Fast Approximate Quadratic Assignment for (Brain) Graph Matching

Joshua T. Vogelstein, John M. Conroy, Louis J. Podrazik, Steven G. Kratzer, Donniell E. Fishkind, R. Jacob Vogelstein, and Carey E. Priebe

Abstract—The quadratic assignment problem (QAP) arises in numerous disparate applications, ranging from traveling salesman problems to various machine vision problems. We are particularly interested in a special case of QAP often called the (weighted) graph matching problem—the process of determining which permutation assigns vertices of one graph to those of another. Our work is motivated by a particular application: brain-graph matching. A brain-graph (or connectome), is a graph in which vertices correspond to (collections of) neurons, and edges are either functional or structural connections between them. Brain-graphs have between $n = \mathcal{O}(10^2)$ and $\mathcal{O}(10^{11})$ vertices, making exact graph matching algorithms computationally infeasible, even for the smallest brains. We cast our brain-graph matching problem as a quadratically constrained quadratic program. Relaxing the quadratic constraints yields a simpler quadratic problem with linear constraints. Our Fast Approximate Quadratic Assignment Problem algorithm, `FAQAP`, finds a local minimum of this problem in $\mathcal{O}(n^3)$ time, outperforming the current state-of-the-art inexact (heuristic) algorithms on a number of QAP benchmarks. Moreover, we prove that our relaxed optimization function has the same solution as the original problem in certain scenarios. Applying `FAQAP` to a synthetic *Caenorhabditis elegans* connectome problem demonstrates that brain-graph matching is much more difficult than many standard QAP benchmarks. Utilizing multiple random restarts, however, often yields optimal performance on this task with ≈ 300 vertices. Unfortunately, the computational complexity of our approach scales too poorly to hope that it can be used for our motivating application. We therefore hope to inspire further development of approximate solutions to these problems. All our code is available from on the first author's website, http://jovo.me.

Index Terms—graph theory, network theory, statistical inference, structural pattern recognition, connectome.

1 INTRODUCTION

GRAPH matching—the process of finding an optimal permutation of the vertices of one graph to align with the vertices of another—is a famously computationally daunting problem [1]. Specifically, graph matching is a kind of $\mathcal{NP}$-hard problem, that is, no known polynomial time algorithm can solve it [2]. Perhaps the most prominent special case of graph matching is the traveling salesman problem [3]. As such, performance of graph matching algorithms are usually evaluated on graphs with $\mathcal{O}(10)$ vertices, or at maximum $\mathcal{O}(100)$ (see [4] for a description of the standard set of benchmarks). Yet, it is increasingly popular to represent large data sets by a graph, and thus increasingly desirable to consider matching large graphs.

The motivating application for this work is *brain-graph matching*. A brain-graph (aka, a connectome) is a graph for which vertices represent (collections of) neurons and edges represent connections between them [5], [6]. Via Magnetic resonance (MR) imaging, one can image the whole brain and estimate connectivity across voxels, yielding a voxelwise connectome with up to $\mathcal{O}(10^6)$ vertices and $\mathcal{O}(10^9)$ edges [7]. Comparing brains is an important step for many neuroscience and neurology inference tasks. For example, it is becoming increasingly popular to diagnose neurological diseases via comparing brain images [8]. To date, however, these comparisons have largely rested on anatomical (e.g., shape) comparisons, not graph comparisons. This is despite the widely held doctrine that many psychiatric diseases are fundamentally "connectopathies", that is, disorders of the connections of the brain [9]–[12]. Currently available tests for connectopic explanation of psychiatric disorders hedge upon first choosing some number of graph invariants to compare across populations. The graph invariant approach to classifying is both theoretically and practically inferior to comparing whole graphs via matching [13].

More generally, state-of-the-art inference procedures for essentially any decision theoretic or inference task follow from constructing interpoint dissimilarity matrices [14]. Thus, we believe that graph matching of large graphs will become a fundamental subroutine of many statistical inference pipelines operating on graphs. Because the number of vertices of these graphs is so large, exact matching is intractable. Instead, we require inexact matching algorithms (also called "heuristics") that will scale polynomially or better [1]. We develop an

- *J.T. Vogelstein, D.E. Fishkind, and C.E. Priebe are with the Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218. E-mail: {joshuav,def,cep}@jhu.edu, {conroyjohnm,ljpodra,sgkratz}@gmail.com, jacob.vogelstein@jhuapl.edu*
- *J.M. Conroy, L.J. Podrazik and S.G. Kratzer are with Institute for Defense Analyses, Center for Computing Sciences, Bowie, MD 20708.*
- *R.J. Vogelstein is with the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723.*

This work was partially supported by the Research Program in Applied Neuroscience.

approach to graph matching based on a relaxation of the quadratic programming problem (QAP). Our approach is only cubic in the number of vertices, and outperforms previously proposed inexact graph matching heuristics.

