

Generating QAP instances with known optimum solution and additively decomposable cost function

Mădălina M. Drugan

the date of receipt and acceptance should be inserted later

Abstract *Quadratic assignment problems* (QAPs) is a NP-hard combinatorial optimization problem. QAPs are often used to compare the performance of meta-heuristics. In this paper, we propose a QAP problem instance generator that can be used for benchmarking for heuristic algorithms. Our QAP generator combines small size QAPs with known optimum solution into a larger size QAP instance. We call these instances *composite QAPs* (cQAPs), and we show that the cost function of composite QAPs is additively decomposable. We give mild conditions for which a composite QAP instance has known optimum solution. We generate cQAP instances using uniform distributions with different bounds for the component QAPs and for the rest of the cQAP elements. Numerical and analytical techniques that measure the difficulty of the composite QAP instances in comparison with other QAPs from the literature are introduced. These methods point out that some cQAP instances are difficult for local search with many local optimum of various values, low epistasis and non-trivial asymptotic behaviour.

Keywords Combinatorial optimization, Meta-heuristics, Quadratic assignment problem, Instance generator, additively decomposable functions, landscape analysis

1 Introduction

The Quadratic assignment problem (QAP) models many real-world problems like computer aided design in the electronics industry, scheduling, vehicle routing, etc. Therefore, the QAP instances are often included in the benchmarks for testing meta-heuristics [6, 17, 19, 19]. Recent extensive reviews on QAPs are given in [4, 13]. Intuitively, QAPs can be described as the (optimal) assignment of a number of facilities to a number of locations. In general, QAP instances are NP hard problems. There are special cases of QAPs that are easy to solve [2, 4, 7, 23]. A problem is considered easy if it is solvable in polynomial time. Exact solutions for hard QAPs are very expensive to obtain even for moderate size QAPs, about 30 facilities.

M.M. Drugan
Computational Modeling group, Artificial Intelligence Lab, Department of Computer Science, Vrije Universiteit Brussel,
Pleinlaan 2, B-1050 Brussels, Belgium
E-mail: mdrugan@vub.ac.be

Meta-heuristic search algorithms based on local search are especially useful for large size problems. Measuring their performance on test problems is best done when the optimum solution for that problem is known.

Let us consider N facilities, a set $\Pi(N)$ of all permutations of $\{1, \dots, N\}$ and the $N \times N$ distance matrix $A = (a_{ij})$, where a_{ij} is the distance between location i and location j . We assume a flow matrix $B = (b_{ij})$ where b_{ij} represents the flow from facility i to facility j . The goal is to minimize the *cost function*

$$c(\pi) = \sum_{i=1}^N \sum_{j=1}^N a_{ij} \cdot b_{\pi_i \pi_j} \quad (1)$$

where π is a permutation of N facilities and π_i is the i -th element of π . It takes quadratic time to evaluate this function. We consider a QAP as a tuple (A, B, s) where s , if known, is the optimum solution. If the optimum solution is not known, we denote the QAP simply with (A, B) . Here, we consider QAP instances that were initially used in [21]. We pose three requirements for the flow and distance matrices. The distance and the flow between the same facilities is 0, $\forall i, a_{ii} = 0$ and $b_{ii} = 0$. The distance and the flow matrices are symmetrical, $\forall i, j, a_{ij} = a_{ji}$ and $b_{ij} = b_{ji}$. All the elements of the distance and flow matrices are non-negative, natural numbers within a range, $\forall i, j, 0 \leq a_{ij} \in \mathbb{N}$ and $0 \leq b_{ij} \in \mathbb{N}$.

Related work. There are several QAP instance generator algorithms [3, 16] that generate hard and large instances of QAPs with a known optimum. They start with an easy QAP instance and iteratively add modifications to the QAP instance in order to make it harder. In these methods, solving QAP instances is associated with solving the bi-partitioning problem. These methods are rather elaborated and restrictive in the type of QAPs generated, and impose strict conditions on the structure of the underlying graph and on the structure variations.

Drezner et al. [5] propose instances of QAPs that are difficult to solve with some heuristics, but easy to solve with exact methods. QAP instances proposed in [5] are simplified versions of instances from [16] where the underlying grid is very sparse and the distribution that positions facilities on this grid is random. Their goal is to generate difficult QAPs with sparse flow matrices. The other type of QAP instance generator from [5] splits the facilities in regions, or clusters. Because of their sparseness, these QAPs are easy to solve with some exact methods [14], even with about 90 facilities, and difficult for heuristics that use the common two facilities exchange operator [15].

Taillard [21] analysed the performance of some heuristics on several QAPs from the literature. He points out that, even though generating uniform randomly distributed values for distance and flow matrices is a popular choice, the resulting QAPs are not interesting. Their exact optimum is difficult to find, but the difference between some easy to find local optima and the global optimum value is small. Moreover, the difference in cost function between the best and the worst cost values is approaching 0 when the number of facilities approaches ∞ . He explained this behaviour by the lack of correlation between local optima. Taillard created a larger variation between the worst and the best local optimum using non-uniform random elements for distance and flow matrices. He concluded that large size QAPs with interesting structure to exploit by heuristic search algorithms are difficult to generate by a problem generator. For an overview of the recent results and developments in the area of random QAPs, we refer to Krokhmal and Pardalos [11].

Main contributions. Our goal is to design a QAP instance generator that creates useful benchmark instances for heuristics like iterated local search which exploit the particularities of the search space. We consider that a QAP benchmark instance should have the following

properties: i) large size, ii) difficult and interesting to solve with both heuristics and exact algorithms, iii) known optimum solution, and iv) not trivial asymptotic behaviour (i.e., the difference between the lower and the upper cost functions does not go to 0 when the number of facilities goes to ∞).

Our solution is to consider several block flow and distance matrices for QAP instances [6]. We aggregate QAPs with computable optimum solutions, into a larger QAP such that the optimum of the resulting QAP is known. We call this class of QAPs, the *composite QAPs* (cQAPs). There are three optimization problems in cQAPs: i) optimizing the component QAPs, ii) optimizing the region outside these components, and iii) a global optimization problem for the entire cQAP. In this way, we will generate multiple local optima considered challenging for heuristics.

The composite QAP has an additively decomposable cost function that can be expressed as the sum of component QAP's cost functions plus two extra terms corresponding to the overlapping parts of the component QAPs and the region outside these component QAPs. Problems with additively decomposable functions, are considered useful test benchmark for meta-heuristic algorithms that explore the structure of the search space [18]. The two extra terms makes the problem difficult for the heuristics and the exact algorithms (i.e., branch-and-bound) that explicitly explore the decomposable structure of the generated QAPs. To hide the block structure of cQAP and its exact solution, we rename the facilities in a random way in order.

We give mild conditions for which cQAPs have the identity permutation as the global optimum solution. We split the analysis of cQAP in two based on the relative positioning of component QAP instances into the component QAP: i) overlapping and ii) non-overlapping. The analysis considers the cyclic form from the permutation group theory, where a cycle here corresponds with a component QAP. The result is the upper and lower bounds for the values in the component and composite QAP instances, but also the relationship between their distributions such that the identical permutation is the global optimum of a cQAP instance. This approach can be considered an exact algorithm where the goal is to find the conditions which easily solves cQAP instances. However, the verification if the global optimum solution is the identical permutation depends on the number of permutations of the component QAP instances into the cQAP instance, making the generation of large cQAPs computationally infeasible. In order to simplify the procedure of generating cQAPs with known global optima, we consider specific generating distributions (i.e., uniform). The conditions on the upper and lower bounds are now easily verifiable, and practical numerical values are proposed.

To generate cQAP instances, a number of parameters need to the set, like the number, the size and the distribution of the component QAPs, and the distributions of the elements in the outside region of cQAP. We use several measurements from the literature (i.e., epistasis) to classify the difficulty of cQAPs for local search algorithms. Accordingly to these measures, cQAP instances are difficult instances for heuristics also when compared with other QAP instances from literature.

Approximating the number and the size of basins of attraction is a good method for landscape analysis [8]. The larger the number and the size of local optima, the more probable it becomes that local search gets stuck before finding the global optimum. We show that even small size QAPs have a lot of local optima and their number increases with the number of facilities. Due to the possible huge number of local optima, we propose an alternative definition and empirical approximation of basins of attraction.

Outline. Section 2 introduces an algorithm that generates composite QAP instances. Section 3 gives the conditions under which these QAPs have the identity permutation as

Algorithm 1: generate_cQAP

Require: d component QAP instances $\{(A^1, B^1, \mathcal{I}), \dots, (A^d, B^d, \mathcal{I})\}$
Require: d allocation strategies $\{\mathcal{S}^1, \dots, \mathcal{S}^d\}$, where $\mathcal{S}^k : \{1, \dots, n^k\} \rightarrow \{1, \dots, N\}$
Require: $\mathcal{H}_A, \mathcal{L}_A, \mathcal{H}_B$ and \mathcal{L}_B discrete distributions
 Calculate N using the allocation strategies
 Initialize A_C, B_C, A_O and B_O with 0s everywhere
 $(A_C, B_C) \leftarrow \text{assemble}(\{(A^1, B^1, \mathcal{I}), \dots, (A^d, B^d, \mathcal{I})\}, \{\mathcal{S}^1, \dots, \mathcal{S}^d\})$
 $(A_O, B_O) \leftarrow \text{fillingUp}(\mathcal{H}_A, \mathcal{L}_A, \mathcal{H}_B, \mathcal{L}_B)$
 $A \leftarrow A_C + A_O; B \leftarrow B_C + B_O$
return (A, B)

optimum solution and Section 4 gives the bounds of uniform distributions for which these conditions hold. In Section 5, we compare some properties of cQAP instances with similar QAP instances from literature. Section 6 uses landscape analysis to show desirable properties of cQAP instances for heuristics. Section 7 concludes this paper.

2 A composite QAP instances generator

In this section, we design an algorithm that generates composite QAP instances from d small size QAP matrices with identity permutation \mathcal{I} as optimum solution, where $\forall i \in \{1, \dots, N\}, \mathcal{I}_i = i$. A straightforward method to transform a component QAP with an optimum solution s into a QAP with identity permutation as optimum solution is to rename the facilities. The algorithm `generate_cQAP` has as input d allocation strategies, \mathcal{S}^i , one for each component QAP, that allocates facilities from a component QAP, (A^i, B^i, \mathcal{I}) , to the composite QAP. The number of facilities N is calculated from the allocation strategies. The third set of inputs is the set of discrete distributions used to fill up the remaining elements of the component QAP, where \mathcal{H}_A and \mathcal{H}_B are the high valued distributions and \mathcal{L}_A and \mathcal{L}_B are the low valued distributions for the matrices A and B of the composite QAP, respectively. In Algorithm 1, the pseudo-code for generating composite QAPs, there are two function calls.

The function `assemble` aggregates the input component QAPs. The composite mask \mathcal{C} keeps track of positions in the matrices of cQAP assigned by the elements of the component QAPs. Then, $c_{ij} = 1$ if the a_{ij} and b_{ij} were assigned in the `assemble` algorithm. Otherwise, $c_{ij} = 0$. The elements from A assigned in `assemble` are denoted with a *masked flow matrix* A_C , where a_{ij} is defined if $c_{ij} = 1$, and a_{ij} is undefined otherwise. Similarly, the elements from B assigned in `assemble` are denoted with a *masked distance matrix* B_C . The corresponding pseudo-code is given in Algorithm 2 and it is explained in Section 2.1.

The function `fillingUp` assigns the positions in cQAP's matrices not assigned with `assemble`. Thus, the elements a_{ij} and b_{ij} are assigned with `fillingUp` if and only if $c_{ij} = 0$. The corresponding pseudo-code is given in Algorithm 3 and it is described in Section 2.2. For ease of notation, we consider an *outside region mask matrix* $\mathcal{O} = -\mathcal{C}$. The elements from A assigned with `fillingUp` are denoted with a masked matrix A_O , and the elements from B assigned with `fillingUp` are denoted with B_O . We show a condition between the elements in this region for which their cost function has the identity permutation as the optimum solution.

Section 2.3 comments on the interplay between the two regions, the component and the outside regions. The algorithm `generate_cQAP` returns the distance matrix $A \leftarrow A_C + A_O$ and the flow matrix $B \leftarrow B_C + B_O$.

2.1 The allocation strategies

Let S^k be an *allocation strategy* or function that assigns facilities from each component QAP, (A^k, B^k, \mathcal{I}) to the facilities of composite QAP. We denote with j the facility from the cQAP that corresponds to the facility i from the component k . Thus, $j \leftarrow S_i^k$ and $S^k : \{1, \dots, n^k\} \rightarrow \{1, \dots, N\}$ where n^k the number of facilities in the k -th component QAP. For each facility from the cQAP, there exists at least one facility of a component QAP. To each facility i from (A^k, B^k, \mathcal{I}) , it corresponds a unique facility t from (A, B) , where $t \leftarrow S_i^k$. The union of all assignment functions is the number of facilities in cQAP, $\cup_{k=1}^d S^k = \{1, \dots, N\}$.

We classify allocation strategies using two criteria: i) by the number of facilities from component QAPs that overlap in the composite QAP, ii) adjacent (or not) sequences of facilities assigned from the component QAPs into the cQAP. When there is no overlap between facilities of the component QAPs, each facility from the cQAP corresponds to exactly one facility from one component QAP. Then, $S^k \cap S^p = \emptyset, \forall k \neq p$. If two facilities from two component QAPs are assigned to a single facility in the cQAP, we add their corresponding values in distance and flow matrices of component QAPs. Then, $\exists k, p, S^k \cap S^p \neq \emptyset$, meaning that the facilities of the component QAPs overlap. To ease the notation, we denote with $S^{k,p}$ the intersection between S^k and S^p .

