A New Greedy Algorithm for the Quadratic Assignment Problem

Theodoros P. Gevezes and Leonidas S. Pitsoulis
Department of Mathematical, Computational and Physical Sciences
Aristotle University of Thessaloniki
541 24 Thessaloniki, Greece

September 30, 2011

Abstract

The classical greedy algorithm for discrete optimization problems where the optimal solution is a maximal independent subset of a finite ground set of weighted elements, can be defined in two ways which are dual to each other. The Greedy-In where a solution is constructed from the empty set by adding the next best element, one at a time, until we reach infeasibility, and the Greedy-Out where starting from the ground set we delete the next worst element, one at a time, until feasibility is reached. It is known that while the former provides an approximation ratio for maximization problems, its worst case performance is not bounded for minimization problems, and vice-versa for the later. However the Greedy-Out algorithm requires an oracle for checking the existence of a maximal independent subset which for most discrete optimization problems is a difficult task. In this work we present a Greedy-Out algorithm for the Quadratic Assignment Problem by providing a combinatorial characterization of its solutions.

Keywords: approximation algorithms, quadratic assignment problem, greedy algorithm

1 Introduction

The greedy algorithm is one of the oldest and simplest algorithms in optimization. While the term *greedy* was coined by Edmonds in [4] for discrete optimization problems, the algorithm is also encountered with names such as *steepest descent* in continuous optimization. There have been several attempts in the past with varying degrees of success, to fully characterize the family of problems for which the greedy algorithm provides the optimal solution. Early results independently by Rado [22], Gale [7] and Edmonds [4], established that discrete optimization problems with linear objective functions for which the greedy algorithm is optimal, are matroids. Korte and Lovász in [14] considered introducing order into the independence systems and defined greedoids which generalize matroids. They have provided necessary and sufficient conditions upon which the greedy algorithm produces the optimal solution on greedoids. Faigle [5] characterized those independence systems on partially ordered sets, where the greedy algorithm is optimal. Probably the most complete characterization of the problem structure where the greedy is optimal for linear objective functions is that of matroid embeddings, introduced by Helman et al. in [10]. Furthermore, an extension of the greedy algorithm to examine more than one element while constructing a solution is given by Hausmann and Korte in [8], while an analysis of its worst case performance for independence systems which are not necessarily matroids is given in [9].

The Quadratic Assignment Problem (QAP) is one of those rare combinatorial optimization problems, where the large amount of research devoted to it did not result in substantial improvements on solving

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heraclitus II. Investing in knowledge society through the European Social Fund.

larger instances to optimality. The computational complexity of solving and approximating the QAP were early settled by Sahni and Gonzalez in [23]. The authors showed through a reduction from the Hamiltonian cycle problem that the QAP is NP-hard and that even finding an ϵ-approximation solution for the QAP is a hard problem, in the sense that the existence of a polynomial ϵ-approximation algorithm implies P=NP. Further inapproximability results were derived by Queyranne [21] for special cases of the problem. Moreover even finding a locally optimal solution of the QAP can be prohibitively hard in the computational sense. In Pardalos, Rendl and Wolkowicz [20] it is shown that the QAP is PLS-Complete for two well known neighborhood structures, namely the Kernighan-Lin [12] and the pair exchange types of neighborhoods. These results suggest that in general a local search algorithm will require computational time which is exponential with respect to the problem size in the worst case. A state of the art survey for the QAP is given by Burkard, Çela, Pardalos and Pitsoulis [1], while a comprehensive book on every aspect of the problem by Çela [3].

In this paper we present a new greedy algorithm for the QAP with a worst case approximation ratio bounded from above by a constant factor. The main component of the algorithm is a procedure that examines if a set of elements contains a solution, which utilizes a new combinatorial characterization of the solutions of the QAP. The paper is organized as follows. In section 2 we present the formulation of the QAP employed in this work. In section 3 we present all the necessary theoretical preliminaries and previous results. The main section of this paper is 4 where necessary and sufficient conditions for a $\{0,1\}$ matrix to be a QAP solution are given, and the problem of deciding whether some $\{0,1\}$ matrix contains a QAP solution is shown to be equivalent to a constrained linear assignment problem. The proposed algorithm and its computational complexity are presented in 5, while in section 6 we provide computational results comparing our greedy algorithm with the classic greedy.

