instances with $j \in \{30, 50\}$ jobs and $m \in \{10, 15, 20\}$ machines (Taillard, 1990).

QAP Landscapes. The quadratic assignment problem (QAP) (Koopmans and Beckmann, 1957) is a permutation problem considered as one of the most difficult to solve NP-hard problems (Sahni and Gonzalez, 1976). QAP consists of allocating n units to n locations in function of a distance matrix $(d_{ij})_{1 \leq i, j \leq n}$ and flow between units matrix $(f_{rs})_{1 \leq r, s \leq n}$. d_{ij} is the distance between locations i and j , and f_{rs} the flow between units r and s . A permutation $x \in S_n$ describing the allocations 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:

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{x_i x_j} \quad (4)$$

x_i represents the i -th element of the solution x . While FSP and QAP share common search spaces S_n , here we equip S_n with the swap neighborhood \mathcal{N}_{swap} , since the swap operator (which consists of exchanging the location of two units) is commonly used for solving QAP with neighborhood-based metaheuristics (Ramkumar et al., 2009; Benlic and Hao, 2013). Each solution x has a neighborhood size of $\binom{n}{2}$. The associated landscape is then defined by $(S_n, \mathcal{N}_{swap}, f)$. We use 8 instances of size $n \in [42, 90]$ from the QAPLIB (Burkard et al., 1997). These instances come from two classes: *lipa* and *ska*. *lipa* instances are asymmetric instances while *ska* instances have rectangular distances and pseudorandom numbers as entries in flow matrices.

In the following, we use these 4 problems (2 pseudo-boolean and 2 permutation ones) to have a more representative panel of landscapes on which to base our analysis of local search strategies. Note that the representations of permutations lead to different landscapes for which some characterization indicators must be redefined. This aspect is discussed in section 5.

3.2. Landscapes analysis

The indicators used for the analysis are presented in section 2.1. Results are reported in table 2, where the two first columns report $\rho(1)$ and 1-ruggedness estimated with a sample of 100,000 initial solutions. Starting from these solutions, 100,000 random walks have been performed to estimate the autocorrelation function. We report the length of walks (in %) where thresholds are reached. Here these thresholds are set to 0.8 and 0.5. When the thresholds are quickly reached, it indicates a low correlation between fitness and distance between solution, and thus a landscape more challenging to explore efficiently using local searches. Similarly, we use thresholds to report k -ruggedness (values 0.1 and 0.25). k -ruggedness 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 context.

The last two columns of the tables report 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. d_{Rd} corresponds to the average distance between random solutions. The neutrality rate is estimated using a sample of 100,000 random pairs of neighboring solutions.

The indicator values computed for each class of landscape lead to general observations, which are used in the next sections to identify relations between algorithms efficiency and landscape properties.

- **NK landscapes:** Autocorrelation and k -ruggedness functions are strongly correlated with the values of K . On the less rugged landscapes, the average distance between local optima is low. Indeed, this distance is much smaller than the average distance between random solutions. Contrary to specific NK models landscapes (Barnett, 1998; Newman and Engelhardt, 1998), the random NK functions used here lead to landscapes having no significant neutrality.
- **UBQP landscapes:** The local ruggedness measure suggests significantly rugged landscapes—one observes similar values on NK landscapes generated with a higher value of K . However, the $\rho(l)$ and k -ruggedness values are very high and are comparable to smooth NK landscapes. This property

Table 2

Landscapes indicator values (approximation by sampling). Left: NK; Right: UBQP, FSP, and QAP.

NK	$\rho(1)$	1-rug.	$\rho(l)$		k -rug.		$d_{LO}(d_{Rd})$	Neut.	UBQP n d	$\rho(1)$	1-rug.	$\rho(l)$		k -rug.		$d_{LO}(d_{Rd})$	neut.
			≤ 0.8	≤ 0.5	≥ 0.1	≥ 0.25						≤ 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%	2048 10	0.9984	0.3%	8.5%	26.4%	5.2%	29.5%	269.7 (1024)	0.038%
128 2	0.9745	1.2%	7.0%	22.0%	7.9%	22.9%	44.9 (64)	0.00%	2048 25	0.9985	0.7%	8.0%	27.6%	5.0%	27.6%	230.7 (1024)	0.022%
128 4	0.9584	3.1%	4.7%	14.2%	3.1%	10.2%	60.9 (64)	0.00%	2048 50	0.9987	0.9%	8.5%	26.6%	5.0%	29.2%	238.1 (1024)	0.013%
128 6	0.9449	5.2%	3.1%	10.2%	2.4%	7.1%	63.5 (64)	0.00%	2048 100	0.9990	1.0%	8.8%	27.7%	4.8%	27.7%	307.2 (1024)	0.011%
128 8	0.9290	7.4%	3.1%	7.9%	1.6%	4.7%	63.9 (64)	0.00%	4096 10	0.9994	0.2%	8.8%	28.2%	4.7%	28.8%	536.1 (2048)	0.027%
128 10	0.9140	9.6%	2.3%	7.0%	1.6%	3.9%	64.0 (64)	0.00%	4096 25	0.9992	0.5%	8.5%	26.3%	5.0%	28.9%	512.0 (2048)	0.013%
128 12	0.8983	11.8%	2.3%	5.5%	0.8%	3.1%	64.0 (64)	0.00%	4096 50	0.9989	0.5%	8.5%	27.2%	4.7%	28.5%	483.5 (2048)	0.010%
256 1	0.9899	0.3%	8.6%	27.4%	16.1%	43.9%	50.2 (128)	0.00%	4096 100	0.9983	0.7%	8.2%	26.7%	5.7%	30.0%	486.6 (2048)	0.007%
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%									

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

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

suggests a high local ruggedness and low global ruggedness. Such landscapes can be viewed as *big valley*. The average distance between local optima confirms this tendency. UBQP landscapes also contain a reduced level of neutrality (less than 0.1%).

