On the symmetry of the Quadratic Assignment Problem through Elementary Landscape Decomposition

Xabier Benavides
xbenavides@bcamath.org
BCAM - Basque Center for Applied Mathematics

Josu Ceberio
josu.ceberio@ehu.eus
UPV/EHU - University of the Basque Country

Leticia Hernando
leticia.hernando@ehu.eus
UPV/EHU - University of the Basque Country

ABSTRACT

When designing meta-heuristic strategies to optimize the quadratic assignment problem (QAP), it is important to take into account the specific characteristics of the instance to be solved. One of the characteristics that has been pointed out as having the potential to affect the performance of optimization algorithms is the symmetry of the distance and flow matrices that form the QAP.

In this paper, we further investigate the impact of the symmetry of the QAP on the performance of meta-heuristic algorithms, focusing on local search based methods. The analysis is carried out using the elementary landscape decomposition (ELD) of the problem under the swap neighborhood. First, we study the number of local optima and the relative contribution of the elementary components on a benchmark composed of different types of instances. Secondly, we propose a specific local search algorithm based on the ELD in order to experimentally validate the effects of the symmetry. The analysis carried out shows that the symmetry of the QAP is a relevant feature that influences both the characteristics of the elementary components and the performance of local search based algorithms.

CCS CONCEPTS
- Mathematics of computing → Combinatorial optimization;
- Computing methodologies → Discrete space search;

KEYWORDS
Quadratic Assignment Problem, Elementary Landscapes, Symmetry

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

1 INTRODUCTION

Formalized by Koopmans and Beckmann [36] in 1957, the quadratic assignment problem (QAP) has been a recurring problem in combinatorial optimization due to its known complexity. In fact, in 1976, Sahni and Gonzalez [44] proved its NP-hardness. The QAP was originally proposed as a mathematical model for the location of a set of indivisible economic activities; however, in recent years, it has demonstrated to have many other real world applications such as facility layout design [24], parallel production scheduling [29], backboard wiring [6] or keyboard configuration [8]. Moreover, some major optimization problems can be expressed as particular cases of the QAP, for example the traveling salesman problem (TSP) [34], the linear ordering problem (LOP) [13] or the DNA fragment assembly problem (DNA-FA) [38]. As a result of this, a number of survey papers and reference books about the QAP have been published over the years [2, 9, 37].

A great number of strategies to solve the QAP have been proposed in the literature. Among them, branch and bound [28] and branch and cut [25] algorithms have proved to be powerful for small problem sizes; however, in large problems, they are no longer viable. As a result, due to the computational limitations, meta-heuristic algorithms [5] postulate as an efficient alternative, though they do not guarantee the optimality of the solutions. Over the years, many different meta-heuristic algorithms have been proposed for the QAP, ranging from local search approaches [4, 49, 53] to population based evolutionary algorithms [1, 23, 27, 40]. Designing new specific algorithms requires a good knowledge of the problem to be solved, so many authors have felt the need to perform extensive prior analyses of the characteristics of the QAP [16, 21, 39, 41].

Nevertheless, due to the inherent complexity of NP-hard problems, it may be difficult to directly study the characteristics of the QAP. In this sense, an alternative strategy can be to decompose the problem into a set of components that provide a framework that eases the analysis [42]. This approach allows us to have a more detailed vision of the problem that can facilitate the detection of useful properties for optimization. Among all the possible decomposition techniques, one available for the QAP is the elementary landscape decomposition (ELD). Proposed by Chicano et al. [19], this decomposition technique allows the QAP to be decomposed as a linear combination of three independent components (elementary landscapes). However, in spite of the growing interest, its impact on the analysis of combinatorial optimization problems and the development of new algorithms have been quite limited. Ceberio et al. [12] used the ELD as a method for the multi-objectivization of the QAP. In that work, a reference to the correlation between the symmetry of the QAP and its ELD decomposition was also highlighted. In fact, the symmetry has been observed to be particularly important in the QAP [22].

In this work, the elementary landscape decomposition is used to better understand the way in which the symmetry of the problem influences the performance of meta-heuristic algorithms, specifically...
local search based methods [48]. For this purpose, we first analyze the components of the decomposition on a benchmark of instances extracted from the QAPLIB library [7]. The analysis focuses on two main issues: the number of local optima and the relative contribution of the components of the decomposition to the structure of the problem. With this analysis, we intend to study the differences between different types of instances according to their symmetry. We build on the obtained results to propose a local search based algorithm that uses the elementary landscape decomposition to efficiently solve the QAP. This algorithm is used to validate the conclusions of the analysis through an experimental study of its performance on a benchmark of instances.

The rest of the paper is organized as follows. In Sections 2 and 3, the quadratic assignment problem and the theory on elementary landscape decomposition are introduced. Section 4 shows the analysis of the QAP through the ELDD and discuses the effects of the symmetry of the instances on the characteristics of the problem. In Section 5, a local search based algorithm that uses the ELDD is proposed and an experimental study of its performance is introduced. Finally, general conclusions and ideas for future work are presented in Section 6.

2 QUADRATIC ASSIGNMENT PROBLEM
The quadratic assignment problem (QAP) [36] consists of \( n \) facilities that need to be assigned to \( n \) available locations, taking into account that a facility can only be assigned to one location, and vice versa. \( d_{ij} \) is defined as the distance between the location \( i \) and location \( j \), and \( h_{pq} \) is defined as the work flow between the facility \( p \) and the facility \( q \). The goal of the QAP is to find the configuration that minimizes the overall communication costs between facilities, computed by the following objective function:

\[
 f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} h_{\sigma(i)\sigma(j)}
\]

where \( \sigma \) is a permutation of size \( n \) and \( \sigma(i) \) denotes the facility assigned to the location \( i \). Therefore, any QAP instance is composed of two elements: a distance matrix \( D = [d_{ij}]_{n \times n} \) that specifies the distances between locations and a flow matrix \( H = [h_{pq}]_{n \times n} \) that denotes the work flows between facilities.

