Search moves in the local optima networks of permutation spaces: the QAP case

Marco Baioletti
University of Perugia
Perugia, Italy
marco.baioletti@unipg.it

Alfredo Milani
University of Perugia
Perugia, Italy
alfredo.milani@unipg.it

Valentino Santucci
University for Foreigners of Perugia
Perugia, Italy
valentino.santucci@unistrapg.it

Marco Tomassini
University of Lausanne
Lausanne, Switzerland
marco.tomassini@unil.ch

ABSTRACT

In this work we analyze, from a qualitative point-of-view, the structure of the connections among the local optima in the fitness landscapes of the Quadratic Assignment Problem (QAP). In particular, we are interested in determining which search moves, intended as pairwise exchanges of permutation items, are beneficial for moving from one optimum to another. Novel algebraic methods are introduced for determining, and measuring the effectiveness, of the exchange moves connecting two given optima. The analysis considers real-like QAP instances whose local optima networks are clustered in communities. The results of the conducted experimentation shows the presence of few preferred search moves that look more effective for moving across intra-community optima, while the same is not so apparent when the optima are taken from different communities.

CCS CONCEPTS

• **Computing methodologies** → **Discrete space search**; **Randomized search**;

KEYWORDS

Fitness Landscape Analysis, Algebraic Evolutionary Computation, Quadratic Assignment Problem

ACM Reference Format:
Marco Baioletti, Alfredo Milani, Valentino Santucci, and Marco Tomassini. 2019. Search moves in the local optima networks of permutation spaces: the QAP case. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3319619.3326849

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3326849

1 INTRODUCTION

Hard combinatorial optimization problems can often be efficiently solved by using metaheuristic approaches, if one is ready to give up strict global optimality. The solutions of these problems can often be naturally represented by permutations of objects such as graph vertices, jobs to be scheduled on given machines, colors of graph nodes, and many others. As a consequence, metaheuristics often manipulate problem solutions represented as permutations and thus it becomes important to investigate the relationships between operations on permutations and their effectiveness in searching a given problem instance solution space. In this study, we use the *Local Optima Networks* (LONs) [21] in order to have a compact representation of a given problem instance fitness landscape. LONs are weighted networks in which vertices represent the local optima of the fitness landscape and the arcs represent probabilities of transitions between optima through their respective basins of attraction. Using the Quadratic Assignment Problem (QAP) as an example, we first generate a number of LONs from the corresponding problem instances and then we partition the LONs, which are complex networks, into the corresponding communities of optima. These communities are used later to determine efficient search moves in the permutation space.

Indeed, the main goal of this work is to verify the existence of – and determine the – preferred search moves that allow to move across different basins of attractions. In particular, we are interested to the landscapes of the QAP using pairwise exchange of items as search moves.

In order to perform such analysis, we start from the algebraic interpretation of combinatorial search spaces first introduced in [26–28] and further investigated in [2, 4, 7, 25]. Here, we extend this algebraic framework by proposing a method for computing the pairs of items – and not only their positions (as previously done in [2]) – to be exchanged in order to move from one permutation to another by using a minimal number of exchange moves. A concise, and quick to compute, representation of such a set of pairs of items is introduced together with a numeric computation of their importance in moving between two given permutations.

These algebraic tools are then applied to the local optima belonging to the considered LONs. By exploiting the clustered structure of the LONs, both an intra-community and an inter-community analyses have been designed. The former aims to detect the emergence of preferred search moves that allow to traverse the basins of the optima belonging to a given community of the LON, while the latter is intended to analyze the connections in the pairs of optima belonging to different communities.

The results of the conducted experimentation show the presence of more effective search moves connecting intra-community optima, while the same observation is much less evident in the inter-community analysis.

The paper is structured as follows. In the next section, we describe the LON concept and the partition of LONs graphs into communities of optima. Section 3 briefly describes the Quadratic Assignment Problem. Next, we present the algebraic representation of the permutation space and the way in which search moves are computed. Section 5 describes the conducted experimentation and the main results obtained. Finally, Section 6 concludes the paper by also drawing future lines of research.

2 LOCAL OPTIMA NETWORKS

The local optima network (LON) model for combinatorial landscapes was first proposed in [21], with follow up work appearing in [23, 32]. In this network-based model, vertices correspond to solutions that are optima of the associated combinatorial problem, and edges correspond to weighted transitions among them. As a first benchmark, the well studied family of abstract landscapes, the Kauffman's *NK* model, was investigated [16]. In this model the ruggedness, and hence the difficulty of the landscape, can be tuned from easy to hard by increasing the K parameter from 1 to $N - 1$. Later work considered *NK* models incorporating *neutrality*,i.e., extended regions of equal or quasi-equal fitness [34]. When neutrality is present, the single-solution local optima are replaced by the local optima plateaus to capture the relevant structure of the search space. Subsequently, more complex and realistic search spaces were studied: the quadratic assignment problem, the flow-shop problem, the number partition problem, and the Euclidean TSP. Initially, weighted edges represented an approximation to the probability of transition between the basins associated to the edges' ends in a given direction. This definition, although informative, produced densely connected networks and required exhaustive sampling of the basins of attraction. A second version, *escape edges* was proposed in [33], which does not require a full computation of the basins. Instead, these edges account for the chances of escaping a local optimum after a controlled mutation (*e.g.* 1 or 2 bit-flips in binary space) followed by hill-climbing (see below).