- **FSP landscapes:** FSP landscapes are characterized as highly rugged, and local optima seem to follow a uniform distribution: the average distance between them is similar to the average distance between random solutions. FSP landscapes also have a high level of neutrality, inversely correlated with the number of machines of the instance.
- **QAP landscapes:** QAP landscapes are also highly rugged, and their local optima seem to be uniformly distributed. These landscapes also contain a significant level of neutrality, but less than FSP landscapes.

4. PNLs parameterization analysis

The experiments presented in this section have three main objectives:

- determine, for each landscape, an adequate parameter value for each considered method,
- study the parameter sensitivity of PNLs (mainly according to λ values),
- identify links between landscape properties and efficient¹ λ values.

¹We call efficient or appropriate parameter value a value that leads the algorithm to good-quality solutions compared to other parameterization.

Table 3

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

Problem	\mathcal{L} values tested (TS)	\mathcal{M} values tested (ILS)
NK	{5, 10, 15, 20}	{1, 5, 10, 15, 20}
UBQP	{5, 10, 15, 20, 30, 40, 50, 60}	{1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80}
QAP	{8, 12, 16, 20, 24, 28, 32}	{1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80}
FSP	{4, 6, 8, 10, 12, 14, 16, 18}	{1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80}

4.1. Experimental protocol

Let us recall that experiments focus on five algorithms: three PNLs (SW, ID_{best} , and ID_{any}) and two classical LS algorithms (ILS and TS). The considered local searches require a single parameter (excluding the stop criterion), and their mode of operation is based on elementary pivoting rules. In this section, we first describe the values used for parameterization analysis. Then, we briefly discuss the stopping criterion and the experimental setup used for the analysis of the results.

Parameter value candidates. We consider ILS_F , an ILS that uses the first improvement both as pivoting rule and acceptance criterion 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 best improvement.

For each problem, we performed a set of experiments, using different \mathcal{M} values. The most effective parameterizations are given in the next subsection and used later for comparison between algorithms. The values tested are reported in table 3.

TS uses a list of bits of size \mathcal{L} and no aspiration mechanism, which ensures a minimal distance between solutions along the walk. Indeed, as we use a landscape-oriented approach, it implies using a distance-based list rather than a solution-based list. The pivoting rule used is best improvement, which is the most commonly used rule within tabu search algorithms. Tested values of \mathcal{L} are summarized in table 3. As for ILS_F , the most effective values are used for comparison between algorithms.

Each PNLs tested here requires only one parameter (λ). For each of them, we consider the following λ values to study their influence on their dynamics.

- Binary landscapes: we consider values of $\lambda \in \{2^k : k \in \llbracket 1, \log_2 |\mathcal{N}| \rrbracket\}$. On NK landscapes, we additionally consider values of $\lambda \in \{4k' : k' \in \llbracket \frac{2^{k-1}}{4} + 1, \frac{2^{k-1}}{4} - 1 \rrbracket\}$, where k is such that 2^k is the best parameterization observed in the previous step, ie. the parameter leading to the best average fitness. For example, if $\lambda = 32$ leads to the best fitness average, we consider all intermediate values in steps of 4 between 20 and 60.
- Permutation landscapes: the values of λ used correspond to different ratios of the size of the neighborhood, as follows: $\lambda \in \{\frac{|\mathcal{N}|}{16}, \frac{|\mathcal{N}|}{15}, \dots, \frac{|\mathcal{N}|}{2}, |\mathcal{N}|\}$.

Ideally, all possible values should be examined, but this would lead to a high number of executions. Although automatic parameterization approaches exist (for example, *irace* as off-line configurator (López-Ibáñez et al., 2016)), we chose not to use them in this study. While they are often useful in leading the search to good quality-solutions without considering the constraints of parameter setting, this kind of process induces new constraints or components that complexifies the analysis of the results and would limit our observations. For example, in a context of off-line configuration, the results can vary according to the configurator under consideration, and it requires to find or generate similar instances to find a good configuration on the instances to solve. In the context of online configuration, the parameter

value is determined dynamically, and the process 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 components of PNLs, thus it is important to limit their number.

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

Stopping criterion. This criterion is determined by a maximal number of solution evaluations. This number is set to 100 million in our experiments, a deliberately high budget so that a sufficient convergence can be observed on the different landscapes. We also extract from the information collected during the runs the best fitness achieved for a reduced budget. In this work, we observe the behavior of the five algorithms after 1 million and 100 million evaluations.

Experimental setup. For each triplet (landscape, method, λ), 100 runs are 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 evaluations.

Doing so allows us to perform a statistical analysis to compare methods. Let \mathcal{S} be the number of times a A method reaches a better local optimum than a B method over 100 confrontations. In the following, we consider that a method A statistically dominates a method B on a set of 100 runs when $\frac{1}{2^{100}} \sum_{i=0}^{\mathcal{S}} \binom{100}{i} \geq 0.95$, ie. $\mathcal{S} \geq 58$ w.r.t. a binomial test (with a p -value less than 0.05).

4.2. PNLs Parameter sensitivity analysis

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

4.2.1. Parameter sensitivity analysis

Considering several parameter values allows us to compare local searches more fairly, limiting the risk that one of them is more efficient because of a better configuration. It also allows observing the evolution of the capacity of a given local search to reach good-quality solutions according to the setting. This analysis can help us to determine the degree of sensitivity of methods to the parameterization and to provide guidelines to design useful automatic parameterization based on landscapes. In the following, we present and discuss the evolution of the average quality of strategies according to their parameter setting on NK landscapes.

Figure 2 illustrates the average fitness obtained with respect to the settings of SW, ID_{best} and ID_{any} on several NK landscapes throughout executions of 100 million evaluations. The average fitness values obtained by SW and ID_{best} evolve similarly. These two PNLs only differ in the pivoting rule applied when at least one improving neighbor is encountered. This similarity could explain the proximity of their dynamics on NK landscapes.

