Maximum Quadratic Assignment Problem:
Reduction from Maximum Label Cover and LP-based Approximation Algorithm

Konstantin Makarychev* Rajsekar Manokaran† Maxim Sviridenko‡

Abstract

We show that for every positive $\varepsilon > 0$, unless $\mathcal{NP} \subset \mathcal{BPQP}$, it is impossible to approximate the maximum quadratic assignment problem within a factor better than $2^{\log^{1-\varepsilon} n}$ by a reduction from the maximum label cover problem. Then, we present an $O(\sqrt{n})$-approximation algorithm for the problem based on rounding of the linear programming relaxation often used in the state of the art exact algorithms.

1 Introduction

In this paper we consider the Quadratic Assignment Problem. An instance of the problem, $\Gamma = (G, H)$ is specified by two weighted graphs $G = (V_G, w_G)$ and $H = (V_H, w_H)$ such that $|V_G| = |V_H|$ (we denote $n = |V_G|$). The set of feasible solutions consists of bijections from V_G to V_H. For a given bijection φ the objective function is

$$\text{value}_{\text{QAP}}(\Gamma, \varphi) = \sum_{(u,v)\in V_G \times V_G} w_G(u, v) w_H(\varphi(u), \varphi(v)). \tag{1}$$

There are two variants of the problem the Minimum Quadratic Assignment Problem and the Maximum Quadratic Assignment Problem (MaxQAP) depending on whether the objective function (1) is to be minimized or maximized. The problem was first defined by Koopmans and Beckman [24] and sometimes this formulation of the problem is referred to as Koopmans-Beckman formulation of the Quadratic Assignment Problem. Both variants of the problem model an astonishingly large number of combinatorial optimization problems such as traveling salesman, maximum acyclic subgraph, densest subgraph and clustering problems to name a few. It also generalizes many practical problems that arise in various areas such as modeling of backboard wiring [33], campus and hospital layout [15, 17], scheduling [21] and many others [16, 25]. The surveys and books [2, 10, 13, 11, 26, 29] contain an in-depth treatment of special cases and various applications of the Quadratic Assignment Problem.

The Quadratic Assignment Problem is an extremely difficult optimization problem. The state of the art exact algorithms can solve instances with approximately 30 vertices, so a lot of research effort was concentrated on constructing good heuristics and relaxations of the problem.

*IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.
†Princeton University, Princeton NJ 08544
‡IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

Previous Results. The Minimum Quadratic Assignment Problem is known to be hard to approximate even under some very restrictive conditions on the weights of graphs G and H [32, 22]. Polynomial time exact [13] and approximation algorithms [22] are known for very specialized instances.

In contrast, MaxQAP seem to be more tractable. Barvinok [8] constructed an approximation algorithm with performance guarantee εn for any $\varepsilon > 0$. Nagarajan and Sviridenko [28] designed $O(\sqrt{n}\log^2 n)$-approximation algorithm by by utilizing approximation algorithms for the minimum vertex cover, densest k-subgraph and star packing problems. For the special case when one of the edge weight functions (w_G or w_H) satisfy the triangle inequality there are combinatorial 4-approximation [3] and LP-based 3.16-approximation algorithms [28]. Another tractable special case is the so-called dense Quadratic Assignment Problem [4]. This special case admits a sub-exponential time approximation scheme and in some cases it could be implemented in polynomial time [4, 20]. On the negative side, APX-hardness of MaxQAP is implied by the APX-hardness of its special cases, e.g. Traveling Salesman Problem with Distances One and Two [30].

An interesting special case of MaxQAP is the Densest k-Subgraph Problem. The best known algorithm by Bhaskara, Charikar, Chlamtac, Feige, and Vijayaraghavan [9] gives a $O(n^{1/4})$ approximation. However, the problem is not even known to be APX-hard (under standard complexity assumptions). Feige [18] showed that the Densest k-Subgraph Problem does not admit a ρ-approximation (for some universal constant $\rho > 1$) assuming that random 3-SAT formulas are hard to refute. Khot [23] ruled out PTAS for the problem under the assumption that $\mathcal{NP}$ does not have randomized algorithms that run in sub-exponential time.

Our Results. Our first result is the first superconstant non-approximability for MaxQAP. We show that for every positive $\varepsilon > 0$, unless $\mathcal{NP} \subset \mathcal{BPQP}$ ($\mathcal{BPQP}$ is the class of problems solvable in randomized quasipolynomial-time), it is impossible to approximate the maximum quadratic assignment problem with the approximation factor better than $2^{\log^{1-\varepsilon} n}$. Particularly, there is no polynomial time poly-logarithmic approximation algorithms for MaxQAP under the above complexity assumption. It is an interesting open question if our techniques can be used to obtain a similar result for the Densest k-Subgraph Problem.

Our second result is an $O(\sqrt{n})$-approximation algorithm based on rounding of the optimal solution of the linear programming relaxation. The LP relaxation was first considered by Adams and Johnson [1] in 1994. As a consequence of our result we obtain a bound of $O(\sqrt{n})$ on the integrality gap of this relaxation that almost matches a lower bound of $\Omega(\sqrt{n}/\log n)$ of Nagarajan and Sviridenko [28]. Note, that the previous $O(\sqrt{n}\log^2 n)$-approximation algorithm [28] was not based on the linear programming relaxation, and therefore no non-trivial upper bound on the integrality gap of the LP was known.

2 Hardness of Approximation

A weighted graph $G = (V, w)$ is specified by a vertex set V along with a weight function $w : V \times V \to \mathbb{R}$ such that for every $u, v \in V$, $w(u,v) = w(v,u)$ and $w(u,v) \geq 0$. An edge $e = (u,v)$ is said to be present in the graph G if $w(u,v)$ is non-zero.

We prove the inapproximability of the MaxQAP problem via an approximation preserving poly-time randomized reduction from the Label Cover problem defined below.

Definition 2.1 (Label Cover Problem)**.** *An instance of the label cover problem denoted by* $\Upsilon = (G = (V_G, E_G), \pi, [k])$ *consists of a graph G on V_G with edge set E_G along with a set of labels*

$[k] = \{0, 1, \ldots k-1\}$. *For each edge $(u,v) \in E_G$, there is a constraint π_{uv}, a subset of $[k] \times [k]$ defining the set of accepted labelings for the end points of the edge. The goal is to find a labeling of the vertices, $\Lambda : V_G \to [k]$ maximizing the total fraction of the edge constraints satisfied. We will denote the optimum of a instance Υ by $\mathrm{OPT}_{LC}(\Upsilon)$. In other words,*

$$\mathrm{OPT}_{LC}(\Upsilon) \stackrel{\mathrm{def}}{=} \max_{\Lambda : V_G \to [k]} \frac{1}{|E_G|} \sum_{(u,v) \in E} I((\Lambda(u), \Lambda(v)) \in \pi_{uv})$$

We will denote the fraction of edges satisfied by a labeling Λ by $\mathrm{value}_{LC}(\Upsilon, \Lambda)$.

The PCP theorem [6, 7], along with the Raz parallel repetition theorem [31] shows that the label cover problem is hard to approximate within a factor of $2^{\log^{1-\varepsilon} n}$.

Theorem 2.2 (see e.g., Arora and Lund [5])**.** *If $\mathcal{NP} \not\subset \mathcal{QP}$, then for every positive $\varepsilon > 0$, it is not possible to distinguish satisfiable instances of the label cover problem from instances with optimum at most $2^{-\log^{1-\varepsilon} n}$ in polynomial time.*

We will show an approximation preserving reduction from a label cover instance to a MaxQAP instance such that: if the label cover instance Υ is completely satisfiable, the MaxQAP instance Γ with have optimum 1; on the other hand, if $\mathrm{OPT}_{\mathrm{LC}}(\Upsilon)$ is at most δ, then no bijection φ obtains a value greater than $O(\delta)$.

Strictly speaking, the problem is not well defined when the graphs G and H do not have the same number of vertices. However, in our reduction, we will relax this condition by letting G have fewer vertices than H, and allowing the map φ to be only injective (i.e., $\varphi(u) \neq \varphi(v)$, for $u \neq v$). The reason is that we can always add enough isolated vertices to G to satisfy $|V_G| = |V_H|$. We also assume that the graphs are unweighted, and thus given an instance Γ consisting of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, the goal is to find an injective map $\varphi : V_G \to V_H$, so as maximize

$$\mathrm{value}_{\mathrm{QAP}}(\Gamma, \varphi) = \sum_{(u,v) \in E_G} I((\varphi(u), \varphi(v)) \in E_H),$$

here $I(\cdot)$ denotes the indicator function. We denote the optimum by $\mathrm{OPT}_{\mathrm{QAP}}(\Gamma)$.

