ALZA, J., BARTLETT, M., CEBERIO, J. and MCCALL, J. 2022. Analysing the fitness landscape rotation for combinatorial optimisation. In *Rudolph, G., Kononova, A.V., Aguirre, H., Kerschke, P., Ochoa, G. and Tušar, T. (eds.) Parallel problem solving from nature (PPSN XVII): proceedings of 17th Parallel problem solving from nature international conference 2022 (PPSN 2022), 10-14 September 2022, Dortmund, Germany*. Lecture notes in computer science, 13398. Cham: Springer [online], pages 533-547. Available from: https://doi.org/10.1007/978-3-031-14714-2_37

Analysing the fitness landscape rotation for combinatorial optimisation.

ALZA, J., BARTLETT, M., CEBERIO, J. and MCCALL, J.

2022

Analysing the Fitness Landscape Rotation for Combinatorial Optimisation

Joan Alza[1[0000−0002−7644−9993]], Mark Bartlett[1[0000−0003−3383−4100]], Josu Ceberio[2[0000−0001−7120−6338]], and John McCall[1[0000−0003−1738−7056]]

[1] National Subsea Centre, Aberdeen, AB21 0BH, Scotland
Robert Gordon University, Aberdeen, AB10 7GJ, Scotland
{j.alza-santos, m.bartlett3, j.mccall}@rgu.ac.uk
[2] University of the Basque Country UPV/EHU, Donostia, 20018, Spain
josu.ceberio@ehu.eus

Abstract. Fitness landscape rotation has been widely used in the field of dynamic combinatorial optimisation to generate test problems with academic purposes. This method changes the mapping between solutions and objective values, but preserves the structure of the fitness landscape. In this work, the rotation of the landscape in the combinatorial domain is theoretically analysed using concepts of discrete mathematics. Certainly, the preservation of the neighbourhood relationship between the solutions and the structure of the landscape are studied in detail. Based on the theoretical insights obtained, landscape rotation has been employed as a strategy to escape from local optima when local search algorithms get stuck. Conducted experiments confirm the good performance of the rotation-based local search algorithms to perturb the search towards unexplored local optima on a set of instances of the linear ordering problem.

Keywords: Landscape Rotation · Combinatorial Optimisation · Group Theory

1 Introduction

In the field of dynamic optimisation, there still remains the issue of translating real-world applications to academia [20]. Researchers often design simplified and generalised benchmark problems for algorithm development in controlled-changing environments, although they often omit important properties of real-world problems [12, 16]. In academia, dynamic problems are generally created using generators that introduce regulated changes to an existing static optimisation problem by means of adjustable parameters. The Moving Peaks Benchmark (MPB) [3] is probably the most popular generator, where a set of n parabolic peaks change in height, width and position in a continuous space $\mathbb{R}^n$.

In the combinatorial domain, the *landscape rotation* is presumably the most popular method to construct dynamic problems for academic purposes [2, 11, 18, 19, 21]. Introduced by Yang and Yao, the XOR dynamic problem generator [18, 19] periodically modifies the mapping between solutions and objective values by means of defined operators (the exclusive-or and the composition operators).

According to [16], the landscape rotation in the binary space progressively permutes the initial problem, preserving important properties of the problem, such as the landscape structure, stable. In fact, the wide use of this strategy comes from its preservation nature, as well as its simplicity for comparing algorithms in controlled-changing environments. However, despite its popularity, a theoretical analysis of the preservation of the landscape structure, and study further applications of this operation in combinatorial optimisation problems, beyond the binary space, are still lacking [16].

This work introduces an analysis of the fitness landscape rotation in combinatorial problems using notions of group and graph theory. The theoretical notations provided investigate the preservation of the neighbourhood relationship between solutions even when the landscape is rotated, and capture the repercussion of rotations in the permutation space. The study is supported by proofs and examples to demonstrate its validity.

Utilising the theoretical insights gained, we experimentally investigate different ways to employ the landscape rotation for the development of advanced local search algorithms. Particularly, the goal is to illustrate the applicability of the landscape rotation to perturb the search of the algorithm when it gets trapped. To that end, two rotation-based algorithms, obtained from [2], are employed and compared to study the exploratory profit of this strategy. Conducted experiments on a set of instances of the Linear Ordering Problem [5, 10] reveal the good performance of rotation-based local search algorithms, and also show the ability of the landscape rotation strategy to reach unexplored local optima. The results obtained are supported by visualisations that illustrate the behaviour of these algorithms by means of Search Trajectory Networks [13].

The remainder of the paper is structured as follows. Section 2 introduces background on the fitness landscape in the combinatorial domain, and provides important properties of the group theory. Section 3 explores the landscape rotation under group actions, and studies the repercussion of rotations in combinatorial fitness landscapes. Section 4 presents two rotation-based algorithms that are studied in the experimentation. Section 5 describes the experimental study, and discusses the applicability of the landscape rotation from the observed results. Finally, section 6 concludes the paper.