Let us now define LONs more formally starting from the well known concept of a fitness landscape. A **fitness landscape** [8, 24] is a triplet (X, N, f) where X is a finite set of feasible solutions *i.e.* a search space; $N : X \longrightarrow 2^X$, a neighborhood structure, is a function that assigns to every $x \in X$ a set of neighbors $N(x)$, and $f : X \longrightarrow \mathbb{R}$ is a fitness function that assigns a real value to the corresponding solutions.

The present study considers search spaces in which feasible solutions can be represented as permutations, e.g., as those generated by QAP instances. For this case, a basic neighborhood structure is the pairwise exchange operation which exchanges any two positions in a permutation, thus transforming it into another permutation. This gives a neighborhood size of $n(n-1)/2$, where n is the problem instance size. Here is how the LON graph for a given problem instance is constructed.

The **nodes** in the network are local optima (LO) in the search space. For a maximization problem, a solution $x \in X$ is a local optimum iff $\forall x' \in N(x)\ f(x') \leqslant f(x)$. For a minimization problem such as QAP, the inequality is reversed. Notice that in this work we do not target specifically neutral fitness landscape with large plateaus. However, this definition of local optima is still relevant for small amounts of neutrality. For fitness landscape with high levels of neutrality, please refer to the definitions of previous work [34] where the nodes are local optima plateaus. LO are extracted using a best-improvement hill-climber (*hc*), as given in Algorithm 1. Thereby, when selecting the fittest neighbor (line 4), ties are broken at random.

Algorithm 1 Best-improvement hill-climbing (maximization)

1: **procedure** HILLCLIMBING
2: $x \leftarrow$ random initial solution
3: **while** $x \neq$ Local Optimum **do**
4: set $x' \in N(x)$, s.t. $f(x') = max_{y \in N(x)} f(y)$
5: **if** $f(x) < f(x')$ **then**
6: $x \leftarrow x'$
7: **end if**
8: **end while**
9: **end procedure**

The **escape edges** in the network are defined according to a distance function $dist$ and a positive integer $D > 0$. The distance function represents the minimal number of moves between two solutions by a given search (mutation) operator. There is an edge e_{ij} between LO_i and LO_j if a solution x exists such that $dist(x, LO_i) \leqslant D$ and $hc(x) = LO_j$. In other words, if LO_j can be reached after mutating LO_i and running hill-climbing from the mutated solution. The weight $\tilde{w}_{ij}$ of this edge is $\tilde{w}_{ij} = \sharp\{x \in X \mid dist(x, LO_i) \leqslant D$ and $hc(x) = LO_j\}$. That is, the number of LO_i mutations that reach LO_j after hill-climbing. This weight can be normalized by the total number of solutions, $\sharp\{x \in X \mid dist(x, LO_i) \leqslant D\}$, within reach at distance D: $w_{ij} = \tilde{w}_{ij} / \sum_j \tilde{w}_{ij}$.

The weighted **local optima network** $G_w = (N, E)$ is the graph where the nodes $n_i \in N$ are the local optima, and there is an edge $e_{ij} \in E$, with weight w_{ij}, between two nodes n_i and n_j if $w_{ij} > 0$. Any operation that does not result in a transition to a neighboring basin of attraction contributes to the weight w_{ii}, i.e., it introduces a self-loop representing the probability that a move causes the search to remain into the basin of the original local optimum. We thus have $w_{ij} + w_{ii} = 1, \forall j \in N(i)$ for normalized weights.

According to the definition of weights, w_{ij} may be different from w_{ji}. Thus, two weights are needed in general, and we have a weighted, oriented transition graph G_w.

2.1 Communities of Optima in LONs

LONs are graphs with a possibly complex structure and, as such, they can be studied with the methods of network science, see e.g., [9]. This is what was done in, e.g., [21, 23, 32, 34]. Among other important features, complex networks usually show a clustered structure;

in other words, one can distinguish in them groups of nodes that are more strongly interconnected between them than with the rest of the network. Those clusters are also called *communities* and are a very important feature of real-world networks such as social networks. It has been often assumed that optima in combinatorial fitness landscapes are either uniformly distributed [14], or they configure what is called a "massif central" or "big valley", that is, the optima are mainly clustered together in a single complex structure [10, 16]. However, recent research has shown that this need not be the case [22]. Actually, in many cases the optima form clusters of different sizes and the size of the clusters, as well as the fitness of the respective optima have important consequences for the performance of local search based metaheuristic [13, 15, 29]. In the present study we use this knowledge as it applies to real-like instances of the QAP problem, which is described in the next section, in order to find moves in permutation space that efficiently connect local optima between them.

3 THE QUADRATIC ASSIGNMENT PROBLEM

The Quadratic Assignment Problem (QAP) [18] is a combinatorial problem in which a set of facilities with given flows has to be assigned to a set of locations with given distances in such a way that the sum of the product of flows and distances is minimized. A solution to the QAP is generally written as a permutation π of the set $\{1, 2, ..., n\}$. The cost associated with a permutation π is given by:

$$C(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$$

where n denotes the number of facilities/locations and $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are referred to as the distance and flow matrices, respectively. The structure of these two matrices characterizes the class of instances of the QAP problem.

The results presented in this article are based on the instance generators proposed in [17] which are in turn inspired by [31]. In [17] the generators were devised for the multi-objective QAP, but are adapted here for the single-objective QAP. For the sake of the present study, we used the *real-like generator* which produces instances where the distance and flow matrices have structured entries. To generate the symmetric distance matrix, N points (integer coordinates) are randomly distributed in a circle of radius 100, and the entries are given by the distances between these N points. The flow matrix is also symmetric with entries following the law $\lceil 10^r \rceil$ where r is a uniform random integer from $[L, U]$. This procedure generates non-uniformly random instances of type *Tai*$\mathrm{nn}b$ which have the so-called "real-like" structure since they resemble the structure of QAP problems found in practical applications.