Informally, our reduction does the following. Given an instance $\Upsilon = (G = (V_G, E_G), \pi, [k])$ of the label cover problem, consider the *label extended* graph H on $V_G \times [k]$ with edges $((u,i) - (v,j))$ for every $(u,v) \in E_G$ and every accepting label pair $(i,j) \in \pi_{uv}$. Every labeling Λ for Υ naturally defines a injective map, φ between V_G and $V_G \times [k]$: $\varphi(u) = (u, \Lambda(u))$. Note that φ maps edges satisfied by Λ onto edges of H. Conversely, given a injection $\varphi : V_G \to V_G \times [k]$ such that $\varphi(u) \in \{u\} \times [k]$ for every $u \in V_G$, we can construct a labeling Λ for Υ satisfying exactly the constraint edges in G which were mapped on to edges of H. However, the additional restriction on the injection is crucial for the converse to hold: an arbitrary injective map might not correspond to any labeling of the label cover Υ.

To overcome the above shortcoming, we modify the graphs G and H as follows. We replace each vertex u in G with a "cloud" of vertices $\{(u,i) : i \in [N]\}$ and each vertex (u,x) in H with a cloud of vertices $\{(u,x,i) : i \in [N]\}$, each index i is from a significantly large set $[N]$. Call the new graphs $\widetilde{G}$ and $\widetilde{H}$ respectively.

For every edge $(u,v) \in E_G$, the corresponding clouds in $\widetilde{G}$ are connected by a random bipartite graph where each edge occurs with probability α. We do this independently for each edge in E_G.

For every accepting pair $(x, y) \in \pi_{uv}$, we copy the "pattern" between the clouds $(u, x, \star)$ and $(v, y, \star)$ in $\widetilde{H}$.

Note, that every solution of the label cover problem $u \mapsto \Lambda(u)$ corresponds to the map $(u, i) \mapsto (u, \Lambda(u), i)$ which maps every "satisfied" edge of $\widetilde{G}$ to an edge of $\widetilde{H}$. The key observation now is that, we may assume that every (u, i) is mapped to some (u, x, i), since, loosely speaking, the pattern of edges between $(u, \star)$ and $(v, \star)$ is unique for each edge (u, v): there is no way to map the *cloud* of u to the *cloud* of u' and the *cloud* of v to the *cloud* of v' (unless $u = u'$ and $v = v'$), so that more than an α fraction of the edges of one cloud are mapped on edges of the other cloud. We will make the above discussion formal in the rest of this section.

Hardness Reduction

Input: A label cover instance $\Upsilon = (G = (V_G, E_G), \pi, [k])$.

Output: A MAXQAP instance $\Gamma = (\widetilde{G}, \widetilde{H})$; $\widetilde{G} = (V_{\widetilde{G}}, E_{\widetilde{G}})$, $\widetilde{H} = (V_{\widetilde{H}}, E_{\widetilde{H}})$.

Parameters: Let N be an integer bigger than $n^4 |E_G| k^5$ and $\alpha = 1/n$.

- Define $V_{\widetilde{G}} = V_G \times [N]$ and $V_{\widetilde{H}} = V_G \times [k] \times [N]$.
- For every edge (u, v) of G pick a random set of pairs $\mathcal{E}_{uv} \subset [N] \times [N]$. Each pair $(i, j) \in [N] \times [N]$ belongs to $\mathcal{E}_{uv}$ with probability α. The probabilities are chosen independently of each other.
- For every edge (u, v) of G and every pair (i, j) in $\mathcal{E}_{uv}$, add an edge $((u, i), (v, j))$ to $\widetilde{G}$. Then
$$E_{\widetilde{G}} = \{((u, i), (v, j)) : (u, v) \in E_G \text{ and } (i, j) \in \mathcal{E}_{uv}\}.$$
- For every edge (u, v) of G, every pair (i, j) in $\mathcal{E}_{uv}$, and every pair (x, y) in π_{uv}, add an edge $((u, x, i), (v, y, j))$ to $\widetilde{H}$. Then
$$E_{\widetilde{H}} = \{((u, x, i), (v, y, j)) : (u, v) \in E_G,\ (i, j) \in \mathcal{E}_{uv} \text{ and } (x, y) \in \pi_{uv}\}.$$

It is easy to see that the reduction runs in polynomial time. The size of the instance produced is nN^2k. In our reduction, both k and N are polynomial in n.

We will now show that the reduction is in fact approximation preserving with high probability. In the rest of the section, we will assume $\Gamma = (\widetilde{G}, \widetilde{H})$ is a MAXQAP instance obtained from a label cover instance Υ using the above reduction with parameteres N and α. Note that Γ is a random variable.

We will first show that if the label cover instance has a good labeling, the MAXQAP instance output by the above reduction has a large optimum. The following claim, which follows from a simple concentration inequality, shows that the graph $\widetilde{G}$ has, in fact, as many edges as expected.

Claim 2.3. *With high probability, $\widetilde{G}$ contains at least $\alpha |E_G| N^2/2$ edges.*

Lemma 2.4 (Completeness)**.** *Let Υ be a satisfiable instance of the Label Cover Problem. Then there exists a map of $\widetilde{G}$ to $\widetilde{H}$ that maps every edge of $\widetilde{G}$ to an edge of $\widetilde{H}$. Thus,* $\mathrm{OPT}_{QAP}(\Gamma) = |E_{\widetilde{G}}|$.

Proof. Let $u \mapsto \Lambda(u)$ be the solution of the label cover that satisfies all constrains. Define the map $\varphi : V_{\widetilde{G}} \to V_{\widetilde{H}}$ as follows $\varphi(u,i) = (u, \Lambda(u), i)$. Suppose that $((u,i),(v,j))$ is an edge in $\widetilde{G}$. Then $(u,v) \in E_G$ and $(i,j) \in \pi_{uv}$. Since the constraint between u and v is satisfied in the instance of the label cover, $(\Lambda(u), \Lambda(v)) \in \pi_{uv}$. Thus, $((u,\Lambda(u),i),(v,\Lambda(v),j)) \in E_{\widetilde{H}}$. □

Next, we will bound the optimum of Γ in terms of the value of the label cover instance Υ. We do this in two steps. We will first show that for a fixed map φ from $V_{\widetilde{G}}$ to $V_{\widetilde{H}}$ the expected value of Γ can be bounded as a function of the optimum of Υ. Note that this is well defined as $V_{\widetilde{G}}$ and $V_{\widetilde{H}}$ are determined by Υ and N (and independent of the randomness used by the reduction). Next, we show that the value is, in fact, tightly concentrated around the expected value. Then, we do a simple union bound over all possible φ to obtain the desired result. In what follows, φ is a fixed injective map from $V_{\widetilde{G}}$ to $V_{\widetilde{H}}$. Denote the first, second and third components of φ by φ_V, φ_{label} and $\varphi_{[N]}$ respectively. Then, $\varphi(u,i) = (\varphi_V(u,i), \varphi_{label}(u,i), \varphi_{[N]}(u,i))$.