2 Background

This section introduces concepts to comprehend the rotation of the fitness landscape in the combinatorial domain, and presents some basics of group theory to analyse the rotation consequences in the next section.

2.1 Combinatorial Fitness Landscape

Formally, a combinatorial optimisation problem is a tuple $P = (\Omega, f)$, where Ω is a countable finite set of structures, called search space, and $f : \Omega \rightarrow \mathbb{R}$ is an objective function that needs to be maximised or minimised. Most combinatorial problems are categorised as NP-Hard [6], which means that there is no algorithm able to solve them in polynomial time. As a result, heuristic algorithms, and especially local search algorithms, have been widely used to solve combinatorial problems [8].

A key assumption about local search algorithms is the *neighbourhood* operator, which links solutions to each other through their similarity. Formally, a neighbourhood $\mathcal{N}$ is a mapping between a solution $x \in \Omega$ and a set of solutions $\mathcal{N}(x)$ after a certain operation in the encoding of x, such that

$$\mathcal{N} : \Omega \rightarrow \mathcal{P}(\Omega),$$

where $\mathcal{P}(\Omega)$ is the power set of Ω. In other words, two solutions x and y are neighbours when a modification in the encoding of x transforms it into y, so $x \in \mathcal{N}(y)$. The neighbourhood operator in combinatorial optimisation usually implies *symmetric* relations, meaning that any operation is reversible, i.e. $x \in \mathcal{N}(y) \Leftrightarrow y \in \mathcal{N}(x)$. This property naturally leads to define *regular* neighbourhoods, which implies the same cardinality of the neighbourhood of every solution in Ω, i.e. each solution has the same number of neighbours.

The fitness landscape in the combinatorial domain can be defined as the combination of combinatorial optimisation problems together with the neighbourhood operator [14]. Formally, the fitness landscape is a triple $(\Omega, f, \mathcal{N})$, where Ω is the search space, f is the objective function and $\mathcal{N}$ stands for the neighbourhood operator. The metaphor of the fitness landscape allows comprehending the behaviour of local search algorithms when solving a combinatorial problem, given a specific neighbourhood operator. In other words, the behaviour of local search algorithms, along with the suitability of different neighbourhood operators, can be studied based on properties of the fitness landscape, such as the number of local optima, global optima, basins of attraction or plateaus. These components are thoroughly described in the following paragraphs.

A local optimum is a solution $x^* \in \Omega$ whose objective value is better or equal than its neighbours' $\mathcal{N}(x^*) \in \Omega$, i.e. for any maximisation problem, $\forall y \in \mathcal{N}(x^*)$, $f(x^*) \geq f(y)$. The number of local optima of a combinatorial problem can be certainly associated to the difficulty of a local search algorithm to reach the global optimum (the local optimum with the best objective value) [7]. Nevertheless, there are other problem features, such as those explained in [15], that are also valid for understanding the dynamics of local search algorithms.

Some works in the combinatorial domain study the basins of attraction of local optima to shape the fitness landscape, and calculate the probability to reach the global optimum [7, 8, 17]. Formally, an attraction basin of a local optimum, $\mathcal{B}(x^*)$, is a set of solutions that lead to the local optimum x^* when a steepest-ascent hill-climbing algorithm is applied, so $\mathcal{B}(x^*) = \{x \in \Omega | a_x = x^*\}$, where a_x is the final solution obtained by the algorithm starting from x. Figuratively, an attraction basin $\mathcal{B}(x^*)$ can be seen as a tree-like directed acyclic graph, where the nodes are solutions, and the edges represent the steepest-ascent movement from a solution to a neighbour. This assumption leads to the following definition.

Definition 1 (Attraction graph). *Let us define an attraction graph to be a directed graph $\mathcal{G}_f(x^*) = (V, E)$, where f is the objective function, $V \subseteq \Omega$ is a set of solutions, and E is a set of directed edges representing the movement from a solution to a neighbour with a better, or equal, objective value. For every solution in the graph, there is an increasing path (sequence of solutions connected by directed edges) until reaching the local optima, such that*

$\forall x \in V, (x = a_1, a_2, \ldots, a_h = x^*)$, where $a_{i+1} \in \mathcal{N}(a_i)$, $(a_i, a_{i+1}) \in E$, and $f(a_i) \leq f(a_{i+1})$ for any maximisation problem.

The fitness landscape can be represented as the collection of all the attraction graphs, such that $O_f = \cup_{x^* \in \Omega^*} \mathcal{G}_f(x^*)$, where x^* is a local optimum of the set composed by local optima $\Omega^* \subset \Omega$, given a triple $(\Omega, f, \mathcal{N})$. Note that a solution (node) may belong to multiple attraction graphs if some neighbours, that belong to different attraction graphs, share the same objective value.