The symmetry of the QAP has been singled out as a relevant feature that might influence the performance of optimization algorithms. Thus, in this work we classify the QAP instances into three different groups: symmetric (both the distance and flow matrices are symmetric\(^1\)), semi-symmetric (either the distance matrix or the flow matrix is symmetric, but not both), and asymmetric (neither of the distance and flow matrices is symmetric).

3 ELEMENTARY LANDSCAPE DECOMPOSITION
A landscape of a combinatorial optimization problem [43] is defined as a triplet \((\Omega, f, N)\), where \( \Omega \) is the search space of the problem, \( f : \Omega \rightarrow \mathbb{R} \) stands for the objective function that measures the fitness value of the solutions and \( N \) denotes a neighborhood function that assigns to each solution \( x \in \Omega \) a set of neighboring solutions to \( x \).

\(^1\)With respect to the main diagonal.

\(N(x) \subset \Omega\). The concepts of local minimum and maximum in a combinatorial optimization problem are defined by the landscape:

- **Local minimum**: Any solution \( x \in \Omega \) such that \( f(x) \leq f(y) \) for all \( y \in N(x) \).
- **Local maximum**: Any solution \( x \in \Omega \) such that \( f(x) \geq f(y) \) for all \( y \in N(x) \).

Thus, landscapes are the basis of local search optimization processes. Among all possible landscapes, there are some that are of particular interest due to their properties: the elementary landscapes [45, 51]. These landscapes were discovered by L.K. Grover [31] when the author noticed that the local search procedures of some combinatorial optimization problems could be modeled by a discrete formula similar to the wave equation used in physics. This formula, known as Grover’s wave equation, makes it possible to calculate the average value of the objective function \( f \) evaluated over all the neighborhood \( N(x) \) based on the fitness value of \( x \):

\[
 \bar{a}_{v}(f(x)) = f(x) + \kappa \frac{\bar{f} - f(x)}{|N(x)|}
\]

where \( \bar{f} \) is the average value of the objective function over the entire search space and \( \kappa \) is a characteristic constant that depends on the landscape. Any landscape that satisfies this equation is known to be elementary. More specifically, a landscape is elementary when its objective function \( f \) is elementary, that is, when \( f \) is an eigenfunction of the Laplacian matrix of the graph induced by \( \Omega \) and \( N \) [45, 51].

Elementary landscapes always satisfy the following [20]:

- \( f(x) < \bar{f} \Rightarrow f(x) < \bar{a}_{v}(f(y)) < \bar{f} \) \( \forall y \in N(x) \)
- \( f(x) = \bar{f} \Rightarrow f(y) = \bar{a}_{v}(f(y)) = \bar{f} \) \( \forall y \in N(x) \)
- \( f(x) > \bar{f} \Rightarrow f(x) > \bar{a}_{v}(f(y)) > \bar{f} \) \( \forall y \in N(x) \)

The above implies that all the local minima have a objective function value that is equal to or less than the average objective function value of the entire search space, while just the opposite happens in the case of the local maxima. These properties ensure that the elementary landscapes have a well known structure, which makes them particularly interesting for dealing with combinatorial optimization problems.

Although not every landscape \((\Omega, f, N)\) is elementary, any landscape whose neighborhood is regular \((N(x) = d) \) for all \( x \in \Omega \) and symmetric \((y \in N(x) \iff x \in N(y))\) for all \( x, y \in \Omega \) can be decomposed as a linear combination of several elementary landscapes which are known as elementary components of the problem. This process is called elementary landscape decomposition (ELD) [19].

3.1 Elementary Landscape Decomposition of the QAP
In the case of the QAP, no neighborhood that produces an elementary landscape is known. However, according to [17], the QAP under the swap neighborhood can be decomposed into a combination of three elementary landscapes. In the swap neighborhood, two solutions are neighbors if one can be transformed into the other by exchanging two items of the solution. Therefore, the landscape used to represent the QAP is \( L = (S_{N}, f, N) \), where \( S_{N} \) is the set
of all the permutations of size \( n \), \( f \) is the objective function given by Equation (1) and \( N \) is the swap neighborhood. In what follows, an explanation of the results developed in [17] is presented in a summarized form. For a more detailed explanation, see [17].

The elementary landscape decomposition of \( L \) is done by searching for a set of elementary functions \( \{ f_1, f_2, f_3, ..., f_m \} \) that form \( m \) elementary landscapes together with the search space and the neighborhood function of \( L \). This set of functions is created by the decomposition of the objective function of \( L \), so it must satisfy \( f(\sigma) = f_1(\sigma) + f_2(\sigma) + f_3(\sigma) + ... + f_m(\sigma) \) for all \( \sigma \in S_n \). In order to facilitate the decomposition, we rewrite Equation (1) as follows:

\[
f(\sigma) = \sum_{i,j,p,q=1}^{n} \phi(i,j,p,q)(\sigma)
\]

where \( \phi(i,j,p,q) \) are defined as:

\[
\phi(i,j,p,q) = \begin{cases} 
\alpha & \text{if } \sigma(i) = p \land \sigma(j) = q \\
\beta & \text{if } \sigma(i) = q \land \sigma(j) = p \\
\gamma & \text{if } \sigma(i) = p \lor \sigma(j) = q \\
\epsilon & \text{if } \sigma(i) = q \lor \sigma(j) = p \\
\xi & \text{if } \sigma(i) \neq p \land \sigma(j) \neq q 
\end{cases}
\]

where \( \alpha, \beta, \gamma, \epsilon, \xi \) are parameters, \( i, j, p, q \) are defined as:

\[
\phi(i,j,p,q) = \begin{cases} 
\alpha & \text{if } \sigma(i) = p \land \sigma(j) = q \\
\beta & \text{if } \sigma(i) = q \land \sigma(j) = p \\
\gamma & \text{if } \sigma(i) = p \lor \sigma(j) = q \\
\epsilon & \text{if } \sigma(i) = q \lor \sigma(j) = p \\
\xi & \text{if } \sigma(i) \neq p \land \sigma(j) \neq q 
\end{cases}
\]

The experimental framework used in this work consists of the following instances:

- 20 instances extracted from the QAPLIB library [7]: 8 symmetric instances (chr15a, chr20a, esa16a, esa16b, had18, had20, rou15, rou20), 8 semi-symmetric instances (lipa20a, lipa20b, lipa30a, lipa30b, tai15b, tai20b, tai25b, tai30b), 4 asymmetric instances (bur26a, bur26b, bur26c, bur26d).

- 4 asymmetric instances specifically generated for the experimentation (xab20a, xab20b, xab20c, xab20d).

The digits in the instance names indicate the size of the problems. Due to the lack of asymmetric instances in the QAPLIB library, we generate 4 additional asymmetric instances in order to have a more diverse benchmark for the analysis. The instance generation technique used is loosely inspired by the one proposed in [47]:

- **Distance matrix (D):** First, \( n \) uniform random points are generated in a \( 100 \times 100 \) rectangle, and the euclidean distance \( e(i, j) \) between every pair of points \( 1 \leq i, j \leq n \) is computed. Then, for every \( 1 \leq i, j \leq n \) such that \( i \neq j \), \( d_{ij} = \sqrt{e(i, j)} \) where \( \sqrt{v} \) is a uniform random value between 0.85 and 1.15. Finally, the matrix is scaled so that \( 0 \leq d_{ij} \leq 100 \) for every \( 1 \leq i, j \leq n \). All the values are rounded to the closest integer.

- **Flow matrix (H):** For every \( 1 \leq i, j \leq n \) such that \( i \neq j \), a uniform random value \( x \) between 0 and 1 is sampled. If \( x > sp \) where \( sp \) is a parameter that indicates the sparsity of the matrix, \( h_{ij} \) is set to a uniform random integer between 0 and 100. Otherwise, \( h_{ij} = 0 \).

The entries on the main diagonals of both matrices are set to 0. Using this technique, we obtain instances that are formed by two asymmetric matrices: a semi-structured distance matrix and a random flow matrix. In order to enhance the diversity of the instances, we generate 4 additional asymmetric instances in order to have a more diverse benchmark for the analysis. The instance generation technique used is loosely inspired by the one proposed in [47]:

- **Distance matrix (D):** First, \( n \) uniform random points are generated in a \( 100 \times 100 \) rectangle, and the euclidean distance \( e(i, j) \) between every pair of points \( 1 \leq i, j \leq n \) is computed. Then, for every \( 1 \leq i, j \leq n \) such that \( i \neq j \), \( d_{ij} = \sqrt{e(i, j)} \) where \( \sqrt{v} \) is a uniform random value between 0.85 and 1.15. Finally, the matrix is scaled so that \( 0 \leq d_{ij} \leq 100 \) for every \( 1 \leq i, j \leq n \). All the values are rounded to the closest integer.

- **Flow matrix (H):** For every \( 1 \leq i, j \leq n \) such that \( i \neq j \), a uniform random value \( x \) between 0 and 1 is sampled. If \( x > sp \) where \( sp \) is a parameter that indicates the sparsity of the matrix, \( h_{ij} \) is set to a uniform random integer between 0 and 100. Otherwise, \( h_{ij} = 0 \).
benchmark, the generated instances are created using different parameter settings: \( sp = 0 \) (xab20a), \( sp = 0.25 \) (xab20b), \( sp = 0.5 \) (xab20c) and \( sp = 0.75 \) (xab20d). The higher the value of \( sp \), the higher the sparsity of the flow matrix.

### 4.2 Number of local optima

The number of local optima of a combinatorial landscape can be related to the difficulty of finding the global optima using local search algorithms [26, 32]; therefore, it can be used as an indirect measure of complexity. However, it is unfeasible to exhaustively calculate the number of local optima of the benchmark instances due to the size of their search spaces. For that reason, in this study the ChaoLee2 estimator that is reviewed in [32] has been considered. This estimation technique approximates the total number of local optima from a sample of local optima that is obtained by performing a basic local search starting from \( M \) random solutions. The considered local search works under the swap neighborhood and chooses the best solution at each step. From the \( M \) initial solutions, \( r \) unique local optima \( \Theta = \{ \sigma_1^*, \sigma_2^*, ..., \sigma_r^* \} \subset S_n \) \((r \leq M)\) are obtained. Based on this sample, the estimator makes a distinction between easy-to-find and hard-to-find optima, which is defined by a \( \delta \) parameter that indicates the minimum number of times an optimum must be observed to be considered easy-to-find. The number of easy-to-find and hard-to-find optima in the sample is then used to estimate the total number of local optima. For the sake of brevity, we do not explain the exact formula used by ChaoLee2, so we refer the interested reader to [32].