Lemma 2.5. *For every injective map* $\varphi : V_{\widetilde{G}} \to V_{\widetilde{H}}$,

$$\mathbb{E}\left[\mathrm{value}_{QAP}(\Gamma,\varphi)\right] \le \alpha |E_G| N^2 \times (\mathrm{OPT}_{LC}(\Upsilon) + \alpha).$$

Proof. Define a probabilistic labeling of G as follows: *for every vertex u, pick a random $i \in [N]$, and assign label $\varphi_{label}(u,i)$ to u i.e., set $\Lambda(u) = \varphi_{label}(u,i)$.* The expected value of the solution to the Label Cover problem equals

$$\begin{aligned}
\mathbb{E}_\Lambda[\mathrm{value}_{\mathrm{LC}}(\Upsilon,\Lambda)] &= \frac{1}{|E_G|} \sum_{(u,v)\in E_G} \mathbb{E}\left[I((\Lambda(u),\Lambda(v)) \in \pi_{uv})\right] \\
&= \frac{1}{|E_G|} \sum_{(u,v)\in E_G} \frac{1}{N^2} \sum_{i,j\in[N]} I((\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{uv}).
\end{aligned}$$

Since $\mathrm{value}_{\mathrm{LC}}(\Upsilon,\Lambda) \le \mathrm{OPT}_{\mathrm{LC}}(\Upsilon)$ for every labeling $u \mapsto \Lambda(u)$,

$$\sum_{(u,v)\in E_G} \sum_{i,j\in[N]} I((\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{uv}) \le |E_G| \cdot N^2 \cdot \mathrm{OPT}_{\mathrm{LC}}(\Upsilon). \tag{2}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\left[\mathrm{value}_{\mathrm{QAP}}(\Gamma,\varphi)\right] &= \mathbb{E}\left[\sum_{((u,i),(v,j))\in E_{\widetilde{G}}} I((\varphi(u,i),\varphi(v,j)) \in E_{\widetilde{H}})\right] \\
&= \sum_{(u,v)\in E_G} \sum_{i,j\in[N]} \Pr\left\{(i,j) \in \mathcal{E}_{uv} \text{ and } (\varphi(u,i),\varphi(v,j)) \in E_{\widetilde{H}}\right\}. \qquad (3)
\end{aligned}$$

Recall, that the goal of the whole construction was to *force* the solution to map each (u,i) to $(u, \varphi_{label}(u,i), i)$. Let $\mathcal{C}_\varphi$ denote the set of quadruples that satisfy this property:

$$\mathcal{C}_\varphi = \{(u,i,v,j) : (u,v) \in E_G \text{ and } \varphi(u,i) = (u, \varphi_{label}(u,i), i),\ \varphi(v,j) = (v, \varphi_{label}(v,j), j)\}\ .$$

If $(u,i,v,j) \in \mathcal{C}_\varphi$, then

$$\begin{aligned}
&\Pr\{(i,j) \in \mathcal{E}_{uv} \text{ and } (\varphi(u,i),\varphi(v,j)) \in E_{\widetilde{H}}\} \\
&\qquad = \Pr\left\{(i,j) \in \mathcal{E}_{uv} \text{ and } (\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{uv}\right\} \\
&\qquad = \Pr\left\{(i,j) \in \mathcal{E}_{uv}\right\} \cdot I((\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{uv}) \\
&\qquad = \alpha \cdot I((\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{uv}).
\end{aligned}$$

If $(u,v) \in E_G$, but $(u,i,v,j) \notin \mathcal{C}_\varphi$, then either $(i,j) \neq (\varphi_{[N]}(u,i), \varphi_{[N]}(v,j))$ or $(u,v) \neq (\varphi_V(u,i), \varphi_V(v,j))$, and hence the events $\{(i,j) \in \mathcal{E}_{uv}\}$ and $\{(\varphi_{[N]}(u,i), \varphi_{[N]}(v,j)) \in \mathcal{E}_{\varphi_V(u,i)\varphi_V(v,j)}\}$ are independent. We have

$$\Pr\left\{(i,j) \in \mathcal{E}_{uv} \text{ and } (\varphi(u,i), \varphi(v,j)) \in E_{\widetilde{H}}\right\} \leq \Pr\left\{(i,j) \in \mathcal{E}_{uv} \text{ and } (\varphi_{[N]}(u,i), \varphi_{[N]}(v,j)) \in \mathcal{E}_{\varphi_V(u,i)\varphi_V(v,j)}\right\} \leq \alpha^2.$$

Now, splitting summation (3) into two parts depending on whether $(u,i,v,j) \in \mathcal{C}_\varphi$, we have

$$\mathbb{E}\left[\text{value}_{\text{QAP}}(\Gamma, (\varphi)\right] \leq \alpha |E_G| N^2 \, \text{OPT}_{\text{LC}}(\Upsilon) + \alpha^2 |E_G| N^2.$$

□

We use the following concentration inequality for Lipschitz functions on the boolean cube.

Theorem 2.6 (McDiarmid [27]). *Let $X_1, \dots X_T$ be independent random variables taking values in the set $\{0,1\}$. Let $f : \{0,1\}^T \to \mathbb{R}$ be a K-Lipschitz function i.e., for every $x, y \in \{0,1\}^T$, $|f(x) - f(y)| \leq K\|x-y\|_1$. Finally, let $\mu = \mathbb{E}\left[f(X_1, \dots, X_n)\right]$. Then for every positive ε,*

$$\Pr\left\{f(X_1, \dots, X_n) - \mu \geq \varepsilon\right\} \leq 2e^{\frac{-2\varepsilon^2}{TK^2}}.$$

Lemma 2.7. *For every injective map $\varphi : V_{\widetilde{G}} \to V_{\widetilde{H}}$,*

$$\Pr\left\{\text{value}_{QAP}(\Gamma, \varphi) - \mathbb{E}\left[\text{value}_{QAP}(\Gamma, \varphi)\right] \geq \alpha N^2\right\} \leq e^{-n^2 Nk}.$$

Proof. The presence of edges in the random graphs $\widetilde{G}$ and $\widetilde{H}$ is determined by the random sets $\mathcal{E}_{uv}$ (where $(u,v) \in E_G$). Thus, we can think of the random variable $\text{value}_{\text{QAP}}(\Gamma, \varphi)$ as of function of the indicator variables X_{uivj}, where X_{uivj} equals 1, if $(i,j) \in \mathcal{E}_{uv}$; and 0, otherwise. To be precise, $\text{value}_{\text{QAP}}(\Gamma, \varphi)$ equals

$$\sum_{\substack{(u,v)\in E_G \\ i,j \in [N]}} X_{uivj} X_{\varphi_V(u,i)\varphi_{[N]}(u,i)\varphi_V(v,j)\varphi_{[N]}(v,j)} I((\varphi_{label}(u,i), \varphi_{label}(v,j)) \in \pi_{\varphi_V(u,i)\varphi_V(v,j)}).$$

Observe, that variables X_{uivj} are mutually independent (we identify X_{uivj} with X_{vjui}). Each $X_{uivj} = 1$ with probability α. Finally, $\text{value}_{\text{QAP}}(\Gamma, \varphi)$ is (k^2+1)-Lipschitz as a function of the variables X_{uivj}. That is, if we change one of the variables X_{uivj} from 0 to 1, or from 1 to 0, then the value of the function may change by at most k^2+1. This follows from the expression above, since for every fixed φ, each X_{uivj} may appear in at most k^2+1 terms (reason: there is one term $X_{uivj} X_{\varphi_V(u,i)\varphi_{[N]}(u,i)\varphi_V(v,j)\varphi_{[N]}(v,j)}$ and at most k^2 terms $X_{u'i'v'j'} X_{\varphi_V(u',i')\varphi_{[N]}(u',i')\varphi_V(v',j')\varphi_{[N]}(v',j')}$, such that $\varphi(u',i') = (u,x,i)$ and $\varphi(v',j') = (v,y,j)$ for some $x, y \in [k]$, since φ is an injective map). McDiarmid's inequality with $T = N^2 \cdot |E_G|$, $K = (k^2+1)$, and $\varepsilon = \alpha N^2$, implies the statement of the lemma. □

Corollary 2.8 (Soundness). *With high probability, the reduction outputs an instance Γ such that*

$$\text{OPT}_{QAP}(\Gamma) \leq \alpha |E_G| N^2 \times (\text{OPT}_{LC}(\Upsilon) + 2\alpha)$$

Remark 2.9. *It is instructive to think, that $2\alpha \ll \text{OPT}_{LC}(\Upsilon)$.*

Proof. The total number of maps from V_G to V_H is $(nN)^{nNk}$. Thus, by the union bound, with probability $1 - o(1)$, for every injective mapping $\varphi : V_G \to V_H$:

$$\text{value}_{\text{QAP}}(\Gamma, \varphi) - \mathbb{E}\left[\text{value}_{\text{QAP}}(\Gamma, \varphi)\right] \leq \alpha N^2.$$

Plugging in the bound for the expected value from Lemma 2.5 gives

$$\text{OPT}_{\text{QAP}}(\Gamma) \leq \alpha |E_G| N^2 \, \text{OPT}_{\text{LC}}(\Upsilon) + \alpha^2 |E_G| N^2 + \alpha N^2.$$

□

Theorem 2.10. *For every positive $\varepsilon > 0$, there is no polynomial time approximation algorithm for the Maximum Quadratic Assignment problem with the approximation factor less than $D = 2^{\log^{1-\varepsilon} n}$ (where n is the number of vertices in the graph) unless $\mathcal{NP} \subset \mathcal{BPQP}$.*

Proof. Assume to the contrary that there exists a polynomial time algorithm A with the approximation factor less than $D = 2^{\log^{1-\varepsilon} n}$ for some positive ε. We use this algorithm to distinguish satisfiable instances of the label cover from at most $1/(4D)$–satisfiable instances in randomized polynomial time, which is not possible (if $\mathcal{NP} \not\subset \mathcal{BPQP}$) according to Theorem 2.2.