In the case that neighbouring solutions have equal objective values, we say that the landscape contains flat structures, called *plateaus*. Formally, a plateau $\Gamma \subseteq \Omega$ is a set of solutions with the same objective value, such that for every pair of solutions $x, y \in \Gamma$, there is a path $(x = a_1, a_2, \ldots, a_k = y)$, where $a_i \in \Gamma, a_{i+1} \in \mathcal{N}(a_i)$ and $f(a_i) = f(a_{i+1})$. The authors in [8] demonstrated that combinatorial problems often contain plateaus, and remark the importance of considering plateaus when working with problems in the combinatorial domain. The authors also differentiate three classes of plateaus, and point out that a plateau composed by multiple local optima can be considered as a single local optimum when applying local search-based algorithms, as their basins of attraction lead to the same plateau.

2.2 Permutation Space

One of the most studied fields in combinatorial optimisation is the permutation space, where solutions of the problem are represented by permutations. Formally, a permutation is a bijection from a finite set, usually composed by natural numbers $\{1, 2, \ldots, n\}$, onto the same set. The search space Ω represents the set of all permutations of size n, called *symmetric group* and denoted as $\mathbb{S}_n$, whose size is $n!$. Permutations are usually denoted using $\sigma, \pi \in \mathbb{S}_n$, except for the identity permutation $e = 12...n$. We direct the interested reader to [4] for a deeper analysis of permutation-based problems.

The similarity between permutations can be specified by permutation distances. The distance between two permutations is the minimum number of operations to convert one permutation into another. Irurozki in [9] studies the Kendall's-τ, Cayley, Ulam and Hamming distance metrics, and suggests some methods to generate new permutations uniformly at random for each distance metric. It is worth mentioning that each distance metric has its own maximum and minimum distances, d_{min} and d_{max}. We direct the interested reader to [9] for more details on permutation distances.

2.3 Landscape Rotation

The fitness landscape rotation has been probably the most popular benchmark generator, for academic purposes, in the combinatorial domain. Introduced as the XOR dynamic problem generator [18, 19], this method periodically applies the rotation operation to alter the mapping between solutions and objective values by means of the exclusive-or (rotation) operator. Formally, given a static binary problem, a rotation degree ρ and the frequency of change τ, the objective value of a solution $x \in \Omega$ is altered by

$$f_t(x) = f(x \oplus M_t),$$

where f_t is the objection function at instance $t = \lceil \frac{i}{\tau} \rceil$, i is the iteration of the algorithm, f is the original (static) objective function, '$\oplus$' is the exclusive-or operator and $M_t \in \Omega$ is a binary mask. The mask M_t is incrementally generated by $M_t = M_{t-1} \oplus T$, where T is a binary template randomly generated containing $\lfloor \rho \times n \rfloor$ number of ones. The initial mask is a zero vector, $M_1 = \{0\}^n$.

Some works in the literature extended the XOR dynamic problem generator to the permutation space [2, 21, 11]. The landscape rotation in the permutation space can be represented as

$$f_t(\sigma) = f(\Pi_t \circ \sigma),$$

where $\sigma \in \mathbb{S}_n$ is a solution, '$\circ$' is the composition operation and Π_t is a permutation mask. The permutation mask is incrementally generated by $\Pi_t = \Pi_{t-1} \circ \pi$, where π is a permutation template generated using the methods in [9], containing $\lfloor d_{max} \times \rho \rfloor$ operations from the identity permutation given a permutation distance. The permutation mask is initialised by the identity permutation, $\Pi_1 = e$.

According to Tínos and Yang [16], the XOR dynamic problem generator changes the fitness landscape according to a permutation matrix, where the neighbourhood relations between solutions are maintained over time. However, as far as we are concerned, these assumptions have never been studied in the permutation space. Hence, this is the motivation of this work.

2.4 Group Theory

The landscape rotation can be represented by group actions, where the search space along with the rotation operation satisfy certain properties. Formally, given a finite set of solutions Ω and a group operation '$\cdot$', $G = (\Omega, \cdot)$ is a group if the closure, associativity, identity, and invertibility properties are satisfied. Mathematically, these fundamental group properties (axioms) are defined as:

- **Closure:** $x, y \in G, x \cdot y \in G$.
- **Associativity:** $x, y, z \in G$, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.
- **Identity:** $i \in G, \forall x \in G$, $x \cdot i = i \cdot x = x$.
- **Invertibility:** $x, x^{-1} \in G, x \cdot x^{-1} = x^{-1} \cdot x = i$.