If the facilities of the component QAPs are assigned to non-adjacent facilities in the composite QAP, then the composite QAP is not diagonal. Assigning the facilities from each component QAP randomly, without replacement, to facilities of cQAP, is an example of non-contiguous assignment of facilities. The resulting cQAP can be transformed in a diagonal QAP by swapping rows and columns.

The choice of allocation strategies is important also in calculating N . A non-overlapping strategy implies that the size, or the number of facilities, of the cQAP is the sum of the sizes of the component QAPs. Then, $N = \sum_{k=1}^d n^k$. For overlapping allocation strategies, N is smaller than the sum of the sizes of component QAPs.

In Algorithm 2, for each pair of facilities $(i, j) \in A^k$ from each k -th component QAP, there assigned a pair of facilities (t, p) in the resulting A_C . We update the values $a_{tp} \in A_C$ and $b_{tp} \in B_C$ with the corresponding values in $a_{ij}^k \in A^k$ and $b_{ij}^k \in B^k$. Then, $a_{tp} \leftarrow a_{tp} + a_{ij}^k$ and $b_{tp} \leftarrow b_{tp} + b_{ij}^k$, respectively. Since A_C and B_C should be symmetrical, we also set $a_{pt} \leftarrow a_{tp}$ and $b_{pt} \leftarrow b_{tp}$. The mask \mathcal{C} is updated to 1 for the pair of indices t and

Algorithm 2: assemble($\{(A^k, B^k, \mathcal{I}) \mid k \leq d\}, \{S^1, \dots, S^d\}$)

```

for k = 1 to d do
  for i = 1 to  $n^k$  do
     $t \leftarrow S_i^k$ 
    for j = i + 1 to  $n^k$  do
       $p \leftarrow S_j^k$ 
       $a_{tp} \leftarrow a_{tp} + a_{ij}^k; a_{pt} \leftarrow a_{tp}$ 
       $b_{tp} \leftarrow b_{tp} + b_{ij}^k; b_{pt} \leftarrow b_{tp}$ 
       $c_{tp} \leftarrow 1; c_{pt} \leftarrow 1$ 
    end for
  end for
end for
return  $(A_C, B_C)$ 

```

Algorithm 3: fillingUp($\mathcal{H}_A, \mathcal{L}_A, \mathcal{H}_B, \mathcal{L}_B$)

```

 $\mathcal{R}_A \leftarrow \emptyset$  and  $\mathcal{R}_B \leftarrow \emptyset$  two ordered sets
 $t \leftarrow 0$ 
for  $i, j = 1$  to  $N$ , and  $c_{ij} = 0$  do
  if  $\alpha < U(0, 1)$  then
    Generate  $a_{ij} \propto \mathcal{H}_A$  and  $b'_t \propto \mathcal{L}_B$ 
  else
    Generate  $a_{ij} \propto \mathcal{L}_A$  and  $b'_t \propto \mathcal{H}_B$ 
  end if
   $\mathcal{R}_A \leftarrow \mathcal{R}_A \cup a_{ij}$  and  $\mathcal{R}_B \leftarrow \mathcal{R}_B \cup b'_t$ 
   $t \leftarrow t + 1$ 
end for
for  $r = 1$  to  $|\mathcal{R}_A|$  do
   $r \leftarrow$  rank of  $a_{ij}$  in  $\mathcal{R}_A$ 
   $b_{ij} \leftarrow b'_t$  with rank  $|\mathcal{R}_A| - r$  in  $\mathcal{R}_B$ 
end for
return  $(A_{\mathcal{O}}, B_{\mathcal{O}})$ 

```

p , $c_{tp} \leftarrow 1$, and $c_{pt} \leftarrow 1$. The returned, yet incomplete cQAP, is added to the output of Algorithm 3.

2.2 The outside region

To preserve the identity permutation as the optimum solution for the cQAP, our algorithm guarantees that the elements in $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$ obey the rearrangement inequality [22].¹ Informally, the rearrangement inequality states that the values in $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$ are either larger or smaller than values in $A_{\mathcal{C}}$ and $B_{\mathcal{C}}$, respectively. Thus, the largest values in $B_{\mathcal{O}}$ correspond to the lowest values in $A_{\mathcal{O}}$ and the lowest values in $B_{\mathcal{O}}$ correspond to the largest values in $A_{\mathcal{O}}$.

Consider that the distribution \mathcal{L}_A samples the lowest values of $A_{\mathcal{O}}$, and the distribution \mathcal{L}_B samples the lowest values of $B_{\mathcal{O}}$. Consider that the distribution \mathcal{H}_A samples the highest values of $A_{\mathcal{O}}$, and the distribution \mathcal{H}_B samples the highest values of $B_{\mathcal{O}}$. We generate α values in $A_{\mathcal{O}}$ from \mathcal{H}_A and $1 - \alpha$ from \mathcal{L}_A . Because of the rearrangement inequality, α values in $B_{\mathcal{O}}$ are generated from \mathcal{L}_A and $1 - \alpha$ are generated from \mathcal{H}_A .

Algorithm 3 presents the pseudo-code for generating the elements in $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$. fillingUp has as input parameters the set of distributions for the outside region, $\mathcal{H}_A, \mathcal{L}_A, \mathcal{H}_B, \mathcal{L}_B$, and the percentage of values generated from \mathcal{H}_A in $A_{\mathcal{O}}$, α .

Let \mathcal{R}_A and \mathcal{R}_B be ordered sets containing all the values from $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$, respectively. Let r be the rank of a_{ij} in \mathcal{R}_A , where $c_{ij} = 0$. If a_{ij} is generated from \mathcal{H}_A , then b_{ij} is generated from \mathcal{L}_B such that the rank of b_{ij} in \mathcal{R}_B is $|\mathcal{R}_A| - r$. Similarly, if a_{ij} is generated from \mathcal{L}_A , then b_{ij} is generated from \mathcal{H}_B such that the rank of b_{ij} in \mathcal{R}_B is $|\mathcal{R}_A| - r$.

¹ Let n variables be generated with any two distributions $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ for which $x_1 \leq \dots \leq x_n$ and $y_1 \geq \dots \geq y_n$. The rearrangement inequality states that $\sum_{i=1}^n x_i \cdot y_i \leq \sum_{i=1}^n x_i \cdot y_{\pi_i}$, for all permutations π .

2.3 Outside vs composite regions

Cela [4] showed that QAP instances where all the elements obey the rearrangement inequality are *easy*. When the component QAPs are degenerated, $n^1 = \dots = n^d = 1$, the cQAP becomes the "easy" QAP of Cela [4]. We consider the composite region \mathcal{C} to be the difficult region of the generated cQAP, whereas the outside region \mathcal{O} is the easy region of the cQAP. By design, the outside region and all the component QAPs are optimized by the identity permutation, but the composite QAP's optimum solution is not, in general, the identity permutation. In the next section, we give conditions for which the generated composite QAP instances have the identity permutation as an optimum solution.

Let m_A be the smallest element for all the component distance matrices A^k , and let m_B be the smallest element for all the component flow matrix B^k . Then, $m_A \leftarrow \min_{k,i,j} \{a_{ij}^k\}$ and $m_B \leftarrow \min_{k,i,j} \{b_{ij}^k\}$. Let M_A be the largest element from all the component distance matrices A^k , and let M_B be the largest element for all the component flow matrices B^k . Then, $M_A \leftarrow \max_{k,i,j} \{a_{ij}^k\}$ and $M_B \leftarrow \max_{k,i,j} \{b_{ij}^k\}$. Let ℓ_A and L_A be the lowest and the highest bounds of the distribution \mathcal{L}_A , and let ℓ_B and L_B be the lowest and the highest bounds of the distribution \mathcal{L}_B . Let h_A and H_A be the lowest and the highest bounds of the distribution \mathcal{H}_A and let h_B and H_B be the lowest and the highest bounds of the distribution \mathcal{H}_B . We take $\ell_A < L_A < h_A < H_A$ and $\ell_B < L_B < h_B < H_B$.

We consider that all the elements in the outside region are either smaller or larger than all the elements in the composite region. Then, when elements are exchanged between the two regions, accordingly to the rearrangement inequality, the cost of the cQAP instance increases. This means that $\forall a_{ij} \in A_C$ and $\forall b_{ij} \in B_C$, we have $L_A \leq a_{ij} \leq h_A$, and $L_B \leq b_{ij} \leq h_B$. If there is not overlap between the component QAP instances, we have $\ell_A \leq L_A < m_A < M_A < h_A < H_A$ and $\ell_B \leq L_B < m_B < M_B < h_B < H_B$. This condition is used in Section 3 to generate cQAP instances with identity permutation as the optimum solution.

3 Designing cQAPs with known global optimum solution

In this section, we study the conditions for which a cQAP has the identity permutation as the optimal solution.

Additively decomposable cost functions of the cQAP instances. Consider the set $\Pi(N)$ of all permutations of N facilities in the flow matrix. In the permutation group theory, permutations are often written in cyclic form. If π is a permutation of facilities, we can write it as $\pi = (\pi^1, \dots, \pi^d)$, where π^k is the k -th cycle containing a set of facilities that can be swapped with each other. These cycles are disjoint subsets. We consider d cycles, each cycle contains the facilities of exactly one component QAP. If there are n^k facilities in the k -th component QAP, the corresponding cycle is a n^k -cycle. The cycle of a component QAP k has the cost function

$$c^k(\pi) = \sum_{i,j,\pi_i,\pi_j \in S^k} a_{ij}^k \cdot b_{\pi_i \pi_j}^k \quad (2)$$

where $k \in \{1, \dots, d\}$, d is the number of component QAPs and a_{ij}^k is an element of the k -th component QAP. Similarly, $b_{\pi_i \pi_j}^k$ is an element of the k -th component QAP. By design, the optimal cost for each cycle is $c^k(\mathcal{I}) = \min_{\pi} c^k(\pi)$.

The cost function of π is now

$$c(\pi) = \sum_{i=1}^N \sum_{j=1}^N a_{ij} \cdot b_{\pi_i \pi_j} = \sum_{k=1}^d c^k(\pi) + R_C(\pi) + R_O(\pi) \quad (3)$$

where

$$R_O(\pi) = \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi_i \pi_j}=0} a_{ij} \cdot b_{\pi_i \pi_j} \quad (4)$$

is a residue defined as the sum of costs for the elements in the outside region. In the sequel, the residue $R_C(\pi)$ is the sum over all overlapping positions of the QAPs and depends on the overlapping pattern. Thus, the composite QAP instances have *additively decomposable cost functions* with two residual terms representing the costs of the overlapping parts and the outside region, respectively.

3.1 Generating non-overlapping composite QAP instances

In this section, we give mild conditions for which a composite QAP with non-overlapping component QAPs has a known global optimum. The cost function of any permutation π from Equation 3 can be formulated as

$$c(\pi) = \sum_{k=1}^d c^k(\pi) + R_O(\pi)$$

where $R_O(\pi)$ as in Equation 4. Since there is no overlap in the component QAPs, then $R_C(\pi) = 0$.

The set $\Pi(N)$ of all possible permutations is split in three disjunct subsets defined by: i) products of cycles over the component QAPs, ii) cycles that completely switch their facilities with other cycles, and iii) the general case where facilities are switched between cycles. We now express the cost function of a permutation using Equation 3.

i) Swapping facilities in their cycles. When a facility belongs to the k -th component, so does its permutation. Thus, $\forall i, k$, we have $i \in S^k$ iff $\pi_i \in S^k$. Let $\pi = (\pi^1, \pi^2, \dots, \pi^d)$ be a subset of permutations expressed as a product of d cycles, where a k -cycle is defined over the facilities of the k -th component QAP as in Equation 2.

ii) Swapping cycles. Let $\pi' = (\pi'^1, \dots, \pi'^d)$ be a permutation of n -cycles, where the facilities of component QAPs are swapped entirely. Thus, if the k -th and the p -th cycles are swapped, then $\forall i \in S^k$ and $\pi'_i \in S^p$. The cost function of a cycle is now

$$c^k(\pi') = \sum_{i,j \in S^k \wedge \pi'_i, \pi'_j \in S^p} a_{ij}^k \cdot b_{\pi'_i \pi'_j}^p \quad (5)$$

iii) Swapping facilities between cycles. For the rest of the permutations, exchanging facilities between cycles results also in swapping facilities between component QAPs and the outside region. This represents the majority of the possible permutations and includes the other two cases, i.e. *i)* and *ii)*, as particular cases. When facilities are swapped between the component QAPs, there are elements swapped between the composite and the outside region. For example, for such a permutation π'' , if $i, j, \pi''_i \in S^k$ and $\pi''_j \in S^p$, then $a_{ij}^k \in A_C$ and $b_{\pi''_i \pi''_j} \in B_O$.

The cost function when both facilities and their permutation belong to the k -th component QAP is

$$c^k(\pi'') = \sum_{i,j,\pi_i'',\pi_j'' \in \mathcal{S}^k} a_{ij}^k \cdot b_{\pi_i''\pi_j''}^k + \sum_{i,j \in \mathcal{S}^k \wedge \pi_i'',\pi_j'' \in \mathcal{S}^p} a_{ij}^k \cdot b_{\pi_i''\pi_j''}^p \quad (6)$$

The residue for the outside region contains three sum of products of elements in the composite and the outside region

$$R_{\mathcal{O}}(\pi'') = \sum_{k=1}^d \sum_{r=1}^d \sum_{\substack{p=1 \\ r \neq k, p \neq r, k}}^d \sum_{\substack{i,j \in \mathcal{S}^k \\ \pi_i'' \in \mathcal{S}^r \wedge \pi_j'' \in \mathcal{S}^p}} a_{ij}^k \cdot b_{\pi_i''\pi_j''}^k + \sum_{k=1}^d \sum_{r=1}^d \sum_{\substack{p=1 \\ r \neq k}}^d \sum_{\substack{t=1 \\ t \neq p}}^d \sum_{\substack{i \in \mathcal{S}^k \wedge j \in \mathcal{S}^r \\ \pi_i'' \in \mathcal{S}^p \wedge \pi_j'' \in \mathcal{S}^t}} a_{ij} \cdot b_{\pi_i''\pi_j''} \quad (7)$$

In the following proposition, we give mild conditions for which a composite QAP instance with non-overlapping component QAP instances has the identity permutation as optimum solution. Intuitively, the proof considers the difference between the identity permutation and another permutation for all three cases presented above.