2 Quadratic Assignment Problems

Quadratic assignment problems (QAPs) were first introduced in 1957 to deal with the location of indivisible economical activities. Since then, QAPs have found a dazzling array of applications and special cases, including, perhaps most famously, the traveling salesman problem [3]. Perhaps due to its complex computational properties—it is $\mathcal{NP}$-hard [15]—QAP has received widespread attention in both the mathematical graph theory and computer science communities [1]. Moreover, the potential span of applications of QAP algorithms is vast, ranging from neural coding [16] to machine vision [17].

Let $A = (a_{ik})$, $B = (b_{jl})$, and $C = (c_{ij}) \in \mathbb{R}^{n \times n}$ be real-valued matrices, $\pi\colon [n] \to [n]$ be a permutation function for n elements, $[n] = \{1, 2, \ldots, n\}$, and Π be the set of all such permutation functions. Koopmans and Beckman [18] introduced QAP in its original form:

$$\text{(KBP)} \quad \underset{\pi}{\text{minimize}} \sum_{i,j \in [n]} a_{\pi(i)\pi(j)} b_{ij} + \sum_{k \in [n]} c_{k\pi(k)} \tag{1a}$$

$$\text{subject to } \pi \in \Pi. \tag{1b}$$

Eq. (1) can be written in terms of a permutation matrix, $P = (p_{ij}) \in \mathcal{P}$, where $\mathcal{P}$ is the set of all permutation matrices:

$$\text{(QAP)} \quad \underset{P}{\text{minimize}} \quad \sum_{i,j,k,l \in [n]} a_{ik} b_{jl} p_{ij} p_{kl} + \sum_{i,j \in [n]} c_{ij} p_{ij} \tag{2a}$$

$$\text{subject to} \quad \sum_{i \in [n]} p_{ij} = 1 \, \forall i \in [n] \tag{2b}$$

$$\sum_{j \in [n]} p_{ij} = 1 \, \forall j \in [n] \tag{2c}$$

$$p_{ij} \in \{0, 1\} \, \forall i, j \in [n]. \tag{2d}$$

Eq. (2) indicates why this problem is a *quadratic* assignment problem. To make the relationship to the quadratic form more explicit, consider the following re-parameterization. Let $H = -A \otimes B$ be the Kronecker product of A and B, that is,

$$H = -\begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & \cdots & a_{nn}B \end{bmatrix}. \tag{3}$$

Then, let $x = vec(P) = (p_{11}, p_{12}, \ldots, p_{1n}, p_{21}, \ldots, p_{nn})^\mathsf{T}$, and let $f = vec(C)$. The objective function of Eq. (2) can therefore be written $x^\mathsf{T} H x + f^\mathsf{T} x$. But what about the constraints? The first two sets of constraints, Eqs. (2b) and (2c), impose that rows and columns must sum to unity. We can construct a matrix $K \in \{0, 1\}^{n^2 \times 2n}$ such that

$$K_{ij} = \begin{cases} 1 & \forall (i, j) : i \in [n] \cap j \in \{i, i+n, \ldots, i+n(n-1)\} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Thus, $Kx = \mathbf{1}$ is equivalent to Eqs. (2b) and (2c), where $\mathbf{1}$ is a column vector of ones of appropriate size. Binary constraints, such as the ones in Eq. (2d), can be achieved by enforcing the following quadratic constraint: $x_i(x_i - 1) = 0$. Together, we can rewrite Eq. (2) as a quadratically constrained quadratic program (QCQP):

$$\text{(QCQP)} \quad \underset{\pi}{\text{minimize}} \quad x^\mathsf{T} H x + f^\mathsf{T} x \tag{5a}$$

$$\text{subject to} \quad Kx = \mathbf{1} \tag{5b}$$

$$x_i(x_i - 1) = 0 \, \forall i \in [n] \tag{5c}$$

3 Graph Matching

A labeled graph $G = (\mathcal{V}, \mathcal{E})$ consists of a vertex set $\mathcal{V}$, where $|\mathcal{V}| = n$ is number of vertices, and an edge set $\mathcal{E}$, where $|\mathcal{E}| \leq n^2$. Note that we are not restricting our formulation to be directed or exclude self-loops. Given a pair of graphs, $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$, where $|\mathcal{V}_1| = |\mathcal{V}_2| = n$, let $\pi : \mathcal{V}_1 \to \mathcal{V}_2$ be a permutation function (bijection), and let Π be the set of all such permutation functions. Now consider the following two closely related problems:

- **Graph Isomorphism (GI):** Does there exist a $\pi \in \Pi$ such that $(u, v) \in \mathcal{E}_1$ if and only if $(\pi(u), \pi(v)) \in \mathcal{E}_2$.
- **Graph Matching (GMP):** Which π (if any) satisfies the above isomorphism criterion?