2 Problem Formulation

In this work we use the quadratic assignment problem formulation as given by Lawler in [19]. Given a four-dimensional array of cost coefficients $C = (c_{ijkl})$ the problem can be stated as

$$\min_{p\in\Pi_{I_n}} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ijp(i)p(j)} \tag{1}$$

where Π_{I_n} is the set of all permutations of some finite set $I_n = \{1, 2, \ldots, n\}$. The QAP is often encountered by the Koopmans and Beckmann [13] formulation, where are given three $n \times n$ real matrices $F = (f_{ij})$, $D = (d_{kl})$ and $B = (b_{ik})$, and the problem can be stated as

$$\min_{p\in\Pi_{I_n}} \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d_{p(i)p(j)} + \sum_{i=1}^{n} b_{ip(i)}. \tag{2}$$

Clearly any instance of a Koopmans-Beckmann QAP can be formulated as an instance of a Lawler QAP by setting

$$c_{ijkl} = \begin{cases} f_{ij}d_{kl}, & \text{for} \quad i \neq j, k \neq l, \\ f_{ii}d_{kk} + b_{ik}, & \text{otherwise.} \end{cases}$$

In this paper we will employ an alternative formulation for the QAP as it is stated in (1) using Kronecker products. Let the *inner product* of two matrices $A, B \in \mathbb{R}^{n\times n}$ be defined by

$$\langle A, B\rangle := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ij}.$$

Also for $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ let their *Kronecker product* be the matrix of order $mp \times nq$

$$A \otimes B := \begin{bmatrix} a_{11}B & a_{12}B & \ldots & a_{1n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \ldots & a_{mn}B \end{bmatrix} \tag{3}$$

Associate with any permutation $p \in \Pi_{I_n}$ an $n \times n$ *permutation matrix* matrix $X = (x_{ij})$ such that

$$x_{ij} = \begin{cases} 1 & \text{if } p(i) = j, \\ 0 & \text{otherwise.} \end{cases}$$

We will denote the index set $\{1, \ldots, n\}$ by I_n and the set of all permutation matrices associated with some set I_n by $\mathbb{X}_n$. Construct a $n^2 \times n^2$ matrix C from the n^4 costs c_{ijkl}, such that the $(ijkl)^{th}$ element corresponds to the $((i-1)n+j, (k-1)n+l)^{th}$ element of C. It is easy to see now that the QAP in (1) is equivalent to a linear assignment problem of dimension n^2, in which C is the cost matrix, with the additional constraint that the $n^2 \times n^2$ permutation matrix which defines a feasible solution must be the Kronecker product of a permutation matrix of order n by itself. In other words we have the following equivalent formulation of the QAP

$$\begin{aligned} & \min \langle C, Y \rangle \\ \text{s.t.} \quad & Y = X \otimes X \\ & X \in \mathbb{X}_n \end{aligned} \tag{4}$$

3 Performance of the Greedy Algorithm

In this section we describe the worst case approximation performance of the greedy algorithm for discrete optimization problems. The theoretical framework on which the analysis is based, is that of independence systems. Let us first state some necessary definitions (see chapter 13 in [15]).

Definition 1. *Given a finite set E and some family of subsets $\mathcal{I} \subset 2^E$, the set system $(E, \mathcal{I})$ is called an* **independence system** *if the following axioms are true:*

i) $\emptyset \in \mathcal{I}$.

ii) If $X \in \mathcal{I}$ *and* $Y \subset X \;\Rightarrow Y \in \mathcal{I}$.

We will refer to the set E as the *ground set*, while the elements of $\mathcal{I}$ are called *independent* and those not in $\mathcal{I}$ are called *dependent*. Maximal independent sets are called *bases* and minimal dependent sets are called *circuits*. Given some independence system $(E, \mathcal{I})$, the *rank* of some $I \subseteq E$ is defined as

$$r(I) := \max\{|X| : X \subseteq I, X \in \mathcal{I}\},$$

while the *lower rank* as

$$lr(I) := \min\{|X| : X \subseteq I, X \in \mathcal{I}, X \cup \{s\} \notin \mathcal{I} \;\; \forall s \in I \backslash X\}.$$

We can interpret the values of $r(E)$ and $lr(E)$ as the cardinality of the maximum cardinality basis and the minimum cardinality basis of the independence system respectively. For an independence system $(E, \mathcal{I})$ define the following family of subsets of E

$$\mathcal{I}^* := \{X \subseteq E : \text{there is a basis } B \text{ of } (E, \mathcal{I}) \text{ such that } X \cap B = \emptyset\}.$$

It is easy to show that $(E, \mathcal{I}^*)$ is also an independence system, which is called the *dual* of $(E, \mathcal{I})$. Given now an independence system $(E, \mathcal{I})$ and some weight function $c : E \to \mathbb{R}_+$, we can state the following two discrete optimization problems with linear objective functions:

MAXIMIZATION PROBLEM FOR $(E, \mathcal{I})$: Given $(E, \mathcal{I}, c)$ find some $X \in \mathcal{I}$ such that $c(X) := \sum_{x \in X} c(x)$ is maximum.

Minimization Problem for $(E, \mathcal{I})$:
Given $(E, \mathcal{I}, c)$ find some X with $r(X) = r(E)$ such that $c(X) := \sum_{x \in X} c(x)$ is minimum.

Most combinatorial optimization problems can be stated as optimization problems on independence systems. In the case of the QAP as it is formulated in (4), the ground set E is all n^4 costs c_{ijkl}, while $\mathcal{I}$ contains any subset of E such that the corresponding indices $(ijkl)$ form a submatrix of the Kronecker product of a permutation matrix of order n by itself. Consider the following two greedy algorithms for the maximization problem on independence systems (see Figures 1 and 2).