Proposition 1 *Let be $\{(A^k, B^k, \mathcal{I}) \mid k = 1, \dots, d\}$ a set of equal sized QAP instances with identity permutation as optimum solution, and Algorithm 1 to generate a composite QAP from these component QAPs. Let the component QAPs be non-overlapping and let the following equations hold*

$$\ell_A < L_A < m_A < M_A < h_A < H_A \quad \text{and} \quad \ell_B < L_B < m_B < M_B < h_B < H_B \quad (8)$$

$$\min\{m_A, m_B\} \cdot (\min\{h_A, h_B\} + \min\{\ell_A, \ell_B\}) > M_A \cdot M_B + \min\{h_A \cdot L_B, L_A \cdot h_B\} \quad (9)$$

$$\sum_{k=1}^d \left(c^k(\mathcal{I}) - c^k(\pi) \right) + \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi_i\pi_j}=0} a_{ij} \cdot (b_{ij} - b_{\pi_i\pi_j}) < 0 \quad (10)$$

where π any permutation. Then, (A, B, \mathcal{I}) has the identity permutation as optimum solution.

Proof Let π be a permutation. The difference between the cost function of π and the cost function of \mathcal{I} is

$$\Delta(c(\mathcal{I}), c(\pi)) = c(\mathcal{I}) - c(\pi) = \sum_{i,j=1}^N a_{ij} \cdot b_{ij} - \sum_{i,j=1}^N a_{ij} \cdot b_{\pi_i\pi_j} = \sum_{i,j=1}^N a_{ij} \cdot (b_{ij} - b_{\pi_i\pi_j})$$

\mathcal{I} is the global optimum solution if $\Delta(\mathcal{I}, \pi) < 0$ for all π .

i) Swapping facilities in their cycles. In this case, $\Delta(c(\mathcal{I}), c(\pi)) < 0$ because, by design:

1. for each component QAP, $\forall k$, the identity permutation is optimal, $c^k(\mathcal{I}) - c^k(\pi) < 0$;
2. the elements in the outside region obey the rearrangement inequality

$$\sum_{c_{ij}=0} a_{ij} \cdot b_{ij} < \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi_i\pi_j}=0} a_{ij} \cdot b_{\pi_i\pi_j}$$

ii) *Swapping cycles.* From Equation 5, we deduce that

$$\Delta(c(\mathcal{I}), c(\pi')) = \sum_{k=1}^d \left(c^k(\mathcal{I}) - c^k(\pi') \right) + \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi'_i \pi'_j}=0} a_{ij} \cdot (b_{ij} - b_{\pi'_i \pi'_j}) \quad (11)$$

This difference is smaller than 0 when Inequality 10 holds.

iii) *Swapping facilities between cycles.* From Equation 6 and 7, the difference in cost function is

$$\begin{aligned} \Delta(c(\mathcal{I}), c(\pi'')) &= \sum_{k=1}^d \sum_{i,j,\pi''_i,\pi''_j \in \mathcal{S}^k} a_{ij}^k \cdot (b_{ij}^k - b_{\pi''_i \pi''_j}^k) + \sum_{k=1}^d \sum_{\substack{p=1 \\ p \neq k}}^d \sum_{\substack{i,j \in \mathcal{S}^k \\ \pi''_i, \pi''_j \in \mathcal{S}^p}} a_{ij}^k \cdot (b_{ij}^k - b_{\pi''_i \pi''_j}^p) \\ &+ \sum_{k=1}^d \sum_{\substack{r=1 \\ r \neq k}}^d \sum_{\substack{p=1 \\ p \neq r, k}}^d \sum_{\substack{i,j \in \mathcal{S}^k \wedge \\ \pi''_i \in \mathcal{S}^r \wedge \pi''_j \in \mathcal{S}^p}} a_{ij}^k \cdot (b_{ij}^k - b_{\pi''_i \pi''_j}^p) + \sum_{k=1}^d \sum_{\substack{r=1 \\ r \neq k}}^d \sum_{\substack{p=1 \\ p \neq r, k}}^d \sum_{\substack{i \in \mathcal{S}^r \wedge j \in \mathcal{S}^p \\ \pi''_i, \pi''_j \in \mathcal{S}^k}} a_{ij} \cdot (b_{ij} - b_{\pi''_i \pi''_j}^k) \\ &+ \sum_{k=1}^d \sum_{\substack{r=1 \\ r \neq k}}^d \sum_{\substack{p=1 \\ p \neq k}}^d \sum_{\substack{t=1 \\ t \neq p}}^d \sum_{\substack{i \in \mathcal{S}^k \wedge j \in \mathcal{S}^r \wedge \\ \pi''_i \in \mathcal{S}^p \wedge \pi''_j \in \mathcal{S}^t}} a_{ij} \cdot (b_{ij} - b_{\pi''_i \pi''_j}) \quad (12) \end{aligned}$$

In the first term of this difference, all facilities are swapped within the same cycle resembling the sum from Equation 6. Thus, like Equation 6, this sum is negative even when it does not contain all the facilities of a component QAP because the facilities exchanged with other cycles are ignored. In the last term from Equation 12, i.e. the fifth term, elements of the outside region are swapped, and the corresponding cost function is negative accordingly to the rearrangement inequality.

In the second term of Equation 12, elements of the component QAPs are swapped. The second and the last terms represent the difference from Equation 11. Although some facilities might be missing from the initial equation, accordingly to Inequality 10, Equation 11 holds for any π' that swaps only cycles. If the number of swaps between the component QAPs increases, then the number of products obtained by swapping elements of the composite and outside region, i.e. the third and fourth terms in Equation 12, decreases.

We now show that the sum of the third and fourth terms from Equation 12 where elements between the composite and the outside region are swapped has a negative cost. Using the rearrangement inequality and Equation 8, we infer that multiplying values from the composite and outside region has a larger result than multiplying values from the same region. Thus, the cost of swapping facilities between the composite and outside region is lower bounded by $\min\{m_A, m_B\} \cdot (\min\{h_A, h_B\} + \min\{\ell_A, \ell_B\})$. In the sequel, the cost of swapping facilities in the composite region is upper bounded by $M_A \cdot M_B$, and the cost of swapping facilities in the outside region is upper bounded by $\min\{h_A \cdot L_B, L_A \cdot h_B\}$. Inequality 9 results directly.

This concludes our proof. \square

In Proposition 1, Inequality 8 ensures that the rearrangement inequality holds. Inequality 10 states that if swapping elements in the outside region generates more variance than swapping cycles then the identity permutation is the global minimum for the subset of permutations where cycles are completely swapped. From Inequality 9 and the rearrangement

inequality, we have that a permutation where facilities are swapped between the outside and the composite region has a higher cost than a permutation where solutions are swapped *in* the composite or *in* the outside region. Note that the condition in Inequality 9 can always be fulfilled by setting the bounds of the distributions \mathcal{H}_A and \mathcal{H}_B high enough.

The result from Proposition 1 allows the combination of interesting QAP instances from QAPLIB. Note that there are needed $d!$ verifications of Inequality 10 in order to decide if the composite QAP has the identity permutation as global minimum. Thus, this exact approach is not practical for large number of component QAP instances.

3.2 Generating composite QAPs with overlap

In general, when the component QAPs overlap, the overlapping regions have a different optimum than the identity permutation. In this section, we give mild conditions for the composite QAP instances to have identity permutation as optimal solution. For ease off the discussion, we consider a simple pattern for the overlapping component QAPs, where each component QAP has the same size n and it is overlapping with exactly two other component QAPs. Each overlap is half of a component QAP's size, $\lfloor n/2 \rfloor$. For each facility of the composite QAP, there are exactly two component QAPs that overlap on. When all d component QAPs overlap, there are d overlapping regions and the total number of facilities is $N = d \cdot n/2$. The flow and distance matrices of all component QAPs are generated with the same distribution \mathcal{D} , where m and M are the lower and upper bounds of \mathcal{D} . The elements of $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$ are generated with the distributions \mathcal{H} and \mathcal{L} . The lower bounds of \mathcal{L} and \mathcal{H} are denoted with ℓ and h , respectively. We denote with L and H the higher bounds of \mathcal{L} and \mathcal{H} , respectively. To make a qualitative distinction between the overlapping and non-overlapping regions, we consider that all non-overlapping values are lower than all overlapping values, $M < 2 \cdot m$. Then,

$$\ell < L < m < M < 2 \cdot m < 2 \cdot M < h < H \quad (13)$$

Now, a $(k, k+1)$ -cycle is defined as the overlap between the k -th and the $k+1$ -th component QAP instances. We denote with $\pi^{k,k+1}$ its permutation. The k -th component QAP contains two cycles corresponding to the two overlapping areas of the k -th component QAP, the $(k-1, k)$ -cycle and the $(k, k+1)$ -cycle. A permutation of the k -th component is $\pi^k = (\pi^{k-1,k}, \pi^{k,k+1})$. Like for the non-overlapping cycles, the permutations are split in three categories based on the distribution of cycles.

i) Swapping facilities in their cycles. Let $\pi = (\pi^{1,2}, \dots, \pi^{d-1,d}, \pi^{d,1})$ be a permutation that can be expressed as d product of cycles. The cost of the $(k, k+1)$ -cycle is

$$c^{k,k+1}(\pi) = \sum_{\substack{i,j \in \mathcal{S}^{k,k+1} \\ \pi_i, \pi_j \in \mathcal{S}^{k,k+1}}} (a_{ij}^k + a_{ij}^{k+1}) \cdot (b_{\pi_i \pi_j}^k + b_{\pi_i \pi_j}^{k+1})$$

The residue cost of the overlapping part from Equation 2 is

$$R_{\mathcal{C}}(\pi) = \sum_{k=1}^d c^{k-1,k}(\pi^{k,k+1}) + c^{k,k+1}(\pi^{k-1,k})$$

where $c^{k-1,k}(\pi^{k,k+1})$ and $c^{k,k+1}(\pi^{k-1,k})$ are the cost function of the non-overlapping parts of the k -th component QAP

$$c^{k-1,k}(\pi^{k,k+1}) = \sum_{\substack{i,j \in \mathcal{S}^{k-1,k} \\ \pi_i, \pi_j \in \mathcal{S}^{k,k+1}}} a_{ij}^k \cdot b_{\pi_i \pi_j}^{k+1}$$

$$c^{k,k+1}(\pi^{k-1,k}) = \sum_{\substack{i,j \in \mathcal{S}^{k,k+1} \\ \pi_i, \pi_j \in \mathcal{S}^{k-1,k}}} a_{ij}^{k+1} \cdot b_{\pi_i \pi_j}^k$$

The cost $c^{k-1,k}$ of the identity permutation $\mathcal{I}^{k,k+1}$ is

$$c^{k-1,k}(\mathcal{I}^{k,k+1}) = \sum_{i,j \in \mathcal{S}^{k-1,k}} a_{ij}^k \cdot b_{i+n/2, j+n/2}^{k+1}$$

ii) Swapping cycles. Consider that the facilities between two cycles are entirely swapped. Let $\pi' = (\pi'^{1,2}, \dots, \pi'^{d-1,d}, \pi'^{d,1})$ be a permutation of the d cycles. When the $(k, k+1)$ -th and the $(p, p+1)$ -th cycles are swapped, if $i \in \mathcal{S}^{k,k+1}$ then $\pi_i \in \pi^{p,p+1}$, and the cost function of this cycle is

$$c^{k,k+1}(\pi') = \sum_{\substack{i,j \in \mathcal{S}^{k,k+1} \\ \pi_i, \pi_j \in \mathcal{S}^{p,p+1}}} (a_{ij}^k + a_{ij}^{k+1}) \cdot (b_{\pi_i \pi_j}^p + b_{\pi_i \pi_j}^{p+1})$$

iii) Swapping facilities between cycles are the largest number of permutations being the general case covering the other two previous cases.

The following proposition gives mild conditions to have \mathcal{I} as exact solution in overlapping cQAPs. The proof of this proposition is given in Appendix A.

Proposition 2 Let $\{(A^k, B^k, \mathcal{I}) \mid k = 1, \dots, d\}$ be d equal sized QAP instances with identity permutations as optimum solutions that are combined into a cQAP using Algorithm 1. Let $\mathcal{S}^{k,p}$ be an assignment function such that: *i)* $\forall k \mid \mathcal{S}^{k,k+1} = \lfloor n/2 \rfloor$, and *ii)* $\forall k$ and $p \notin \{k-1, k+1\}$, $\mathcal{S}^{k,p} = \emptyset$. Inequalities 13, 14 and 15 hold

$$m \cdot (h + \ell) > 4 \cdot M^2 + h \cdot L \quad (14)$$

$$\sum_{k=1}^d (c^k(\mathcal{I}) - c^k(\pi)) + \sum_{c_{ij}=0 \wedge c_{\pi_i \pi_j}=0} a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j}) \quad (15)$$

$$\sum_{k=1}^d \left(c^{k-1,k}(\mathcal{I}^{k,k+1}) - c^{k-1,k}(\pi^{k,k+1}) + c^{k,k+1}(\mathcal{I}^{k+1,k+2}) - c^{k,k+1}(\pi^{k+1,k+2}) \right) < 0$$

where $c^k(\pi)$ as in Equation 2.

Then, (A, B, \mathcal{I}) has the identity permutation as exact solution.

To decide if a generated cQAP has the identity permutation as optimum solution, we need $d!$ evaluations of Inequality 15 corresponding to all combinations of the component QAPs on the diagonal of cQAP.