Both GI and GM are computationally difficult. GM is at least as hard as GI, since solving GM also solves GI, but not vice versa. It is not known whether GI is in complexity class $\mathcal{P}$ [19]. In fact, GI is one of the few problems for which, if $\mathcal{P} \neq \mathcal{NP}$, then GI might reside in an intermediate complexity class called $\mathcal{GI}$-complete. GM, however, is known to be $\mathcal{NP}$-hard. Yet, for large classes of GI and GM problems, linear or polynomial time algorithms are available [20]. Moreover, at worst, it is clear that GI is only "moderately exponential," for example, $\mathcal{O}(\exp\{n^{1/2+o(1)}\})$ [21]. Unfortunately, even when linear or polynomial time GI or GM algorithms are available for special cases of graphs, the constants are typically unbearably large. For example, if all vertices have degree less than k, there is a linear time algorithm for GI. However, the hidden constant in this algorithm is $512k^3!$ [22].

Because we are interested in solving GM for graphs with $\mathcal{O}(10^6)$ or more vertices, exact GM solutions will be computationally intractable. As such, we develop a fast inexact graph matching algorithm. Our approach is based on formulating GM as a quadratic assignment problem.

4 Graph Matching as a QAP

The weighted graph matching problem (WGMP) can be formulated as a QAP. Let A and B correspond to the adjacency matrix representations of two graphs that we desire to match, and assume that $|V(A)| = |V(B)| = n$. We therefore have the following problem:

$$
\begin{aligned}
\text{(WGMP)} \quad & \underset{\pi \in \Pi}{\operatorname{argmin}} \sum_{i,j \in [n]} (a_{\pi(i)\pi(j)} - b_{ij})^2 = && \text{(6a)}\\
& \underset{P \in \mathcal{P}}{\operatorname{argmin}} \left\| PAP^{\mathsf{T}} - B \right\|_F = && \text{(6b)}\\
& \underset{P \in \mathcal{P}}{\operatorname{argmin}}\, tr(PAP^{\mathsf{T}} - B)^{\mathsf{T}}(PAP^{\mathsf{T}} - B) = && \text{(6c)}\\
& \underset{P \in \mathcal{P}}{\operatorname{argmin}} -tr(B^{\mathsf{T}}PAP^{\mathsf{T}}). && \text{(6d)}
\end{aligned}
$$

Note that Eq. (6d) demonstrates that WGMP is indeed a QAP, although the C matrix has been dropped. Our approach follows from relaxing the quadratic/binary constraint of the QCQP formulation of QAP (5). Specifically, we relax the binary constraint to a non-negative constraint. Thus, instead of constraining our search space to permutation matrices, the feasible space becomes the doubly stochastic matrices, that is, all matrices whose rows and columns both sum to unity, and whose elements are all non-negative, $\mathcal{D} = \{P : P^{\mathsf{T}}\mathbf{1} = P\mathbf{1} = \mathbf{1}, P \succeq 0\}$, where $\succeq$ indicates an element-wise inequality. Because the equalities in Eq. (6) follow from P being a permutation matrix, relaxing the constraints for different formulations yields different optimization problems. We relax the binary constraints to non-negative constraints in the trace formulation of the QAP, Eq. (6d):

$$
\begin{aligned}
\text{(FAQAP)} \quad & \underset{P}{\text{minimize}} && -tr(B^{\mathsf{T}}PAP^{\mathsf{T}}) && \text{(7a)}\\
& \text{subject to} && P \in \mathcal{D}. && \text{(7b)}
\end{aligned}
$$

FAQAP, which stands for "Fast Approximate Quadratic Assignment Problem", is therefore a quadratic problem with linear constraints, meaning that relatively standard solvers may be employed.

Importantly, the convex hull of permutation matrices is the set of doubly stochastic matrices, implying that this is a "natural" relaxation in a very meaningful sense. Moreover, although the objective function $-tr(B^{\mathsf{T}}PAP^{\mathsf{T}})$ is quadratic, it is not necessarily convex. This follows from computing the Hessian of the objective function with respect to P:

$$\nabla^2_P = B \otimes A + B^{\mathsf{T}} \otimes A^{\mathsf{T}}, \qquad (8)$$

which is not necessarily positive definite ($\otimes$ indicates the Kronecker product). This means that the solution space will potentially be multimodal, making initialization important. With this in mind, below, we describe an algorithm to find a local optimum of FAQAP.