4 SEARCH MOVES

In this work we consider the search space of the permutations of the set $\{1, \ldots, n\}$, connected among them by search moves formed by sequences of pairwise exchanges of permutation items (or simply *exchanges* from now on).

As introduced in [1–3, 5, 6, 26, 27], such structured space has an interesting algebraic interpretation that is here exploited in order to concisely represent the exchanges involved in moving from one permutation to another.

After recalling the required background in Section 4.1, we introduce, in Section 4.2, an algebraic method that allows to identify and concisely represent the exchange moves between two permutations. Clearly, this method works for two generic permutations, though we will employ it later on in order to find the connecting paths among the local optima in a LON.

4.1 Algebraic structure of the permutation space

A permutation of the set $[n] = \{1, 2, \ldots, n\}$ is a bijective function $\pi : [n] \to [n]$. Usually, a permutation π is interpreted as an ordering of items, therefore $\pi(i)$ indicates the item at position i in the ordering. The set of all the permutations of $[n]$ is denoted by $\mathcal{S}_n$. Permutations can be composed by means of the operator $\circ$. Given $\pi, \rho \in \mathcal{S}_n$, $\sigma = \pi \circ \rho$ is defined as $\sigma(i) = \pi(\rho(i))$, for all $i \in [n]$. $\mathcal{S}_n$ forms a group with respect to $\circ$, called *symmetric group*. Its neutral element is the identity permutation e defined as $e(i) = i$, for all $i \in [n]$, while the inverse of $\pi \in \mathcal{S}_n$ is the permutation π^{-1} defined as $\pi^{-1}(i) = j$ if and only if $\pi(j) = i$. Note that $\circ$ is associative and non-commutative.

The group $\mathcal{S}_n$ is finitely generated, i.e., there exists a subset of permutations, called *generators*, such that any other permutation can be factorized in a product of finitely many generators. $\mathcal{S}_n$ has multiple generating sets, each of them encoding a different kind of search move (see [2, 7, 30]). Here, we consider the generating set *EXC* $\subset \mathcal{S}_n$ defined as *EXC* $= \{\epsilon_{ij} : 1 \leq i < j \leq n\}$, where ϵ_{ij} is the identity permutation with the items i and j exchanged. Given a generic $\pi \in \mathcal{S}_n$, the composition

$$\pi \circ \epsilon_{ij} \tag{1}$$

swaps the items at positions i and j in π. For example, let $\pi = \langle 31542 \rangle$, then $\pi \circ \epsilon_{15} = \langle 31542 \rangle \circ \langle 52341 \rangle = \langle 21543 \rangle$. Hence, the $n(n-1)/2$ generators in *EXC* algebraically express the operations required in order to move in the permutation space by means of exchanges.

An important concept for any finitely generated group is its associated Cayley graph. For $\mathcal{S}_n$ generated by *EXC*, the Cayley graph $\mathcal{CG}$ is the labeled graph whose vertices are all the permutations in $\mathcal{S}_n$ and there is an arc from π to ρ labeled by $\epsilon_{ij} \in$ *EXC* if and only if $\rho = \pi \circ \epsilon_{ij}$. Therefore, $\mathcal{CG}$ actually represents the permutation search space equipped with its neighborhood structure as previously introduced in Section 2.

For any pair of permutations, the shortest paths between them in $\mathcal{CG}$ represent the sequences of exchanges required to transform one permutation into the other. Formally, let $\pi, \rho \in \mathcal{S}_n$ and let us choose the shortest path from π to ρ with arc-labels $\epsilon_{i_1 j_1}, \epsilon_{i_2 j_2}, \ldots, \epsilon_{i_k j_k}$, then the following equivalence holds

$$\pi \circ \epsilon_{i_1 j_1} \circ \epsilon_{i_2 j_2} \circ \ldots \circ \epsilon_{i_k j_k} = \rho. \tag{2}$$

By left-composing both sides of the equivalence by π^{-1} we obtain

$$\pi^{-1} \circ \rho = \epsilon_{i_1 j_1} \circ \epsilon_{i_2 j_2} \circ \ldots \circ \epsilon_{i_k j_k}. \tag{3}$$

In general, there are multiple shortest paths between two permutations in the Cayley graph. Anyway, equation (3) continues to hold if we replace its right hand side with a sequence of generators from any other shortest path from π to ρ. Therefore, the permutation $\pi^{-1} \circ \rho$

is, in some sense, the difference between ρ and π. For this reason it is also denoted as $\rho \ominus \pi$ (see [27]).

Often, we are given two endpoint permutations π and ρ and we are required to compute one of the sequences of exchanges that transform π into ρ. From equation (3), this is equivalent to one of the factorizations of $\rho \ominus \pi = \pi^{-1} \circ \rho$ in terms of the generators in *EXC*.

For instance, this factorization can be computed using the algorithm *RandSS* introduced in [2], which is based on the widely known concept of cycles decomposition of a permutation [19]. Indeed, any permutation can be decomposed in a product of cycles. A k-cycle of a generic permutation π is a sequence of k items $(\pi(i_0), \dots, \pi(i_{k-1}))$ such that, for any $0 \leq j < k$, the item $\pi(i_j)$ appears at position $\pi(i_{(j-1) \bmod k})$ in π. For example, the permutation $\langle 26745831 \rangle$ has the following cycles decomposition $(1268)(37)(4)(5)$.