4 Generating composite QAP instances using uniform distributions

In this section, we generate cQAP instances such that the conditions from Proposition 1 and Proposition 2 hold without additional evaluations of the relationships between distributions. The component QAPs are independently generated with an uniform random distribution, \mathcal{D} , the same for all the components. Let \mathcal{L} and \mathcal{H} be uniform independent distributions, and let the values of \mathcal{L} and \mathcal{H} be uniform randomly located in $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$. We give mild conditions on the upper and lower bounds of the uniform distributions and we exemplify them with numerical values. Note that even though component QAPs and the elements in the outside region are generated by uniform random distributions, the values of the corresponding cQAP instances does not belong to a uniform random distribution.

4.1 Composite QAP instances with non-overlapping component QAPs

We study how the inequalities from Proposition 1 depend on the bounds of the uniform distributions. To compare cQAP instances with the uniform randomly generated QAPs of Taillard [21], let the flow and the distance matrices be generated by the same distribution.

Let the two terms from Inequality 10 be denoted as the variance of the composite region and of the outside region

$$\Delta_{\mathcal{C}} = \sum_{k=1}^d c^k(\mathcal{I}) - c^k(\pi)$$

$$\Delta_{\mathcal{O}} = R_{\mathcal{O}}(\mathcal{I}) - R_{\mathcal{O}}(\pi) = \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi_i \pi_j}=0} a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j})$$

We explicitly compute the values of $\Delta_{\mathcal{C}}$ and $\Delta_{\mathcal{O}}$ in order to calculate the sing of the sum $\Delta_{\mathcal{O}} + \Delta_{\mathcal{C}}$. If $\Delta_{\mathcal{O}} + \Delta_{\mathcal{C}}$ is non-negative, the identity permutation is the global optimum solution.

Consider that there are $L - \ell + 1$ values in \mathcal{L} , $\ell = s_0, s_1, \dots, s_{L-\ell} = L$, and $H - h + 1$ uniformly generated values in \mathcal{H} , $(h = t_0), t_1, \dots, (t_{H-h} = H)$. Let's assume that $L - \ell = H - h$. With a perfect random generator, in any row and column of $A_{\mathcal{O}}$ and $B_{\mathcal{O}}$ values of \mathcal{L} and of \mathcal{H} are equally represented.

4.1.1 The variance in the outside region, $\Delta_{\mathcal{O}}$

The value of the identity permutation \mathcal{I} is the minimum value and does not depend on α , which represents the distribution of low and high values. Accordingly with the rearrangement inequality, the lowest values are multiplied with the highest values. Assuming that all the values of the distributions \mathcal{L} and \mathcal{H} are uniformly distributed in the outside region, the outside region has the approximative value of

$$R_{\mathcal{O}}(\mathcal{I}) = \sum_{c_{ij}=0} a_{ij} \cdot b_{ij} \approx \frac{|A_{\mathcal{O}}|}{H - h + 1} \cdot \left(\sum_{i=0}^{L-\ell} s_i \cdot t_{H-h-i} \right) \quad (16)$$

Let π be a permutation obtained from \mathcal{I} by swapping w facilities. As a result, in the outside region $|A_{\mathcal{O}}|$ a non-overlapping pattern for a cQAP instance, there are a p percent swapped elements, $p \cdot |A_{\mathcal{O}}| = w \cdot (N - d)$, where $p \in [0, 1]$ and the number of elements in the outside region in both the flow and distance matrix is $|A_{\mathcal{O}}| = N^2 - d \cdot n^2$.

i) $\alpha = 0.5$. Then, the elements in the flow and distance matrices are equally generated from low and high distributions, \mathcal{L} and \mathcal{H} . The swapped elements are randomly distributed in $B_{\mathcal{O}}$ and, thus, the cost of the outside region, $R_{\mathcal{O}}(\pi)$, is upper bounded by

$$R_{\mathcal{O}}(\pi) \leq \frac{|A_{\mathcal{O}}|}{L - \ell + H - h + 2} \cdot \left(\sum_{i=0}^{L-\ell} s_i + \sum_{j=0}^{H-h} t_j \right)^2$$

Assuming that $(1-p) \cdot |A_{\mathcal{O}}|$ percent of the outside region is optimized and the remaining $p \cdot |A_{\mathcal{O}}|$ percent of the outside region is uniform randomly positioned in the matrix, the cost of the outside region is approximated by

$$R_{\mathcal{O}} \approx \frac{(1-p) \cdot |A_{\mathcal{O}}|}{L - \ell + 1} \cdot \left(\sum_{i=0}^{L-\ell} s_i \cdot t_{L-\ell-i} \right) + \frac{p \cdot |A_{\mathcal{O}}| \cdot \left(\sum_{i=0}^{L-\ell} s_i + \sum_{j=0}^{H-h} t_j \right)^2}{L - \ell + H - h + 2}$$

Given a certain value for p , the variance in the outside region is

$$\Delta_{\mathcal{O}} \approx \frac{p \cdot |A_{\mathcal{O}}|}{L - \ell + 1} \cdot \left(\sum_{i=0}^{L-\ell} s_i \cdot t_{L-\ell-i} \right) - \frac{p \cdot |A_{\mathcal{O}}| \cdot \left(\sum_{i=0}^{L-\ell} s_i + \sum_{j=0}^{H-h} t_j \right)^2}{L - \ell + H - h + 2} \quad (17)$$

ii) $\alpha = 1.0$. The values in the distance matrix are generated by \mathcal{H} and the values in the flow matrix are generated by \mathcal{L} . The cost of the outside region, $R_{\mathcal{O}}(\pi)$ is upper bounded by

$$R_{\mathcal{O}}(\pi) \leq \frac{|A_{\mathcal{O}}|}{(L - \ell + 1) \cdot (H - h + 1)} \cdot \left(\sum_{i=0}^{L-\ell} s_i \right) \cdot \left(\sum_{j=0}^{H-h} t_j \right)$$

If we assume that $(1-p) \cdot |A_{\mathcal{O}}|$ percent of the outside region is minimized, then the cost of the outside region is approximated by

$$R_{\mathcal{O}} \approx \frac{(1-p) \cdot |A_{\mathcal{O}}|}{L - \ell + 1} \cdot \left(\sum_{i=0}^{L-\ell} s_i \cdot t_{L-\ell-i} \right) + \frac{p \cdot |A_{\mathcal{O}}| \cdot \left(\sum_{i=0}^{L-\ell} s_i \right) \cdot \left(\sum_{j=0}^{H-h} t_j \right)}{(H - h + 1) \cdot (L - \ell + 1)}$$

The variance in the outside region is about

$$\Delta_{\mathcal{O}} \approx \frac{p \cdot |A_{\mathcal{O}}|}{L - \ell + 1} \cdot \left(\sum_{i=0}^{L-\ell} s_i \cdot t_{L-\ell-i} \right) - \frac{p \cdot |A_{\mathcal{O}}| \cdot \left(\sum_{i=0}^{L-\ell} s_i \right) \cdot \left(\sum_{j=0}^{H-h} t_j \right)}{(H - h + 1) \cdot (L - \ell + 1)} \quad (18)$$

iii) ζ . To compare our composite QAP instances with Drezner et al. [5]'s instances, we consider a certain amount of 0's in the low distribution of the outside region \mathcal{L} , where $\zeta \in \{0, 0.5, 1.0\}$. $\zeta = 0$ means that there are no 0's in the outside region, and $\zeta = 1.0$ means that there are only 0's in \mathcal{L} .

Intuitively, the variance of an outside region with lots of 0s is smaller than the variance of an outside region with no 0s. In both Inequalities 17 and 18, if the amount of 0's increases, then $\Delta_{\mathcal{O}}$ decreases. If all low values in $A_{\mathcal{O}}$ are 0's, i.e. $\zeta = 1.0$, then the variance in the outside region is also 0.

4.1.2 The variance in the composite region

The minimum cost of all d component QAPs are approximately equal because all the values are generated from the same uniform distribution. These costs could be increased by: the imperfection of the random generator, the limited size of the component QAP, and the large number of sampled values. Consider that there are $M - m + 1$ values in \mathcal{D} , such that $(m = v_0), v_1 \dots, (v_{M-m} = M)$. Let $|A_C|$ be the number of elements in the composite region, where in $|A_C| = d \cdot n \cdot (n - 1)$ and $N = d \cdot n$.

Following the same line of reasoning, the maximum variance in these distributions is

$$\Delta_C < \sum_{k=1}^d c^k(\mathcal{I}) - \frac{|A_C|}{(M - m + 1)^2} \left(\sum_{i=0}^{M-m} v_i \right)^2 \quad (19)$$

The elements of the component matrices are uniform randomly generated and positioned. Thus, when the component QAPs are optimized and $N \rightarrow \infty$, we have

$$\frac{|A_C|}{(M - m + 1)^2} \left(\sum_{i=0}^{M-m} v_i \right)^2 \approx \sum_{k=1}^d c^k(\mathcal{I})$$

and $\Delta_C \rightarrow 0$.

4.1.3 A numerical example

For the previous discussion, we conclude that if $N \rightarrow \infty$, then Δ_C is approaching 0 and Δ_O has a negative value. Furthermore, Δ_O is more negative for $\alpha = 0.5$ than for $\alpha = 1.0$.

Let's consider the following numerical values: i) $m = 20$ and $M = 49$, ii) $h = 80$ and $H = 99$, and iii) $\ell = 1$ and $L = 20$. Thus $L - \ell = H - h = 20$. These bounds are chosen such that they cover a large number of values between 0 and 99, which are also the bounds of the uniform randomly generated QAPs of Taillard [21].

Let $n = 8$ be the number of facilities in component QAPs, where $d \geq 2$. Further, $s_i = i$ and $t_i = i + 80$, where $i \in \{0, \dots, 19\}$. Consider two values for $\alpha \in \{0.5, 1.0\}$. Note that the variance in the outside region is the same for $\alpha = 0$ and $\alpha = 1.0$ because both segregate the high and low distributions in the outside region A_O and B_O .

In non-overlapping cQAPs, where $|A_O| = N^2 - d \cdot n^2 = d \cdot (d - 1) \cdot n^2$, if $n = 8$ and $d = 2, 3, \dots$, then $|A_O| = 128, 384, 768, \dots$. The cost of the outside region should be well approximated by Equation 16, $R_O(\mathcal{I}) = |A_O| \cdot \frac{1 \cdot 99 + \dots + 20 \cdot 80}{20} = |A_O| \cdot 906.5$.

If $\alpha = 0.5$, then using Equation 17 we have that $\Delta_O \approx p \cdot |A_O| \cdot 906.5 - p \cdot |A_O| \cdot 6703.2 \approx -p \cdot |A_O| \cdot 5807.7$. Note that the second term it is negative and dominates Δ_O .

If $\alpha = 1.0$, then using Equation 18 we have that $\Delta_O \approx p |A_O| \cdot 906.5 - p \cdot |A_O| \cdot 15680^2/40 = -p \cdot |A_O| \cdot 6145653.5$. Again, the negative term is the largest in the equation. Thus, Δ_O for $\alpha = 1.0$ is much smaller than for $\alpha = 0.5$.

For $d = 2, 3, \dots$, we have $|A_C| = d \cdot n \cdot (n - 1) = 112, 168, 224, \dots$. Then, from Equation 19 we have that $\Delta_C < \sum_{k=1}^d c^k(\mathcal{I}) - |A_C| \cdot 29241$.

Thus, if $\alpha = 0.5$ then $\Delta_C + \Delta_O < 0$ and the identity permutation is the optimal solution. Otherwise, we need to explicitly compute the sum $\Delta_C - \Delta_O$. We found that the condition $\Delta_C + \Delta_O < 0$ is mild because from 100 generated cQAPs with different set of parameters α and ζ , less than 1% did not fulfil it.

4.2 Composite QAP instances with overlapping component QAPs

Inequality 15 from Proposition 2, that is checked following the same type of analysis like in Section 4.1. Using the definition of $\Delta_{\mathcal{C}}$ and $\Delta_{\mathcal{O}}$, we denote with

$$\Delta'_{\mathcal{C}} = \Delta_{\mathcal{C}} + \sum_{k=1}^d c^{k,k+1} (\mathcal{I}^{k+1,k+2}) - c^{k,k+1} (\pi^{k+1,k+2}) + c^{k-1,k} (\mathcal{I}^{k,k+1}) - c^{k-1,k} (\pi^{k,k+1})$$

$$\Delta'_{\mathcal{O}} = \Delta_{\mathcal{O}}$$

Note that if $N \rightarrow \infty$, then $\Delta'_{\mathcal{C}} \rightarrow 0$.

Consider L , H and D , as before. The variance in the outside region for overlapping cQAPs is equal with the variance in the outside region of the non-overlapping cQAPs, $\Delta'_{\mathcal{O}} = \Delta_{\mathcal{O}}$. The number of solutions in the outside region for $o = n/2$ and $N = d \cdot n/2$, is $|A_{\mathcal{O}}| = N^2 - 3 \cdot d \cdot n^2/4 = d \cdot (d-3) \cdot n^2/4$.

Additionally to $\Delta_{\mathcal{C}}$, $\Delta'_{\mathcal{C}}$ contains terms of the overlapping regions. All the elements of the composite regions are generated uniform randomly. Then,

$$\sum_{k=1}^d c^{k-1,k} (\pi^{k,k+1}) \leq \frac{n^2}{4 \cdot d^2} \cdot \left(\sum_{i=1}^d v_i \right)^2$$

Let $|A_{\mathcal{C}}| = d \cdot n \cdot (n-1)$ as before. Thus,

$$\Delta'_{\mathcal{C}} < \sum_{k=1}^d \left(c^k (\mathcal{I}) + c^{k-1,k} (\mathcal{I}^{k,k+1}) + c^{k,k+1} (\mathcal{I}^{k+1,k+2}) \right) - \frac{2 \cdot |A_{\mathcal{C}}| - d \cdot n^2}{2 \cdot d^2} \left(\sum_{i=1}^d v_i \right)^2 \quad (20)$$

Again, if $\Delta'_{\mathcal{C}} \rightarrow 0$, when $N \rightarrow \infty$.