There is another property, the **commutativity**, that is not fundamental for the definition of a group. A group is said to be commutative when $x, y \in G$, $x \cdot y = y \cdot x$. It is worth mentioning that the commutation property holds in the binary space, but it does not in the permutation space.

3 Analysis of the Fitness Landscape Rotation

In this section, we aim to theoretically analyse some important consequences of rotating the landscape in the combinatorial domain using group properties. To that end, we will study the (i) the neighbourhood relation preservation after rotating the landscape, and (ii) the repercussion of rotations in landscapes encoded by permutations.

3.1 Neighbourhood Analysis

The properties of the fitness landscape rotation can be studied using notions of group theory, i.e. the rotation (exclusive-or '$\oplus$' and composition '$\circ$') operators can be generalised to the group operation '$\cdot$'. Following previous notations, we can demonstrate the preservation of neighbourhood relations between solution before and after a rotation without loss of generality.

Theorem 1. *Given a group G, let $\mathcal{N}(x) \in G$ be the neighbourhood of x and $t \in G$ the mask used to rotate the space. We say that the neighbourhood relations are preserved iff $\mathcal{N}(t \cdot x) \Leftrightarrow t \cdot \mathcal{N}(x)$.*

Proof. Let $x, y \in G$ be two neighbouring solution in the group, such that $x \in \mathcal{N}(y)$ is a neighbour of y (and vice versa). We can define the neighbourhood operation as $c(i,j) \cdot x \in \mathcal{N}(x)$, where $c(i,j)$ is an operation in the encoding of a solution. For example, $c(i,j)$ can represent the swap of the elements i and j, or the insertion of the element at position i into the position j.

Based on this notation, we can define the fundamental group properties (identity, invertibility and associativity) as

$$c(i,j) \cdot e = e \cdot c(i,j) = c(i,j) \tag{1}$$

$$c(i,j) \cdot c(i,j)^{-1} = c(i,j)^{-1} \cdot c(i,j) = e \tag{2}$$

$$x = c(i,j) \cdot x \cdot c(i,j)^{-1} \tag{3}$$

Note that $c(i,j)^{-1} = c(j,i)$ represents the inverse operation of $c(i,j)$. Assuming that the landscape is rotated using the template $t \in G$, we can say that Ω and $t \cdot \Omega$ are defined independently. In the following, we aim to prove that $t \cdot \mathcal{N}(x) \Leftrightarrow \mathcal{N}(t \cdot x)$.

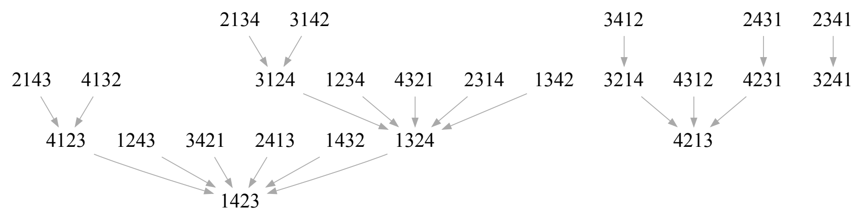
First, we must ensure that the rotation of the landscape preserves the neighbourhood relation between solutions, so $\mathcal{N}(t \cdot x) \subset t \cdot \mathcal{N}(x)$. It can be demonstrated in the following way.

$$\begin{aligned} x &\in \mathcal{N}(y) \\ x &= c(i,j) \cdot y \\ t \cdot x &= t \cdot c(i,j) \cdot y \\ c(i,j)^{-1} \cdot t \cdot x &= c(i,j)^{-1} \cdot t \cdot c(i,j) \cdot y \\ c(i,j)^{-1} \cdot t \cdot x &= t \cdot y \end{aligned} \tag{4}$$

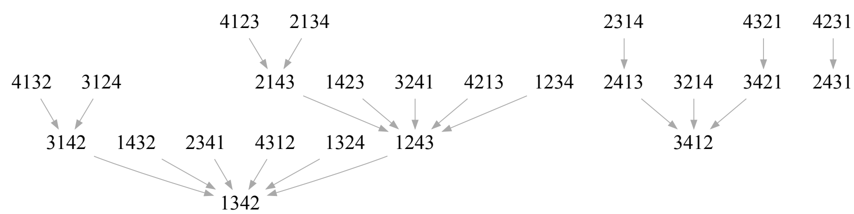
Note that the last step in the equation is given by the Equation 3. Considering that $c(i,j)^{-1} = c(j,i)$ and $c(i,j) \cdot x \in \mathcal{N}(x)$, we can say that $c(i,j)^{-1} \cdot t \cdot x \in \mathcal{N}(t \cdot x)$. Therefore, given the symmetry of the neighbourhood operator, we can prove that the rotation of the neighbourhood is a subset of the neighbourhood of the rotated $t \cdot x$, so $t \cdot y \in \mathcal{N}(t \cdot x)$.