Compared to SW and ID_{best} , ID_{any} behavior evolves differently depending on the value of λ . When

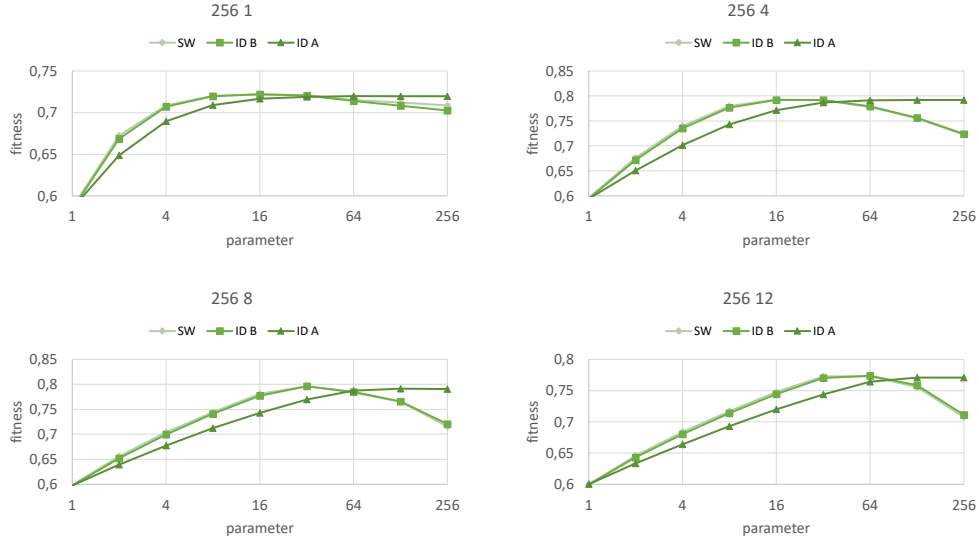


Fig. 2. Fitness variation on NK landscapes ($N = 256$) during executions with different parameter values (λ_{SW} , λ_{IDb} , λ_{IDa} for SW, ID_{best} , ID_{any} , respectively).

this value is high, ID_{any} is more effective in finding good-quality 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 PNLSSs, and sharply increases the appropriate proportion 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 search process requires a higher intensification rate and, thus, a lower diversification rate.

In general, considering more neighbors in PNLSSs increases the likelihood of meeting an improving neighbor. Indeed, when the search reaches higher solutions in the landscape, this limits the diversification rates and consequently promotes the search in good areas of the landscape. When epistasis is low, the range of λ values that lead on average to the best solutions is larger. For the three PNLSSs, a positive correlation exists between the value leading to the best average fitness and the value of K (and therefore the ruggedness rates).

4.2.2. Summary of best sample sizes for PNLSS

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

NK landscapes: The table 4 presents the most appropriate λ values found on NK landscapes. The statistical analysis of the results is not always significant for this experiment. Then, when several λ values lead to results that are not statistically comparable, the smaller value is reported.

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

Table 4

Parameter values that leads to bests average fitness over 100 executions for each couple (landscape, method) on NK landscapes: 1 million evaluations (left), and 100 million evaluations (right).

NK	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	NK	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	NK	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	NK	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 1	8	8	16	5	20	256 1	16	16	96	15	20
128 2	12	16	72	15	15	256 2	16	16	88	10	15	128 2	12	16	40	10	20	256 2	24	24	96	5	20
128 4	16	16	56	5	5	256 4	16	16	72	5	5	128 4	16	16	40	5	20	256 4	20	20	192	1	120
128 6	16	20	80	5	5	256 6	24	24	64	5	5	128 6	16	16	56	1	15	256 6	24	24	184	5	20
128 8	24	24	64	1	5	256 8	32	32	88	5	5	128 8	20	20	72	1	15	256 8	32	32	112	1	15
128 10	32	28	112	1	5	256 10	36	40	112	5	5	128 10	24	32	120	1	10	256 10	36	40	184	1	15
128 12	32	36	64	1	5	256 12	48	48	152	1	5	128 12	28	36	96	1	10	256 12	44	52	184	1	15
512 1	24	24	128	20	50	1024 1	24	24	128	15	50	512 1	16	16	256	20	50	1024 1	16	16	256	15	50
512 2	24	24	128	15	50	1024 2	24	24	128	20	50	512 2	16	24	128	5	50	1024 2	24	24	256	5	50
512 4	24	24	128	15	50	1024 4	24	24	128	20	50	512 4	24	24	128	5	50	1024 4	24	24	256	5	50
512 6	24	32	128	10	40	1024 6	24	32	128	20	50	512 6	24	32	256	5	30	1024 6	32	32	256	5	50
512 8	32	40	128	10	20	1024 8	40	40	128	15	40	512 8	40	40	256	5	30	1024 8	40	48	256	5	40
512 10	48	48	128	5	15	1024 10	48	56	128	15	30	512 10	56	48	256	5	20	1024 10	56	64	256	10	30
512 12	64	64	256	5	10	1024 12	56	64	256	10	20	512 12	64	64	256	5	15	1024 12	72	80	256	5	20

fication on rugged landscapes. Globally, the most appropriate values of λ_{SW} 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.

The ideal number of neighbors to evaluate (λ_{IDa}) for ID_{any} is higher than for the other two PNLSSs, 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 ruggedness), indicating a higher need for intensification on such landscapes.

The most appropriate number of perturbations \mathcal{M} applied between the intensifying phases (descents) of ILS_F is higher on smooth landscapes. A more significant 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, enabling the search to explore new areas of the landscape. Lastly, adequate \mathcal{M} values often decrease when the number of solution evaluations increases, except on very smooth landscapes (when $K = 1$). On such landscapes, a high rate of diversification should be maintained to allow transitions between huge basins of attraction. For more rugged landscapes, the best value tested is generally the one of the smallest considered ($\mathcal{M} = \{1, 5\}$): a few perturbations are sufficient to increase the possibility of reaching unencountered good-quality optima (the number of local optima being higher, and their basin of attraction smaller).