4.2.1 A numerical example

Our numerical example considers for the outside region the same parameters as in Section 4.1, $\ell = 1$, $L = 20$, $h = 80$ and $H = 99$. For the composite region, let $\mathcal{D} \propto [21, 39]$ the uniform distribution from which the component QAPs are generated. Thus, $2 \cdot \mathcal{D} \propto [42, 78]$ and Inequalities 13, and 14 hold. We have that $\Delta_{\mathcal{C}}$ is about the same as in Inequation 19, and $\Delta_{\mathcal{O}}$ is like in Inequalities 18 and 17. The identity permutation is thus the optimum solutions for these cQAPs.

In Table 1, we give an example of a composite QAP instance where two component QAP instances of size $n^1 = n^2 = 8$ overlap in 4 facilities. We consider $\alpha = 0.5$ to have a maximum variance of the elements in the outside region. To compare our generator with Drezner et al. [5]'s generator, we also consider a certain amount of 0's in \mathcal{L}_A , i.e. ζ , which also increases the variance of the elements in the outside region. In the example from Table 1, we take $\zeta = 0.5$ meaning that half of the lower values in $A_{\mathcal{O}}$ are 0's, and half of the lower values in $B_{\mathcal{O}}$ are 0's.

Note that even though the component QAPs and the outside region are generated with uniform random distributions, the values in the resulting cQAP are not uniform randomly distributed.

The composite distance matrix A															
0	45	42	47	21	20	23	22	1	0	0	1	24	21	22	24
45	0	43	41	22	23	24	20	93	0	94	0	22	20	21	22
42	43	0	46	21	24	21	24	7	97	5	98	22	22	20	22
47	41	46	0	22	22	21	21	98	2	6	94	20	22	24	22
21	22	21	22	0	46	42	46	24	22	23	20	96	94	91	9
20	23	24	22	46	0	42	41	20	22	21	22	90	98	1	0
23	24	21	21	42	42	0	43	22	23	21	21	93	1	0	0
22	20	24	21	46	41	43	0	20	23	21	20	97	0	90	8
1	93	7	98	24	20	22	20	0	45	48	45	22	20	23	22
0	0	97	2	22	22	23	23	45	0	42	45	21	22	23	20
0	94	5	6	23	21	21	21	48	42	0	40	22	21	21	21
1	0	98	94	20	22	21	20	45	45	40	0	21	20	24	20
24	22	22	20	96	90	93	97	22	21	22	21	0	42	47	45
21	20	22	22	94	98	1	0	20	22	21	20	42	0	46	42
22	21	20	24	91	1	0	90	23	23	21	24	47	46	0	42
24	22	22	22	9	0	0	8	22	20	21	20	45	42	42	0

The composite flow matrix B															
0	91	3	22	40	21	47	90	1	23	24	20	20	23	41	22
91	0	42	22	91	20	97	46	43	22	23	20	24	22	5	22
3	42	0	23	92	24	98	46	44	20	22	23	23	22	97	23
22	22	23	0	23	4	24	23	22	2	8	6	45	44	24	42
40	91	92	23	0	20	43	1	5	20	22	20	24	20	46	20
21	20	24	4	20	0	23	21	21	43	44	47	9	7	22	5
47	97	98	24	43	23	0	95	94	23	21	22	21	23	40	21
90	46	46	23	1	21	95	0	44	22	20	20	23	20	7	22
1	43	44	22	5	21	94	44	0	24	24	21	20	24	97	21
23	22	20	2	20	43	23	22	24	0	42	43	9	8	20	95
24	23	22	8	22	44	21	20	24	42	0	45	97	94	20	1
20	20	23	6	20	47	22	20	21	43	45	0	95	96	20	92
20	24	23	45	24	9	21	23	20	9	97	95	0	46	20	44
23	22	22	44	20	7	23	20	24	8	94	96	46	0	21	42
41	5	97	24	46	22	40	7	97	20	20	20	20	21	0	22
22	22	23	42	20	5	21	22	21	95	1	92	44	42	22	0

Table 1 A composite QAP with $N = 16$ facilities where the two component QAPs, $n^1 = n^2 = 8$, overlap in 4 facilities. In the composite region, $m_A = m_B = 21$ and $M_A = M_B = 39$. In the outside region, $\ell_A = \ell_B = 1$, $L_A = L_B = 19$, $h_A = h_B = 80$, $H_A = H_B = 99$, $\alpha = 0.5$ and $\zeta = 0.5$.

5 The difficulty of the QAP instances

Our goal is to design instances of QAPs that are difficult to solve with exact methods and in the same time have structure to be exploited with heuristics. There are specific measures to compute the hardness of QAP instances for each of these two techniques.

Empirical difficulty measures. *Dominance* is a measure of the amplitude of variance of the flow and distance matrices. High flow dominance means low epistasis [15]. We denote with d_B the dominance for the flow matrix, and with d_A the dominance for the distance matrix.

The *ruggedness coefficient* [1] is a normalization of the *autocorrelation coefficient* of the cost function when QAP is explored with local search. By definition, the auto-correlation coefficient is $\epsilon = \frac{2 \cdot \langle c^2 \rangle - \langle c \rangle^2}{\langle (c(\pi) - c(\pi'))^2 \rangle}$, where $\langle (c(\pi) - c(\pi'))^2 \rangle$ the average value of $(c(\pi) - c(\pi'))^2$ over all *neighbourhood* permutations, π and π' . The ruggedness coefficient is $\varphi = 100 - \frac{400}{n-2} \cdot (\epsilon - \frac{n}{4})$ with the assumption that $\epsilon > n/4$. A φ close to 0 indicates a flat landscape suited for local search. A large φ , close to 100, indicates a steep landscape with a lot of local optima suited for heuristics which restart the local search.

A complexity measure for a QAP emerges from the combination of the ruggedness φ and the *dominance* of a QAP (A, B) defined as a vector $(\min\{d_A, d_B\}, \max\{d_A, d_B\})$. A dominance vector (u, v) is deemed better than another dominance vector (u', v') if and

only if $u > u'$ and $v > v'$. Thus, a difficult QAP instance has both large dominance and ruggedness coefficients.

To investigate the asymptotic behaviour of QAPs, a fourth indicator calculates the difference between the optimum value (or a known feasible solution) and the value of the solution generated with the inverse permutation of that solution. Here, we consider the inverse of the optimum (or, if optimum is unknown, the best known solution) an approximation of the worst solution of an instance of QAP. Thus, for cQAPs with the optimum solution \mathcal{I} , we assume that the reverse of the identity permutation, \mathcal{I}^{-1} , is an approximation of the worst solution.

Exact methods. We compute the running times of the small size cQAP instances, i.e., $n \leq 32$, using the exact method branch and bound [13]. The lower bounds are used in exact algorithms to guarantee the optimum and, in the same time, to avoid the total enumeration of the solution space. The advantage of the lower bounds is that there are simple to compute even for large size QAP instances, i.e. $n \leq 250$. Their disadvantage is the gap between the true optimum cost and the approximation given by the lower bound increases with the number of facilities, making the approximation unreliable.

5.1 The empirical difficulty of cQAP instances

In this section, we study the impact of different parameters on the empirical difficulty of the cQAP instances. For each parameter setting, 50 cQAP instances were generated. The significance of the difference between results is computed with Wilcoxon non-parametric two sample tests. We investigate the difficulty of a cQAP instance given: i) the distribution of low and high values, ii) the number of facilities in the cQAP instance, iii) the number of facilities in the component QAPs, and iv) the number of facilities that overlap between component QAP.

The distribution in the outside regions. Let $n = 8$, $o = 4$ and $N = \{16, \dots, 152\}$ be the parameters from the example in Section 4.2. In Figure 1, we present on the top, the ruggedness φ and the dominance of the flow and distance matrices for $\alpha = 0.5$. In the same figure, on the bottom, the properties of cQAP instances with $\alpha = 1.0$ are shown. In each figure, the number of 0's in \mathcal{L} is varied, $\zeta = \{0, 0.5, 1.0\}$.

In Figure 1 a), the ruggedness is rapidly increasing with N towards its maximum value 1. If $N > 80$, then the ruggedness values are close to 100, $\varphi > 99$. In Figure 1 d), when $\alpha = 1.0$, the ruggedness is close to 1 for all N 's, qualifying these cQAP instances as more difficult than when $\alpha = 0.5$. But, in Figure 1 c) and f), the dominance of the flow matrices is lower for $\alpha = 1$ than for $\alpha = 0.5$. Note that for $\alpha = 0.5$ the dominance values in the two matrices are approximatively equal, and they slightly increase with N . For $\alpha = 1.0$, the distance dominance, d_A , is constantly increasing, and the flow dominance, d_B , is constantly decreasing. In conclusion, there is no clear classification for these cQAP instances into easy and hard instances.

Except for Figure 1 e), the number of 0s in the outside regions, ζ , does not have a large effect on the ruggedness φ or the dominance measures. For $\zeta = 1.0$, when N increases, the variance in the distance dominance d_A increases and the variance in d_B decreases. Since d_B does not depend on ζ , the values in the flow dominance are all generated from \mathcal{H} .

The difference between the identity and the inverse permutation. Figure 2 shows the cost function for the optimum solution and the difference between this optimum costs and the costs of the inverse permutation. In Figure 2 c), for $\alpha = 1.0$, there is a very small difference between the cost of the identity permutation and its inverse indicating a poor

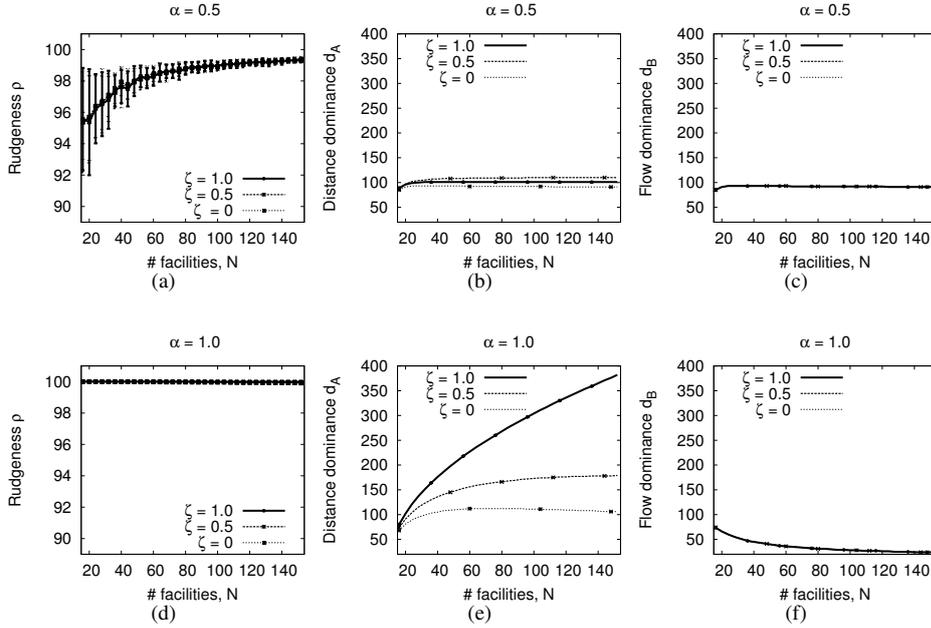


Fig. 1 The properties of cQAPs when $n = 8$, $o = \lfloor n/2 \rfloor$, $\zeta \in \{0, 0.5, 1.0\}$, and $N = \{16, \dots, 152\}$. On the top, $\alpha = 0.5$, on the bottom, $\alpha = 1.0$. We show: a) and d) the ruggedness, b) and e) and the dominance of the distance matrices, and c) and f) the dominance of the flow matrices.

asymptotic behaviour, $N \rightarrow \infty$. In Figure 2 a), when $\alpha = 0.5$, we have $\Delta(c(\mathcal{I}), c(\mathcal{I}^{-1})) > 3e + 6$, the maximum value being about 20 times larger than for the cQAP instances with $\alpha = 1.0$. Figures 2 b) and d) show that the optimum cost functions for both values $\alpha \in \{0.5, 1.0\}$ are about the same for the same N and $\zeta \in \{0, 1.0\}$.

Note the small standard deviations for all the measures except the ruggedness. We conclude that mixing high and low values in the outside regions of A and B does not increase the ruggedness of cQAP but increases the difference between local and global optimum solutions of the generated cQAPs. The percentage of 0s has a big impact on the optimum cost $c(\mathcal{I})$. Less 0s means smaller variance in A and B , and thus a small difference $\Delta(c(\mathcal{I}), c(\mathcal{I}^{-1}))$. A large amount of 0s increases the distance dominance but decreases the optimum cost. A small ζ decreases d_A but increases $c(\mathcal{I})$.

Based on these results, we recommend $\alpha = 0.5$ and $\zeta = 0.5$.

The overlap. The experiments in Figure 3 considers three values for the overlap $o \in \{0, \lfloor n/4 \rfloor, \lfloor n/2 \rfloor\}$. In Figure 3, for the top figures a) and b), we show φ and $c(\mathcal{I})$ for the non-overlapping cQAPs, $o = 0$, with large variance, $\alpha = 0.5$. For the same number of facilities, the ruggedness of this non-overlapping cQAPs is slightly, still significantly, smaller than the ruggedness of overlapping cQAPs. The optimum cost is smaller for non-overlapping cQAPs as compared with overlapping cQAPs because the values in the overlapping parts are larger than the values in the non-overlapping regions. In conclusion, we recommend $o = \lfloor n/2 \rfloor$.