Algorithm: Greedy-In

`Input :` independence system $(E, \mathcal{I})$, cost function $c : E \to \mathbb{R}$
`Output:` set $X \in \mathcal{I}$

1. Sort E such that $c(e_1) \geq c(e_2) \geq \ldots \geq c(e_{|E|})$
2. $X := \emptyset$
3. **for** $i = 1, \ldots, |E|$ **do**
4.   **if** $X \cup \{e_i\} \in \mathcal{I}$ **then**
5. $X := X \cup \{e_i\}$
6.   **end if**
7. **end for**
8. **return** X

Figure 1: The Greedy-In algorithm for maximization problems

Algorithm: Greedy-Out

`Input :` independence system $(E, \mathcal{I})$, cost function $c : E \to \mathbb{R}$
`Output:` set $X \in \mathcal{I}$

1. Sort E such that $c(e_1) \leq c(e_2) \leq \ldots \leq c(e_{|E|})$
2. $X := E$
3. **for** $i = 1, \ldots, |E|$ **do**
4.   **if** $X - \{e_i\}$ contains a basis **then**
5. $X := X - \{e_i\}$
6.   **end if**
7. **end for**
8. **return** X

Figure 2: The Greedy-Out algorithm for maximization problems

We can see that in the Greedy-In we insert the best elements in a partial solution until it becomes infeasible, while in the Greedy-Out we extract the worst elements from the whole ground set until we reach a feasible solution. For the minimization problem on independence systems we could also state the above greedy algorithms by reversing the order upon which the ground set E is sorted. It is also apparent that the computationally intensive tasks are to check whether a feasible solution remains feasible after the addition of an element, and whether a given set of elements contains a feasible solution. We have the following two important results regarding the performance of the greedy algorithm.

Theorem 1 ([9]). *Let $(E,\mathcal{I})$ be an independence system. For $c : E \to \mathbb{R}_+$ let $G(E,\mathcal{I},c)$ denote a solution found by* Greedy-In *for the* Maximization Problem, *while $OPT(E,\mathcal{I},c)$ is the optimum solution. Then*

$$q(E,\mathcal{I}) := \min_{X\subseteq E} \frac{lr(X)}{r(X)} \leq \frac{G(E,\mathcal{I},c)}{OPT(E,\mathcal{I},c)} \leq 1$$

for all $c : E \to \mathbb{R}_+$. There is a cost function where the upper bound is attained.

Noting that the Greedy-In for some independence system $(E,\mathcal{I})$ corresponds to Greedy-Out for the dual $(E,\mathcal{I}^*)$, we can derive the dual version of Theorem 1 as follows.

Theorem 2 ([16]). *Let $(E,\mathcal{I})$ be an independence system. For $c : E \to \mathbb{R}_+$ let $G(E,\mathcal{I},c)$ denote a solution found by* Greedy-Out *for the* Minimization Problem, *while $OPT(E,\mathcal{I},c)$ is the optimum solution. Then*

$$1 \leq \frac{G(E,\mathcal{I},c)}{OPT(E,\mathcal{I},c)} \leq \max_{X\subseteq E} \frac{|X| - lr^*(X)}{|X| - r^*(X)} := \rho(E,\mathcal{I})$$

for all $c : E \to \mathbb{R}_+$, where $r^*(X)$ and $lr^*(X)$ are the rank and the lower rank respectively as defined on the dual (E, I^*). There is a cost function where the upper bound is attained.

Based on the above theorems, we can see that the values of $q(E,\mathcal{I})$ and $\rho(E,\mathcal{I})$ provide theoretical bounds regarding the approximation performance of a greedy algorithm for a maximization and a minimization problem respectively. Note that the Greedy-In procedure does not provide an approximation bound for minimization problems and the Greedy-Out for maximization problems.

All the greedy approaches that have appeared in the literature for the QAP are essentially Greedy-In approaches, therefore with no lower bound on their worst case performance. This is mainly because step 4 of the Greedy-Out algorithm in Figure 2 requires an oracle, which must decide whether or not a subset of assignments $(ijkl)$ contains a QAP solution matrix, and this is a non-trivial task. We present for the first time a solution to this problem and thereby an implementation of the Greedy-Out algorithm for the QAP.

4 Structural Properties of $X \otimes X$

In this section we present some structural properties of a binary matrix Y that is the Kronecker product of a permutation matrix by itself. Necessary and sufficient conditions are derived, which provide the means of constructing a Greedy-Out algorithm for the QAP. Throughout this section we assume that we have a matrix $Y \in \{0,1\}^{n^2\times n^2}$, while its $((i-1)n+j,(k-1)n+l)^{th}$ element is denoted as y_{ijkl} in order to be in direct analogy with the n^4 costs c_{ijkl} that define a QAP instance.