5 Fast Approximate Quadratic Assignment Problem Algorithm

Our algorithm, called `FAQAP`, has three components:

I. Choose an initial estimate.
II. Find a local solution to Eq. (7).
III. Project that solution onto the set of permutation matrices.

Below, we provide details for each component.

I: Find a suitable initial position $P^{(0)} \in \mathcal{D}$ While any doubly stochastic matrix would be a feasible initial point, two choices seem natural: (i) the "flat doubly stochastic matrix," $J = \mathbf{1} \cdot \mathbf{1}^{\mathsf{T}}/n$, which is the center of the feasible region, and (ii) the identity matrix, which is a permutation matrix. We elect to use the flat matrix as our default initial starting point.

II: Find a local solution to Eq. (7) As mentioned above, Eq. (7) is a quadratic problem with linear equality and boundary constraints. A number of off-the-shelf algorithms are readily available for finding local optima in such problems. We utilize the Frank-Wolfe (FW) algorithm, which is both (i) a kind of projection descent algorithm and (ii) a successive linear programming algorithm. The FW algorithm was originally designed for solving quadratic problems with linear (equality and/or inequality) constraints [23]. It later became used more generally for nonlinear programming problems [24]. Specifically, it can be used to solve optimization problems of the following form:

$$
\begin{aligned}
\text{(FWP)} \quad & \underset{x}{\text{minimize}} && f(x) && \text{(9a)}\\
& \text{subject to} && x \in \mathcal{S}, && \text{(9b)}
\end{aligned}
$$

where $\mathcal{S} \subset \mathbb{R}^d$ is a polyhedral set (a set described by linear constraints) and the function $f : \mathcal{S} \to \mathbb{R}$ is continuously differentiable. Generally, the Frank-Wolfe algorithm consists of the following steps. Given an initial position, $x^{(0)}$, recurse the following steps until some convergence criterium (e.g., computational budget) is met. Let $\widetilde{x}^{(i)}$ be the optimal solution that minimizes the first-order Taylor series approximation of f around $x^{(i)}$ such that $\widetilde{x}^{(i)} \in \mathcal{S}$. Next, let $x^{(i+1)}$ minimize $f(x)$ such that $x^{(i+1)}$ is on the line segment from $x^{(i)}$ to $\widetilde{x}^{(i)}$ in $\mathcal{S}$.

Below we provide a detailed view of applying FW to FAQAP. Let our objective function be that of Eq. (7a), $f(P) = tr(B^{\mathsf{T}}PAP^{\mathsf{T}})$. Given an initial position, $P^{(0)}$, iterate the following four steps.

Step 1: Compute the gradient $\nabla f(P^{(i)})$*:* The gradient f with respect to P is given by

$$\nabla f(P^{(i)}) = -AP^{(i)}B^{\mathsf{T}} - A^{\mathsf{T}}P^{(i)}B. \qquad (10)$$

Step 2: Compute the direction $\widetilde{P}^{(i+1)}$*:* The direction is given by the argument that minimizes a first-order Taylor series approximation to $f(P)$ around the current estimate, $P^{(i)}$. The first-order Taylor series approximation to $f(P)$ is given by

$$\widetilde{f}^{(i)}(P) \triangleq f(P^{(i)}) + \nabla f(P^{(i)})^{\mathsf{T}}(P - P^{(i)}). \qquad (11)$$

Thus, Step 2 of FW is

$$\widetilde{P}^{(i+1)} = \operatorname*{argmin}_{P \in \mathcal{D}} f(P^{(i)}) + \nabla f(P^{(i)})^\mathsf{T}(P - P^{(i)}) \tag{12a}$$

$$= \operatorname*{argmin}_{P \in \mathcal{D}} \nabla f(P^{(i)})^\mathsf{T} P. \tag{12b}$$

As it turns out, Eq. (12b) can be solved as a *Linear Assignment Problem* (LAP). The details of LAPs are well known [3], so we relegate them to the appendix. Suffice it to say here, LAPs can be solved via the "Hungarian Algorithm", named after three Hungarian mathematicians [25]–[27]. Modern variants of the Hungarian algorithm are cubic in n, that is, $\mathcal{O}(n^3)$, or even faster in the case of sparse or otherwise structured graphs [3], [28].