The number of facilities. In Figure 3 bottom figures c) and d), the influence of the size of component QAPs is studied. When $\alpha = 1.0$ the outside region has a minimal effect on

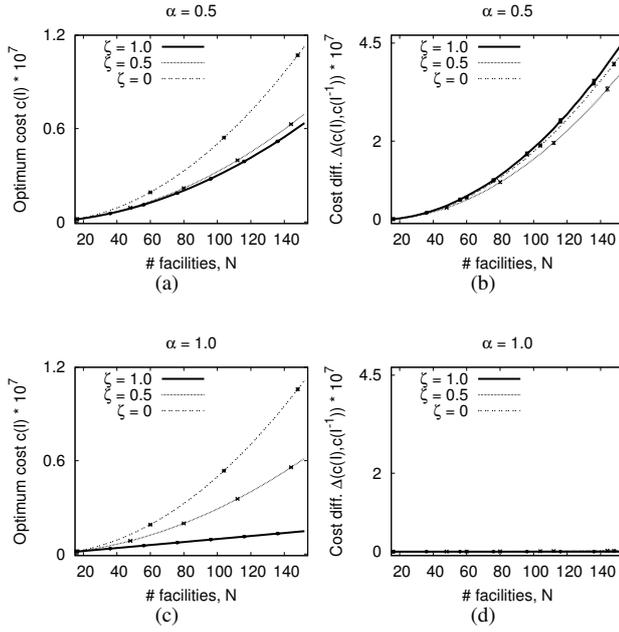


Fig. 2 a) and c) The difference between the optimum solution \mathcal{I} and its inverse, and b) and d) the optimum cost value $c(\mathcal{I})$.

the behaviour of the generated cQAPs and the differences in cost functions are very large. When $\alpha = 0.5$, the outside region has a dominating effect on the empirical difficulty of the cQAP instances that is about the same for different values of n . Of course, the values of n are rather small in this experiment. For larger n the effect of the composite region in the difficulty measures could be larger. We recommend n as large as possible, e.g. $n = 8$.

To conclude, a difficult cQAP has: i) a large number of facilities, ii) large size component QAPs, iii) a mixed distribution of low and high values in the outside regions. Then, the cQAP instance has: i) high ruggedness, ii) large variance in the flow and distance matrices, and iii) large difference between the optimal and the other solutions.

5.2 Comparing the empirical difficulty of QAP instances

For comparison purposes, in Table 2, we show the properties of thirteen symmetrical QAPs, 9 from QAPLIB and 2 QAP instances from [5] and 2 composite QAP instances, *cqap24* and *cqap56*. Accordingly to [1], *tai64c* and *esc32c* are the most interesting QAPs because their high values for ruggedness φ , and dominance d_A and d_B . But, the difference between the maximum and (an approximation of) the minimum cost is only 30 for *esc32c* and only 37 times larger than the maximum bound M_B for *tai64c*, indicating many solutions with the same value. The next two instances, *dre24* and *dre90*, have very high values for ruggedness φ and distance dominance, d_A , but low flow dominance, $d_B \leq 60$. The first four QAPs are very sparse to obtain large dominance values for the distance matrix. *tai25a* has high

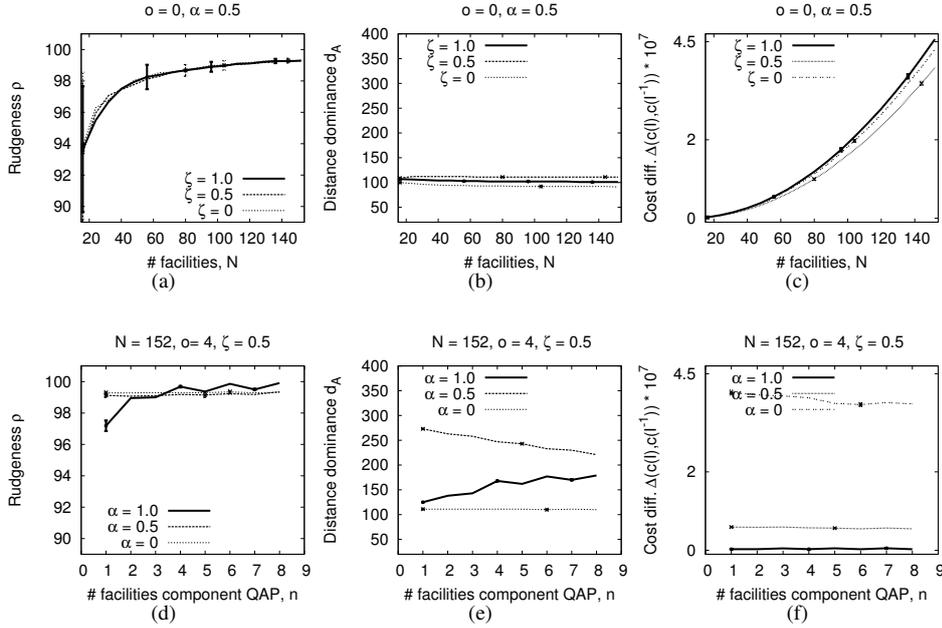


Fig. 3 The top figures: a) the ruggedness and b) the optimum cost for cQAPs with non-overlapping component QAPs, $o = 0$, $n = 8$, and $\alpha = 0.5$. The bottom figures: c) the distance dominance and d) the difference in cost functions for cQAPs with variable size component QAPs $n \in \{1, \dots, 8\}$, where $\alpha \in \{0.5, 1.0\}$, $\zeta = 0.5$, $o = \lfloor n/2 \rfloor$ and $N = 152$.

ruggedness values φ and low values for dominance $d_B = 62$ and $d_A = 64$. The last six QAPs have low values for ruggedness $\varphi < 90$.

As we have remarked in the previous section, an important factor for the ruggedness coefficient is the distribution of the high and low values in distance and flow matrices. For example, five of these QAP instances, *sko100a*, *tho30*, *wil100*, *kra30b*, and *ste36b*, have unbalanced large values for either distance or flow matrices. The exception is *tai64c* that has very large ruggedness value $\varphi > 99$, large values for both flow and distance dominance but also the highest bounds for the flow matrix, which, we think, causes the large flow dominance. QAPs with low ruggedness values $\varphi < 90$, have also low distance dominance values $d_A < 60$. The exception is *ste36b* that has both high distance and flow dominance because of its sparseness. Note that the distance between the minimum solution and its inverse is 0 for *sko100a*, *wil100* and *ste36b* indicating many solutions with the same value.

It is interesting to note that one of Drezner et al. [5]'s algorithms that generates QAPs can be thought of cQAPs with $o = 0$, $\alpha = 1.0$ and $\zeta = 1.0$ and with very large values for the flow matrix. We have shown that cQAPs with these parameters for α and ζ and o are the easiest parameters. This means that these Drezner-like QAPs are easier than the other cQAPs.

Let's compare *tai25a*, Taillard's uniformly generated QAP, with similar cQAP instances. The *tai25a* instance has the same bounds for A and B , since $m_A = m_B = 1$ and $M_A = M_B = 99$. For $N = 25$, *tai25a* has a larger ruggedness value than the corresponding cQAP instance *cqap24* but the dominance values for *tai25a* are smaller than for *cqap24*.

Properties of some QAP instances from literature										
name	φ	d_A	d_B	optim/feasib	Δ	Δ/sol	M_A	m_A	M_B	m_B
<i>tai64c</i>	100	482	128	1855928	3700112	1.6	1	0	1e5	3125
<i>esc32g</i>	100	850	69	6	30	5	3	0	4	0
<i>dre90</i>	99.7	554	48	1838	8524	4.6	10	0	10	1
<i>dre24</i>	99.5	303	60	396	1988	5.02	10	0	10	1
<i>tai25a</i>	97.9	64	62	1167256	278028	0.23	99	0	99	0
<i>cqap24</i>	96.3	102	93	304636	488834	1.6	99	0	99	0
<i>cqap56</i>	100	108	93	1189196	4342610	3.65	99	0	99	0
<i>sko100a</i>	89	51	107	152002	0	0	18	1	10	0
<i>tho30</i>	88	59	138	149936	64890	0.43	11	1	247	0
<i>wil100</i>	86	51	65	273038	0	0	18	1	9	0
<i>nug30</i>	84	53	112	6124	1900	0.31	9	1	10	0
<i>kra30b</i>	77	50	150	91420	42760	0.47	415	50	4	0
<i>ste36b</i>	67	101	400	15852	0	0	73	1	316	0

Table 2 Properties of QAP 12 instances sorted by ruggedness values, φ , in decreasing order, where M_A is the largest value in A , m_A is the lowest value in A , M_B is the highest value from B and m_B is the lowest value from B .

For a fair comparison between cQAPs, Drezner's and Taillard's QAPs, we limit the size of cQAPs used to $N = \{12, \dots, 56\}$. We conveniently denote these cQAPs with *cqapN*. The difficulty of *cqap24* and *cqap56* are given in Table 2. As a general observation, it is difficult to rank the different QAPs using only the above properties. In the next section, we compare the properties of several uniform randomly generated QAPs from Taillard, i.e. *tai25a*, and several sparse QAPs from Drezner et al., i.e. *dre24*, are also included.

5.3 Comparing QAP instances with heuristic and exact methods

In this section, we show that the generated composite QAP instances can be difficult for both the exact and heuristic methods, even more difficult than the random QAP instances from Taillard [20]. The difficulty of a QAP instance for the branch and bound algorithm measures how well the lower bounds techniques approximate the optimum solution. In Table 3, we compare our composite QAP instances and the random QAP instances, considered difficult for both exact and heuristics, using two lower bounds and two heuristic methods. There are 12 random instances with the number of facilities between 12 and 100 and there are 13 composite QAP instances with the number of facilities between 16 and 80. The gap is calculated as

$$gap = \frac{feasible_solution - best_solution}{feasible_solution} * 100\%$$

where the *feasible_solution* is the best solution returned by an algorithm and *best_solution* is the best solution known so far, or the optimum solution, or a lower bound. The used software is downloaded from the QAPLIB homepage (a Quadratic Assignment Problem Library) <http://www.seas.upenn.edu/qaplib/>.

The gap between the best (known) solution and the Gilmore-Lawyer lower bound [9] is given in third column, and the gap with the elimination lower bound [10] is given in the fourth column. Note that the gap of the composite QAP instances is higher than for the random instances indicating that these instances are difficult for branch and bound methods. The fifth and the sixth columns in Table 3 give the output for the Grasp [12] and the tabu search [20], respectively. Also for these two heuristics, the gap between the best known

Exact and heuristic methods for QAP instances					
name	Opt/ feas	Gap %			
		Gilm	Elim	GRASP	Tabu
<i>tai12a</i>	224416	12.70	13.94	2.04	8.47
<i>tai15a</i>	388214	15.64	16.28	0.20	14.87
<i>tai17a</i>	491812	16.08	16.86	1.94	10.19
<i>tai20a</i>	703482	17.46	18.15	1.54	6.97
<i>tai25a</i>	1167256	17.55	18.04	2.53	4.73
<i>tai30a</i>	1818146	17.24	17.48	1.81	2.82
<i>tai35a</i>	2422002	19.44	19.83	2.60	1.16
<i>tai40a</i>	3139370	20.59	20.86	2.63	0.38
<i>tai50a</i>	4938796	21.96	22.23	3.10	1.47
<i>tai60a</i>	7205962	22.91	23.15	3.23	1.90
<i>tai80a</i>	13499184	23.48	23.63	2.70	2.57
<i>tai100a</i>	21052466	24.83	24.96	2.73	2.81
<i>cqap16</i>	171444	17.86	42.65	5.50	5.79
<i>cqap20</i>	238754	16.76	55.01	13.28	4.58
<i>cqap24</i>	312308	18.65	63.22	31.20	20.29
<i>cqap28</i>	388832	18.98	69.36	41.26	23.89
<i>cqap32</i>	462050	19.76	77.40	48.51	33.32
<i>cqap36</i>	565198	19.60	78.43	53.07	38.39
<i>cqap40</i>	687634	18.83	78.00	55.27	46.52
<i>cqap44</i>	787718	19.67	80.93	58.91	53.24
<i>cqap48</i>	905046	19.83	83.64	61.53	54.55
<i>cqap52</i>	1046514	18.95	83.13	63.82	60.86
<i>cqap60</i>	1337714	17.89	85.50	66.35	63.54
<i>cqap72</i>	1809722	16.89	88.08	70.47	68.55
<i>cqap80</i>	2162198	16.46	89.62	72.34	71.22

Table 3 The performance of exact and heuristics methods on 25 QAP instances measured as the gap between the given solution (or the proposed lower bound) and the best known solution.

solution and the solution returned by the heuristics is more than 10-20 times higher for the composite QAP than for the random QAP instances. Furthermore, the gap increases constantly with the number of facilities for composite QAPs whereas for the random QAPs the gap is rather constant with N .

We conclude that composite QAP instances are meaningful test benchmark for exact and heuristics techniques. In the following section, we give more insights on why these QAP instances are difficult for the heuristic methods.

6 Landscape analysis

We analyse the landscape and properties of QAP instances using *multi-restart local search*. Intuitively, local search starts from an initial solution and iteratively generates new solutions using a neighbourhood strategy. Each step, a solution that improves over the existing best-so-far solution is chosen. The algorithm stops when there is no improvement possible. Best improvement explores *all* the individuals in the neighbourhood of a solution and selects the best solution that improves over the initial solution and all the other visited solutions. Because LS can be stuck in local optima, the *multi-restart LS* (MLS) algorithm restarts LS multiple times from one uniform randomly chosen initial solution.

QAPs are minimization permutation problems. A suitable neighbourhood operator for QAPs is the 2-exchange swapping operator that swaps the position of two different facili-

ties. This operator is attractive because of its linear time to compute the change in the cost function with the condition that the matrices A and B are symmetrical. The size of the neighbourhood increases quadratically with the number of facilities. For this experiment, we restart the local search $M = 10^5$ times.