Given that $Y = X \otimes X$ for some $X \in \mathbb{X}_n$, the first observation about Y that follows directly from the definition of the Kronecker product, is that it has a *symmetry* in the sense that

$$y_{ijkl} = x_{ik}x_{jl} = x_{jl}x_{ik} = y_{jilk}, \tag{5}$$

for $i \neq j$ and $k \neq l$, with elements

$$y_{iikk} = x_{ik}x_{ik} \tag{6}$$

which correspond to the linear terms in (2). To facilitate the discussion that follows, we partition $Y \in \{0,1\}^{n^2\times n^2}$ into two types of $n\times n$ submatrices. Let $X^{(ik)} = (x^{(ik)}_{jl})$ be the submatrix of Y where $x^{(ik)}_{jl} := y_{ijkl},\ \ j,l \in I_n$ while $Z^{(jl)} = (z^{(jl)}_{ik})$ be the submatrix of Y where $z^{(jl)}_{ik} := y_{ijkl},\ \ i,k \in I_n$. Observe that we have the following relationship between these elements

$$x^{(ik)}_{jl} = z^{(jl)}_{ik}, \text{ for all } i,j,k,l \in I_n. \tag{7}$$

Furthermore, for some matrix S let $|S|$ denote the number of its nonzero elements.

In the following theorem we provide an alternative characterization of the Kronecker product of a permutation matrices by itself by stating necessary and sufficient conditions on the structure of a $\{0,1\}$ matrix.

Theorem 3. *Let $Y \in \{0,1\}^{n^2 \times n^2}$. Then $Y = X \otimes X$ for some $X \in \mathbb{X}_n$ if and only if the following conditions are true*

i) $|Y| = n^2$

ii) if $y_{iikk} = 0$ *then* $X^{(ik)} = \mathbf{0}$ *and* $Z^{(ik)} = \mathbf{0}$

iii) if $y_{ijks} = 0$ *or* $y_{iskl} = 0$ *for all* $s \in I_n$ *then* $X^{(ik)} = \mathbf{0}$ *and* $Z^{(ik)} = \mathbf{0}$

Proof. Necessity is shown first. Consider some $Y = X \otimes X$ for some $X \in \mathbb{X}_n$. Condition $i)$ follows directly from the definition of the Kronecker product. For condition $ii)$ if $y_{iikk} = 0$, since $y_{iikk} = x_{ik}x_{ik}$ we will have that $x_{ik} = 0$. Since

$$y_{iskt} = x_{ik}x_{st} = y_{sitk}, \ \ \forall s, t \in I_n,$$

we have that $y_{sitk} = y_{iskt} = 0$ for all $s, t \in I_n$ or equivalently $X^{(ik)} = Z^{(ik)} = \mathbf{0}$. Finally for condition $iii)$, if $y_{ijks} = 0$ for all $s \in I_n$ then $x_{ik}x_{js} = 0$ for all $s \in I_n$. But since X is a permutation matrix there exists some $s \in I_n$ such that $x_{js} = 1$ which implies that $x_{ik} = 0$. Therefore, as previously, $X^{(ik)} = Z^{(ik)} = \mathbf{0}$. Similarly if $y_{iskl} = 0$ for all $s \in I_n$ then $X^{(ik)} = Z^{(ik)} = \mathbf{0}$.

Sufficiency is established now. Let m be the number of nonzero $X^{(ik)}$ submatrices of Y, that is $m = |M|$ for $M := \{X^{(ik)} : X^{(ik)} \neq \mathbf{0}, i, k \in I_n\}$. If $m > n$ since $|Y| = n^2$ there exists some nonzero $X^{(ik)}$ such that $|X^{(ik)}| < n$. Therefore, it must contain some zero row, say j, that is $x^{(ik)}_{js} = y_{ijks} = 0$ for all $s \in I_n$. By condition $iii)$ then $X^{(ik)} = \mathbf{0}$ which is a contradiction. If $m < n$ there exists some i such that $X^{(ij)} = \mathbf{0}$ for all $j \in I_n$, therefore $y_{iijj} = x^{(ij)}_{ij} = 0$ for all $j \in I_n$. By condition $ii)$ this implies that $Z^{(ij)} = \mathbf{0}$ or equivalently $z^{(ij)}_{st} = y_{sitj} = 0$ for all $j, s, t \in I_n$ and because of (7), $x^{(st)}_{ij} = y_{sitj} = 0$ for $j, s, t \in I_n$, which by condition $iii)$ implies that $X^{(st)} = \mathbf{0}$ for all $s, t \in I_n$, that is $Y = \mathbf{0}$ which is a contradiction. We can therefore conclude that $m = n$. So Y has n nonzero submatrices $X^{(ik)}$ while as it has already been demonstrated there is no i (or k) such that $X^{(ik)} = \mathbf{0}$ for all $k \in I_n$ (or $i \in I_n$). This implies that exists a permutation matrix $\bar{X} = (\bar{x}_{ik}) \in \mathbb{X}_n$ such that

$$\bar{x}_{ik} = \begin{cases} 1 & \text{if } X^{(ik)} \neq \mathbf{0}, \\ 0 & \text{if } X^{(ik)} = \mathbf{0}. \end{cases} \tag{8}$$