Let Υ be an instance of the label cover. Using the reduction described above transform Υ to an instance of MaxQAP Γ. Run the algorithm A on Γ. *Accept* Υ, if the value $A(\Gamma)$ returned by the algorithm is at least $|E_{\widetilde{G}}|/D$. *Reject* Υ, otherwise. By Lemma 2.4, if Υ is satisfiable, then $\text{OPT}_{\text{QAP}}(\Gamma) = |E_{\widetilde{G}}|$ and, hence $A(\Gamma) \geq |E_{\widetilde{G}}|/D$. Thus we always accept satisfiable instances. On the other hand, if the instance Υ is at most $1/(4D)$–satisfiable, then, by Corollary 2.8, with high probability

$$\text{OPT}_{\text{QAP}}(\Gamma) \leq \alpha |E_G| N^2 (\text{OPT}_{\text{LC}}(\Upsilon) + 2\alpha) < |E_{\widetilde{G}}|/D,$$

the second inequality follows from $|E_{\widetilde{G}}| \geq \alpha |E_G| N^2/2$ (see Claim 2.3). Therefore, with high probability, we reject Υ. □

3 LP Relaxation and Approximation Algorithm

We now present a new $O(\sqrt{n})$ approximation algorithm slightly improving on the result of Nagarajan and Sviridenko [28]. The new algorithm is surprisingly simple. It is based on a rounding of a natural LP relaxation. The LP relaxation is due to Adams and Johnson [1]. Thus we show that the integrality gap of the LP is $O(\sqrt{n})$.

Consider the following integer program. We have assignment variables x_{up} between vertices of the two graphs that are indicator variables of the events "*u maps to p*", and variables y_{upvq} that are indicator variables of the events "*u maps to p and v maps to q*". The LP relaxation is obtained by dropping the integrality condition on variables.

LP Relaxation

$$
\begin{array}{lll}
\max & \displaystyle\sum_{\substack{u,v\in V_G\\ p,q\in V_H}} w_G(u,v)w_H(p,q)y_{upvq} & \\
& \sum_{p\in V_H} x_{up} = 1, & \text{for all } u\in V_G;\\
& \sum_{u\in V_G} x_{up} = 1, & \text{for all } p\in V_H;\\
& \sum_{u\in V_G} y_{upvq} = x_{vq}, & \text{for all } u\in V_G,\ p,q\in V_H;\\
& \sum_{p\in V_H} y_{upvq} = x_{vq}, & \text{for all } u,v\in V_G,\ q\in V_H;\\
& y_{upvq} = y_{vqup}, & \text{for all } u,v\in V_G,\ p,q\in V_H;\\
& x_{up}\in[0,1], & \text{for all } u\in V_G,\ p\in V_H;\\
& y_{upvq}\in[0,1], & \text{for all } u\in V_G,\ p\in V_H.
\end{array}
$$

Approximation Algorithm

1. We solve the LP relaxation and obtain an optimal solution (x^*, y^*). Then we pick random subsets of vertices $L_G \subset V_G$ and $L_H \subset V_H$ of size $\lfloor n/2 \rfloor$. Let $R_G = V_G \setminus L_G$ and $R_H = V_H \setminus L_H$. In the rest of the algorithm, we will care only about edges going from L_G to R_G and from L_H to R_H; and we will ignore edges that completely lie in L_G, R_G, L_H or R_H.

2. For every vertex u in the set L_G, we pick a vertex p in L_H with probability x^*_{up} and set $\widetilde{\varphi}(u) = p$ (recall that $\sum_p x^*_{up} = 1$, for all u; with probability $1 - \sum_{p\in L_H} x^*_{up}$ we do not choose any vertex for u). Then for every vertex $p \in L_H$, which is chosen by at least one element u, we pick one of these u's uniformly at random; and set $\varphi(u) = p$ (in other words, we choose a random $u \in \widetilde{\varphi}^{-1}(p)$ and set $\varphi(u) = p$). Let $\widetilde{L}_G \subset L_G$ be the set of all chosen u's.

3. We now find a permutation $\psi : R_G \to R_H$ so as to maximize the contribution we get from edges from $\widetilde{L}_G$ to R_G i.e., to maximize the sum

$$\sum_{\substack{u\in\widetilde{L}_G\\ v\in R_G}} w_G(u,v)w_H(\varphi(u),\psi(v)).$$

   This can be done, since the problem is equivalent to the maximum matching problem between the sets R_G and R_H where the weight of the edge from v to q equals

$$\sum_{u\in\widetilde{L}_G} w_G(u,v)w_H(\varphi(u),q).$$

4. Output the mapping φ for vertices in $\widetilde{L}_G$, mapping ψ for vertices in R_G, and an arbitrary mapping for vertices in $L_G \setminus \widetilde{L}_G$, consistent with φ and ψ.

3.1 Analysis of the Algorithm

Theorem 3.1. *The approximation ratio of the algorithm is $O(\sqrt{n})$.*

While the algorithm is really simple, the analysis is more involved. Let LP^* be the value of the LP solution. To prove that the algorithm gives $O(\sqrt{n})$-approximation, it suffices to show that

$$\mathbb{E}\left[\sum_{\substack{u\in L_G\\ v\in R_G}} w_G(u,v)w_H(\varphi(u),\psi(v))\right] \geq \frac{LP^*}{O(\sqrt{n})}. \tag{4}$$

We split all edges of graph G into two sets: heavy edges and light edges. For each vertex $u \in V_G$, let $\mathcal{W}_u$ be the set of $\lceil\sqrt{n}\rceil$ vertices $v \in V_G$ with the largest weight $w_G(u,v)$. Then,

$$LP^* = \sum_{\substack{u\in V_G\\ v\in V_G\setminus\mathcal{W}_u}} \sum_{p,q\in V_H} y^*_{upvq} w_G(u,v)w_H(p,q) + \sum_{\substack{u\in V_G\\ v\in\mathcal{W}_u}} \sum_{p,q\in V_H} y^*_{upvq} w_G(u,v)w_H(p,q).$$

Denote the first term by LP^*_I and the second by LP^*_{II}. Instead of working with ψ, we explicitly define two new bijective maps ν_I and ν_{II} from R_G to R_H and prove, that

$$\mathbb{E}\left[\sum_{\substack{u\in \widetilde{L}_G\\ v\in R_G}} w_G(u,v)w_H(\varphi(u),\nu_I(v))\right] \geq \frac{LP^*_I}{O(\sqrt{n})};\ \text{and}\ \mathbb{E}\left[\sum_{\substack{u\in \widetilde{L}_G\\ v\in R_G}} w_G(u,v)w_H(\varphi(u),\nu_{II}(v))\right] \geq \frac{LP^*_{II}}{O(\sqrt{n})}.$$

These two inequalities imply the bound we need, since the sum (4) is greater than or equal to each of the sums above. Before we proceed, we state two simple lemmas we need later (see the appendix for the proofs).

Lemma 3.2. *Let S be a random subset of a set V. Suppose that for $u \in V$, all events $\{u' \in S\}$ where $u' \neq u$ are jointly independent of the event $\{u \in S\}$. Let s be an element of S chosen uniformly at random (if $S = \varnothing$, then s is not defined). Then $\Pr\{u = s\} \geq \Pr\{u \in S\}/(\mathbb{E}[|S|]+1)$.*

Lemma 3.3. *Let S be a random subset of a set L, and T be a random subset of a set R. Suppose that for $(l,r) \in L \times R$, all events $\{l' \in S\}$ where $l' \neq l$ and all events $\{r' \in T\}$ where $r' \neq r$ are jointly independent of the event $\{(l,r) \in S \times T\}$. Let s be an element of S chosen uniformly at random, and let t be an element of T chosen uniformly at random. Then,*

$$\Pr\{(l,r) = (s,t)\} \geq \frac{\Pr\{(l,r)\in S\times T\}}{(\mathbb{E}[|S|]+1)\times(\mathbb{E}[|T|]+1)}$$

(here (s,t) is not defined if $S = \varnothing$ or $T = \varnothing$).