It is important to note that: (i) e is the only permutation with exactly n cycles (of length 1), and (ii) an exchange of items belonging to the same cycle breaks the cycle into two new cycles, thus increasing the number of cycles by one. Therefore, a factorization can be obtained by iteratively choosing and applying an exchange that breaks one of the cycles of the incumbent permutation till this one becomes the identity. The sequence of exchanges, in reverse order of application, is thus a factorization of the input permutation. For further details see [2].

4.2 Computation of the search moves

Here, given two generic endpoint permutations π and ρ, our objective is to concisely represent all the pairs of items that are required to be exchanged in at least one of the alternative exchange sequences that transform π into ρ.

Clearly, such alternative exchange sequences exactly correspond to the shortest paths between π and ρ in the Cayley graph induced by the generating set *EXC*.

Therefore, the idea is to use the cycles decomposition of $\rho \ominus \pi$ in order to concisely represents all the exchanges we are looking for. However, we have to take into account a subtle aspect of the algebraic application of an exchange generator. Indeed, the expression in equation (1) corresponds to swap the items at positions i and j in π, while what we would like is a way to identify the items that are exchanged, independently of their positions.

By considering that any permutation π is a bijection from positions to items, we can exchange two generic items i and j in π by exchanging the items at positions i and j in π^{-1} and inverting back the obtained permutation. Algebraically, this corresponds to computing

$$\left(\pi^{-1} \circ \epsilon_{ij}\right)^{-1} = \epsilon_{ij} \circ \pi, \tag{4}$$

where the equivalence derives from a widely known group property and the fact that $\epsilon_{ij}^{-1} = \epsilon_{ij}$ for any $\epsilon_{ij} \in EXC$.

The expression in equation (4) says that, in order to exchange the items i and j in a given permutation π, we simply have to left-compose π by ϵ_{ij}. This result can be clearly extended to exchanges' sequences longer than one. Hence, given $\pi, \rho \in \mathcal{S}_n$, the sequence of exchanges from π to ρ, such that any exchange encodes the items to be swapped – and not their positions – can be derived from the equivalence

$$\epsilon_{i_k j_k} \circ \epsilon_{i_2 j_2} \circ \dots \circ \epsilon_{i_1 j_1} \circ \pi = \rho, \tag{5}$$

which, by right composing both sides by π^{-1}, implies

$$\rho \circ \pi^{-1} = \epsilon_{i_k j_k} \circ \epsilon_{i_2 j_2} \circ \dots \circ \epsilon_{i_1 j_1}. \tag{6}$$

Using the difference notation, $\rho \circ \pi^{-1} = \pi^{-1} \ominus \rho^{-1}$.

Therefore, the pairs of items to be exchanged, in at least one shortest path between π and ρ, are exactly all the pairs of items appearing in each cycles of $\pi^{-1} \ominus \rho^{-1}$. The cycles decomposition of $\pi^{-1} \ominus \rho^{-1}$ can be computed in $\Theta(n)$ time and $\Theta(n)$ space. Note that, since the number of different shortest paths connecting two permutations is exponential in the distance between the endpoint permutations, our representation is compact and quick to compute.

For the sake of clarity, we provide an illustrative example. Given the permutations π and ρ such that $\pi^{-1} \ominus \rho^{-1}$ has the cycles decomposition $(1268)(37)(4)(5)$, then the pairs of items to be exchanged, in at least one shortest path from π to ρ in the Cayley graph, are: $(1, 2)$, $(1, 6)$, $(1, 8)$, $(2, 6)$, $(2, 8)$, $(6, 8)$, and $(3, 7)$.

Beside representing the exchanges of items, an additional requirement for our experimentation is to count, for each exchange, in how many shortest paths it appears. Clearly, this quantifies the importance of such an exchange in transforming π into ρ.

Given the cycles decomposition of $\pi^{-1} \ominus \rho^{-1}$, this quantity depends on: (i) the number of different factorizations that are possible for any cycle in the decomposition, and (ii) the distance between the two items in the cycles where they appear.

In the following, we provide some approximated formulae and tabulations of such quantities. These have been derived by considering recursive counting procedures. We will investigate exact counting procedures in a future work.

Formally, let $\pi^{-1} \ominus \rho^{-1}$ be formed by the cycles $c_1, c_2, \dots, c_L$, and considering the exchange ϵ_{ij} between items belonging to the cycle c_k, then the approximated number $Q(\epsilon_{ij})$ of shortest paths, between π and ρ, where ϵ_{ij} appears is given by

$$Q_k(\epsilon_{ij}) = \left(\prod_{t \neq k} R(\text{length}(c_t))\right) \cdot S\left(\text{length}(c_k), \Delta_{ij}^{(c_k)}\right) \cdot M!, \tag{7}$$

where: $R(l)$ is the (approximated) number of factorizations of a cycle of length l; $S(l, \Delta)$ is the (approximated) number of factorizations of the cycle, of length l, which contains the two items to exchange at distance Δ between them; and M is the number of cycles larger than 1.

Equation (7) can be described as follows. The items i and j can only appear in one of the cycles. Therefore, we are free to decompose the other cycles in any one of their possible factorizations (first factor involving the function R), while, for the cycle containing i and j, we have to count only the factorizations containing ϵ_{ij} (second factor involving the function S). Finally, the third factor is due to the fact that the cycles in a decomposition commute.