Step 3: Compute the step size $\alpha^{(i)}$ Given $\widetilde{P}^{(i+1)}$, the new direction is given maximizing the *original* optimization problem, Eq. (7), along the line segment from $P^{(i)}$ to $\widetilde{P}^{(i+1)}$ in $\mathcal{D}$.

$$\alpha^{(i)} = \operatorname*{argmin}_{\alpha \in [0,1]} f(P^{(i)} + \alpha^{(i)} \widetilde{P}^{(i)}). \tag{13}$$

This can be performed exactly, because f is a quadratic function.

Step 4: Update $P^{(i)}$ Finally, the new estimated doubly stochastic matrix is given by

$$P^{(i+1)} = P^{(i)} + \alpha^{(i)} \widetilde{P}^{(i+1)}. \tag{14}$$

Stopping criteria Steps 1–4 are iterated until some stopping criterion is met (computational budget limits, $P^{(i)}$ stops changing much, or $\nabla f(P^{(i)})$ is close to zero). These four steps collectively comprise the FW algorithm for solving Eq. (7).

III: Projecting onto the set of permutation matrices Let $P^{(I)}$ be the doubly stochastic matrix resulting from the final iteration. We project $P^{(I)}$ onto the set of permutation matrices, yielding

$$\hat{P} = \operatorname*{argmin}_{P \in \mathcal{P}} -\langle P^{(I)}, P \rangle, \tag{15}$$

where $\langle \cdot, \cdot \rangle$ is the usual Euclidean inner product, i.e., $\langle X, Y \rangle \overset{\triangle}{=} tr(X^\mathsf{T} Y) = \sum_{ij} x_{ij} y_{ij}$. Note that Eq. (15) is a LAP (again, see appendix for details).

6 RESULTS

6.1 FAQAP solves QAP in certain special cases

Because FAQAP relaxes the constraints of QAP, in certain important special cases, the minimum of FAQAP will be identical to the minimum of QAP. This insight leads to the following theorem:

Theorem 1. *If A and B are the adjacency matrices of simple graphs (symmetric, hollow, and binary) that are isomorphic to one another, then the minimum of FAQAP is equal to the minimum QAP.*

Proof: Because any feasible solution to QAP is also a feasible solution to FAQAP, we must only show that the optimal solution to FAQAP can be no better than the optimal solution to QAP. Let $A = PBP^\mathsf{T}$, so that $\langle A, PBP^\mathsf{T} \rangle = 2m$, where m is the number of edges in A. If FAQAP could achieve a lower objective value, then it must be that there exists a $D \in \mathcal{D}$ such that $\langle A, DBD^\mathsf{T} \rangle > \langle A, PBP^\mathsf{T} \rangle = 2m$ (remember that we are minimizing the negative Euclidean inner product). For that to be the case, it must be that $(DBD^\mathsf{T})_{ij} \geq 1$ for some (u, v). That this is not so may be seen by the submultiplicativity of the ℓ_∞ norm: $\|Px\|_\infty \leq \|P\|_\infty \|x\|_\infty$. Applying this twice (once for each permutation matrix multiplication) yields our result. □

6.2 Algorithm Complexity and leading constants

As mentioned above, GM is computationally difficult; even those special cases for which polynomial time algorithms are available, the leading constants are intractably large for all but the simplest cases. We therefore determined the average complexity of our algorithm *and* the leading constants. Figure 1 suggests that our algorithm is not just cubic in time, but also has very small leading constants ($\approx 10^{-7}$ seconds), making using this algorithm feasible for even reasonably large graphs.

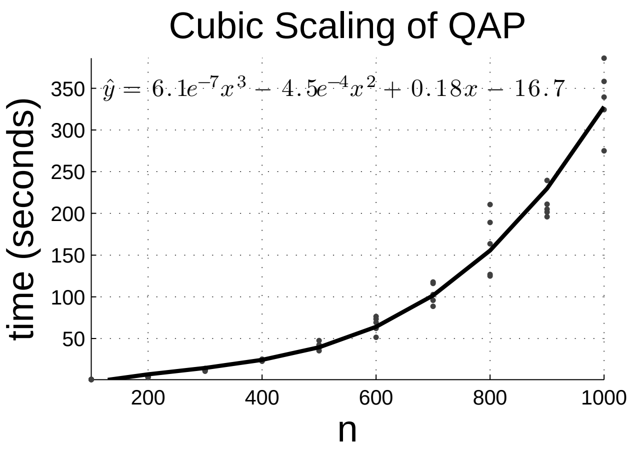

Fig. 1: Performance of FAQAP as function of number of vertices. Data was sampled from an Erdös-Rényi model with $p = log(n)/n$. Each dot represents a single simulation. The solid line is the best fit cubic function. Note the leading constant is 10^{-7} seconds.