The first map ν_I is a random permutation between R_G and R_H. Observe, that given subsets L_G and L_H, the events $\{\widetilde{\varphi}(u) = p\}$ are mutually independent for different u's and the expected size of $\widetilde{\varphi}^{-1}(p)$ is at most 1, here $\widetilde{\varphi}^{-1}(p)$ is the preimage of p (recall the map $\widetilde{\varphi}$ may have collisions, and hence $\widetilde{\varphi}^{-1}(p)$ may contain more than one element). Thus, by Lemma 3.2 applied to the set $\widetilde{\varphi}^{-1}(p) \subset L_G$,

$$\Pr\{\varphi(u) = p \mid L_G, L_H\} \geq \Pr\{\widetilde{\varphi}(u) = p \mid L_G, L_H\}/2 = \begin{cases} x^*_{up}/2, & \text{if } u \in L_G \text{ and } p \in L_H;\\ 0, & \text{otherwise.}\end{cases}$$

For every $u,v \in V_G$ and $p,q \in V_H$, let $\mathcal{E}_{upvq}$ be the event $\{u \in L_G, v \in R_G, p \in L_H, q \in R_H\}$. Then,

$$\Pr\{\mathcal{E}_{upvq}\} = \Pr\{u \in L_G, v \in R_G, p \in L_H, q \in R_H\} = \frac{1}{16} - o(1).$$

Thus, the probability that $\varphi(u) = p$ and $\nu_I(u) = q$ is $\Omega(x^*_{up}/n)$. We have

$$\begin{aligned}
\mathbb{E}\left[\sum_{\substack{u\in L_G\\ v\in R_G}} w_G(u,v) w_H(\varphi(u),\nu_I(v))\right] &\geq \Omega(1) \times \sum_{u,v\in V_G}\sum_{p,q\in V_H} \frac{x^*_{up}}{n}\, w_G(u,v)w_H(p,q)\\
&\geq \Omega(1) \times \sum_{p,q\in V_H} w_H(p,q) \sum_{u\in V_G} x^*_{up} \sum_{v\in\mathcal{W}_u} \frac{w_G(u,v)}{n}\\
&\geq \Omega(1) \times \sum_{p,q\in V_H} w_H(p,q) \sum_{u\in V_G} x^*_{up} \frac{\min\{w_G(u,v) : v\in\mathcal{W}_u\}}{\sqrt{n}}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
LP^*_I &= \sum_{p,q\in V_H} w_H(p,q) \sum_{u\in V_G} x^*_{up}\left(\sum_{v\in V_G\setminus\mathcal{W}_u} \frac{y^*_{upvq}}{x^*_{up}} w_G(u,v)\right)\\
&\leq \sum_{p,q\in V_H} w_H(p,q) \sum_{u\in V_G} x^*_{up} \max\{w_G(u,v) : v \in V_G\setminus\mathcal{W}_u\}\\
&\leq \sum_{p,q\in V_H} w_H(p,q) \sum_{u\in V_G} x^*_{up} \min\{w_G(u,v) : v\in\mathcal{W}_u\}.
\end{aligned}$$

We now define ν_{II}. For every $v \in V_G$, let

$$l(v) = \operatorname{argmax}_{u\in V_G}\left\{\sum_{p,q\in V_H} w_G(u,v)w_H(p,q)y_{upvq}\right\}.$$

We say that $(l(v), v)$ is a heavy edge. For every $u \in L_G$, let

$$\mathcal{R}_u = \{v \in R_G : l(v) = u\}.$$

All sets $\mathcal{R}_u$ are disjoint subsets of R_G. We now define a map $\widetilde{\nu}_{II} : \mathcal{R}_u \to R_H$ independently for each $\mathcal{R}_u$ for which $\widetilde{\varphi}(u)$ is defined (even if $\varphi(u)$ is not defined). For every $v \in \mathcal{R}_u$, and $q \in R_H$, define

$$z_{vq} = \frac{y^*_{u\widetilde{\varphi}(u)vq}}{x^*_{u\widetilde{\varphi}(u)}}.$$

Observe, that $\sum_{v\in\mathcal{R}_u} z_{vq} \leq 1$ for each $q \in R_H$ and $\sum_{q\in R_H} z_{vq} \leq 1$ for each $v \in \mathcal{R}_u$. Hence, for a fixed $\mathcal{R}_u$, the vector $(z_{vq} : v \in \mathcal{R}_u, q \in R_H)$ lies in the convex hull of integral partial matchings between $\mathcal{R}_u$ and R_H. Thus, the fractional matching $(z_{vq} : v \in \mathcal{R}_u, q \in R_H)$ can be represented as a convex combination of integral partial matchings. Pick one of them with the probability proportional to its weight in the convex combination. Call this matching $\widetilde{\nu}^u_{II}$. Note, that $\widetilde{\nu}^u_{II}$ is injective and that the supports of $\widetilde{\nu}^{u'}_{II}$ and $\widetilde{\nu}^{u''}_{II}$ do not intersect if $u' \neq u''$ (since $\mathcal{R}_{u'} \cap \mathcal{R}_{u''} = \varnothing$).

Let $\widetilde{\nu}_{II}$ be the union of $\widetilde{\nu}^u_{II}$ for all $u \in L_G$. The partial map $\widetilde{\nu}_{II}$ may not be injective and may map several vertices of R_G to the same vertex q. Thus, for every q in the image of R_G, we pick uniformly at random one preimage v and set $\nu_{II}(v) = q$. We define ν_{II} on the rest of R_G arbitrarily.

Fix L_G, L_H and $R_G = V_G \setminus L_G$, $R_H = V_H \setminus L_H$. Let $u \in L_G$, $v \in \mathcal{R}_u$, $p \in L_H$ and $q \in R_H$. We want to estimate the probability that $\varphi(u) = p$ and $\nu_{II}(v) = q$. Observe, that given sets L_G and L_H, the event $\{\widetilde{\varphi}(u) = p \text{ and } \widetilde{\nu}_{II}(v) = q\}$ is independent of all events $\{\widetilde{\varphi}(u') = p\}$ for $u' \neq u$ and all events $\{\widetilde{\nu}_{II}(v') = q\}$ for $v' \notin \mathcal{R}_u$. The expected size of $\widetilde{\nu}^{-1}_{II}(q)$ is at most 1, since

$$\sum_{u' \in L_G} \sum_{v' \in \mathcal{R}_{u'}} \Pr\left\{\widetilde{\nu}^{u'}_{II}(v') = q\right\} \leq \sum_{u' \in L_G} \sum_{v' \in \mathcal{R}_{u'}} \sum_{p' \in L_H} x^*_{u'p'} y^*_{u'p'v'q} / x^*_{u'p'} \leq \sum_{v' \in V_G} \sum_{p' \in V_H} y^*_{l(v')p'v'q} = \sum_{v' \in V_G} x^*_{v'q} \leq 1.$$

Therefore, by Lemma 3.3,

$$\Pr\{\varphi(u) = p \text{ and } \nu_{II}(v) = q \ | \ L_G, L_H, u \in L_G, v \in \mathcal{R}_u, p \in L_H, q \in H\} \geq \Pr\{\widetilde{\varphi}(u) = p \text{ and } \widetilde{\nu}_{II}(v) = q \ | \ L_G, L_H, u \in L_G, v \in \mathcal{R}_u, p \in L_H, q \in R_H\}/4 = y^*_{upvq}/4.$$