According to [32], when all the obtained local optima are different \((r = M)\), the ChaoLee2 method does not work. In these cases, the strategy followed is to take a random sample of solutions, and the proportion of local optima in the sample is used as an estimator of the proportion of local optima in the entire search space. This alternative estimation method is called CountOptima.

Until now, we have assumed that all the local optima are independent from each other, so we have purposely put aside some aspects that greatly influence the structure of the landscapes. One of the most important factors is the plateaus formed by local optima [33]. A plateau in a combinatorial landscape is a set of solutions \( \Psi = \{ \sigma_1 \} \subset S_n \) \((1 \leq \sigma_1 \leq M)\) such that, for every pair of solutions \( x, y \in \Psi \), satisfies \( f(x) = f(y) \) and there is a path \((x = a_1, a_2, ..., a_k = y)\) such that \( a_i \in P \) and \( a_{i+1} \in N(a_i) \). A plateau that is formed by multiple local optima can be considered as a unique local optimum when applying local search based algorithms, so a landscape in which most of the local optima are clustered on plateaus may be much less rugged than the estimates suggest.

Thus, in order to validate the obtained results, the local optima found in the sample \( \Theta \) are grouped into plateaus. Given any pair of local optima \( \sigma_1^*, \sigma_2^* \in \Theta \), we consider that \( \sigma_1^* \) and \( \sigma_2^* \) are part of the same plateau if \( f(\sigma_1^*) = f(\sigma_2^*) \) and there is a path \((\sigma_1^* = a_1^*, a_2^*, ..., a_k^* = \sigma_2^*)\) such that \( a_i^* \in \Theta \) and \( a_{i+1}^* \in N(a_i^*) \). Although considering only the local optima found in the sample speeds up the computations, the results are less precise since we could consider as different plateaus two sets of solutions that are connected by a path of local optima \((b_1^*, b_2^*, ..., b_l^*)\) such that \( b_i \notin \Theta \). Nevertheless, this strategy provides an approximate upper bound for the number of plateaus in the sample in a reasonable amount of time.

#### 4.2.1 Results

The number of local optima of each of the elementary landscapes for the benchmark instances are estimated based on a sample of 200,000 local searches per instance and landscape. The cutoff parameter of the ChaoLee2 estimator is set to \( \delta = 10 \) according to the recommendations of [15]. When 100% of the encountered local optima appear only once in the sample, the alternative CountOptima method is used based on 200,000 random solutions. The obtained results are shown in Table 1.

As can be observed, the CountOptima method estimates that 100% of the solutions in the search space of \( L_1 \) are local optima in all the symmetric and semi-symmetric benchmark instances. Although this could lead us to believe that \( L_1 \) is a very rugged landscape, this is not really the case. According to [50], the objective function of \( L_1 \) turns out to be constant \((f_1(\sigma_1) = f_1(\sigma_2)\) for all \( \sigma_1, \sigma_2 \in S_n \)) when at least one of the matrices that form the instance is symmetric, which explains why every solution in the search space of the symmetric and semi-symmetric instances is a local optimum itself. This affirmation has already been formally demonstrated in [50], but, for the sake of completeness, we provide an alternative mathematical proof in Appendix A.

Table 2 shows the results for the plateaus formed by the local optima found in the samples. As the method used only computes an upper bound for the number of plateaus, when a landscape is constant it may return misleading results (more than 1 plateau). Therefore, to avoid confusion, we omit the \( L_1 \) landscape in symmetric and semi-symmetric instances when grouping the local optima.

In asymmetric benchmark instances, \( L_1 \) seems to be a very rugged landscape, since the estimated number of local optima and plateaus is much higher than in the rest of the elementary landscapes. Moreover, as not all the solutions in the search space of \( L_1 \) are local optima, we also know that in these cases \( L_1 \) is not constant. This is the major difference with respect to symmetric and semi-symmetric instances.

The results also show that, independently of the symmetry of the instance, the \( L_3 \) landscape is, in general, much less rugged than \( L_1 \) and \( L_2 \). In fact, in almost all the symmetric and semi-symmetric benchmark instances, \( L_3 \) seems to have just 1 plateau formed by local optima. Two exceptions are the exc16 instances, in which the estimated proportion of local optima in the search space is 100%. However, we suspect that in these cases the objective function of \( L_3 \) is constant, just like in the case of \( L_1 \).

### 4.3 Contribution of the elementary components

The second analysis focuses on the study about the relative contribution of the elementary components of the decomposition to the structure of the problem. The aim is to find out the elementary landscapes that have the greatest influence in each type of instance. For this purpose, we first compute the spectral amplitudes [18, 35, 46] of the eigenvalues of \( f_1, f_2 \) and \( f_3 \) for the benchmark instances. The spectral amplitudes can be considered as a measure of the relative contribution of each of the elementary landscapes to the variance of the general objective function \( f \), and are calculated as:

\[
W_i = \frac{f_i^2 - \overline{f_i}^2}{f^2 - \overline{f}^2}
\]  

(8)
Table 1: Estimated number of local optima. In the cases where CountOptima is used (italic), the results are shown as the estimated percentage of local optima in the search space.