We will now show that each of these n nonzero submatrices $X^{(ik)}$ of Y is a permutation matrix of order n. Consider any nonzero submatrix $X^{(ik)}$. If there exists row j (column l) such that

$$x^{(ik)}_{jl} = y_{ijkl} = 0, \text{ for all } l \in I_n \ (j \in I_n),$$

i.e. it is a zero row (column), then by condition $iii)$ we have that $X^{(ik)} = \mathbf{0}$ which is a contradiction. Therefore each nonzero submatrix $X^{(ik)}$ of Y does not contain a zero row or column, which implies that $|X^{(ik)}| \geq n$. Combined with the fact that $|Y| = n^2$ we have that each of these n nonzero submatrices is a permutation matrix. Given permutation matrix $\bar{X}$ from (8), consider any $i, j, k, l \in I_n$ such that $\bar{x}_{ik} = \bar{x}_{jl} = 1$ which implies that $X^{(ik)} \neq \mathbf{0} \neq X^{(jl)}$. We will show that the only nonzero element in row j and in column l of submatrix $X^{(ik)}$ is the $(jl)^{th}$ element. For any $s \in I_n$ where $s \neq l$ we have that $X^{(js)} = \mathbf{0}$ which implies that

$$x^{(js)}_{js} = y_{jjss} = 0,$$

so by condition $ii)$ we have that $Z^{(js)} = \mathbf{0}$ or equivalently $z^{(js)}_{ik} = 0$ for any $s \neq l$. Since, from (7), $z^{(js)}_{ik} = x^{(ik)}_{js}$ we have that $x^{(ik)}_{jl} = 1$. Similarly for column l of matrix $X^{(ik)}$. This implies that every nonzero submatrix $X^{(ik)}$ of Y is equal to $\bar{X}$. Considering this fact, it follows that

$$x^{(ik)}_{jl} = y_{ijkl} = 1.$$

Therefore $Y = \bar{X} \otimes \bar{X}$. ☐

Corollary 1. *Let $Y = X \otimes X$ for some permutation matrix $X \in \mathbb{X}_n$. The element $y_{ijkl} = 0$ if $i = j$ and $k \neq l$ or $i \neq j$ and $k = l$ for all $i, j, k, l \in I_n$.*

Proof. The element y_{iikl}, where $k \neq l$, belongs to the $X^{(ik)}$ submatrix of Y. If $y_{iikk} = 0$ then by condition $ii)$ of Theorem 3 we have $X^{(ik)} = 0$. If $y_{iikk} = 1$, then since $y_{iikk} = x_{ik}x_{ik}$ we have that $x_{ik} = 1$ which implies that $X^{(ik)} = X$. So any element in row i and in column $l \neq k$ of $X^{(ik)}$ submatrix is zero. Similarly for the element y_{ijkk}, where $i \neq j$. □

Definition 2. *For $X, Y \in \{0,1\}^{n\times n}$ we say that X* **contains** *Y and write $X > Y$ if and only if*

$$x_{ij} \geq y_{ij}, \text{ for all } i, j \in I_n.$$

Definition 3. *Any matrix $Y \in \{0,1\}^{n^2\times n^2}$ satisfying the conditions $i) - iii)$ of Theorem 3 is called a* **QAP matrix**. *A matrix $Z \in \{0,1\}^{n^2\times n^2}$ is called* **QAP-feasible matrix** *if and only if $Z > Y$ for some QAP matrix Y.*

Let us define the following decision problem, which can be thought of as a linear assignment problem with an additional constraint:

> Pairs Linear Assignment Problem (PLAP):
> Given some $Z \in \{0,1\}^{n\times n}$ and $\mathcal{I} \subseteq \{\{(i,k),(j,l)\} : (i,k),(j,l) \in I_n \times I_n\}$ find some permutation matrix $X \in \mathbb{X}_n$ such that:
>
> i) $Z > X$
>
> ii) $x_{ik}x_{jl} = 0$ for all $\{(i,k),(j,l)\} \in \mathcal{I}$
>
> or decide that no such matrix exists.

Denote an instance of the Pairs Linear Assignment Problem by $P(Z,\mathcal{I})$. Note that this problem is equivalent to a matching problem on a complete bipartite graph $G(V_1 \cup V_2, E)$, with the additional constraint that a subset of $E(G) \times E(G)$ cannot appear in the solution. The following theorem can now be stated.