Then, we must prove the inverse statement, i.e. the rotated neighbourhood derives in the rotation of the neighbourhood, $t \cdot \mathcal{N}(x) \subset \mathcal{N}(t \cdot x)$. It can be demonstrated in the following way.

$$\begin{aligned} &x \in \mathcal{N}(t \cdot y) \\ &x = c(i,j) \cdot t \cdot y \\ &x = c(i,j) \cdot [c(i,j)^{-1} \cdot t \cdot c(i,j)] \cdot y \\ &x = e \cdot t \cdot c(i,j) \cdot y \end{aligned} \tag{5}$$



(a) Original landscape



(b) Landscape rotated by $t = 1342$

Fig. 1: Illustrative visualisation of the landscape structure before and after a rotation.

In this case, after considering $c(i,j) \cdot x \equiv \mathcal{N}(x)$, we use the inverse property of the neighbourhood operation (Equation 3) and the identity property (Equation 2) to demonstrate that $x = t \cdot c(i,j) \cdot y$, so $x \in t \cdot \mathcal{N}(y)$. Therefore, we can prove that $x \in t \cdot G$ derives in $x \in G$.

In summary, by showing the symmetry of the rotation operation, we can confirm that the neighbourhood relations between solutions are preserved.

3.2 Repercussion of the Landscape Rotation

The authors in [16] mention that the topological structure of the fitness landscape must be analysed to comprehend the behaviour of the algorithms. In order to study the repercussion of changes in the space, we will use the definition of the attraction graphs (Definition 1) to represent the topological features of the fitness landscape. In the following, we will consider the permutation space ($\mathbb{S}_n$), the swap operation (2-exchange operator) and the steepest-ascent hill-climbing algorithm (saHC) to represent the fitness landscape, and more precisely, the collection of attraction graphs.

Note that the preservation of the neighbourhood relations (previous section) is independent of the algorithm and objective function. Thus, we can demonstrate that the landscape structure is preserved even if solutions are rearranged. Using graph theory notations, we could point out that attraction graphs are *isomorphic* (structurally equivalent) to themselves in the rotated environment, such that $O_f \cong O_{f_t}$, where O_{f_t} is the set of attraction graphs that composes the

Table 1: Example of the number of exchanges between attraction graphs of all possible rotation masks generated by the Cayley distance metric.

t (d_C)	**Exchanges**	t (d_C)	**Exc.**	t (d_C)	**Exc.**
1234 (0)	0 (original)	1423 (2)	10	4213 (2)	13
1243 (1)	4	2143 (2)	16	4321 (2)	14
1324 (1)	8	2341 (2)	14	2314 (3)	12
1432 (1)	8	2431 (2)	14	2413 (3)	12
2134 (1)	16	3124 (2)	12	3142 (3)	12
3214 (1)	12	3241 (2)	13	3421 (3)	13
4231 (1)	14	3412 (2)	12	4123 (3)	14
1342 (2)	10	4132 (2)	14	4312 (3)	13

rotated fitness landscape f_t. Despite the conservation of the landscape structure, all solutions are mapped to different positions when the landscape is rotated. Let us illustrate these concepts with the following example.

Figure 1 displays the landscape of a permutation problem of size $n = 4$ as a collection of attraction graphs produced by a given objective function f. These images illustrate the preservation of the landscape structure. For example, the attraction graph on the right side of both images will always contain two solutions, such that $\mathcal{G}_f(3241)$ and $\mathcal{G}_{f_t}(1324 \circ 3241 = 2431)$ are isomorphic.

It is worth noting that the landscape rotation alters the mapping between solutions and objective values. Hence, we can conclude that, since solutions are rearranged at different positions in the fitness landscape, the objective values are preserved. In other words, the fitness landscapes before and after a rotation, O_f and O_{f_t}, are *equal* in terms of objective values, e.g. $f(3241) = f_t(2431)$.

In order to measure the impact of the rearrangement after a rotation, we can use the total number of solution exchanges between attraction graphs. This assumption is motivated by the fact that, for a local search-based algorithm, it is more likely to "escape" from an attraction graph when a rotation implies a big number of solution exchanges between graphs. Continuing with the previous example, Table 1 summarises the total number of solution exchanges between attraction graphs for all the possible rotations generated at a given Cayley distance (d_C). This distance metric uses the smallest number of swaps assumption to generate permutations uniformly at random [1]. The table entries demonstrate that the rotation degree, measured as the Cayley distance, is not necessarily proportional to the number of solution exchanges between attraction graphs in the permutation space. For example, the average number of solution exchanges for each Cayley distance in Table 1 reflects that rotating to $d_C = 1$ produces 10.3 exchanges, $d_C = 2$ produces 12.9 exchanges and $d_C = 3$ produces 12.6 exchanges, on average, respectively. Hence, the use of permutation distances as a rotation degree should be used with caution, since medium rotations can be severe, in terms of the total number of solution exchanges between attraction graphs.