Like \mathcal{M} values for ILS_F , the length of the tabu list that leads to the best average fitness is lower on rugged landscapes. On smooth landscapes, \mathcal{L} increases with N : the most appropriate length of the tabu list increases with the size of NK landscapes. There are tabu bits whose value cannot change, which ensures a distance of \mathcal{L} between two solutions separated by \mathcal{L} steps. A higher value induces a greater distance between solutions encountered during the search, and thus a greater diversification. Adequate values of \mathcal{L} are generally higher for a budget of 100 million evaluations. As observed with ILS_F parameter values, a higher parameter value seems more appropriate on smooth landscapes.

For UBQP, QAP, and FSP landscapes, the best settings for each method are reported in table 5. Results are more difficult to interpret since landscapes are of different structures and derived from various combinatorial problems. We restrict our analysis to PNLSS methods to simplify the reading of the paper.

Table 5

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

	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	Inst.	λ_{SW}	λ_{IDb}	λ_{IDa}	\mathcal{M}	\mathcal{L}	
UBQP	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
QAP	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
		sko42	78	86	861	5	20	sko72	2	170	852	5	32
	100 m.	lipa70a	8	185	1207	5	24	sko49	130	147	1176	5	16
		lipa80a	8	243	3160	5	12	sko56	32	128	1540	5	24
		lipa90a	16	250	4005	5	24	sko64	8	224	2016	5	24
		sko42	78	128	861	5	16	sko72	2	170	2556	5	24
FSP	1 m.	30 10 01	58	54	217	1	10	50 10 01	153	153	408	1	18
		30 15 01	96	108	435	1	6	50 15 01	153	163	612	1	16
		30 20 01	87	108	435	1	6	50 20 01	188	175	1225	1	14
	100 m.	30 10 01	145	145	435	1	12	50 10 01	188	153	350	1	16
		30 15 01	145	145	435	1	12	50 15 01	272	245	2450	1	12
		30 20 01	174	145	435	1	12	50 20 01	408	350	2450	1	12

UBQP landscapes: With a budget of 1 million evaluations, the best PNLs parameter values are generally stable on most landscapes. Only λ_{SW} fluctuates over three landscapes. On other landscapes, SW and ID_{best} systematically maximize their performance on similar settings. The optimal settings are similar to those observed on smooth NK landscapes in terms of optimal neighborhood proportions to consider.

When we consider a budget of 100 million evaluations, the best settings remain stable for each method on every UBQP landscape.

QAP landscapes: 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 approximately equivalent. For ID_{best} , this size is generally larger, which can be explained by the selection of the first improving encountered neighbor. This leads to a less intensifying search than SW , which selects the best-encountered neighbor (with a fixed sample size value). Considering a fixed instance size, the sample of neighbors that leads SW and ID_{best} to best results is generally smaller on *sko* landscapes than on *lipa* landscapes. The value which maximizes PNLs performance is probably not only related to the size and ruggedness of the landscapes.

As on the other landscapes, ID_{any} requires a huge sample of neighbors to obtain its best average fitness. On some landscapes, mainly *lipa* ones, this 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 an 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 ID_{any} requires an extended sample size to be effective, which goes away from the intent of PNLs.

For 100 million evaluations, SW and ID_{best} require a smaller λ value than for 1 million evaluations

on `lipa` landscapes. The value of λ for ID_{best} does not change on `lipa` landscapes but increases on `sko`, indicating a stronger need for intensification to achieve good-quality solutions on these landscapes. The most appropriate parameterization for ID_{any} does not evolve among `lipa` landscapes with this higher number of evaluations. This size increases on larger `sko` landscapes, indicating a limited need for diversification to drive the search towards good-quality solutions.

FSP landscapes: 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. On landscapes with a significant level of neutrality, a small λ value is sufficient to quickly navigate on 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 PNLSS. Likely to ID_{best} and SW , this value is negatively correlated with the neutrality rate.

The values of λ leading SW and ID_{best} to the best average fitness are generally higher when 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 the ones used when the budget equals 1 million evaluations. As with the other two PNLSS, the sample size is stable when $j = 30$, and also when $j = 50$.

The results of this section show that among the PNLSS, SW and ID_{best} often require similar settings to maximize their effectiveness. The appropriate size of the neighborhood sample seems to be positively correlated with the landscape ruggedness. ID_{any} requires a larger sample size than the other two PNLSS, 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 studied budgets. The convergence levels mostly vary according to the landscape properties and especially its size and ruggedness.

5. Empirical comparison of PNLSS

This section is dedicated to providing a comparative study of the three PNLSS versions. Results are also compared with ILS_F and TS results. We execute each method using the parameter value determined in the experiments reported in the previous section. We follow the experimental protocol described in the previous section.

The tables containing values indicate the best average fitness obtained by the considered methods on each landscape. For each pair (landscape, method), we report a percentage that corresponds to the deviation of the average fitness obtained by the method from the best average fitness obtained on the landscape. The dominance between the methods is reported following the color code described in Table 6.

For a landscape and a method A , the colors in the tables indicate the number of methods statistically dominated by A (shades of red) and the number of methods that dominate A (shades of blue) according to a binomial test (p -value < 0.05), as described in the previous section. Note that the p -value is defined for each comparison; therefore, this possibility of error is present for each pair of methods compared.

For each instance, we also report the difference (in %) between each method and the best average fitness obtained.