Theorem 4. *For some $Y \in \{0,1\}^{n^2\times n^2}$ let $\mathcal{I}(Y) := \{\{(i,k),(j,l)\} : y_{ijkl} = 0\}$. Then $P(X^{(ik)},\mathcal{I}(Y))$ is feasible for any $i \in I_n$ and some $k \in I_n$ if and only if $P(X^{(ik)},\mathcal{I}(Y))$ is feasible for some $i, k \in I_n$.*

Proof. Necessity is trivial. For sufficiency, consider some $X \in \mathbb{X}_n$ such that $X^{(ik)} > X$ and $x_{ik}x_{jl} = 0$ for all pairs $\{(i,k),(j,l)\} \in \mathcal{I}(Y)$. We will show that

$$X^{(st)} > X, \ \ \forall s, t \in I_n \text{ such that } x_{st} = 1.$$

Consider any $s \neq i$ and take $X^{(st)}$ for $x_{st} = 1$. For $x_{ab} = 1$ if $x^{(st)}_{ab} = 0$ it implies that $y_{satb} = 0$ which in turn implies that $x_{st}x_{ab} = 0$ by the definition of $\mathcal{I}(Y)$. Since $x_{st}x_{ab} = 1$ though, we have to have $x_{st} = 0$ which is a contradiction to our original hypothesis. □

In what follows we provide a characterization of QAP-feasible matrices.

Theorem 5. *For $Y \in \{0,1\}^{n^2\times n^2}$ there are $i, k \in I_n$ such that $P(X^{(ik)},\mathcal{I}(Y))$ has a solution $X \in \mathbb{X}_n$ if and only if $Y > X \otimes X$.*

Proof. For some $(satb)^{th}$ nonzero element of $X \otimes X$ we have that $x_{st} = x_{ab} = 1$, and consider y_{satb}. If $y_{satb} = 0$ we have that $\{(s,t),(a,b)\} \in \mathcal{I}(Y)$, and since X is a solution to $P(X^{(ik)},\mathcal{I}(Y))$ we have that $x_{st}x_{ab} = 0$ which is a contradiction to the choice of the $(satb)^{th}$ element. So $y_{satb} = 1$ for any $(satb)^{th}$ element of $X \otimes X$ which has a value one, implying that $Y > X \otimes X$. Sufficiency is proved in a similar way. □

Theorems 3 and 5 provide characterizations for QAP and QAP-feasible matrices respectively, and will be used in the Greedy-Out algorithm for the QAP. Note that the combination of Theorems 5 and 4 suggests that in order to check whether some $Y \in \{0,1\}^{n^2 \times n^2}$ is a QAP-feasible matrix, it is enough to check if there is some $k \in I_n$ for some $i \in I_n$, such that for the given matrix $X^{(ik)}$ the problem $P(X^{(ik)}, \mathcal{I}(Y))$ has a solution.

5 The greedy-out-QAP Algorithm

The proposed Greedy-Out algorithm for the QAP is shown in Figure 3. The algorithm for a given nonnegative cost matrix $C \in \mathbb{R}_+^{n^2 \times n^2}$, computes a QAP matrix Y that is a Greedy-Out solution for that instance.

In line 1 we initialize Y to be a matrix of ones. In every iteration of the loop defined in lines 2 through 14 a subset of the elements of Y is set to zero, such that Y will remain QAP-feasible. In lines 3 and 4 we choose the largest element c_{ijkl} of C and set it to zero. This reflects the Greedy-Out behavior of the algorithm where we exclude the next worst element from the ground set. Even if the corresponding assignment $y_{ijkl} = 0$ is not possible because it would imply that Y will cease to be QAP-feasible, c_{ijkl} would not be considered in subsequent iterations. In lines 6 and 7 we temporarily set y_{ijkl} to zero, as well as and any other element of Y so dictated by the last two conditions of Theorem 3. For the resulting matrix, the corresponding Pairs Linear Assignment Problem is checked for feasibility in line 8. If there is a feasible solution, then by Theorem 5 we know that the aforementioned assignments transform Y into a QAP-feasible matrix, thereby they are made permanent in line 9. In the loop defined in lines 10 through 12, we set the corresponding elements of C to zero to reflect the assignments made to the elements of Y in line 7. The algorithm terminates when $|Y| = n^2$, which means that Y is a QAP matrix. The termination of the

Algorithm: Greedy-Out-QAP

```
Input :  cost matrix C ∈ R^{n²×n²}
Output:  QAP matrix Y ∈ {0,1}^{n²×n²}

1.  Y := 1^{n²×n²}
2.  while |Y| > n² do
3.    choose the greatest element c_ijkl of C
4.    c_ijkl := 0
5.    Ȳ := Y
6.    ȳ_ijkl := 0
7.    apply recursively the conditions ii) and iii) of Theorem 3 to Ȳ
8.    if ∃i,k ∈ I_n such that P(X^(ik), I(Ȳ)) has a solution then
9.      Y := Ȳ
10.     for s,t,a,b = 1,...,n do
11.       c_stab := c_stab ȳ_stab
12.     end for
13.   end if
14. end while
15. return Y
```

Figure 3: The Greedy-Out algorithm for the QAP

algorithm is guaranteed by steps 4 and 7, since C has a finite number of elements.

All the steps of the Greedy-Out algorithm can be performed in polynomial time apart from step 8 that requires the solution of the decision problem PLAP, which, as it will be proved in the next theorem, is not likely to be polynomially solvable.