4 Rotation as a Perturbation Strategy

From the theoretical insights gained from the previous section analysis, we suggest the landscape rotation as a perturbation strategy for local search-based algorithms to react when algorithms get trapped in poor quality local optima. The rotation action can be used to relocate a stuck algorithm's search, ideally into a different attraction graph, by means of the permutation distance that controls the magnitude of the perturbation. Note that, generally, a local optimum in the original landscape is not mapped to a local optimum in the rotated landscape. This assumption motivates its usage to reach unexplored local optima. In short, rotation-based local search algorithms can be summarised as follows:

1: Run the local search algorithm until reaching a local optimum, x^*.
2: The rotation operation is applied to relocate the algorithm at the solution $t \cdot x^*$. Ideally, the algorithm will reach a new local optimum, such that $t \cdot x^* \subseteq \mathcal{G}_{f_t}(y^*)$.
3: This process is repeated until meeting the stopping criterion.

In [2], we presented two rotation-based perturbation strategies: a depth-first and a width-first strategy. These methods differ in the way they use the rotation operation. If we apply these strategies into the steepest-ascent hill climbing algorithm (saHC), the depth-first algorithm (saHC-R1) will use the rotation operation to move the search away, and continue the search from a new position until getting trapped again. Then, both local optima are compared, and the search is relocated to the best found solution. On the other hand, the width-first algorithm (saHC-R2) applies the rotation operation for some iterations, and then continues to search from a new position (undo the rotation) until it gets stuck again. Unlike saHC-R1, this strategy does not relocate the search to the best solution found, which encourages continuous exploration of different graphs. For more details, see [2].

5 Results and Discussion

This section evaluates the performance of the proposed strategies to solve the Linear Ordering Problem (LOP) [5, 10]. This problem aims to maximise the entries above the main diagonal of a given matrix $B = [b_{i,j}]_{n \times n}$. The objective is to find a permutation σ that orders the rows and columns of B, such that

$$\arg\max_{\sigma \in \mathbb{S}_n} f(\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{\sigma_i, \sigma_j}.$$

The specific instances used in the experimentation of this work are obtained from the supplementary material web[3] presented in [8]. The web contains 12 LOP instances: eight instances of size 10, and four instances of size 50. The parameters employed in the rotation-based algorithms [1] are summarised in Table 2. Note that the rotation degree is designed to start with big rotations, and exponentially decreasing it to intensify the search. The motivation of this

[3] `http://www.sc.ehu.es/ccwbayes/members/leticia/AnatomyOfAB/instances/InstancesLOP.html`

Table 2: Parameter settings for the experimental study.

Parameter	Value
Rotation degree	$d = \left\vert d_{max} e^{\left(\frac{\ln\left(\frac{d_{min}}{d_{max}}\right)}{S}\right) i} \right\vert$
Distance metric	Cayley distance
Number of repetitions	30
Stopping criterion	$10^3 n$ iterations

strategy is to balance the intensification-diversification trade off based on the search process of the algorithm.

Obtained results and instance properties are summarised in Table 3, where the best found objective values, the number of rotations and the number (and percentage) of visited local optima are shown for each algorithm. The total number of local optima for each instance has been obtained from [8]. However, due to the incompleteness of the number of local optima for instances of size $n = 50$, we only show the number of local optima explored for these instances, without showing the percentages of local optima explored.

The results show the good performance of rotation-based algorithms, as well as their explorability ability. Both algorithms are able to find the same optimal solutions (except for N-be75oi), in terms of objective values, but they differ in the number of rotations and local optima explored. The percentages represent the ability of the algorithms to explore the attraction graphs that compose the landscape. saHC-R2 always performs more rotations than saHC-R1, and thus, it can find a larger (or the same) number of local optima than saHC-R1. Therefore, we can say that saHC-R2 tends to be more exploratory than saHC-R1.

In order to illustrate the influence of the rotation degree on the search of the algorithms, Table 4 shows the number of rotations and local optima reached by each algorithm, over the 30 runs, for each Cayley distance, on Instance 8.

Table 3: Information of the instances, and results of the rotation-based algorithms on LOP instances. Percentages for instances of size $n = 50$ are not available, since their total number of local optima is unknown.