The approximated number of different factorizations of an l-cycle can be computed by the following recurrence:

$$\begin{aligned} &R(1) = 1 \\ &R(2) = 1 \\ &R(l) = \textstyle\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} R(j-i) R(l-j+i). \end{aligned} \tag{8}$$

Moreover, we have been able to computationally tabulate the approximated quantities R and S for cycle lengths up to 11 (this is enough for our experimentation). This quantities are provided in Table 1, where l indicates the cycle length, while the S values are provided following the order $S(l,1),\ldots,S(l,l-1)$.

Table 1: Tabulation of the functions R and S

l	$R(l)$	$S(l,\Delta)$
1	1	–
2	1	1
3	3	2, 2
4	14	8, 5, 8
5	85	45, 23, 23, 45
6	621	315, 144, 117, 144, 315
7	5236	2589, 1103, 796, 796, 1103, 2589
8	49680	24182, 9818, 6561, 5818, 6561, 9818, 24182
9	521721	251493, 98627, 62454, 51178, 51178, 62454, 98627, 251493
10	5994155	2872419, 1097626, 668017, 517642, 478071, 517642, 668017, 1097626, 2872419
11	74701055	35674928, 13363000, 7889908, 5864570, 5117644, 5117644, 5864570, 7889908, 13363000, 35674928

5 ANALISYS OF THE SEARCH MOVES IN THE COMMUNITIES OF OPTIMA

The main goal of this paper is to verify the existence of preferred search moves that allow to move across basins of a fitness landscape. In particular, we are interested in the landscapes of the QAP using pairwise exchange of items as search moves.

In order to perform our investigation, we have generated four QAP instances as described in Section 3 (see also [17]): `KCso11rl-1`, `KCso11rl-2`, `KCso11rl-3`, and `KCso11rl-4`. All of them have size $n = 11$ and belong to the family of "real-like" instances. The relatively small size makes it possible to exhaustively build the complete LON, while the real-like structure should result in communities of strongly interconnected optima.

Given an instance, after computing its LON, two different community finding algorithms from the R-based package *igraph*[12] have been executed in order to obtain two different clusterizations of the optima. The algorithms used are *FastGreedy* and *WalkTrap* which are based on a hierarchical agglomeration algorithm and on random walks respectively. We use two methods because different community detection algorithms usually give similar but different partitions of the graph nodes, given their heuristic character. For the sake of example, Figure 1 depicts the clusterization obtained by *WalkTrap* on the LON of `KCso11rl-1`.

Once the optima are clustered into communities [20], two different analyses have been performed: *intra-community* and *inter-community* analyses. Both of them are based on the scheme depicted in Algorithm 2.

The function *AnalyzeExchanges* of Algorithm 2 takes as input a set S of pairs of permutations and returns a (upper triangular) matrix Z such that Z_{ij} counts in how many paths, connecting pairs of solutions in S, appears the exchange ϵ_{ij}. Therefore, Z_{ij} measures how much it is important to swap the items i and j in order to move between the permutations in S. Clearly, when the pairs in S are local optima, these quantities identifies the exchanges that are more prone to allow moves across the basins of a fitness landscape.

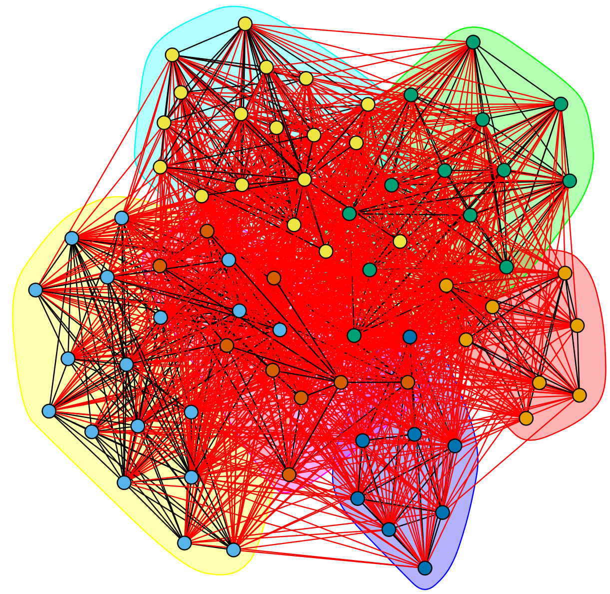

Figure 1: Clusterization of `KCso11rl-1` obtained by *WalkTrap*

Algorithm 2 Computation of the importance of the exchanges in order to move among a set of permutations

```
function AnalyzeExchanges(S ⊂ 𝒮_n × 𝒮_n)
    Z_ij ← 0 for all 1 < i < j ≤ n
    for all pairs of permutations π, ρ ∈ S do
        Compute the cycles decomposition of π^{-1} ⊖ ρ^{-1}
        for all cycles c of π^{-1} ⊖ ρ^{-1} do
            for all pairs of items i, j ∈ c do
                Z_ij ← Z_ij + Q_k(ε_ij)          ▷ see equation (7)
            end for
        end for
    end for
    return Z
end function
```

The intra-community and inter-community analyses differ for the set of optima provided in input to *AnalyzeExchanges*.

Intra-Community Analysis. For each one of the k communities in a LON, *AnalyzeExchanges* is run on all the pairs of local optima in the community. Hence, k matrices of Z-values are produced, one per community.