We are now ready to estimate the value of the solution:

$$\begin{aligned}
\mathbb{E}\left[\sum_{\substack{u \in L_G \\ v \in R_G}} w_G(u,v) w_H(\varphi(u), \nu_{II}(v))\right] &\geq \mathbb{E}_{L_G, L_H}\left[\sum_{\substack{u \in L_G \\ v \in \mathcal{R}_u}} \sum_{\substack{p \in L_H \\ q \in R_H}} \frac{y^*_{upvq}}{4}\, w_G(u,v) w_H(p,q)\right] \\
&= \frac{1}{4}\, \mathbb{E}_{L_G, L_H}\left[\sum_{v \in R_G : l(v) \in L_G} \sum_{\substack{p \in L_H \\ q \in R_H}} y^*_{l(v)pvq}\, w_G(u,v) w_H(p,q)\right] \\
&= \frac{1}{4} \sum_{v \in V_G} \sum_{p,q \in V_H} \Pr\left\{\mathcal{E}_{l(v)pvq}\right\}\ y^*_{l(v)pvq}\, w_G(l(v),v) w_H(p,q) \\
&= \frac{1}{64 + o(1)} \sum_{v \in V_G} \sum_{p,q \in V_H} y^*_{l(v)pvq}\, w_G(l(v),v) w_H(p,q) \\
&\geq \frac{1}{65} \sum_{v \in V_G} \max_{u \in V_G} \left\{\sum_{p,q \in V_H} y^*_{upvq}\, w_G(u,v) w_H(p,q)\right\} \\
&\geq \frac{1}{65} \sum_{v \in V_G} \frac{1}{|\mathcal{W}_v|} \sum_{u \in \mathcal{W}_v} \left(\sum_{p,q \in V_H} y^*_{upvq}\, w_G(u,v) w_H(p,q)\right) \\
&= \frac{1}{65} \times \frac{LP^*_{II}}{\lceil \sqrt{n}\, \rceil}.
\end{aligned}$$

This finishes the proof.

3.2 De-randomized algorithm

We now give a de-randomized version of the approximation algorithm. We will need a slightly weaker LP relaxation than we used before.

LP Relaxation

$$
\begin{array}{lll}
\max & \displaystyle\sum_{\substack{u,v\in V_G\\ p,q\in V_H}} w_G(u,v)w_H(p,q)y_{upvq} & \\
& \sum_{p\in V_H} x_{up} \leq 1, & \text{for all } u \in V_G; \\
& \sum_{u\in V_G} x_{up} \leq 1, & \text{for all } p \in V_H; \\
& \sum_{u\in V_G} y_{upvq} \leq x_{vq}, & \text{for all } u \in V_G,\ p,q \in V_H; \\
& \sum_{p\in V_H} y_{upvq} \leq x_{vq}, & \text{for all } u,v \in V_G,\ q \in V_H; \\
& y_{upvq} = y_{vqup}, & \text{for all } u,v \in V_G,\ p,q \in V_H; \\
& x_{up} \in [0,1], & \text{for all } u \in V_G,\ p \in V_H; \\
& y_{upvq} \in [0,1], & \text{for all } u \in V_G,\ p \in V_H.
\end{array}
$$

This LP is obtained from the original LP by replacing equalities "=" with inequalities "≤" in the first four constraints. The integrality gap of the new LP is the same as of the original LP. In fact, given a feasible solution x^*, y^* of the new LP we can always increase the values of some variables to get a feasible solution x^{**}, y^{**} of the original LP (then $x^{**} \geq x^*$ and $y^{**} \geq y^*$ component-wise).

Theorem 3.4. *There exists a polynomial time (deterministic) algorithm that given an instance Γ of* MaxQAP *consisting of two weighted graphs $G = (V_G, w_G)$, $H = (V_H, w_H)$ and a solution (x^*, y^*) to the LP, of cost LP^*, outputs a bijection $\varphi : V_G \to V_H$ such that*

$$\text{value}_{QAP}(\Gamma, \varphi) \geq \frac{LP^*}{O(\sqrt{n})}.$$

Proof. The existence of the map φ follows from Theorem 3.1. We have already established that either $LP_I^* \geq LP_{II}^*$ (see Theorem 3.1 for definitions) and then

$$\sum_{u,v\in V_G} \sum_{p,q\in V_H} \frac{x_{up}^*}{n} w_G(u,v) w_H(p,q) \geq C_{r.alg} \frac{LP^*}{\sqrt{n}};$$

or $LP_{II}^* \geq LP_I^*$ and then there exists a map $\varphi_{r.alg} : V_G \to V_H$ (returned by the randomized algorithm) and a disjoint set of stars $\mathcal{S} = \{(u, \mathcal{R}_u)\}$ (each with the center in the vertex $u \in V_G$ and leaves $\mathcal{R}_u \subset V_G$) such that

$$\sum_{(u,\mathcal{R}_u)\in\mathcal{S}} \sum_{v\in\mathcal{R}_u} w_G(u,v) w_H(\varphi_{r.alg}(u), \varphi_{r.alg}(v)) \geq C_{r.alg} \frac{LP^*}{\sqrt{n}},$$

for some universal constant $C_{r.alg}$. We consider these cases separately.

I. First, assume that $LP_I^* \geq LP_{II}^*$. Find a bijection $\varphi : V_G \to V_H$ that maximizes

$$\sum_{u \in V_G} \left[\frac{1}{n} \sum_{\substack{v \in V_G \\ q \in V_H}} w_G(u,v) w_H(\varphi(u), q) \right].$$

We find the bijection by solving the maximum matching problem between V_G and V_H, where the cost of mapping $u \mapsto p$ equals

$$\frac{1}{n} \sum_{\substack{v \in V_G \\ q \in V_H}} w_G(u,v) w_H(\varphi(u), q).$$

Then we find a bijection $\nu : V_G \to V_H$ that maximizes

$$\sum_{u,v \in V_G} w_G(u,v) w_H(\varphi(u), \nu(v)).$$

Again, we do this by solving the maximum matching problem, where now the cost of mapping $v \mapsto q$ equals

$$\sum_{u \in V_G} w_G(u,v) w_H(\varphi(u), q).$$

Since for a random permutation ν_I the maximum is at least $C_{r.alg} LP^*/\sqrt{n}$, we get

$$\sum_{u \in V_G} \sum_{v \in V_G} w_G(u,v) w_H(\varphi(u), \nu(v)) \geq C_{r.alg} \frac{LP^*}{\sqrt{n}}. \tag{5}$$

We now use the greedy deterministic MAX CUT approximation algorithm[1] to partition V_G into two sets L_G and R_G so as to maximize

$$\sum_{u \in L_G} \sum_{v \in R_G} w_G(u,v) w_H(\varphi(u), \nu(v)).$$

The cost of cutting an edge (u,v) is $w_G(u,v)w_H(\varphi(u), \nu(v))$. The cost of the obtained solution is at least a half of (5). We now use the greedy deterministic MAX DICUT (directed maximum cut) approximation algorithm[2] to partition V_H into sets L_H and R_H so as to maximize

$$\sum_{\substack{u \in L_G \\ \varphi(u) \in L_H}} \sum_{\substack{v \in R_G \\ \varphi(v) \in R_H}} w_G(u,v) w_H(\varphi(u), \nu(v)) = \sum_{\substack{p \in L_H \\ \varphi^{-1}(p) \in L_G}} \sum_{\substack{q \in R_H \\ \nu^{-1}(q) \in L_H}} w_G(\varphi^{-1}(p), \nu^{-1}(q)) w_H(p,q).$$

[1]The greedy MAX CUT algorithm picks vertices from the set V_G in an arbitrary order and puts them in the sets L_G or R_G. Thus, at every step t all vertices are partitioned into three groups $L_G(t)$, $R_G(t)$ and a group of not yet processed vertices $U_G(t)$. If the weight of edges going from v to $R_G(t)$ is greater than the weight of edges going from v to $L_G(t)$, then the algorithm adds v to L_G, otherwise to R_G. The algorithm maintain the following invariant: at every step the weight of cut edges is greater than or equal to the weight of uncut edges. Thus, in the end, the weight of cut edges is at least a half of the total weight of all edges.

[2]The greedy MAX DICUT algorithm first finds an undirected maximum cut (A_G, B_G) using the greedy MAX CUT algorithm. The cost of the undirected maximum cut is at least a half of the total weight of all edges. Then, it outputs the cut (A_G, B_G), if more edges are directed from A_G to B_G than from B_G to A_G, it outputs the cut (B_G, A_G), otherwise. The cost of the directed cut is at least a quarter of the total weight of all directed edges.