Instance	LO	saHC-R1			saHC-R2		
		Obj. Value	Rotations	LO (%)	Obj. Value	Rotations	LO (%)
Instance 1	13	1605	2015	13 (100%)	1605	2636	13 (100%)
Instance 2	24	1670	2011	24 (100%)	1670	2629	24 (100%)
Instance 3	112	4032	1956	104 (92.8%)	4032	2545	106 (94.6%)
Instance 4	129	3477	1988	121 (93.8%)	3477	2565	125 (96.9%)
Instance 5	171	32952	2093	169 (98.8%)	32952	2712	171 (100%)
Instance 6	226	40235	1954	215 (95.1%)	40235	2571	220 (97.3%)
Instance 7	735	22637	2138	683 (92.9%)	22637	2742	716 (97.4%)
Instance 8	8652	513	2528	6063 (70%)	513	3351	6887 (79.6%)
N-be75eec	> 500	236464	8527	62143 (−%)	236464	10737	75404 (−%)
N-be75np	> 500	716994	8306	36046 (−%)	716994	10364	58423 (−%)
N-be75oi	> 500	111171	8507	80001 (−%)	111170	10698	96371 (−%)
N-be75tot	> 500	980516	8437	31550 (−%)	980516	10606	53450 (−%)

Table 4: Number of rotations and reached local optima by each algorithm on Instance 8.

		Cayley distance								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
saHC-R1	**Rotations**	19594	19304	10851	7416	5623	4525	3778	3279	1452
	LO	678	1968	2500	2679	2725	2633	2438	2234	1227
saHC-R2	**Rotations**	27680	26090	14191	9432	7039	5633	4676	4027	1746
	LO	736	3809	4014	3719	3337	3037	2761	2528	1346

Remember that the rotation degree describes an exponential decrease as the search progresses[4].

The table shows that, although the number of rotations exponentially decays, the highest exploratory behaviour of the algorithms holds on medium-small distances, i.e. both algorithms find more local optima when the rotation operates at $d_C = \{3, 4, 5\}$. This performance matches with the example in Table 1, where rotating to medium distances is sufficient to perturb the search of algorithms to different attraction graphs.

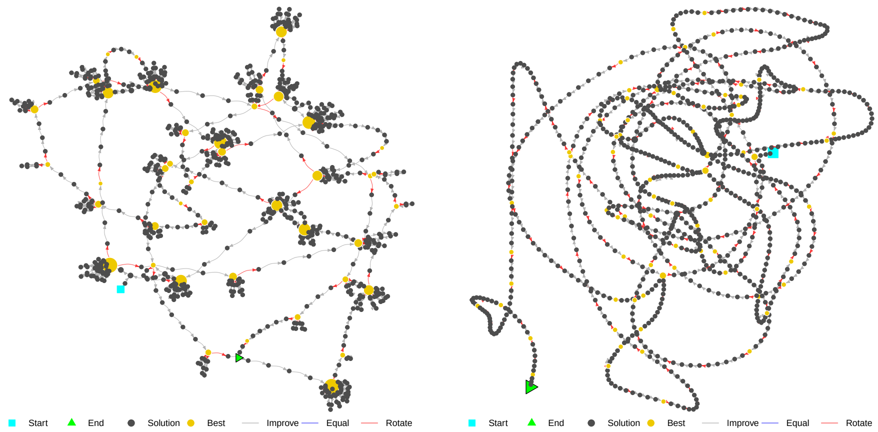
In order to visually represent and analyse the evolution of the algorithms, we use the Search Trajectory Networks (STNs) tool [13], a directed-graph-based model that displays search spaces in two or three dimensions. Figure 2 displays a single run of each algorithm on Instance 8 using STNs. The colours in the figures highlight the starting and ending points of the search (blue and green nodes), the best found solutions (yellow nodes) and the rotation operations (red edges), respectively. The entire experimentation is available online[5].

The plots show the behaviour of each algorithm in a two-dimensional space. The left plot shows the behaviour of saHC-R1, where the algorithm always applies the rotation operation from the best solution found. This visualisation gives an insight of the structure of the landscape, since the algorithm is able to explore the paths that compose the attraction graphs. On the other hand, Figure 2b shows a continuous search of saHC-R2, where the algorithm moves through attraction graphs, meaning that it rarely gets stuck on the same local optimum after a rotation. This behaviour can be comprehended by the fact that saHC-R2 does not rotate from the best solution found, but from the last local optimum found. That said, we can conclude that saHC-R1 outperforms in instances with few but deep attraction graphs, while saHC-R2 outperforms in instances composed of many attraction graphs.

Finally, it is worth noting the presence of a plateau composed of local optima in Instance 8, i.e. multiple local optima have the same objective value, which turns out to be the optimal value. The figures confirm that both algorithms can detect and deal with plateaus. Interestingly, Figure 2a shows that some local optima that form the plateau are visibly larger, which means that saHC-R1 visits them several times. From previous statements, we can deduce that Instance 8 is composed of neighbouring local optima with the same objective value, and also that saHC-R2 reaches more local optima than saHC-R1.

[4] The repercussion of the rotation degree in other instances is available online.