Inter-Community Analysis. In this case, *AnalyzeExchanges* is executed only once by taking as input all the pairs LO_i, LO_j of local optima such that LO_i belongs to a different community with respect to LO_j.

Since the optima inside a community are more likely to be strongly connected among them, our expectation is to observe more "concentrated" values on the Z matrices produced by the intra-community analysis than on those obtained from the inter-community analysis.

5.1 Experimental results

Since we have considered 4 instances and 2 community finding algorithms, we have run our intra-community and inter-community analyses on 8 clusterized LONs.

Furthermore, we have computed the Gini index on every Z matrix produced. The Gini index is a measure of statistical dispersion very popular in economy [11]. It takes as input a set of non-negative values and returns a number in $[0, 1]$. As extreme cases, it returns 0 when all the input values are equal and 1 when all the values are zero except one. Clearly, since in our scenario is practically impossible to have all the "mass" concentrated in a single Z-value, a Gini index around 0.5 has been empirically verified to produce a "concentrated" distribution among the exchanges.

In Figure 2, we provide illustrations of the Z matrices computed on the LON of the `KCsol1rl-1` instance clustered by means of the algorithm *WalkTrap*. The first six heatmaps correspond to the Z matrices of the six communities found by the intra-community analysis, while the last one depicts the inter-community Z matrix. The lower triangular part of every heatmap is set to zero, while the color of the every entry (i, j), such that $i < j$, is hotter when the corresponding Z_{ij} value is larger. Moreover, the caption of every heatmap provides the value of the Gini index computed on the corresponding matrix. For the sake of space, the heatmaps of the other instances are provided online at this link `https://bit.ly/2FScdTB`.

Both the graphical illustrations and the Gini indices of Figure 2 indicate the emergence of (relatively) few preferred pairs of items to exchange in order to move between intra-community optima. However, as expected, such observation is much less evident on the output of the inter-community analysis. Also the Gini index of the inter-community matrix is less than half of the smallest Gini index obtained by the intra-community analyses.

The same indications seem to be confirmed by the data provided in Table 2. Here, for each pair "instance + community finding algorithm", it is provided: the number of communities in the LON, the average and the standard deviation of the intra-community Gini indexes, and the inter-community Gini index. Even here, the inter-community Gini index is sensibly smaller than the average intra-community Gini index in all the instances and regardless of the community finding algorithm. The only exception is the LON of the instance `KCsol1rl-2` clusterized by means of *WalkTrap*. Note anyway that, in this case, the LON has only two communities. Therefore, it looks likely that, more clustered is a LON easier is to observe the emergence of few search moves that interconnect different local optima in the landscape.

6 CONCLUSION AND FUTURE WORK

In this work, we have proposed an experimental analysis aiming to determine the existence of preferred search moves that interconnect local optima belonging to a same community or different communities of a given clusterized LON. The LONs have been exhaustively computed on some small and real-like instances of the Quadratic Assignment Problem. Nevertheless, this clusterized structure can be found also in other important combinatorial optimization problems, thereby it is likely that the present analysis can be extended to other general cases.

Our investigation revealed that a small number of preferred search moves are more important for moving across optima belonging to the same community, while this result is less apparent when the optima belong to different communities. To the best of our knowledge, this is the first study aiming to determine the search moves – and not only their quantities – that interconnect different basins of attractions in a fitness landscape. The investigation has been possible by using together consolidated fitness landscape analysis concepts, network analysis tools, and algebraic methods.

This work represents a first step of a research which can be extended along several possible lines like, for instance, the application to other permutation problems and the study of larger instances by using local optima sampling and approximated methodologies. Furthermore, this work can be the basis for a deeper investigation which can be used to improve the performances of metaheuristic search algorithmS.

ACKNOWLEDGMENTS

The research described in this work has been partially supported by: the research grant "Fondi per i progetti di ricerca scientifica di Ateneo 2019" of the University for Foreigners of Perugia under the project "Algoritmi evolutivi per problemi di ottimizzazione e modelli di apprendimento automatico con applicazioni al Natural Language Processing"; and by RCB-2015 Project "Algoritmi Randomizzati per l'Ottimizzazione e la Navigazione di Reti Semantiche" and RCB-2015 Project "Algoritmi evolutivi per problemi di ottimizzazione combinatorica" of Department of Mathematics and Computer Science of University of Perugia. Finally, we thank S. Verel for the use of his program for computing the QAP instances LON.

REFERENCES

[1] Marco Baioletti, Alfredo Milani, and Valentino Santucci. 2015. Linear Ordering Optimization with a Combinatorial Differential Evolution. In *Proc. of 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*. 2135–2140. https://doi.org/10.1109/SMC.2015.373

[2] Marco Baioletti, Alfredo Milani, and Valentino Santucci. 2016. An Extension of Algebraic Differential Evolution for the Linear Ordering Problem with Cumulative Costs. In *Proc. of 14th International Conference on Parallel Problem Solving from Nature - PPSN XIV*. 123–133. https://doi.org/10.1007/978-3-319-45823-6_12

[3] M. Baioletti, A. Milani, and V. Santucci. 2017. Algebraic Particle Swarm Optimization for the permutations search space. In *Proc. of 2017 IEEE Congress on Evolutionary Computation (CEC 2017)*. 1587–1594. https://doi.org/10.1109/CEC.2017.7969492

[4] M. Baioletti, A. Milani, and V. Santucci. 2018. Algebraic Crossover Operators for Permutations. In *2018 IEEE Congress on Evolutionary Computation (CEC 2018)*. 1–8. https://doi.org/10.1109/CEC.2018.8477867