The *basin of attraction* [8] of a local optimum X is the set of independent identically uniformly distributed restarting points for the local search for which the optimum is in X . As performance indicator, we count the number of times the minimum solution is found. Section 6.1 shows that this definition of basins of attraction is not informative even for small size QAPs because of the large number of basins of attraction in their landscape. In Section 6.2, we propose an alternative definition of basins of attraction that reduce the number of basins of attraction by counting as one the same optimum value. In Section 6.3, we use empirical measures to approximate the size of indecisive basins of attraction.

We compare the basins of attraction for three QAP generators: i) Taillard [21], ii) Drezner [5] and iii) the composite QAPs with the parameters recommended in the previous section. All these three methods generate QAPs of various sizes using uniform randomly generated numbers distributed with some, simple or rather elaborated, strategy in the flow and distance matrices. Taillard's QAPs have matrices with uniform randomly distributed elements. Drezner's QAPs represent grids with uniform randomly generated points with a rather small amount of connections in the grid. The component QAPs and the elements in the outside region of cQAP are also uniform randomly generated. Therefore, for all these QAPs, we assume a *random configuration for basins of attraction*.

6.1 Basins of attraction

To approximate the number of basins of attraction, B , we count the frequency of their occurrence, where β_j is the number of local optima that are reached exactly j times. Like in [8], we assume that the distribution of β_j follows a family of parametrized distributions Law_γ . The expected values $\beta_{j,\gamma} = \mathbb{E}_\gamma[\beta_j]$ are computed using

$$\beta_{j,\gamma} = L \cdot \frac{\Gamma(j+\gamma)}{j! \cdot \Gamma(j+\gamma)} \cdot \frac{\Gamma(L \cdot \gamma)}{\Gamma((L-1) \cdot \gamma)} \cdot \frac{\Gamma((L-1) \cdot \gamma + M - j)}{\Gamma(L \cdot \gamma + M)} \cdot \frac{M!}{(M-j)!}$$

where L is the number of basins of attraction visited so far, and M is the number of uniform randomly sampled solutions used to restart LS. To simplify this equation, we assume that γ is a positive integer. Then, $\Gamma(\gamma) = \gamma!$. After re-grouping the terms of the above equation, we have

$$\beta_{j,\gamma} = L \cdot \frac{\binom{M}{j} \cdot \binom{L}{1}}{\binom{L+M}{j+1}}$$

where $\binom{M}{j} = \frac{M!}{j! \cdot (M-j)!}$. The values $\beta_{j,\gamma}$ are used to calculate B , where $\beta_{j,\gamma}$ is the solution of the equation

$$\frac{\sum_{j=1}^{\infty} \beta_{j,\gamma}}{M} = \frac{1 - (1 + \frac{M}{\gamma \cdot B})^{-\gamma}}{\gamma}$$

Here, $\gamma \in \{1, 2, 3\}$ is chosen to minimize the difference between β_j and $\beta_{j,\gamma}$ in the χ^2 test, $\sum_j (\beta_j - \beta_{j,\gamma})^2 / \beta_{j,\gamma}$. The probability that at least one point of the searching space lies in a basin of attraction is

$$p_{B,M} \rightarrow_{L \rightarrow \infty} \exp -a^{-1}$$

where a is the solution for $M = [L^2 \cdot a]$.

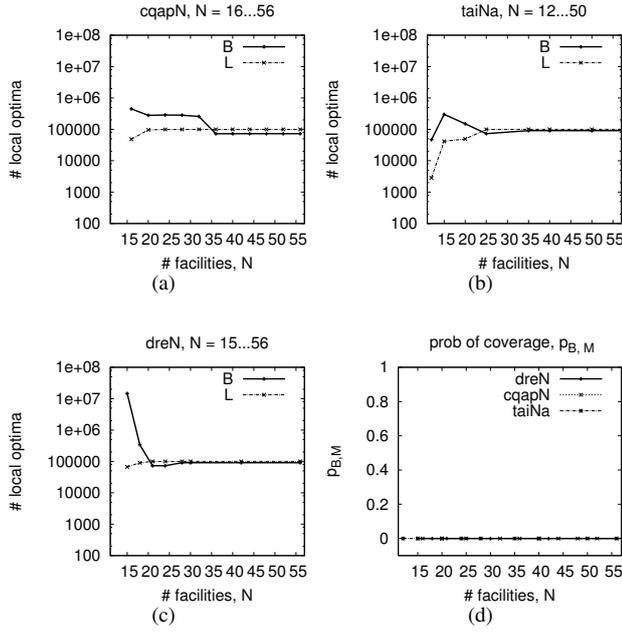


Fig. 4 The estimated number of basins of attraction B versus the visited basins of attraction L for $M \in (0, 10^5]$ on three types of QAPs: a) *cqapN*, where $N = \{12, \dots, 56\}$, b) *tailNa*, where $N = \{12, \dots, 50\}$ and c) *dreN*, where $N = \{15, \dots, 56\}$. d) The probabilities that all local optima of the search space were sampled in the standard basins, $p_{B,M}$.

Some experimental results. In Figure 4 on the top, we compare the number of *approximated basins of attraction*, B , versus the number of *visited basins of attraction*, L , for the three QAPs. We set $M = 10^5$ and $\gamma = \{1, 2, 3\}$. For larger $N > 20$, for all the QAP instances, the values of estimated, B , and visited, L , basins of attractions are about the same 10^5 . The covering probability $p_{B,M}$ for the three QAPs is basically 0.

Thus, even for a small number of facilities, QAP instances are difficult with lots of local optima, the restarting points almost always is located in an unique basin of attraction. We conclude that this definition of basin of attraction is not very informative for the prediction of the size and the number of local optima of QAPs.

6.2 Fitness indecisive basins of attraction

The number of basins of attraction is reduced by observing that LS cannot discriminate between solutions that have equal fitness values but different representations. These solutions, with the same value, can belong, or not, to the same plateau and, in general, they are difficult to explore and to store.

A fitness indecisive set of basin of attraction, *in short* indecisive basin of attraction, is the set of points for which the local optimum solutions obtained by restarting LS from these points have the same cost (or fitness value).

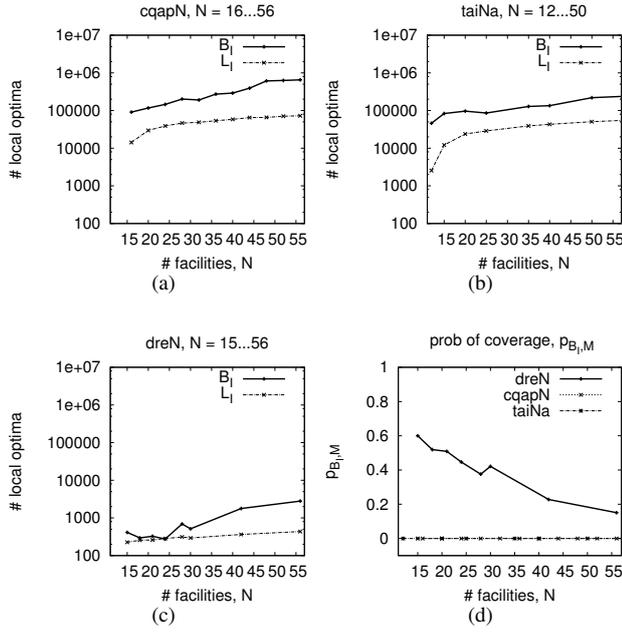


Fig. 5 The estimated, B_I , vs the visited, L_I , number of *indecisive* basins of attraction for $M = 10^5$ on three type of QAPs: a) *cqapN*, b) *taiNa* and c) *dreN*. d) The probabilities that all local optima of the search space were sampled in the indecisive basins, $p_{B_I, M}$.

The number and the size of indecisive basins of attraction are computed with the same algorithm as the regular basins of attraction, with the logical difference that now the equal local optimum values are counted as the same indecisive basin of attraction. We denote with B_I the number of estimated indecisive basins of attraction, with L_I the number of visited indecisive basins of attraction, and with $p_{B_I, M}$ the coverage probability of B_I .

Experimental results. On the bottom of Figure 5 the number of approximated indecisive basins of attraction, B_I , versus the number of already visited indecisive basins of attraction L for the three QAPs are compared. Note that even for the small sized *cqap16* and *tai15a*, there are lots of indecisive basins of attraction, $L_I \approx 10^4$, which is about $M/10$, the coverage probability is again about 0. We deduce that $M = 10^5$ LS restarts are not enough for landscape analysis of *cqapN* and *taiNa*.

Note that for *dre15*, the coverage probability $p_{B_I, M} = 0.6$ is large and B_I is close to L_I meaning that $M = 10^5$ is a reasonable number of restarts for MLS. Furthermore, *dreN* has non-zero coverage probabilities $p_{B_I, M} > 0.2$ even for medium size QAPs, $N \leq 56$, and the lowest B_I . Comparing these results with the results the standard basins of attractions, i.e. B , L and $p_{B, M}$, we deduce that Drezner's QAPs has lots of basins of attraction with the same value of its local optima, which can be grouped in much fewer indecisive basins of attraction.

In conclusion, for the same number of facilities N , *cqapN* and *taiNa* are more difficult than *dreN*.

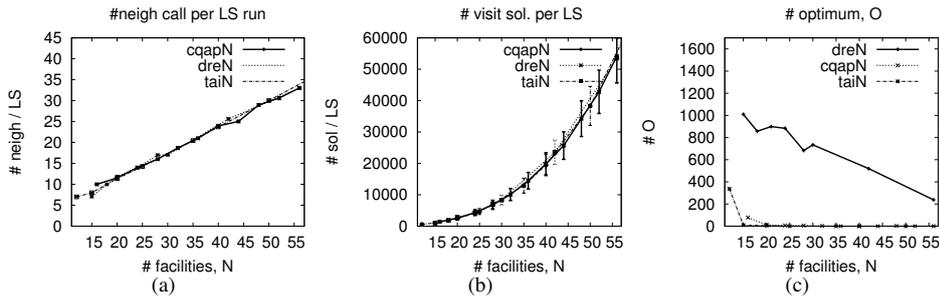


Fig. 6 a) The number of calls of the neighbourhood function per local search run. b) The number of solutions visited during a local search run. c) The number of times a solution with the best value in all runs is found. Each of the three tests is run on three type of QAPs: *cqapN*, *dreN*, and *taiN*.

6.3 Empirical analysis of basins of attraction

The number of calls of the neighbourhood function in a *LS*, $\#\mathcal{N}$, is an empirical approximation of the size of basins of attraction. The number of swaps generated in a *LS* run is the number of called neighbourhoods multiplied with the number of solutions in a neighbourhood, $\#\mathcal{N} \cdot \binom{N}{2}$. This measure gives an approximation of the time necessary for a *LS* to converge to a local optimum.

Figure 6 a) presents the average number of calls of neighbourhoods for the three types of QAPs with different N . Note that QAPs with the same number of facilities have about the same number of calls for the neighbourhood function. For the same number of facilities, the number of visited solutions during a *LS* run, Figure 6 b), is about the same for all QAPs. In Figure 6 a), the larger N is, the more times the neighbourhood function is called. This is reflected also in Figure 6 b) where the number of visited solutions during a *LS* run increases when N increases.

The percentage of times *LS*'s local optimum is equal to the best value found in all runs from Figure 6 c) is a good indicator of the performance of *MLS* algorithms. The identity permutation is a solution for *dreN* type of QAPs, whereas for Taillard's QAPs, *taiN* the (near)optimal solutions are known from *QAPLIB*'s home-page. From $M = 10^5$ *LS* runs, for $N > 12$, *MLS* almost never finds the solutions with best value. Unlike for *cqapN* and *taiN*, for all the instances *dreN*, where $N \leq 56$, *MLS* finds an optimal solution within $M = 10^5$ *LS* runs. As expected, the larger the number of facilities in *dreN*, the lower the number of times *MLS* finds the best-so-far solution.

We conclude that the tested cQAPs are more difficult than Drezner's QAP instances that have lots of local optima with the same value. In the following section, we show that cQAPs are more interesting for heuristics than Taillard's QAPs that are uniform randomly generated and have no correlations between the basin of attractions.

7 Conclusions

We propose a QAP instance generator that aggregates small size QAP instances whose exact solutions can be easily computed into a larger size QAP instance. We show that the resulting QAP instances, we call them composite QAP instances (cQAPs), have additively decomposable cost functions, which are known to be interesting for metaheuristics. The component

QAPs can overlap in the resulting cQAP. The elements from the composite QAP instance outside these component QAPs are generated using the rearrangement inequality.

We give some conditions for which a cQAP instance has the identity permutation as global optimal solution when all the component QAPs have the identity permutation as the optimal solution and the elements in the outside region have the identity permutation as the minimal solution. We assume that these optimal solutions are computed in feasible time by imposing mild conditions on: i) the lower and upper bounds of the generating distributions for the component and the outside regions, and ii) the relationship between the distribution of the two regions. We provide concrete numerical values for such bounding uniform distributions. To obtain cQAPs with different permutations as the global solution, we rename the facilities in a random way.

We analyse the properties of the cQAP instances for various input parameters using different analytical and experimental techniques. The empirical difficulty of QAPs for local search is measured using: i) the variations in the flow and distance matrices, ii) the auto-correlation in and in-between the two matrices, and iii) the difference between the cost of the optimum solution and its inverse permutation. Overall, the generated cQAP instances are classified as difficult for local search if they have: i) number of facilities, ii) large size component QAPs, and iii) an outside region with high variance. Further, it is difficult to rank the cQAP instances generated with fine tuned parameters. We compare our QAP instance generator with other QAP instance generators from literature. We show that the composite QAP instances are also difficult for exact methods.

To measure the difficulty of cQAPs for local search-based heuristics, we approximate the size and the number of basins of attraction with numerical and empirical techniques. We analytically show that a conventional definition of basins of attraction is too strict for practical use because of the large number of basins of even for the QAP instances with small number of facilities. Assuming that there are many plateaus and/or local optima of the same value, we propose fitness indecisive basins of attraction that count as one the local optima with the same value. The empirical approximations of basins of attraction usually use the indecisive basins of attraction. We show that the cQAP instances have a lot of indecisive basins of attraction, unlike other QAP instances from literature, classifying these instances as difficult.