Theorem 6. *PLAP is NP-Complete.*

Proof. It is clear that PLAP belongs to NP. We show that the DISJOINT MATCHINGS (DM) decision problem polynomially transforms to PLAP, where the former has been proved to be NP-Complete in [6].

The DISJOINTS MATCHINGS problem is the following. Given two bipartite graphs G_1 and G_2 with the same set of vertices $V = V_1 \cup V_2$, such that $V_1 \cap V_2 = \emptyset$, find two perfect matchings M_1 and M_2 such that $M_1 \cap M_2 = \emptyset$. Given an instance of DM, where without loss of generality let $|V_1| = |V_2| = n$, construct the following instance of PLAP. Construct matrices $Z_1, Z_2 \in \{0,1\}^{n \times n}$ such that

$$z_{ij}^{(l)} := \begin{cases} 1 & \text{if } (i,j) \in E(G_l), \\ 0 & \text{otherwise.} \end{cases}$$

where $Z_l = (z_{ij}^{(l)})$ for $l = 1, 2$, and denote the composition of Z_1, Z_2 as

$$Z := \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix}.$$

Define $\mathcal{I}$ as

$$\mathcal{I} := \{(i,j)(i+n, j+n) : z_{ij}^{(1)} = z_{ij}^{(2)} \text{for all } i, j \in I_n\}.$$

If $P(Z, \mathcal{I})$ has a feasible solution $X \in \mathbb{X}_{2n}$, then by construction

$$X := \begin{bmatrix} X_1 & 0 \\ 0 & X_2 \end{bmatrix},$$

and the permutation matrices $X_1, X_2 \in \mathbb{X}_n$ will correspond to disjoint perfect matchings for G_1 and G_2 respectively. □

Taking into consideration Theorem 2 and the fact that the QAP can easily be stated as a minimization problem on an independence system, the above result is to be expected. If PLAP was polynomially decidable, then GREEDY-OUT would be a polynomial-time constant approximation ratio algorithm for the QAP, contradicting the complexity result of Sahni and Gonzalez in [23].

6 Computational Experiment

In this section we will present computational results on the performance of both greedy algorithms for the QAP, as it is formulated in 4. In order to compare the GREEDY-OUT algorithm to the GREEDY-IN we performed a computational experiment on a set of randomly generated QAP instances. Specifically, for each value of $n = 1, \ldots, 18$, we generated 100 random input cost matrices $C \in \mathbb{R}_+^{n^2 \times n^2}$. Both algorithms were coded in Fortran and compiled with the `ifort` Intel Fortran compiler version 12.0. The computations were performed on a SGI Altix 450 shared memory architecture cluster, running SUSE Linux Enterprise server, with 16 Intel Itanium® II 1.67 GHz 64-bit processors, with a total of 32 GB shared memory. The results can be seen in Figure 4, where a plot of the values of

$$\rho = \frac{\text{GREEDY-IN solution}}{\text{GREEDY-OUT solution}}$$

are given for every problem in the instance set. We can see that the GREEDY-OUT consistently produces better quality solutions than GREEDY-IN for all values of n, except for $n = 2$ where both algorithms seem to produce the same solution values on the average. For values of $n = 9, \ldots, 18$ the GREEDY-OUT produces better quality solutions for the vast majority of the instances, while we can observe that its superiority in terms of solution quality over the GREEDY-IN tends to diminish as n approaches large values. This can be

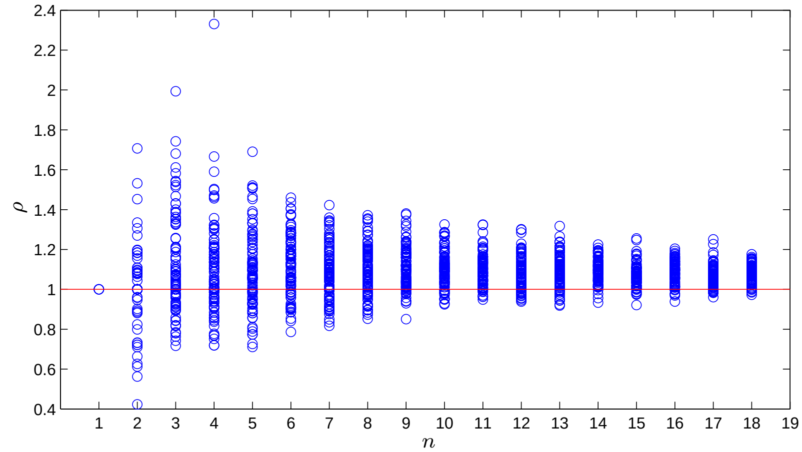

Figure 4: Greedy-In vs. Greedy-Out for the QAP

seen as a consequence of the asymptotic behavior of the QAP, which states that the relative ratio between the worst and best values of the objective function converges to 1, as the size of the problem approaches infinity [2]. Furthermore, in Figure 5 we can see the relationship between the instance size n and the percentage of instances p where each algorithm produces the best solution. So for $n = 19$ for example, the Greedy-Out algorithm finds the best solution in 90% of the cases while the Greedy-In in 15% of the cases. The fact that the sum of the instances for both algorithms is less than 100% for some n is due to the existence of some instances for which both algorithms produce the same solution value. We can observe that the percentage of instances for which the Greedy-Out produces better solution than Greedy-In tends to increase as n approaches large values.