[5] Marco Baioletti, Alfredo Milani, and Valentino Santucci. 2018. Automatic Algebraic Evolutionary Algorithms. In *Proc. of Int. Workshop on Artificial Life and Evolutionary Computation (WIVACE 2017)*. Springer International Publishing, Cham, 271–283. https://doi.org/10.1007/978-3-319-78658-2_20

[6] Marco Baioletti, Alfredo Milani, and Valentino Santucci. 2018. Learning Bayesian Networks with Algebraic Differential Evolution. In *Proc. of 15th Int. Conf. on Parallel Problem Solving from Nature – PPSN XV*. Springer International Publishing, Cham, 436–448. https://doi.org/10.1007/978-3-319-99259-4_35

[7] Marco Baioletti, Alfredo Milani, and Valentino Santucci. 2018. MOEA/DEP: An Algebraic Decomposition-Based Evolutionary Algorithm for the Multiobjective Permutation Flowshop Scheduling Problem. In *Proc. of European Conference on Evolutionary Computation in Combinatorial Optimization - EvoCOP 2018*. Springer International Publishing, Cham, 132–145. https://doi.org/10.1007/978-3-319-77449-7_9

[8] Marco Baioletti and Valentino Santucci. 2017. Fitness Landscape Analysis of the Permutation Flowshop Scheduling Problem with Total Flow Time Criterion. In *Computational Science and Its Applications – ICCSA 2017*. Springer International Publishing, Cham, 705–716. https://doi.org/10.1007/978-3-319-62392-4_51

[9] Albert-László Barabási et al. 2016. *Network science*. Cambridge university press.

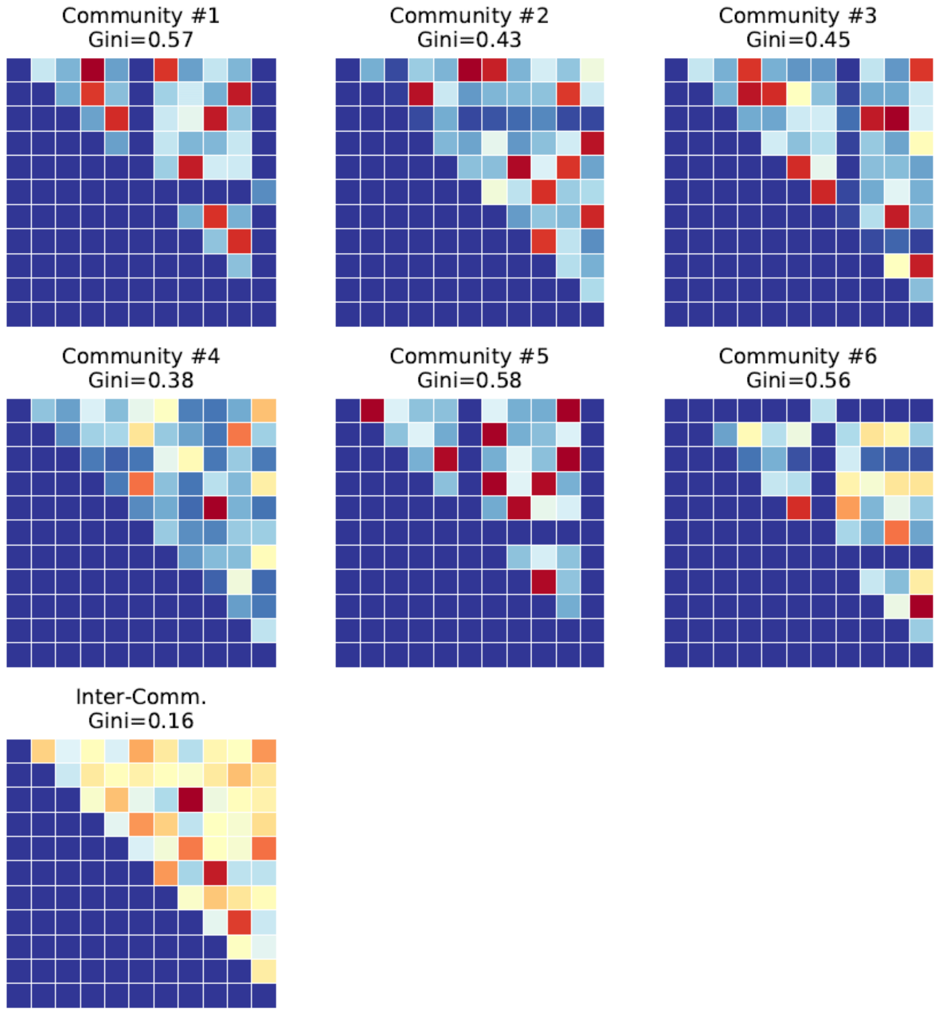

Figure 2: Heatmaps of the instance `KCso11rl-1` clustered by *WalkTrap*

Table 2: Gini Indexes for every clustered LON

Instance	Community Finding Algorithm	#Communities	Intra-Community Gini Index	Inter-Community Gini Index
KCso11rl-1	FastGreedy	6	0.56 ± 0.14	0.16
KCso11rl-1	WalkTrap	6	0.50 ± 0.09	0.16
KCso11rl-2	FastGreedy	6	0.46 ± 0.11	0.12
KCso11rl-2	WalkTrap	2	0.26 ± 0.15	0.14
KCso11rl-3	FastGreedy	8	0.44 ± 0.10	0.18
KCso11rl-3	WalkTrap	6	0.41 ± 0.13	0.17
KCso11rl-4	FastGreedy	5	0.57 ± 0.08	0.16
KCso11rl-5	WalkTrap	5	0.58 ± 0.08	0.16

[10] Kenneth D Boese, Andrew B Kahng, and Sudhakar Muddu. 1994. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16, 2 (1994), 101–113.