To conclude, the generated cQAP instances are both difficult QAP instances for exact methods and heuristics and have known global optimum solution, making these instances attractive to optimize with meta-heuristics.

References

1. E. Angel and V. Zissimopoulos. On the hardness of the quadratic assignment problem with metaheuristics. *Journal of Heuristics*, 8(4):399–414, 2002.
2. R. E. Burkard, E. Çela, V. M. Demidenko, N. N. Metelski, and G. J. Woeginger. Perspectives of easy and hard cases of the quadratic assignment problems. Technical report, Institute of Mathematics Technical University Graz, Austria, 1997.
3. R. E. Burkard, S. E. Karisch, and F. Rendl. QAPLIB a quadratic assignment problem library. *Journal of Global Optimization*, 10:391–403, 1997.
4. E. Çela. *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishers, Dordrecht, 1998.
5. Z. Drezner, P. M. Hahn, and É. D. Taillard. Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. *Annals of Operations Research*, 139(1):65–94, 2005.

6. M. M. Drugan. Instance generator for the quadratic assignment problem with additively decomposable cost function. In *IEEE Congress on Evolutionary Computation (CEC'13)*, pages 2086–2093. IEEE, 2013.
7. G. Erdoğan and B. Tansel. Two classes of quadratic assignment problems that are solvable as linear assignment problems. *Discret. Optim.*, 8(3):446–451, August 2011.
8. J. Garnier and L. Kallel. Efficiency of local search with multiple local optima. *SIAM J. Discrete Math.*, 15(1):122–141, 2001.
9. P. C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal on Applied Mathematics*, 10:305–331, 1962.
10. S. W. Hadley, F. Rendl, and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research*, 17:727–739, 1992.
11. P. Krokhmal and P. Pardalos. Random assignment problems. *European Journal of Operational research*, 194(1):1–17, 2009.
12. Y. Li, P. M. Pardalos, and M. G. C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In P. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 237–261, 1994.
13. E. M. Loiola, N. M. M. de Abreu, P.O. Boaventura-Netto, P. Hahn, and T. Querido. An analytical survey for the quadratic assignment problem. *European J. of Operational Research*, 176(2):657–690, 2007.
14. A. Marzetta and A. Brütingger. A dynamic-programming bound for the quadratic assignment problem. In *COCOON'99*, volume LNCS 1627, pages 339–348. Springer-Verlag Berlin, 1999.
15. P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352, 2000.
16. G. Palubeckis. An algorithm for construction of test cases for the quadratic assignment problem. *Informatika, Lith. Acad. Sci.*, 11(3):281–296, 2000.
17. F. Puglierin, M. M. Drugan, and M. Wiering. Bandit-inspired memetic algorithms for solving quadratic assignment problems. In *IEEE Congress on Evolutionary Computation (CEC'13)*, pages 2078–2085, 2013.
18. P. Siarry and Z. Michalewicz, editors. *Advances in Metaheuristics for Hard Optimization*. Springer, 2008.
19. T. Stützle and S. Fernandes. New benchmark instances for the qap and the experimental analysis of algorithms. In *EvoCOP*, pages 199–209, 2004.
20. É. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.
21. É. D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location science*, 3:87–105, 1995.
22. A. Wayne. Inequalities and inversions of order. *Scripta Mathematica*, 12(2):164–169, 1946.
23. S.E. Wright. New linearizations of quadratic assignment problems. *Comput. Oper. Res.*, 39(11), 2012.

A Proof of Proposition 2

Let $\Delta(c(\mathcal{I}), c(\pi))$ be the difference in the cost functions of the identity permutation and of a permutation π , as before.

i) Swapping facilities in cycles. First, let's consider the permutations that can be expressed as d product of $(k, k + 1)$ -cycles. The difference in cost function is

$$\begin{aligned} \Delta(c(\mathcal{I}), c(\pi)) = & \sum_{k=1}^d \left(c^k(\mathcal{I}) - c^k(\pi_k) \right) + \sum_{i,j \wedge c_{ij}=0 \wedge c_{\pi_i \pi_j}=0} a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j}) \quad (21) \\ & + \sum_{k=1}^d \left(c^{k,k+1}(\mathcal{I}^{k+1,k+2}) - c^{k,k+1}(\pi^{k+1,k+2}) + c^{k-1,k}(\mathcal{I}^{k,k+1}) - c^{k-1,k}(\pi^{k,k+1}) \right) \end{aligned}$$

The first sum corresponds to swapping facilities within component QAPs, and the second term corresponds to swapping elements of the outside region. These two sums also encountered when computing the cost of swapping facilities in the non-overlapping cQAPs in Proposition 1, and thus they are negative. The third sum comes from swapping facilities between the elements of the same overlapping region. If Inequality 15 holds, then $\Delta(c(\mathcal{I}), c(\pi)) < 0$.

ii) *Swapping cycles.* When cycles are completely switched, in most of the cases, there are also elements that are switched between the composite and outside region.

Consider facilities exchanged between the overlapping regions of the k -th component QAP: $i, \pi_j \in \mathcal{S}^{k-1, k}$, and $j, \pi_i \in \mathcal{S}^{k, k+1}$. Then, $a_{ij} \in A_C$, and $b_{\pi_i \pi_j} \in B_C$. Let's consider a facility in the $k+1$ -th component QAP such that $t \in \mathcal{S}^{k+1, k+2}$ and $\pi_t \notin \mathcal{S}^{k-2, k-1}$. Then, $a_{jt} \in A_C$ but $b_{\pi_j \pi_t} \in B_C$. If $\pi_t \in \mathcal{S}^{k-2, k-1}$, then $a_{jt} \in A_C$ and $b_{\pi_j \pi_t} \in B_C$.

Consider facilities exchanged between the overlapping regions that do not belong to the same component QAP: $i, \pi_j \in \mathcal{S}^{k, k+1}$ and $j, \pi_i \in \mathcal{S}^{p, p+1}$, where $k \neq p-1$ and $k \neq p+1$. Then, $a_{ij} \in A_C$, and $b_{\pi_i \pi_j} \in B_C$. Let's consider a facility in the $k+1$ -th component QAP such that $t \in \mathcal{S}^{k+1, k+2}$ and $\pi_t \notin \mathcal{S}^{p+1, p+2}$. Then, $a_{jt} \in A_C$ but $b_{\pi_j \pi_t} \in B_C$. If $\pi_t \in \mathcal{S}^{p+1, p+2}$, then $a_{jt} \in A_C$ and $b_{\pi_j \pi_t} \in B_C$.

Thus, if elements from the k -th component are exchanged with elements from the p -th QAP component, then there are also elements which are switched between these two QAP components and the outside region, unless these QAP components are neighbours, this case being discussed previously. For example, consider that $\exists i, \pi_i, j, \pi_j, r, \pi_r$, such that $i, r \in \mathcal{S}^{k, k+1}$, $j \in \mathcal{S}^{k+1, k+2}$, $\pi_i, \pi_j \in \mathcal{S}^{p, p+1}$, and $\pi_r \in \mathcal{S}^{t, t+1}$. Then $a_{ij}, a_{ir}, a_{jr} \in A_C$, $b_{\pi_i \pi_j} \in B_C$, but $b_{\pi_i \pi_r}, b_{\pi_j \pi_r} \in B_C$.

Accordingly to rearrangement inequality and Inequality 13, swapping elements between the outside and the composite region is more negative than swapping elements between the non-overlapping and overlapping regions. We discuss this case more in the next paragraph.

The permutations of cycles for which there is no exchange between the composite and outside region shifts all the facilities with the same quantity. The difference in cost function is as in Equation 21. If Inequality 15 holds, then $\Delta(c(\mathcal{I}), c(\pi)) < 0$.

iii) *Exchanging facilities between cycles.* Unlike in the previous two cases, there are elements swapped between the composite and the outside region. Thus, we have the following additional sums:

1. Facilities are swapped between the overlapping and the non-overlapping region. Then,

$$(a_{ij}^k + a_{ij}^{k+1}) \cdot (b_{ij}^k + b_{ij}^{k+1} - b_{\pi_i \pi_j}^{p+1}) =$$

$$a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}^{p+1}) + a_{ij}^{k+1} \cdot (b_{ij}^{k+1} - b_{\pi_i \pi_j}^{p+1}) + a_{ij}^k \cdot b_{ij}^{k+1} + a_{ij}^{k+1} \cdot b_{ij}^k$$

and

$$a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}^p - b_{\pi_i \pi_j}^{p+1}) = a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}^p) - a_{ij}^k \cdot b_{\pi_i \pi_j}^{p+1}$$

2. Facilities are swapped between the overlapping and the outside region. Then,

$$(a_{ij}^k + a_{ij}^{k+1}) \cdot (b_{ij}^k + b_{ij}^{k+1} - b_{\pi_i \pi_j}) =$$

$$a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}) + a_{ij}^{k+1} \cdot (b_{ij}^{k+1} - b_{\pi_i \pi_j}) + a_{ij}^k \cdot b_{ij}^{k+1} + a_{ij}^{k+1} \cdot b_{ij}^k$$

and

$$a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j}^k - b_{\pi_i \pi_j}^{k+1}) = a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j}^k) - a_{ij} \cdot b_{\pi_i \pi_j}^{k+1}$$

The difference in the cost function is now

$$\Delta(c(\mathcal{I}), c(\pi)) = \sum_{k=1}^d \sum_{i, j, \pi_i, \pi_j \in \mathcal{S}^k} a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}^k) + \sum_{k=1}^d \sum_{\substack{r=1 \\ r \neq k}}^d \sum_{\substack{i, \pi_i \in \mathcal{S}^k \\ j, \pi_j \in \mathcal{S}^r}} a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j})$$

$$+ \sum_{k=1}^d \sum_{r=1}^d \sum_{\substack{p=1 \\ p \neq r}}^d \sum_{\substack{i, j \in \mathcal{S}^k \\ \pi_i \in \mathcal{S}^r \wedge \pi_j \in \mathcal{S}^p}} a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}) + \sum_{k=1}^d \sum_{r=1}^d \sum_{\substack{p=1 \\ p \neq r}}^d \sum_{\substack{i \in \mathcal{S}^r \wedge j \in \mathcal{S}^p \\ \pi_i, \pi_j \in \mathcal{S}^k}} a_{ij} \cdot (b_{ij} - b_{\pi_i \pi_j}^k)$$

$$+ \sum_{k=1}^d \sum_{\substack{p=1 \\ p \neq k}}^d \sum_{\substack{i, j \in \mathcal{S}^k \\ \pi_i, \pi_j \in \mathcal{S}^p}} a_{ij}^k \cdot (b_{ij}^k - b_{\pi_i \pi_j}^p) + \sum_{k=1}^d \sum_{\substack{p=1 \\ p \neq k}}^d \sum_{\substack{i, j \in \mathcal{L}^{k, k+1} \wedge \\ \pi_i, \pi_j \in \mathcal{S}^{p, p+1}}} \left(a_{ij}^k \cdot (b_{ij}^{k+1} - b_{\pi_i \pi_j}^{p+1}) + a_{ij}^{k+1} \cdot (b_{ij}^k - b_{\pi_i \pi_j}^p) \right)$$

$$+ \sum_{k=1}^d \sum_{\substack{i, j \in \mathcal{S}^{k, k+1} \\ \pi_i, \pi_j \in \mathcal{S}^{k, k+1}}} \left(a_{ij}^k \cdot b_{ij}^{k+1} + a_{ij}^{k+1} \cdot b_{ij}^k \right) - \sum_{k=1}^d \sum_{\substack{i \in \mathcal{S}^{k-1, k} \\ j \in \mathcal{S}^{k, k+1} \\ \pi_i, \pi_j \in \mathcal{S}^{p, p+1}}} a_{ij}^k \cdot b_{\pi_i \pi_j}^{p+1}$$

$$- \sum_{p=1}^d \sum_{\substack{r=1 \\ r \neq p}}^d \sum_{\substack{i \in \mathcal{S}^p \\ j \in \mathcal{S}^r \\ \pi_i, \pi_j \in \mathcal{S}^{p,p+1}}} a_{ij} \cdot b_{\pi_i \pi_j}^{p+1}$$

The first five terms of the above equation are common with the cost differences of non-overlapping cQAPs from Equation 12. The first term is part of $c^k(\mathcal{I}) - c^k(\pi)$, and in the second term elements in the outside region are swapped. The third and the fourth terms swap elements between the component QAPs and the output region. The fourth term swaps elements between component QAPs being a part of $c^k(\mathcal{I}) - c^k(\pi')$.

The last five terms are specific for the cQAPs with overlapping component QAPs. The sixth and the seventh terms exchange facilities between different overlapping regions. Let's now consider the eighth and ninth terms of the difference. The exchanges between the non-overlapping and overlapping regions are by design positive since the smallest elements from the non-overlapping region are swapped with the largest elements from the overlapping region. The last term of the difference is negative and it is given by exchanging elements between the outside and the overlapping region.

To summarize, if elements are switched from the overloading region to the outside region, then there are also elements switch back from the outside region to the non-overlapping region. Following the same reasoning as in proving Inequality 9 from Proposition 1, the bounds for these exchanges are $m \cdot (h + \ell) > 4 \cdot M^2 + L \cdot h$, which means that Inequality 14 holds. The difference between the two bounds is only the term $4 \cdot M^2$ instead of M^2 like in Inequality 9 because we consider the more restrictive case where the lowest element from \mathcal{H} is exchanged with the lowest element from the composite region or the lowest element from \mathcal{L} , in which case $h \cdot m/2 < a_{ij} \cdot b_{\pi_i \pi_j}^{p+1} < M^2$.

This concludes our proof. \square