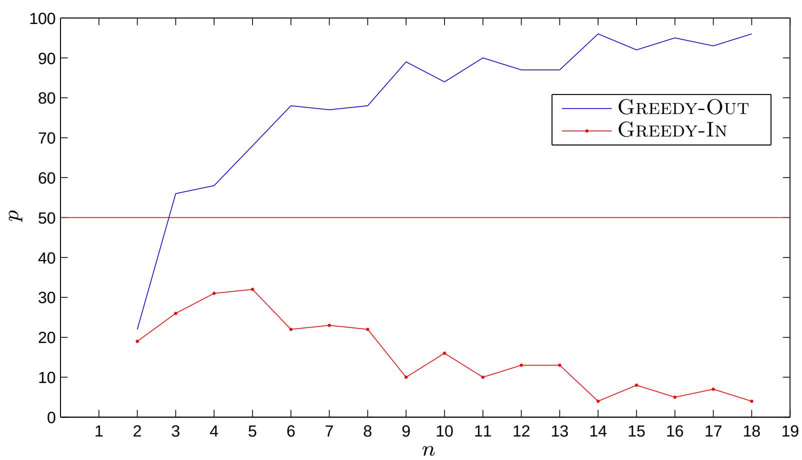

Figure 5: Greedy-In vs. Greedy-Out for the QAP

References

[1] R.E. Burkard, E. Çela, P.M. Pardalos, and L.S. Pitsoulis. The quadratic assignment problem. In Ding-Zhu Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 3, pages 241–337. Kluwer Academic Publishers, 1998.

[2] R.E. Burkard and U. Fincke. Probabilistic asymptotic properties of some combinatorial optimization problems. *Discrete Applied Mathematics*, 12:21–29, 1985.

[3] E. Çela. *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[4] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1:127–136, 1971.

[5] U. Faigle. The greedy algorithm for partially ordered sets. *Discrete Mathematics*, 28:153–159, 1979.

[6] A.M. Frieze. Complexity of a 3-dimensional assignment problem. *European Journal of Operations Research*, 13:161–164, 1983.

[7] D. Gale. Optimal assignments in an ordered set: an application of matroid theory. *Journal of Combinatorial Theory*, 4:176–180, 1968.

[8] D. Hausmann and B. Korte. K-greedy algorithms for independence systems. *Mathematical Methods of Operations Research*, 22(1):219–228, 1978.

[9] D. Hausmann, B. Korte, and T.A. Jenkyns. Worst case analysis of greedy type algorithms for independence systems. *Mathematical Programming Study*, 12:120–131, 1980.

[10] P. Helman, B.M.E. Moret, and H.D. Shapiro. An exact characterization of greedy structures. *SIAM Journal of Discrete Mathematics*, 6(2):274–283, 1993.

[11] A. Kammerdiner, P. Krokhmal and P.M. Pardalos. On the Hamming distance in combinatorial optimization problems on hypergraph matchings. *Optimization Letters*, 4(3):609–617, 2010.

[12] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Journal*, 49:291–307, 1972.

[13] T. C. Koopmans and M. J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[14] B. Korte and L. Lovász. Mathematical structures underlying greedy algorithms. In *Fundamentals of Computation Theory*, volume 117 of *Lecture Notes in Computer Science*, pages 205–209. Springer, 1981.

[15] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms (second edition)*. Springer, Berlin, Germany, 2002.

[16] B. Korte and C.L. Monma. Some remarks on a classification of oracle-type algorithms. *International Series on Numerical Mathematics*, 46:195–215, 1979.

[17] P. Krokhmal, D. Grundel and P.M. Pardalos. Asymptotic Behavior of the Expected Optimal Value of the Multidimensional Assignment Problem. *Mathematical Programming*, 109(2-3):525–551, 2007.

[18] P. Krokhmal and P.M. Pardalos. Random Assignment Problems. *European Journal of Operational Research*, 194(1):1–17, 2009.

[19] E. L. Lawler. The quadratic assignment problem. *Management Science*, 9:586–599, 1963.

[20] P.M. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: A survey and recent developments. In P.M. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 1–42. American Mathematical Society, 1994.

[21] M. Queyranne. Performance ratio of heuristics for triangle inequality quadratic assignment problems. *Operations Research Letters*, 4:231–234, 1986.

[22] R. Rado. A note on independence functions. *Proceedings of the London Mathematical Society*, 7:300–320, 1957.

[23] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the Association of Computing Machinery*, 23:555–565, 1976.