6.3 QAP benchmarks

Having demonstrated the theoretical properties of our approach, we are also interested in assessing its computational properties in comparison with other the previous state-of-the-art. We therefore compare FAQAP to other approaches using a selection of the QAP benchmark library [4]. Specifically, [29] created a path following algorithm (PATH) based on a convex and concave relaxation of QAP. In that manuscript, the authors considered 16 datasets from the QAP benchmark, the same 16 datasets as were used in [30], which are known to be "particularly difficult". PATH was shown to outperform other state-of-the-art algorithms on 14 of 16 tests. Specifically, PATH was compared to the Quadratic Programming Bound approach (QGP) of [31], the graduated assignment algorithm (GRAD) of [32], and Umeyama's algorithm (U) [33]. Because either PATH or QBP outperformed GRAD and U on every dataset, Table 1 shows the performance of FAQAP versus PATH and QBP. FAQAP outperforms both of the previous state-of-the-art inexact cubic algorithms on 12 out of 16 benchmarks.

TABLE 1: Comparison of FAQAP with Minimum Solution and Previous State-of-the-Art. The best (lowest) solution is in **bold**. The number of vertices for each problem is the number in its name (second column).

#	Problem	Min	FAQAP	PATH	QBP
1	chr12c	11156	**13072**	18048	20306
2	chr15a	9896	27584	**19086**	26132
3	chr15c	9504	17324	**16206**	29862
4	chr20b	2298	**3068**	5560	6674
5	chr22b	6194	**8482**	8500	9942
6	esc16b	292	320	300	**296**
7	rou12	235528	**253684**	256320	278834
8	rou15	354210	**371458**	391270	381016
9	rou20	725522	**743884**	778284	804676
10	tai10a	135028	157954	**152534**	165364
11	tai15a	388214	**397376**	419224	455778
12	tai17a	491812	**529134**	530978	550862
13	tai20a	703482	**734276**	753712	799790
14	tai30a	1818146	**1894640**	1903872	1996442
15	tai35a	2422002	**2460940**	2555110	2720986
16	tai40a	3139370	**3227612**	3281830	3529402

6.4 Multiple Restarts

Although FAQAP outperformed PATH on 13 of 16 benchmarks, it was annoying to us to not be the best cubic time approximate QAP solver available. In PATH, the algorithm finds the minimum of a convex path between two extremes, F_0 and F_1. Similarly, QBP finds the minimum of a convex program. Our approach, on the other hand, does not construct a convex problem to solve, rather, it chooses an initial starting point and then finds a local optimum (note that the initial position of the PATH algorithm could also be variable, because F_0 is not convex as they assert, so their starting point depends on their initialization).

The non-convexity of our approach is both a feature and a bug. It is a bug, of course, because it means that the solution depends on the initial condition. It is a feature, however, if (i) we have some reason to believe that better solutions exist (many algorithms efficiently compute relatively tight lower bounds [34]), and (ii) we can efficiently search the space of initial conditions. Although we lack any supporting theory of optimality, we do know how to sample feasible starting points. Specifically, we desire that our starting points are "near" the flat matrix. Therefore, we can sample $K \in \mathcal{D}$, a random doubly stochastic matrix using 10 iterations of Sinkhorn balancing [35], and let our initial guess be $P^{(0)} = (J + K)/2$, where J is the doubly flat matrix. We can therefore use any number of restarts with this approach.

Table 2 shows the performance of running FAQAP 3 and 100 times, reporting only the best result, and comparing it to the best performing result from Table 1. It only required three restarts to outperform all other cubic algorithms on all the benchmarks. Moreover, after 100 restarts, FAQAP finds the absolute minimum. Note that restarting FAQAP multiple times is still cubic, although with an arbitrary number of random restarts, stating that it is cubic is somewhat meaningless.

TABLE 2: Comparison of FAQAP with Minimum Solution and the best result from Table 1, called BEST. Benchmarks for which 100 restarts of FAQAP found the global minimum are **bold**.