<table>
<thead>
<tr>
<th></th>
<th>chr15a</th>
<th>chr20a</th>
<th>esc16a</th>
<th>esc16b</th>
<th>had18</th>
<th>had20</th>
<th>rou15</th>
<th>rou20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Instances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>L2</td>
<td>72.730</td>
<td>3.628.387</td>
<td>47.831.068</td>
<td>9.999.900.000</td>
<td>81.995</td>
<td>454.134</td>
<td>150.772</td>
<td>6.153.458</td>
</tr>
<tr>
<td>L3</td>
<td>4</td>
<td>8</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>1.152</td>
<td>768</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Semi-symmetric Instances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
<td>100.00 %</td>
</tr>
<tr>
<td>L2</td>
<td>9.311.635</td>
<td>10.453.678</td>
<td>167.744.885.090</td>
<td>198.743</td>
<td>5.787</td>
<td>28.443</td>
<td>300.217</td>
<td>3.188.513</td>
</tr>
<tr>
<td>L3</td>
<td>9.999.900.000</td>
<td>32</td>
<td>0.00 %</td>
<td>48</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Approximate upper bound for the number of plateaus in the samples compared to the number of local optima (in brackets). L1 is omitted in symmetric and semi-symmetric instances.

<table>
<thead>
<tr>
<th></th>
<th>chr15a</th>
<th>chr20a</th>
<th>esc16a</th>
<th>esc16b</th>
<th>had18</th>
<th>had20</th>
<th>rou15</th>
<th>rou20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Instances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>L2</td>
<td>41.892 (41.892)</td>
<td>189.658 (189.736)</td>
<td>191.971 (199.581)</td>
<td>199.803 (199.998)</td>
<td>15.533 (34.165)</td>
<td>49.842 (95.928)</td>
<td>61.009 (61.009)</td>
<td>193.591 (193.591)</td>
</tr>
<tr>
<td>L3</td>
<td>1 (4)</td>
<td>1 (8)</td>
<td>200.000 (200.000)</td>
<td>200.000 (200.000)</td>
<td>1 (1.152)</td>
<td>1 (768)</td>
<td>1 (1)</td>
<td>1 (1)</td>
</tr>
<tr>
<td>Semi-symmetric Instances</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>L2</td>
<td>192.030 (192.312)</td>
<td>191.735 (191.739)</td>
<td>199.977 (199.979)</td>
<td>198.743 (198.743)</td>
<td>5.125 (5.125)</td>
<td>17.942 (17.942)</td>
<td>81.067 (81.067)</td>
<td>181.475 (181.475)</td>
</tr>
<tr>
<td>L3</td>
<td>199.691 (199.998)</td>
<td>1 (32)</td>
<td>200.000 (200.000)</td>
<td>1 (48)</td>
<td>1 (2)</td>
<td>1 (1)</td>
<td>1 (1)</td>
<td>1 (1)</td>
</tr>
</tbody>
</table>

Asymmetric Instances

<table>
<thead>
<tr>
<th></th>
<th>bur26a</th>
<th>bur26b</th>
<th>bur26c</th>
<th>bur26d</th>
<th>xab20a</th>
<th>xab20b</th>
<th>xab20c</th>
<th>xab20d</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>5.458.734</td>
<td>999.900.000</td>
<td>6.971.361</td>
<td>1.999.900.000</td>
<td>16.750.465</td>
<td>11.065.476</td>
<td>13.993.626</td>
<td>7.358.769</td>
</tr>
<tr>
<td>L2</td>
<td>279.060</td>
<td>1.147.427</td>
<td>145.420</td>
<td>716.890</td>
<td>261.008</td>
<td>240.978</td>
<td>484.125</td>
<td>222.968</td>
</tr>
<tr>
<td>L3</td>
<td>13.823</td>
<td>21.324.218</td>
<td>41.400</td>
<td>7.763.110</td>
<td>7</td>
<td>6</td>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>

where \( W_i \) is the spectral amplitude that measures the contribution of \( L_i \). By definition, \( W_1 + W_2 + W_3 = 1 \).

Once the contribution of each of the elementary components to the overall fitness is quantified, the next step is to study the relationship between the local optima of \( L \) and the local optima of the elementary landscapes. In particular, we estimate the percentage of the local optima of \( L \) that are also local optima of \( L_1, L_2 \) or \( L_3 \), for each of the benchmark instances. This measure provides information about the relative contribution of the elementary landscapes to the local optima of the overall problem, which is relevant when working with local search based algorithms.

4.3.1 Results. The exact values of the spectral amplitudes for the benchmark instances are calculated as explained in [18]. The obtained results are shown as heat maps in Figure 1.

As the results show, the \( L_3 \) landscape seems to be especially important on symmetric and semi-symmetric benchmark instances. In fact, in almost all the cases, the spectral amplitude of \( L_3 \) is 0.7 or higher. On the other hand, the contribution of the \( L_3 \) landscape seems to be remarkable on asymmetric benchmark instances, since its spectral amplitude is higher than 0.5 in half of them.

Figure 1: Spectral amplitudes. The red color indicates that the amplitude is closer to 0, while the blue color indicates that the amplitude is closer to 1.
The experiments carried out show that the symmetry of the instances has an important impact on the difficulty and relative contribution of the elementary landscapes that form the decomposition of the QAP. All these characteristics might be exploited when designing specific meta-heuristic approaches.

We have proven that in symmetric and semi-symmetric instances the objective function of \( L_1 \) is constant. Therefore, for optimization purposes these types of instances are composed of just two elementary landscapes. Among the non-constant landscapes, \( L_2 \) is generally the elementary landscape that has the greatest relative contribution to both the fitness and the local optima of the problem. Thus, it might be good idea to focus on \( L_2 \) when optimizing symmetric and semi-symmetric instances. In asymmetric instances, however, there is not a clearly predominant elementary landscape. This suggests that in these cases all the elementary landscapes should be considered during optimization, especially \( L_2 \) and \( L_3 \).