Table 6

Legend for tables of results reporting statistical dominance by a binomial test. Red shading indicates the number of times one method statistically dominates the other methods. Blue shading indicates the number of times a method is statistically dominated by other methods.

t	Dominates t methods					Dominated by t methods				
	4	3	2	1	0	0	1	2	3	4
color										

Table 7

Comparison of three PNLs variants, TS, and ILS on NK landscapes. Statistical dominance between methods and best average fitness achieved and deviations to this fitness for each method (in %). Left: budget of 1 million evaluations; Right: budget of 100 million evaluations.

instance	dominates				dominated				avgmax	Δ to avgmax					instance	dominates				dominated				avgmax	Δ to avgmax								
	SW	ID _B	ID _A	TS	ILS	SW	ID _B	ID _A		TS	ILS	SW	ID _B	ID _A		TS	ILS	SW	ID _B	ID _A	TS	ILS	SW		ID _B	ID _A	TS	ILS					
128 1									0.7245	0.04	0.06	0.01	1.35	0.00	128 1												0.7245	0.00	0.00	0.00	1.10	0.00	
128 2									0.7415	0.00	0.05	0.38	1.20	0.33	128 2												0.7424	0.00	0.00	0.04	0.74	0.02	
128 4									0.7922	0.03	0.10	0.00	4.94	0.13	128 4												0.7959	0.00	0.00	0.00	0.08	0.00	
128 6									0.7915	0.00	0.12	0.36	4.97	0.62	128 6												0.8004	0.00	0.01	0.11	0.67	0.08	
128 8									0.7883	0.00	0.08	0.58	4.05	0.73	128 8												0.8021	0.00	0.06	0.45	1.22	0.51	
128 10									0.7796	0.00	0.16	0.56	2.75	0.81	128 10												0.7936	0.00	0.16	0.49	1.36	0.53	
128 12									0.7689	0.00	0.27	1.03	2.13	0.90	128 12												0.7819	0.00	0.03	0.44	1.15	0.55	
256 1									0.7210	0.00	0.04	0.55	1.28	0.49	256 1												0.7220	0.01	0.00	0.29	1.41	0.27	
256 2									0.7430	0.00	0.08	0.68	3.00	0.68	256 2												0.7444	0.04	0.00	0.24	2.61	0.25	
256 4									0.7877	0.00	0.14	0.40	4.91	0.49	256 4												0.7933	0.00	0.00	0.92	1.39	0.18	
256 6									0.7921	0.05	0.00	0.68	2.33	0.58	256 6												0.8045	0.14	0.00	1.52	0.31	0.47	
256 8									0.7819	0.00	0.23	0.47	1.22	0.75	256 8												0.7961	0.12	0.00	0.57	0.83	0.66	
256 10									0.7741	0.00	0.33	0.63	0.65	1.07	256 10												0.7862	0.00	0.02	2.49	1.05	0.60	
256 12									0.7633	0.00	0.10	0.56	0.57	0.95	256 12												0.7754	0.02	0.00	2.56	1.25	0.57	
512 1									0.7044	0.00	0.08	0.59	0.52	0.51	512 1												0.7078	0.00	0.03	0.57	1.00	0.54	
512 2									0.7448	0.00	0.05	0.71	1.92	0.64	512 2												0.7509	0.00	0.00	0.77	2.57	0.74	
512 4									0.7751	0.00	0.12	0.84	1.31	0.72	512 4												0.7860	0.00	0.04	0.75	0.19	0.69	
512 6									0.7825	0.00	0.22	0.75	1.37	0.80	512 6												0.7989	0.00	0.06	0.57	0.30	0.61	
512 8									0.7781	0.00	0.36	0.76	1.18	0.86	512 8												0.7939	0.00	0.05	0.56	1.13	0.67	
512 10									0.7684	0.00	0.24	0.55	0.82	0.62	512 10												0.7829	0.00	0.05	0.49	0.86	0.62	
512 12									0.7581	0.00	0.05	2.89	0.59	0.38	512 12												0.7720	0.00	0.01	0.49	0.80	0.64	
1024 1									0.7127	0.00	0.12	1.10	1.07	1.09	1024 1													0.7163	0.00	0.03	1.11	1.56	1.05
1024 2									0.7474	0.00	0.13	1.29	2.70	1.19	1024 2													0.7522	0.00	0.01	1.26	3.30	1.26
1024 4									0.7749	0.00	0.24	1.28	3.22	1.33	1024 4													0.7878	0.00	0.07	0.99	2.84	1.03
1024 6									0.7762	0.00	0.23	0.90	2.57	0.04	1024 6													0.7949	0.00	0.07	0.70	0.63	0.74
1024 8									0.7711	0.00	0.22	0.75	2.15	1.39	1024 8													0.7901	0.00	0.17	0.54	0.65	0.65
1024 10									0.7625	0.00	0.26	0.55	1.73	0.67	1024 10													0.7793	0.00	0.09	0.46	0.68	0.52
1024 12									0.7535	0.00	0.15	0.64	1.41	0.93	1024 12													0.7694	0.00	0.06	0.39	0.53	0.49

5.1. NK landscapes

We present the statistical dominance relations between the local search variants and their average fitness on NK landscapes in the table 7. Considering 1 million evaluations, SW obtains the best average fitness on these landscapes and dominates other local searches in most cases. Although ID_{best} is close to SW in its behavior and ability to find good-quality solutions, it is regularly dominated by the latter.

The ability of SW and ID_{best} to achieve good solutions on average is closer when considering a budget of 100 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. SW and ID_{best} have an overall similar behavior, although it seems they converge toward solutions of similar quality after 100 million evaluations, SW reaches good solutions faster (on average). Another aspect tends to confirm this thought: as the size of landscapes increases, SW becomes rarely dominated by ID_{best}, and on the largest landscapes, ID_{best} is systematically dominated by SW. Lastly, there is no clear correlation between the comparative efficiency of SW and ID_{best}, and the ruggedness of landscapes.