The cost of a directed edge (p,q) is $w_G(\varphi^{-1}(p),\nu^{-1}(q))w_H(p,q)$, if $\varphi^{-1}(p)\in L_G$, $\nu^{-1}(q)\in R_G$; and 0 otherwise. The cost of the obtained solution is at least 1/8 of (5). Thus

$$\sum_{\substack{u\in L_G:\varphi(u)\in L_H\\ v\in R_G:\nu(v)\in R_H}} w_G(u,v)w_H(\varphi(u),\nu(v)) \geq \frac{C_{r.alg}}{8}\frac{LP^*}{\sqrt{n}}. \tag{6}$$

Note that we do not require that $|L_G|=|L_H|$ or that $|R_G|=|R_H|$. We output the map that maps $u\in L_G$ to $\varphi(u)$ if $\varphi(u)\in L_H$; and $v\in R_G$ to $\nu(v)$ if $\nu(v)\in R_H$. It maps the remaining vertices in an arbitrary way. The cost of the solution is at least (6).

II. We now assume that there exists a collection of disjoint stars $\mathcal{S}=\{(u,\mathcal{R}_u)\}$ (each with the center in the vertex $u\in V_G$ and leaves $\mathcal{R}_u\subset V_G$) and a map $\varphi_{r.alg}:V_G\to V_H$ such that

$$\sum_{(u,\mathcal{R}_u)\in\mathcal{S}}\sum_{v\in\mathcal{R}_u} w_G(u,v)w_H(\varphi_{r.alg}(u),\varphi_{r.alg}(v)) \geq C_{r.alg}\frac{LP^*}{\sqrt{n}}. \tag{7}$$

Define the LP volume of sets $S\subset V_G$, $T\subset V_H$ as follows:

$$\mathrm{vol}_{LP}(S,T)=\sum_{\substack{u\in S\\ v\in V_G}}\sum_{p,q\in V_H} w_G(u,v)w_H(p,q)y^*_{upvq}+\sum_{u,v\in V_G}\sum_{\substack{p\in T\\ q\in V_H}} w_G(u,v)w_H(p,q)y^*_{upvq}.$$

If $S_1,\dots,S_k$ is a partition of V_G and $T_1,\dots,T_k$ is a partition of V_H, then

$$\sum_{i=1}^k \mathrm{vol}_{LP}(S_k,T_k)=2LP^*,$$

since on the left hand side every term of the LP is counted twice. Particularly,

$$\sum_{(u,\mathcal{R}_u)\in\mathcal{S}} \mathrm{vol}_{LP}(\{u\}\cup\mathcal{R}_u,\varphi_{r.alg}(\{u\}\cup\mathcal{R}_u))=2LP^*.$$

Plugging in (7), we get

$$\sum_{(u,\mathcal{R}_u)\in\mathcal{S}}\left(2\sum_{v\in\mathcal{R}_u} w_G(u,v)w_H(\varphi_{r.alg}(u),\varphi_{r.alg}(v))-\frac{C_{r.alg}}{\sqrt{n}}-\mathrm{vol}_{LP}(\{u\}\cup\mathcal{R}_u,\varphi_{r.alg}(\{u\}\cup\mathcal{R}_u))\right)\geq 0.$$

This inequality implies that there exists one star $(u^*,\mathcal{R}_{u^*})$ such that

$$2\sum_{v\in\mathcal{R}_{u^*}} w_G(u^*,v)w_H(\varphi_{r.alg}(u^*),\varphi_{r.alg}(v))\geq\frac{C_{r.alg}}{\sqrt{n}}\ \mathrm{vol}_{LP}(\{u^*\}\cup\mathcal{R}_{u^*},\varphi_{r.alg}(\{u^*\}\cup\mathcal{R}_{u^*})).$$

We find a star $(u,\mathcal{R})$ and an injective map $\varphi:\{u\}\cup\mathcal{R}\to V_H$ satisfying this inequality. We do this as follows: For every choice of u and $\varphi(u)$, we solve the maximum partial matching problem where the cost of assigning $v\mapsto q$ equals

$$2w_G(u,v)w_H(\varphi(u),q)-\\ -\frac{C_{r.alg}}{\sqrt{n}}\left[\sum_{u'\in V_G}\sum_{p',q'\in V_H} w_G(u',v)w_H(p',q')y^*_{u'p'vq'}+\sum_{u',v'\in V_G}\sum_{p'\in V_H} w_G(u',v')w_H(p',q)y^*_{u'p'v'q}\right].$$

The set of matched vertices v is the set of leaves of the star; u is the center.

We fix the solution to be φ on $(u, \mathcal{R}_u)$. We remove the star $(u, \mathcal{R}_u)$ from the graph G and its image $(\varphi(u), \varphi(\mathcal{R}_u))$ from the graph H. We repeat the algorithm recursively for the remaining graphs. To estimate the cost of the solution, observe that the value of the LP decreases by

$$\begin{aligned}
&\sum_{\substack{u,v\in V_G\\ p,q\in V_H}} w_G(u,v)w_H(p,q)y^*_{upvq} - \sum_{\substack{u,v\in V_G\setminus(\{u\}\cup\mathcal{R}_u)\\ p,q\in V_H\setminus(\{\varphi(u)\}\cup\varphi(\mathcal{R}_u))}} w_G(u,v)w_H(p,q)y^*_{upvq}\\
&\qquad\le \sum_{\substack{u\in(\{u\}\cup\mathcal{R}_u),v\in V_G\\ p,q\in V_H}} w_G(u,v)w_H(p,q)y^*_{upvq} + \sum_{\substack{u,v\in V_G\\ p\in(\{\varphi(u)\}\cup\varphi(\mathcal{R}_u)),q\in V_H}} w_G(u,v)w_H(p,q)y^*_{upvq}\\
&\qquad\quad + \sum_{\substack{u\in V_G,v\in(\{u\}\cup\mathcal{R}_u)\\ p,q\in V_H}} w_G(u,v)w_H(p,q)y^*_{upvq} + \sum_{\substack{u,v\in V_G\\ p\in V_H,q\in(\{\varphi(u)\}\cup\varphi(\mathcal{R}_u))}} w_G(u,v)w_H(p,q)y^*_{upvq}\\
&\qquad= 2\,\mathrm{vol}_{LP}(\{u\}\cup\mathcal{R}_u, \varphi(\{u\}\cup\mathcal{R}_u)),
\end{aligned}$$

while the profit we get from mapping $(u, \mathcal{R}_u) \mapsto (\varphi(u), \varphi(\mathcal{R}_u))$ is at least

$$\frac{C_{r.alg}}{2\sqrt{n}}\,\mathrm{vol}_{LP}(\{u\}\cup\mathcal{R}_u, \varphi(\{u\}\cup\mathcal{R}_u)).$$

Hence, the approximation ratio is at least $C_{r.alg}/(4\sqrt{n})$. □

4 Conclusion

There are many open problems and research directions for MaxQAP. Developing and computational testing of various semidefinite programming (SDP) relaxations for the MaxQAP is an active area of research [2, 14]. Feige [19] constructed a set of instances with integrality gap $\Omega(n^{1/3-\varepsilon})$ where $\varepsilon > 0$ is an arbitrary constant for a natural SDP relaxation while our $O(\sqrt{n})$-approximation algorithm implies an upper bound on the integrality gap since the SDP relaxation is stronger than the linear programming one. Closing this gap is an interesting open problem. Another promising approach is applying methods recently introduced in papers by Charikar, Hajiaghayi, Karloff [12] and by Bhaskara, Charikar, Chlamtac, Feige, and Vijayaraghavan [9].

5 Acknowledgement

We would like to thank anonymous referees for valuable comments and suggestions.

References

[1] W. P. Adams and T. A. Johnson, Improved Linear Programming-based Lower Bounds for the Quadratic Assignment Problem, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 16, 1994, 43-77.

[2] K. Anstreicher, Recent advances in the solution of quadratic assignment problems, ISMP, 2003 (Copenhagen). Math. Program. 97, 2003, no. 1-2, Ser. B, 27-42.

[3] E. Arkin, R. Hassin and M. Sviridenko, Approximating the Maximum Quadratic Assignment Problem, Information Processing Letters, 77, 2001, pp. 13-16.

[4] S. Arora, A. Frieze and H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arrangement problems, Mathematical Programming, 92(1), 2002, 1-36.