As we can see, symmetric and semi-symmetric instances seem to have similar characteristics. This is due to the fact that any semi-symmetric instance can be converted into a symmetric instance without modifying its fitness landscape [22]. Thus, semi-symmetric instances are, in fact, special cases of the symmetric QAP.

In addition to the aforementioned differences, we have also found some interesting characteristics that are common to all the considered types of instances. For example, \( L_3 \) has in general fewer local optima (and plateaus formed by local optima) than the rest of the elementary landscapes regardless of the symmetry of the instance, although this feature is especially evident in symmetric and semi-symmetric problems. This characteristic makes \( L_3 \) a particularly easy landscape for local search optimization.

5 LOCAL SEARCH BASED ALGORITHM

Based on the analysis made in the previous section, we propose a specific local search based algorithm [48] that uses the elementary landscape decomposition of the problem to efficiently optimize the QAP. Our main goal is to use this algorithm to experimentally verify the conclusions of the previous analysis. To this end, in this section we explain the proposed method and study its performance on a benchmark of instances.

5.1 Variable Function Search

One of the main problems of the local search algorithms is that they can get stuck in poor quality local optima. Because of that, when developing local search based algorithms, most of the work consists of finding strategies to efficiently escape from local optima in order to reach better solutions. In this work, we propose a strategy that is based on the elementary landscape decomposition of the QAP. Specifically, the proposed local search based algorithm, called Variable Function Search (VFS), consists of the following steps:

1. Starting from a random solution \( \sigma \in S_n \), a basic local search is applied. The considered local search works under the swap neighborhood (\( N \)) and selects the best solution at each step until a local optimum \( \sigma^* \in S_n \) is reached.
2. In order to escape from the local optimum, the swap neighborhood \( N(\sigma^*) \) is explored according to the objective functions of the decomposition \( (f_1, f_2, f_3) \).
3. If the algorithm finds a solution \( \sigma^* \in N(\sigma^*) \) that is better than \( \sigma^* \) in at least one of the objective functions of

---

**Figure 2:** Estimated percentage of the local optima of \( L \) that are also local optima of \( L_1, L_2 \) or \( L_3 \). The red color indicates that the percentage is closer to 0\%, while the blue color indicates that the percentage is closer to 100\%. \( L_1 \) is omitted in symmetric and semi-symmetric instances.

In all cases, the least important landscape is \( L_1 \). As we have already explained, the objective function of this landscape is constant in symmetric and semi-symmetric instances (\( W_1 = 0 \)), and its contribution in asymmetric benchmark instances is very limited, with a spectral amplitude that ranges from 0.02 to 0.03.

Next, we estimate the percentage of the local optima of \( L \) that are also local optima of \( L_1, L_2 \) or \( L_3 \) for each of the benchmark instances from the local optima found in the samples of Section 4.2. As the \( L_1 \) landscape is entirely composed of local optima in symmetric and semi-symmetric instances (constant objective function), we omit this case in order to avoid confusion. The obtained results are shown as heat maps in Figure 2.

In the view of the results, \( L_2 \) shares a significant number of local optima with \( L \) in the majority of symmetric and semi-symmetric benchmark instances. However, the percentage of shared local optima varies greatly from one instance to another, ranging from as low as 0.03\% to as high as 100.00\%. Be that as it may, it seems that the \( L_2 \) landscape has the most important influence on the local optima of \( L \) in these types of instances. The only exceptions in which the \( L_1 \) landscape also shares a high percentage of the local optima with \( L \) (100%) are the esc16 instances, but, as we have already mentioned, we suspect that this is because \( L_3 \) is constant.

In asymmetric benchmark instances, on the other hand, none of the elementary landscapes shares a significant number of local optima with \( L \) (< 1.00\% in all cases). Therefore, in these cases there does not seem to be a single elementary landscape that predominantly influences the local optima of the overall problem.

4.4 Discussion

The experiments carried out show that the symmetry of the instances has an important impact on the difficulty and relative contribution of the elementary landscapes that form the decomposition...
the decomposition, it returns to step 1 with $\sigma = \sigma'$. If there is more than one neighbor solution that satisfies this condition, the algorithm selects the best one among them according to $f$.

2.2. Otherwise, the algorithm stops and returns the best solution found according to $f$.

In order to avoid cycles, the VFS uses a tabu list [30] that stores the most recent neighborhood movements. Those movements cannot be undone until they leave the tabu list.

Although the VFS has been designed in the context of the QAP, it can be easily adapted to any problem for which an elementary landscape decomposition is known. This can be done by simply modifying the neighborhood and objective functions.

### 5.2 Experimental Study

In what follows, an experimental study is carried out in order to check whether the symmetry of the instances affects the performance of the proposed local search based algorithm. To this end, we compare the VFS with a classical Tabu Search (TS) [30] on the same benchmark of instances explained in Section 4.

For an instance of size $n$, the maximum tabu list size is set to $n$ in both algorithms. With respect to the stopping criterion, a number of solution evaluations have been set: $1000n^2$. Taking this into account, each pair algorithm-instance is run 100 times. In Table 3, we show for each case the median of the obtained objective function values and relative errors.

The median results show that the TS is equal to or better than the VFS in all the symmetric benchmark instances. In semi-symmetric and asymmetric benchmark instances, however, both algorithms seem to have a more similar performance. In fact, in these cases the VFS outperforms the TS in a significant number of instances (3 semi-symmetric instances, 4 asymmetric instances). Although we cannot affirm that these differences are exclusively due to the symmetry of the instances, we firmly believe that it is an important factor.