[11] Lidia Ceriani and Paolo Verme. 2012. The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *The Journal of Economic Inequality* 10, 3 (01 Sep 2012), 421–443. https://doi.org/10.1007/s10888-011-9188-x

[12] Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems (2006), 1695. http://igraph.org

[13] Fabio Daolio, Marco Tomassini, Sébastien Vérel, and Gabriela Ochoa. 2011. Communities of minima in local optima networks of combinatorial spaces. *Physica A: Statistical Mechanics and its Applications* 390, 9 (2011), 1684–1694.

[14] J Garnier and L Kallel. 2001. How to detect all maxima of a function. In *Theoretical aspects of evolutionary computing*. Springer, 343–370.

[15] Sebastian Herrmann, Gabriela Ochoa, and Franz Rothlauf. 2016. Communities of local optima as funnels in fitness landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. ACM, 325–331.

[16] S. A. Kauffman. 1993. *The Origins of Order*. Oxford University Press, New York.

[17] J. Knowles and D. Corne. 2003. Instance Generators and Test Suites for the Multiobjective Quadratic Assignment Problem. In *Proceedings of the Evolutionary Multi-Criterion Optimization Conference (EMO 2003) (LNCS)*. Springer, 295–310.

[18] T. C. Koopmans and M. Beckmann. 1957. Assignment Problems and the Location of Economic Activities. *Econometrica* 25, 1 (1957), 53–76.

[19] Serge Lang. 2002. *Algebra*. Vol. 211. Springer.

[20] Paolo Mengoni, Alfredo Milani, and Yuanxi Li. 2018. Community graph elicitation from students' interactions in virtual learning environments. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 10962 LNCS. 414–425. https://doi.org/10.1007/978-3-319-95168-3_28

[21] Gabriela Ochoa, Marco Tomassini, Sebástien Verel, and Christian Darabos. 2008. A Study of NK Landscapes' Basins and Local Optima Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*. ACM, 555–562.

[22] Gabriela Ochoa and Nadarajen Veerapen. 2016. Deconstructing the big valley search space hypothesis. In *Evolutionary Computation in Combinatorial Optimization*. Springer, 58–73.

[23] G. Ochoa, S. Verel, and M. Tomassini. 2010. First-Improvement vs. Best-Improvement Local Optima Networks of NK Landscapes. In *Parallel Problem Solving from Nature - PPSN XI (Lecture Notes in Computer Science)*, Vol. 6238. Springer, 104–113.

[24] C.M. Reidys and P.F. Stadler. 2002. Combinatorial landscapes. *SIAM review* 44, 1 (2002), 3–54.

[25] Valentino Santucci, Marco Baioletti, Gabriele Di Bari, and Alfredo Milani. 2019. A Binary Algebraic Differential Evolution for the MultiDimensional Two-Way Number Partitioning Problem. In *Evolutionary Computation in Combinatorial Optimization*. Springer International Publishing, Cham, 17–32. https://doi.org/10.1007/978-3-030-16711-0_2

[26] Valentino Santucci, Marco Baioletti, and Alfredo Milani. 2014. A Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem with Total Flow Time Criterion. In *Proc. of 13th International Conference on Parallel Problem Solving from Nature – PPSN XIII*. Springer, Cham, 161–170. https://doi.org/10.1007/978-3-319-10762-2_16

[27] V. Santucci, M. Baioletti, and A. Milani. 2016. Algebraic Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem With Total Flowtime Criterion. *IEEE Transactions on Evolutionary Computation* 20, 5 (2016), 682–694. https://doi.org/10.1109/TEVC.2015.2507785

[28] V. Santucci, M. Baioletti, and A. Milani. 2016. Solving permutation flowshop scheduling problems with a discrete differential evolution algorithm. *AI Communications* 29, 2 (2016), 269–286. https://doi.org/10.3233/AIC-150695

[29] Valentino Santucci and Alfredo Milani. 2011. Particle Swarm Optimization in the EDAs Framework. In *Soft Computing in Industrial Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 87–96. https://doi.org/10.1007/978-3-642-20505-7_7

[30] Tommaso Schiavinotto and Thomas Stützle. 2007. A review of metrics on permutations for search landscape analysis. *Computers & Operations Research* 34, 10 (2007), 3143–3153.

[31] É. D. Taillard. 1995. Comparison of iterative searches for the quadratic assignment problem. *Location Science* 3, 2 (1995), 87 – 105.

[32] Marco Tomassini, Sébastien Verel, and Gabriela Ochoa. 2008. Complex-Network Analysis of combinatorial spaces: The NK landscape case. *Phys. Rev. E* 78, 6 (2008), 066114.

[33] S. Verel, F. Daolio, G. Ochoa, and M. Tomassini. 2012. Local Optima Networks with Escape Edges. In *Proceedings of the International Conference on Artificial Evolution, EA-2011 (Lecture Notes in Computer Science)*, Vol. 7401. Springer, 49–60.

[34] S. Verel, G. Ochoa, and M. Tomassini. 2011. Local Optima Networks of NK Landscapes with Neutrality. *IEEE Transactions on Evolutionary Computation* 15, 6 (2011), 783–797.