[5] S. Arora and C. Lund. Hardness of Approximations. In Approximation Algorithms for NP-hard Problems, Dorit Hochbaum, Ed. PWS Publishing , 1996.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and the hardness of approximation problems, Journal of the ACM 45 (3).

[7] S. Arora, and S. Safra, Probabilistic checking of proofs: A new characterization of NP, Journal of the ACM 45 (1): 70122.

[8] A. Barvinok, Estimating L^∞ norms by L^{2k} norms for functions on orbits. Found. Comput. Math. 2 (2002), no. 4, 393–412.

[9] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige and A. Vijayaraghavan, Detecting High Log-Densities – an $O(n^{1/4})$ Approximation for Densest k-Subgraph, to appear in Proceedings of STOC 2010.

[10] R.E. Burkard, E. Cela, P. Pardalos and L.S. Pitsoulis, The quadratic assignment problem, In Handbook of Combinatorial Optimization, D.Z. Du, P.M. Pardalos (Eds.), Vol. 3, Kluwer Academic Publishers, 1998, 241-339.

[11] R.E. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, SIAM Philadelphia, 2009.

[12] M. Charikar, M. Hajiaghayi, H. Karloff, Improved Approximation Algorithms for Label Cover Problems, In Proceedings of ESA 2009, pp. 23–34

[13] Eranda Cela, The Quadratic Assignment Problem: Theory and Algorithms, Springer, 1998.

[14] Y. Dong and H. Wolkowicz,A Low-Dimensional Semidefinite Relaxation for the Quadratic Assignment Problem, Mathematics of Operations Research 34 (2009), pp. 1008-1022.

[15] J. Dickey and J. Hopkins, Campus building arrangement using TOPAZ, Transportation Science, 6, 1972, pp. 59–68.

[16] H. Eiselt and G. Laporte, A combinatorial optimization problem arising in dartboard design, Journal of Operational Research Society, 42, 1991, pp. 113–118.

[17] A. Elshafei, Hospital layout as a quadratic assignment problem, Operations Research Quarterly, 28, 1977, pp. 167–179.

[18] U. Feige, Relations between average case complexity and approximation complexity, in Proceedings of STOC 2002, pp. 534-543.

[19] U. Feige, private communication, 2009.

[20] A. Frieze and R. Kannan, Quick approximation to matrices and applications. Combinatorica 19 (1999), no. 2, 175–220.

[21] A. Geoffrion and G. Graves, Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach, Operations Research, 24 (1976), 596-610.

[22] R. Hassin, A. Levin and M. Sviridenko, Approximating the minimum quadratic assignment problems, ACM Transactions on Algorithms 6(1), (2009).

[23] S. Khot, Ruling out PTAS for graph min-bisection, densest subgraph and bipartite clique, In Proceedings of FOCS 2004, pp. 136145.

[24] T. C. Koopmans and M. Beckman, Assignment problems and the location of economic activities, Econometrica 25 (1957), pp. 5376.

[25] G. Laporte and H. Mercure, Balancing hydraulic turbine runners: A quadratic assignment problem, European Journal of Operations Research, 35 (1988), 378-381.

[26] E.M. Loilola, N.M.M. De Abreu, P.O. Boaventura-Netto, P.M. Hahn, and T. Querido, A survey for the quadratic assignment problem, Invited Review, European Journal of Operational Research, 176, 657-690, 2006.

[27] C. McDiarmid, Concentration. Probabilistic methods for algorithmic discrete mathematics, 195–248, Algorithms Combin., 16, Springer, Berlin, 1998.

[28] V. Nagarajan and M. Sviridenko, On the maximum quadratic assignment problem, Mathematics of Operations Research 34(4), pp. 859-868 (2009), preliminary version appeared in Proceedings of SODA 2009, pp. 516-524.

[29] P. Pardalos and H. Wolkowitz, eds., Proceedings of the DIMACS Workshop on Quadratic Assignment Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 16, 1994.

[30] C. H. Papadimitriou and M. Yannakakis, The traveling salesman problem with distances one and two, Mathematics of Operations Research 18 (1993), pp. 1 - 11.

[31] R. Raz, A Parallel Repetition Theorem, SIAM Journal on Computing, 27, 1998, pp. 763–803.

[32] M. Queyranne, Performance ratio of polynomial heuristics for triangle inequality quadratic assignment problems, Operations Research Letters, 4, 1986, 231-234.

[33] L. Steinberg, The backboard wiring problem: a placement algorithm. SIAM Rev. 3 1961 37–50.

[34] Qing Zhao, Stefan E. Karisch, Franz Rendl and Henry Wolkowicz, Semidefinite Programming Relaxations for the Quadratic Assignment Problem, Journal Journal of Combinatorial Optimization 2(1998), pp. 71-109.

A Appendix

Lemma 3.2 *Let S be a random subset of a set V. Suppose that for $u \in V$, all events $\{u' \in S\}$ where $u' \neq u$ are jointly independent of the event $\{u \in S\}$. Let s be an element of S chosen uniformly at random (if $S = \varnothing$, then s is not defined). Then* $\Pr\{u = s\} \geq \Pr\{u \in S\} / (\mathbb{E}[|S|] + 1)$.

Proof. We have

$$\Pr\{u = s\} = \Pr\{u \in S\} \times \mathbb{E}\left[\frac{1}{|S|} \quad | \quad u \in S\right].$$

By Jensen's inequality $\mathbb{E}[1/|S| \ \mid \ u \in S] \geq 1/\mathbb{E}[|S| \ \mid \ u \in S]$. Moreover,

$$\mathbb{E}[|S| \ \mid \ u \in S] = \mathbb{E}[|S \setminus \{u\}| \ \mid \ u \in S] + 1 = \mathbb{E}[|S \setminus \{u\}|] + 1 \leq \mathbb{E}[|S|] + 1.$$

□

Lemma 3.3 *Let S be a random subset of a set L, and T be a random subset of a set R. Suppose that for $(l, r) \in L \times R$, all events $\{l' \in S\}$ where $l' \neq l$ and all events $\{r' \in T\}$ where $r' \neq r$ are jointly independent of the event $\{(l, r) \in S \times T\}$. Let s be an element of S chosen uniformly at random, and let t be an element of T chosen uniformly at random. Then,*

$$\Pr\{(l, r) = (s, t)\} \geq \frac{\Pr\{(l, r) \in S \times T\}}{(\mathbb{E}[|S|] + 1) \times (\mathbb{E}[|T|] + 1)}$$

(here (s, t) is not defined if $S = \varnothing$ or $T = \varnothing$).

Proof. We have

$$\Pr\{(l, r) = (s, t)\} = \Pr\{(l, r) \in S \times T\} \times \mathbb{E}\left[\frac{1}{|S| \cdot |T|} \quad | \quad (l, r) \in S \times T\right].$$

Note, that if $(l, r) \in S \times T$, then $S \neq \varnothing$ and $T \neq \varnothing$ and hence $1/(|S| \cdot |T|)$ is well defined. By Jensen's inequality (for the convex function $t \mapsto (1/t)^2$),

$$\mathbb{E}\left[\frac{1}{|S| \cdot |T|} \quad | \quad (l, r) \in S \times T\right] =$$
$$\mathbb{E}\left[\left(\frac{1}{\sqrt{|S| \cdot |T|}}\right)^2 \quad | \quad (l, r) \in S \times T\right] \geq \left(\frac{1}{\mathbb{E}\left[\sqrt{|S| \cdot |T|} \quad | \quad (l, r) \in S \times T\right]}\right)^2.$$

Then,

$$\begin{aligned}
\mathbb{E}\left[\sqrt{|S|\cdot|T|} \quad | \quad (l,r)\in S\times T\right] &= \mathbb{E}\left[\sqrt{(|S\setminus\{l\}|+1)(|T\setminus\{r\}|+1)} \quad | \quad (l,r)\in S\times T\right] \\
&= \mathbb{E}\left[\sqrt{(|S\setminus\{l\}|+1)(|T\setminus\{r\}|+1)}\right] \\
&\leq \mathbb{E}\left[\sqrt{(|S|+1)(|T|+1)}\right] \\
&\leq \sqrt{\mathbb{E}\left[|S|+1\right]\mathbb{E}\left[|T|+1\right]},
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality. This finishes the proof. □