### 5.3 Statistical Analysis

In order to further compare the VFS and the TS, we carry out a statistical analysis using the Bayesian signed-rank test [3, 10], which is the Bayesian equivalent of the Wilcoxon test [52]. This technique considers the experimental data and computes the expected probability of each algorithm being the best among all the compared methods. The used implementation is available in the *scmamp* R package [11].

As it is our aim to study the differences between different types of instances, instead of performing one global statistical analysis we carry out three independent analyses, one per instance type (symmetric, semi-symmetric, asymmetric). The experimental data used in the statistical analyses is composed of the relative errors obtained in the experimentation of Section 5.2. The Bayesian signed-rank test requires the definition of the region of practical equivalence (Rope), that is, the interval in which the performance of two algorithms is considered equivalent. In this work, we consider that the performance of two algorithms is equivalent if the difference between their relative errors is smaller than $10^{-6}$. The results of the performed statistical analyses are shown in Figure 3.

#### Table 3: Median of the fitness and relative errors obtained in 100 runs of the algorithms. The best algorithm for each of the benchmark instances is highlighted in green. If there is a tie, both algorithms are highlighted in yellow. The best-known solutions for *xab20a*, *xab20b*, *xab20c* and *xab20d* are the best solutions found in the experimentation.

<table>
<thead>
<tr>
<th></th>
<th>VFS</th>
<th>Tabu Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best known</td>
<td>Fitness</td>
<td>Rel. Error</td>
</tr>
<tr>
<td>chr15a</td>
<td>9.896</td>
<td>0.014240</td>
</tr>
<tr>
<td>chr20a</td>
<td>2.192</td>
<td>0.090715</td>
</tr>
<tr>
<td>esc16a</td>
<td>68</td>
<td>0.029412</td>
</tr>
<tr>
<td>esc16b</td>
<td>292</td>
<td>0.000000</td>
</tr>
<tr>
<td>had18</td>
<td>5.358</td>
<td>0.002240</td>
</tr>
<tr>
<td>had20</td>
<td>6.922</td>
<td>0.003756</td>
</tr>
<tr>
<td>rou15</td>
<td>354.210</td>
<td>0.000000</td>
</tr>
<tr>
<td>rou20</td>
<td>725.522</td>
<td>0.003771</td>
</tr>
<tr>
<td>lipa20a</td>
<td>3.683</td>
<td>0.000000</td>
</tr>
<tr>
<td>lipa20b</td>
<td>27.076</td>
<td>0.000000</td>
</tr>
<tr>
<td>lipa30a</td>
<td>13.178</td>
<td>0.000000</td>
</tr>
<tr>
<td>lipa30b</td>
<td>151.426</td>
<td>0.000000</td>
</tr>
<tr>
<td>tai20b</td>
<td>122.453.319</td>
<td>0.0013482</td>
</tr>
<tr>
<td>tai25b</td>
<td>344.355.646</td>
<td>0.136489</td>
</tr>
<tr>
<td>tai30b</td>
<td>657.117.13</td>
<td>0.124027</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>VFS</th>
<th>Tabu Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best known</td>
<td>Fitness</td>
<td>Rel. Error</td>
</tr>
<tr>
<td>bur26a</td>
<td>5.426.679</td>
<td>0.001607</td>
</tr>
<tr>
<td>bur26b</td>
<td>3.817.852</td>
<td>0.002697</td>
</tr>
<tr>
<td>bur26c</td>
<td>5.426.795</td>
<td>0.001911</td>
</tr>
<tr>
<td>bur26d</td>
<td>3.821.225</td>
<td>0.002563</td>
</tr>
<tr>
<td>xab20a</td>
<td>847.472</td>
<td>0.005883</td>
</tr>
<tr>
<td>xab20b</td>
<td>465.155</td>
<td>0.000952</td>
</tr>
<tr>
<td>xab20c</td>
<td>382.155</td>
<td>0.001418</td>
</tr>
<tr>
<td>xab20d</td>
<td>96.815</td>
<td>0.010040</td>
</tr>
</tbody>
</table>

Figure 3: Results of the Bayesian signed-rank test for each instance type shown as simplex plots.

In short, the points in the simplex plots represent a sampling of the posterior distribution of the probability of win-lose-tie. That is, the closer a point is to the VFS vertex, the higher the probability that the VFS has a better performance, and vice versa. The same applies for the TS and Rope vertices, where the last one indicates...
that both algorithms are practically equivalent. The dashed lines delimit the dominance region of each possibility, that is, the area where the highest probability corresponds to its vertex. In order to better visualize the obtained results, the expected probabilities for each possibility (TS-VFS-Rope) are shown in Table 4.

As we can see, different situations can be observed depending on the symmetry of the instances. In symmetric instances, the TS is the best algorithm with a 0.4978 probability, while in asymmetric instances the VFS is better with a 0.4085 probability. In semi-symmetric instances, however, both algorithms seem to have a similar performance, since according to the statistical analysis the TS and the VFS are practically equivalent with a 0.4546 probability. As the spread of the points in the three plots is quite low, there is almost no uncertainty about the results of the analysis, so we can confirm that the VFS and the TS have different performances on the different sets of benchmark instances. As previously stated, this could be due to the symmetry of the instances.