#	Problem	Min	FAQAP_{100}	FAQAP_3	BEST
1	chr12c	11156	12176	13072	13072
2	chr15a	9896	**9896**	17272	19086
3	chr15c	9504	10960	14274	16206
4	chr20b	2298	2786	3068	3068
5	chr22b	6194	7218	7876	8482
6	esc16b	292	**292**	294	296
7	rou12	235528	**235528**	238134	253684
8	rou15	354210	356654	371458	371458
9	rou20	725522	730614	743884	743884
10	tai10a	135028	135828	148970	152534
11	tai15a	388214	391522	397376	397376
12	tai17a	491812	496598	511574	529134
13	tai20a	703482	711840	721540	734276
14	tai30a	1818146	1844636	1890738	1894640
15	tai35a	2422002	2454292	2460940	2460940
16	tai40a	3139370	3187738	3194826	3227612

6.5 Brain-Graph Matching

A "connectome" is a brain-graph in which vertices correspond to (collections of) neurons, and edges correspond to connections between them. The *Caenorhabditis elegans* (*C. elegans*) is a small worm (nematode) with 302 labeled vertices. We consider the subgraph with 279 somatic neurons that form edges with other neurons [36], [37]. Two distinct kinds of edges exist between vertices: chemical and electrical "synapses" (edges). Any pair of vertices may have several edges of each type. Moreover, some of the synapses are hyper-edges amongst more than two vertices. Thus, the connectome of a *C. elegans* may be thought of as a weighted multi-hypergraph, where the weights are the number of edges of each type. FAQAP natively operates on weighted or unweighted graphs. We therefore conducted the following synthetic experiments.

Let $A_{ij;z} \in \{0, 1, 2, \dots\}$ be the number of synapses from neuron i to neuron j of type z (either chemical c or electrical e), and let $A_z = \{A_{ij;z}\}_{i,j\in[279]}$ for $z \in \{e, c\}$ correspond to the electrical or chemical connectome. To generate synthetic data, we let $B_z^{(k)} = Q_z^{(k)} A_z {Q_z^{(k)}}^\mathsf{T}$, for some $Q_z^{(k)}$ chosen uniformly at random from $\mathcal{P}$, effectively shuffling the vertex labels of the connectome.

TABLE 3: Brain-graph matching summary statistics for both the chemical and electrical connectome.

	unit	chemical	electrical
Accuracy	%	100 (0)	59 (0.30)
Restarts	#	3 (0)	25 (6.7)
Solution Time	sec.	42 (0.42)	79 (20)

Then, we tried to graph match A_z to $B_z^{(k)}$, for $z \in \{e, c\}$ and for $k \in [10]$, that is, we repeat the experiment 10 times. We define accuracy by the value of our objective function, $f(P_z^{(k)})$. To evaluate the impact of multiple restarts, for both connectomes, we restarted `FAQAP` up to 30 times. Specifically, our stopping criteria on the number of restarts was either (i) perfect assignment or (ii) 30 restarts.

Table 1 shows the mean (standard deviation) of accuracy, number of restarts, and solution time for both connectomes. For the chemical connectome, `FAQAP` always found the optimal solution. For the electrical one, the mean number of restarts was < 30 times, meaning that `FAQAP` sometimes found the optimal solution. The properties of these connectomes are analyzed in [37], but it remains unclear to us why the chemical connectome was so much easier to graph match than the electrical one. Both ran very quickly, only requiring tens of seconds. Note that the number of vertices in this brain-graph matching problem is about an order of magnitude larger than the largest of the 16 benchmarks used above.

To investigate the performance of `FAQAP` on undirected graphs, we repeated the above analysis using binarized symmeterized versions of the graphs ($A_{ij;z} = 1$ if and only if $A_{ij;z} \geq 1$ or $A_{ji;z} \geq 1$). The resulting summary statistics are nearly identical to those presented in Table 1, although speed increased by greater than a factor of two.

7 Discussion

This work presents a fast approximate quadratic assignment problem algorithm `FAQAP` for approximately solving large (brain-) graph matching problems. Our key insight was to relax the binary constraint of the QCQP to its continuous and non-negative counterpart—the doubly stochastic matrix—which is the convex hull of the original feasible region. We then demonstrate that the solution to our relaxed optimization problem, FAQAP, is identical to that for QAP whenever the two graphs are simple and isomorphic to one another. Numerically, we demonstrated that not only is `FAQAP` cubic in time, but also its leading constants are quite small. Moreover, it achieves better performance than previous state-of-the-art cubic-time algorithms on 12 of the 16 standard QAP benchmarks. Finally, it solved a brain-graph matching problem, which has an order of magnitude more vertices than any of the 16 QAP benchmarks.

Fortunately, our work is not done. Even with very small leading constants for this algorithm, as n increases, the computational burden gets quite high. For example, extrapolating the curve of Figure 1, this algorithm would take about 2 years to finish (on a standard laptop from 2011) when $n = 20,000$. We hope to be able to approximately solve FAQAP on graphs much larger than that, given that the number of neurons in even a fly brain, for example, is $\mathcal{O}(10^5)$. Therefore, more efficient implementations are of interest.