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. A higher budget does not significantly increase its effectiveness compared to other local searches. ID_{any} is the least efficient variant among the PNLs, which must be due to the excessive

Table 8

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

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

diversification induced by the strategy while selecting deteriorating neighbors (i.e., randomly). On NK landscapes, its convergence seems particularly slow and could be a consequence of a poor intensification around areas of interest of the landscapes. Indeed, in such areas, the neighborhood only contains a few improving neighbors, and ID_{any} would tend to perform random moves too frequently, escaping from these promising areas. Nevertheless, this allows ID_{any} to be effective in so-called easier instances.

TS is the least effective local search on most landscapes, even if it dominates ILS_F on a few landscapes. It is also the only local search tested that does not systematically lead to the best encountered local optimum on the easiest landscape. ILS_F tends to be more efficient on smooth landscapes than on rugged landscapes where it is dominated many times. 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 reaches the best local optimum encountered, which is not the case for PNLs variants. On most landscapes, ILS_F often statistically dominates TS but rarely the three PNLs variants.

5.2. UBQP landscapes

Table 8 provides the statistical dominance between the methods as well as the average fitness obtained on UBQP landscapes. For a budget of 1 million evaluations, ILS_F globally leads 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, with extreme density values, the ability of SW and ILS_F to achieve good-quality solutions is equivalent. On the other two same-size landscapes, SW is less effective and statistically dominated by ID_{any}.

Considering PNLs variants, ID_{best} tends to outperform SW which tends to outperforms ID_{any}. TS is particularly ineffective in achieving good-quality solutions when considering a budget of 1 million evaluations. However, for 100 million evaluations, TS is the most efficient on the largest landscapes. Let us notice that in our study, considered UBQP landscapes are significantly larger than NK landscapes. The low quality of the solutions returned by TS after 1 million evaluations is probably due to the small budget compared to the size of the landscapes. Indeed TS is the local search that evaluates the most significant 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-quality solutions.

Table 9

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

FSP		SW	ID _B	ID _A	TS	ILS											
1 m.	30 10											avg _{min}	SW	ID _B	ID _A	TS	ILS
	30 15											1994.6	0.00	0.01	0.14	0.28	0.31
	30 20											2419.6	0.04	0.00	0.34	0.43	0.07
	50 10											2750.3	0.05	0.00	0.35	0.50	0.37
	50 15											3033.7	0.29	0.24	0.00	0.96	0.27
	50 20											3400.1	0.02	0.00	0.16	0.84	0.51
100 m.	30 10											3903.9	0.01	0.00	0.26	0.68	0.52
	30 15											1985.4	0.00	0.01	0.05	0.28	0.23
	30 20											2400.1	0.00	0.00	0.21	0.67	0.21
	50 10											2742.9	0.00	0.00	0.15	0.39	0.15
	50 15											3025.0	0.19	0.18	0.00	0.19	0.03
	50 20											3357.2	0.00	0.03	0.57	0.53	1.02
											3879.5	0.11	0.00	0.20	0.30	0.48	

Remember that the ruggedness indicators applied to UBQP landscapes indicate a locally rugged but globally smooth structure. This generally smooth structure tends to make them easier to solve than NK landscapes of equivalent size. On these landscapes, PNLs are efficient in quickly finding good solutions thanks to their high number of steps. However, the results for a higher budget indicate that these PNLs are less suitable than TS to intensify the search in relevant areas of these landscapes. TS is probably efficient on UBQP landscapes because their low level of diversification allows the search to stay near the main area of interest.

5.3. FSP landscapes

We present the statistical dominance between the proposed local searches and their average fitness on FSP landscapes in the table 9. SW and ID_{best} are never statistically comparable with each other. Even their average fitness values do not bring information on the possible superiority of one of these methods. These two methods are never statistically outperformed, except on the largest landscape with the highest rate of neutrality ($j = 50, m = 10$). On this landscape, ID_{any} is the most efficient method, although it is the least effective PNLs variant on the other FSP landscapes. SW and ID_{best} systematically dominate ILS_F, except on the landscapes with the highest rate of neutrality ($m = 10$) where the algorithms are statistically equivalent. This tendency could result from the significant rate of neutrality that prevents strict descents of ILS_F from reaching good-quality solutions. The high level of neutrality could explain why $\mathcal{M} = 1$ is always the best parameterization. Using a single random move as perturbation helps the ILS to efficiently explore plateaus. Although they are generally dominated by ID_{best} and SW, ILS_F and ID_{any} regularly dominate TS, which leads to the worst solutions on these landscapes.

Anyhow, the primary information is the global superiority of SW and ID_{best} variants over the other studied local searches. On these landscapes containing plateaus, PNLs are probably less constrained by neutrality, which would partially explain their higher efficiency. Unlike ILS_F, if a non-strict local optimum is reached, 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-quality solutions on FSP landscapes.

5.4. QAP landscapes

Table 10 reports the statistical dominance between the local searches proposed and their average fitness attained over QAP landscapes. On 1ipa landscapes, SW and ID_{best} statistically mostly dominate the other

Table 10

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

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

strategies. The average fitness values attained by ID_{best} slightly outperform those obtained by SW. Also, TS dominates ID_{any}, which sometimes dominates ILS_F.

On sko landscapes, ID_{any} reaches better solutions than other variants. On the smallest sko landscapes, the efficiency of ID_{best} and ID_{any} is similar. On the largest sko landscapes, ILS_F is the second most effective search (outperformed by ID_{any}). Finally, TS is the least efficient on landscapes sko.

We observe that the relative efficiency of PNLs variants on QAP landscapes dramatically depends on the type of considered instance. Nevertheless, results of PNLs variants often outperform TS and ILS_F on most of the instances considered.

PNLSs are consistently more efficient than ILS_F and TS on NK, QAP, and FSP landscapes for the two tested budgets. On UBQP landscapes, PNLs are competitive with a low budget since the partial neighborhood allows them to reach good-quality 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 efficiency. The selection strategy of deteriorating neighbors of ID_{any} induces a strong diversification that regularly prevents it from converging sufficiently toward good-quality solutions. Nevertheless, this strategy is suitable for some landscapes.