Although we can only hypothesize the reason why this happens, we think that it may be mainly due to some of the characteristics that have been investigated in Section 4. For example, as the VFS uses the objective functions of the elementary landscapes to escape from the local optima, when one of the elementary landscapes is constant the escape ability of the algorithm may worsen. Moreover, when Ł shares a high percentage of the local optima with an elementary landscape, the escape ability may also be impacted. All these factors combined could explain why the performance of the VFS is worse in symmetric and semi-symmetric instances compared to asymmetric instances.

### 6 CONCLUSIONS AND FUTURE WORK

The symmetry of the distance and flow matrices is just one of the characteristics of QAP instances. Nevertheless, in this work we have seen that it has a great influence on the difficulty and importance of the elementary components that form the elementary landscape decomposition. As a result, the symmetry of the instances also appears to be very important when deciding which meta-heuristic strategies should be used to solve a particular QAP problem.

However, it is important to remark that this is only an exploratory research, so there is still much work to be done to measure the real influence of the symmetry of the QAP on the performance of optimization algorithms. On the one hand, the experiments on which we have based this paper have been performed on a relatively small benchmark of instances. Therefore, in the future, the number of instances considered in the experimentation should be increased in order to verify the conclusions of the analysis. On the other hand, this work has been mainly focused on local search based algorithms, so it would be interesting to extend the analysis to other types of meta-heuristics such as, for example, evolutionary algorithms.

### APPENDIX A

In this appendix, we prove that the objective function of $L_1$ is constant when the distance matrix $D$ or the flow matrix $H$ is symmetric.

As the demonstration is identical in both cases, we focus on the first one (symmetric distance matrix, $d_{a,b} = d_{b,a}$). First, we rewrite Equation (4) as follows:

$$f_i(\sigma) = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \sum_{c=1}^{n-1} \sum_{d \neq c+1}^{n} g(a,b),(c,d)(\sigma)$$

where

$$g(a,b),(c,d)(\sigma) = \frac{d_{a,b}h_{c,d}^1}{2n} + \frac{\phi_1(a,b)(c,d)}{2n}$$

and

$$\phi_1(a,b)(c,d)$$

with $a \neq b, c \neq d$ and $1 \leq a, b, c, d \leq n$. If we can prove that $g(a,b),(c,d)(\sigma)$ always has the same value regardless of $\sigma$, then we prove that $f_i$ is a constant function and, therefore, $L_1$ is a constant landscape. To this end, we analyze the five possible situations when $D$ is symmetric:

- $\sigma(a) = c \land \sigma(b) = d$. The summands of $g(a,b),(c,d)(\sigma)$ are:

<table>
<thead>
<tr>
<th>Summand</th>
<th>Case</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{a,b}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
<td>$\sigma(i) = p \land \sigma(j) = q$</td>
<td>$d_{a,b}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
</tr>
<tr>
<td>$d_{b,a}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
<td>$\sigma(i) = q \land \sigma(j) = p$</td>
<td>$d_{b,a}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
</tr>
<tr>
<td>$d_{a,b}h_{d,c} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
<td>$\sigma(i) = q \land \sigma(j) = p$</td>
<td>$d_{a,b}h_{d,c} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
</tr>
<tr>
<td>$d_{b,a}h_{d,c} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
<td>$\sigma(i) = p \land \sigma(j) = q$</td>
<td>$d_{b,a}h_{d,c} \frac{\phi_1(a,b)(c,d)}{2n}$</td>
</tr>
</tbody>
</table>

Therefore, we have

$$g(a,b),(c,d)(\sigma) = d_{a,b}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n} + d_{b,a}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$$

For the sake of brevity, from now on we omit the explicit explanation of each of the summands. Their values can be calculated from Equation (7) based on the parameters of $\phi_1$.

- $\sigma(a) = d \land \sigma(b) = c$.

$$g(a,b),(c,d)(\sigma) = d_{a,b}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n} + d_{b,a}h_{c,d} \frac{\phi_1(a,b)(c,d)}{2n}$$

Thus, we have proved that $g(a,b),(c,d)(\sigma)$ is a constant for all $1 \leq a, b, c, d \leq n$ with $a \neq b$ and $c \neq d$. That is, the value of $g(a,b),(c,d)(\sigma)$ is independent of $\sigma$. Therefore, $f_i$ is constant.

### ACKNOWLEDGMENTS

This work has been partially funded by the Elkartek Program (KK-2020/00049), Consolidated Groups (IT1244-19) and BERC 2018-2021 program by the Basque Government, the PID2019-106453GA-I00/AEI/10.3309/SO110011033 project and the BCAM Severo Ochoa excellence accreditation SEV-2017-0718 from the Spanish Ministry of Science and Innovation, and the European Research Council H2020 (the EMPHATIC project).
REFERENCES


for a change: A tutorial for comparing multiple classifiers through Bayesian anal-


imization: Overview and Conceptual Comparison. ACM Comput. Surv. 35 (01
2001), 268–308.


assignment problem library. European Journal of Operational Research 55 (02
1992), 115–119.

tionen mittels quadratischer Zuordnungsprobleme. Mathematical Methods of


ranking analysis. 324–325.


Attraction Basins: Breaking with the Intuition. Evolutionary Computation 27 (05
2019), 1–32.

Problem. 1571–1578.


Parallel computing 17, 4-5 (1991), 443–455.

44 (03 2001).

Parallel computing 23 (01 1997), 55–76.


[19] Luca Maria Gambardella and D. Taillard. 2000. Ant Colonies for the Quadratic

Parallel computing 17, 4-5 (1991), 443–455.

tary landscape decompositions. Evolutionary Intelligence
44 (03 2001).

On the symmetry of the QAP through ELD GECCO '21 Companion, July 10–14, 2021, Lille, France