[5] `https://zenodo.org/record/6406825\#.YkcxaW7MI-Q`



(a) saHC-R1 **(b)** saHC-R2

Fig. 2: Search Trajectory Networks of rotation-based algorithms on Instance 8.

6 Conclusions and Future Work

Landscape rotation has been widely used to generate dynamic problems for academic purposes due to its preserving nature, where important properties of the problem are maintained. In this article, we study the preservation of the landscape structure using group actions, and based on the insights gained, we suggest using the rotation operation to relocate the search of local search-based algorithms when they get stuck. The experiments carried out show the good application of the landscape rotation to perturb the search of the local search algorithm through unexplored local optima. Obtained results also illustrate that 'medium' rotations can cause a big repercussion, when it comes to the number of rearranged solutions.

This work can be extended in several ways. First and most obvious, there are some landscape properties that have been ignored in this manuscript, such as the number and size of attraction graphs, or the frontier and the centrality of local optima [8]. These assumptions, along with the consideration of problem properties, such as the symmetries of the problem instance, could lead to very different outcomes, where the landscape rotation may be less applicable. Finally, this work has considered the swap operator and the steepest-ascent hill-climbing algorithm to construct the attraction graphs in the Linear Ordering Problem. This study can be naturally extended to other combinatorial optimisation problems, as well as other ways to represent the landscape, such as the insertion operation or the first-improvement hill-climbing heuristic.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation (PID2019-106453GA-I00/AEI/10.13039/501100011033), and the Elkartek Program (SIGZE, KK-2021/00065) from the Basque Government.

References

1. Alza, J., Bartlett, M., Ceberio, J., McCall, J.: On the definition of dynamic permutation problems under landscape rotation. In: Proc. GECCO. p. 1518–1526 (2019)
2. Alza, J., Bartlett, M., Ceberio, J., McCall, J.: Towards the landscape rotation as a perturbation strategy on the quadratic assignment problem. In: Proc. GECCO. p. 1405–1413 (2021)
3. Branke, J.: Evolutionary optimization in dynamic environments. Springer Science & Business Media (2002)
4. Ceberio, J.: Solving Permutation Problems with Estimation of Distribution Algorithms and Extensions Thereof. Ph.D. thesis, UPV/EHU (2014)
5. Ceberio, J., Mendiburu, A., Lozano, J.A.: The linear ordering problem revisited. EJOR **241**(3), 686–696 (2015)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co. (1979)
7. Hernando, L., Mendiburu, A., Lozano, J.A.: An evaluation of methods for estimating the number of local optima in combinatorial optimization problems. Evolutionary Computation **21**(4), 625–658 (2013)
8. Hernando, L., Mendiburu, A., Lozano, J.A.: Anatomy of the attraction basins: Breaking with the intuition. Evolutionary Computation **27**(3), 435–466 (2019)
9. Irurozki, E.: Sampling and learning distance-based probability models for permutation spaces. Ph.D. thesis, UPV/EHU (October 2014)
10. Martí, R., Reinelt, G.: The linear ordering problem: exact and heuristic methods in combinatorial optimization, vol. 175. Springer Science & Business Media (2011)
11. Mavrovouniotis, M., Yang, S., Yao, X.: A benchmark generator for dynamic permutation-encoded problems. In: PPSN. pp. 508–517 (2012)
12. Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation pp. 1 – 24 (2012)
13. Ochoa, G., Malan, K.M., Blum, C.: Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics. Applied Soft Computing (2021)
14. Reidys, C.M., Stadler, P.F.: Combinatorial Landscapes. Working Papers 01-03-014, Santa Fe Institute (Mar 2001)
15. Tayarani-N., M.H., Prügel-Bennett, A.: On the landscape of combinatorial optimization problems. IEEE TEVC **18**(3), 420–434 (2014)
16. Tinós, R., Yang, S.: Analysis of fitness landscape modifications in evolutionary dynamic optimization. Information Sciences **282**, 214–236 (2014)
17. Verel, S., Daolio, F., Ochoa, G., Tomassini, M.: Local Optima Networks with Escape Edges. In: International Conference on Artificial Evolution. pp. 10 – 23 (2011)
18. Yang, S.: Non-stationary problem optimization using the primal-dual genetic algorithm. In: Proc. CEC. vol. 3, pp. 2246–2253 (2003)
19. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Comput. p. 815–834 (2005)
20. Yazdani, D., Cheng, R., Yazdani, D., Branke, J., Jin, Y., Yao, X.: A survey of evolutionary continuous dynamic optimization over two decades—part b. IEEE TEVC **25**(4), 630–650 (2021)
21. Younes, A., Calamai, P., Basir, O.: Generalized benchmark generation for dynamic combinatorial problems. In: Proc. GECCO. pp. 25–31 (2005)