6. Conclusion

In this article, we studied partial neighborhood local search (PNLS) algorithms. In particular, we proposed the sample walk (SW) and compared it to the intensification/diversification walk (ID_{walk}), which is declined in two variants ID_{best} and ID_{any}. These PNLs were compared with two widely used local searches, the tabu search (TS) and the iterated local search (ILS). We aimed to observe the differences in the ability of PNLs to achieve good local optima compared to TS and ILS, as well as the differences induced by the variation of the pivoting rules on the search capacity to reach good-quality solutions.

To achieve this objective, we conducted an extensive empirical analysis of PNLs. In particular, we considered: (1) large instances taken from four combinatorial optimization problems: NK landscapes, UBQP, FSP, and QAP; (2) a parameterization analysis; (3) two computational budgets. Furthermore, all results were analyzed in the light of indicators characterizing the structure of fitness landscapes.

A significant part of the experiments was dedicated to the parameterization of PNLs. Apart from the stopping criterion, PNLs require to set a single parameter that directly affects the balance between

intensification and diversification of the search. We tested several parameter values to determine the right balance with respect to landscape characteristics. Generally, SW and ID_{best} require a similar parameterization to be efficient. The PNLs parameterizations that lead to the best solutions, on average, evolve slightly according to the budget and the ruggedness. Moreover, there is a broad range of values for which the PNLs are effective, suggesting that such algorithms are easy to configure automatically.

These experiments also highlighted the values that lead PNLs toward the best solutions on each landscape for both computational budgets, allowing us to compare more efficiently the different local search algorithms. Comparisons with different budgets show that considering a random sample of the neighborhood often allows PNLs to quickly reach good-quality solutions compared to ILS and TS. SW and ID_{best} often lead to solutions close in quality, even if one can observe some variations of efficiency on a few landscapes.

We can point out several research directions following this work. Improving our knowledge about the landscape structure would probably allow us to understand the unexplained differences in the behavior of PNLs. That is notably the case for permutation-based landscapes, where simple indicators are not sufficient to highlight the differences in landscape structure fully. A way of better characterizing landscapes could be by using network structures, such as local optima networks (Ochoa et al., 2014), to emphasize the characteristics that can prevent local search algorithms from reaching better solutions.

Since PNLs are efficient for different computational budget, it would be informative to assess their ability to reach good-quality solutions in an any-time optimization context (Zilberstein, 1996).

SW regularly leads to good-quality solutions and is a particularly simple local search for which many improvement possibilities exist. Integrating SW as a component of more sophisticated metaheuristics could quickly lead to diversified solutions of rather good quality. In general, a combination with exact methods could lead to much more effective intensification in the right areas of the landscapes.

References

- Barnett, L., 1998. Ruggedness and neutrality-the NKp family of fitness landscapes. In *Artificial Life VI: Proceedings of the sixth international conference on Artificial life*, MIT press, pp. 18–27.
- Basseur, M., Goëffon, A., 2013. Hill-climbing strategies on various landscapes: an empirical comparison. In *Proceedings of the 15th annual Conference on Genetic and Evolutionary Computation*, ACM, pp. 479–486.
- Basseur, M., Goëffon, A., 2014. On the efficiency of worst improvement for climbing NK-landscapes. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, pp. 413–420.
- Basseur, M., Goëffon, A., Lardeux, F., Saubion, F., Vigneron, V., 2014. On the attainability of NK landscapes global optima. In *Seventh Annual Symposium on Combinatorial Search*, AAAI Press, Palo Alto, California, pp. 28–34.
- Basseur, M., Goëffon, A., Traverson, H., 2015. Exploring non-neutral landscapes with neutrality-based local search. In *International Conference on Learning and Intelligent Optimization*, Springer, pp. 165–169.
- Baumann, O., Schmidt, J., Stieglitz, N., 2019. Effective search in rugged performance landscapes: A review and outlook. *Journal of Management* 45, 1, 285–318.
- Benlic, U., Hao, J.K., 2013. Breakout local search for the quadratic assignment problem. *Applied Mathematics and Computation* 219, 9, 4800–4815.
- Beyer, H.G., 1994. Towards a theory of evolution strategies: Results for $(1, +\lambda)$ -strategies on (nearly) arbitrary fitness functions. In *International Conference on Parallel Problem Solving from Nature*, Springer, pp. 57–67.
- Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J., 2009. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing* 8, 2, 239–287.
- Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)* 35, 3, 268–308.

- Burkard, R.E., Karisch, S.E., Rendl, F., 1997. QAPLIB—a quadratic assignment problem library. *Journal of Global Optimization* 10, 4, 391–403.
- Engelbrecht, A.P., 2007. *Computational Intelligence: An Introduction*. John Wiley & Sons.
- Ersoy, E., Özcan, E., Uyar, A.Ş., 2007. Memetic algorithms and hyperhill-climbers. In Springer (ed.), *Multidisciplinary International Conference on Scheduling: Theory and Applications*, pp. 159–166.
- Gary, M.R., Johnson, D.S., 1979. Computers and intractability: A guide to the theory of NP-completeness.
- Guerine, M., Rosseti, I., Plastino, A., 2020. A hybrid data mining heuristic to solve the point-feature cartographic label placement problem. *International Transactions in Operational Research* 27, 2, 1189–1209.
- Hansen, P., Mladenović, N., 2006. First vs. best improvement: An empirical study. *Discrete Applied Mathematics* 154, 5, 802–817.
- Hao, J.K., 2012. Memetic algorithms in discrete optimization. In *Handbook of Memetic Algorithms*. Springer, pp. 73–94.
- Hoos, H.H., Stützle, T., 2015. Stochastic local search algorithms: An overview. In *Handbook of Computational Intelligence*. Springer, pp. 1085–1105.
- Kauffman, S.A., Weinberger, E.D., 1989. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology* 141, 2, 211–245.
- Koopmans, T.C., Beckmann, M., 1957. Assignment problems and the location of economic activities, JSTOR. pp. 53–76.
- Kvitek, D.J., Sherlock, G., 2011. Reciprocal sign epistasis between frequently experimentally evolved adaptive mutations causes a rugged fitness landscape. *PLOS Genetics* 7, 4, e1002056.
- Liefoghe, A., Derbel, B., Verel, S., Aguirre, H., Tanaka, K., 2017. Towards landscape-aware automatic algorithm configuration: preliminary experiments on neutral and rugged landscapes. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, pp. 215–232.
- López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3, 43–58.
- Lourenço, H.R., Martin, O.C., Stützle, T., 2019. Iterated local search: Framework and applications. In *Handbook of Metaheuristics*. Springer, pp. 129–168.
- Malan, K.M., Engelbrecht, A.P., 2013. A survey of techniques for characterising fitness landscapes and some possible ways forward. *Information Sciences* 241, 148–163.
- Malan, K.M., Engelbrecht, A.P., 2014. Fitness landscape analysis for metaheuristic performance prediction. In *Recent advances in the theory and application of fitness landscapes*. Springer, pp. 103–132.
- Marmion, M.E., Dhaenens, C., Jourdan, L., Liefoghe, A., Verel, S., 2011. NILS: a neutrality-based iterated local search and its application to flowshop scheduling. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, pp. 191–202.
- Michalewicz, Z., 2012. Quo vadis, evolutionary computation? In *IEEE World Congress on Computational Intelligence*, Springer, pp. 98–121.
- Moura, A., 2019. A model-based heuristic to the vehicle routing and loading problem. *International Transactions in Operational Research* 26, 3, 888–907.
- Murata, T., Ishibuchi, H., Tanaka, H., 1996. Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering* 30, 4, 1061–1071.
- Neveu, B., Trombettoni, G., Glover, F., 2004. ID Walk: A candidate list strategy with a simple diversification device. In *International Conference on Principles and Practice of Constraint Programming*, Springer, pp. 423–437.
- Newman, M.E., Engelhardt, R., 1998. Effects of selective neutrality on the evolution of molecular species. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 265, 1403, 1333–1338.
- Ochoa, G., Tomassini, M., Vérel, S., Darabos, C., 2008. A study of NK landscapes’ basins and local optima networks. In *Proceedings of the 2018 Annual Conference on Genetic and Evolutionary Computation*, ACM, pp. 555–562.
- Ochoa, G., Verel, S., Daolio, F., Tomassini, M., 2014. Local optima networks: A new model of combinatorial fitness landscapes. In *Recent Advances in the Theory and Application of Fitness Landscapes*. Springer, pp. 233–262.
- Ochoa, G., Verel, S., Tomassini, M., 2010. First-improvement vs. best-improvement local optima networks of NK landscapes. *Parallel Problem Solving from Nature*, PPSN XI, 104–113.
- Palubeckis, G., 2004. Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research* 131, 1–4, 259–282.
- Pitzer, E., Affenzeller, M., 2012. A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent*

- Engineering Systems*. Springer, pp. 161–191.
- Ramkumar, A., Ponnambalam, S., Jawahar, N., 2009. A new iterated fast local search heuristic for solving QAP formulation in facility layout design. *Robotics and Computer-Integrated Manufacturing* 25, 3, 620–629.
- Rechenberg, I., 1984. The evolution strategy. a mathematical model of darwinian evolution. In *Synergetics – from Microscopic to Macroscopic Order*. Springer, pp. 122–132.
- Sahni, S., Gonzalez, T., 1976. P-complete approximation problems. *Journal of the ACM* 23, 3, 555–565.
- Sala, R., Müller, R., 2020. Benchmarking for metaheuristic black-box optimization: Perspectives and open challenges. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, pp. 1–8.
- Sánchez-Oro, J., Pantrigo, J.J., Duarte, A., 2014. Combining intensification and diversification strategies in VNS. an application to the vertex separation problem. *Computers & Operations Research* 52, 209–219.
- Sörensen, K., 2015. Metaheuristics — the metaphor exposed. *International Transactions in Operational Research* 22, 1, 3–18.
- Sorensen, K., Sevaux, M., Glover, F., 2018. *A History of Metaheuristics*, Springer. pp. 791–808.
- Srinivas, J., 2019. *Some Metaheuristic Optimization Schemes in Design Engineering Applications*, CRC Press. p. 1.
- Taillard, E., 1990. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research* 47, 1, 65–74.
- Taillard, E., 1993. Benchmarks for basic scheduling problems. *European Journal of Operational Research* 64, 2, 278–285.
- Tari, S., Basseur, M., Goëffon, A., 2017. Sampled walk and binary fitness landscapes exploration. In *International Conference on Artificial Evolution (Evolution Artificielle)*, Springer, pp. 47–57.
- Weinberger, E., 1990. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics* 63, 5, 325–336.
- Whitley, D., Howe, A., Hains, D., 2013. Greedy or not? best improving versus first improving stochastic local search for MAXSAT. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI Press, pp. 940–946.
- Wright, S., 1932. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth International Congress of Genetics*, Vol. 1, pp. 356–366.
- Xu, J., Zhang, J., 2014. Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis. In *Proceedings of the 33rd Chinese Control Conference*, IEEE, pp. 8633–8638.
- Yafrani, M.E., Martins, M.S., Krari, M.E., Wagner, M., Delgado, M.R., Ahiod, B., Lüders, R., 2018. A fitness landscape analysis of the travelling thief problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, pp. 277–284.
- Zilberstein, S., 1996. Using anytime algorithms in intelligent systems. *AI magazine* 17, 3, 73–83.