Additional future work might generalize `FAQAP` in a number of ways. First, many (brain) graphs of interest will be errorfully observed [40], that is, vertices might be missing and putative edges might exhibit both false positives and false negatives. Explicitly dealing with this error source is both theoretically and practically of interest [13]. Second, for many brain-graph matching problems, the number of vertices will not be the same across the brains. Recent work from [29], [38] and [39] suggest that extensions in this direction would be both relatively straightforward and effective. Third, the most "costly" subroutine is LAP. Fortunately, LAP is a quadratic optimization problem with linear constraints. A number of parallelized optimization strategies could therefore potentially be brought to bear on this problem [41]. Fourth, our matrices have certain special properties, namely sparsity, which makes more efficient algorithms (such as "active set" algorithms) readily available for further speed increases. Fourth, for brain-graphs, we have some prior information that could easily be incorporated in the form of vertex attributes. For example, position in the brain, cell type, etc., could be used to measure "dissimilarity" between vertices. The WGMP could easily incorporate these dissimilarities, in fact, the original QAP formulation already encodes them via the matrix C; that matrix was simply dropped when WGMP was originally proposed. Finally, although this approach natively operates on both unweighted and weighted graphs, multi-graphs are a possible extension.

In conclusion, this manuscript has presented an algorithm for approximately solving the quadratic assignment problem that is fast, effective, and easily generalizable. Yet, the $\mathcal{O}(n^3)$ complexity remains too slow to actually solve our problems of interest. To facilitate further development and applications, all the code and data used in this manuscript is available from the first author's website, http://jovo.me.

Appendix

The standard way of writing a Linear Assignment Problem (LAP) is

$$\text{(LAP)} \quad \underset{\pi}{\text{minimize}} \sum_{u,v \in [n]} a_{u\pi(v)} b_{ij} \tag{16a}$$

$$\text{subject to } \pi \in \Pi, \tag{16b}$$

which can be written equivalently in a number of ways using the notion of permutation matrix introduced above:

$$\underset{P\in\mathcal{P}}{\operatorname{argmin}} \|PA-B\|_F = \tag{17a}$$

$$\underset{P\in\mathcal{P}}{\operatorname{argmin}}\, tr(PA-B)^{\mathsf{T}}(PA-B) = \tag{17b}$$

$$\underset{P\in\mathcal{P}}{\operatorname{argmin}} -tr(PAB^{\mathsf{T}}) = \underset{P\in\mathcal{P}}{\operatorname{argmin}} -\langle P^{\mathsf{T}}, AB^{\mathsf{T}}\rangle = \tag{17c}$$

$$\underset{P\in\mathcal{P}}{\operatorname{argmin}} -\langle P, AB^{\mathsf{T}}\rangle, \tag{17d}$$

where $\langle\cdot,\cdot\rangle$ is the usual Euclidean inner product, i.e., $\langle X,Y\rangle \overset{\triangle}{=} tr(X^{\mathsf{T}}Y) = \sum_{ij} x_{ij}y_{ij}$. While the objective function and the first two constraints of LAP are linear, the binary constraints make solving even this problem computationally tricky. Nonetheless, in the last several decades, there has been much progress in accelerating algorithms for solving LAPs, starting with exponential time, all the way down to $\mathcal{O}(n^3)$ for general LAPs, and even faster for certain special cases (e.g., sparse matrices) [3], [28].

That Eq. (12b) is a LAP is evident by considering Eq. (17d). If $A = \nabla_P^{(i)}$ and $B = I$ (the $n\times n$ identity matrix), then Eq. (12b) is identical to Eq. (17d).

To solve a LAP, consider a continuous relaxation of LAP, specifically, relaxing the permutation matrix constraint to a doubly stochastic matrix constraint:

$$\text{(rLAP)} \quad \underset{P}{\text{minimize}} \quad -\langle P, AB^{\mathsf{T}}\rangle \tag{18a}$$

$$\text{subject to} \quad P\in\mathcal{D}. \tag{18b}$$

As it turns out, solving rLAP is equivalent to solving LAP.

Proposition 1. *LAP and rLAP are equivalent, meaning that they have the same optimal objective function value.*

Proof: Although this proposition is typically proven by invoking total unimodularity, we present a simpler proof here. Let P' be a solution to LAP and let $P = \sum_{i\in[k]} \alpha_i P^{(i)}$ be a solution to rLAP for some positive integer k, permutation matrices $\{P^{(i)}\}_{i\in[k]}$, and positive real numbers $\{\alpha_i\}_{i\in[k]}$ such that $\sum_{i\in[k]}\alpha_i = 1$. Note that

$$\langle P, AB^{\mathsf{T}}\rangle = \langle \sum_{i\in[k]} \alpha_i P^{(i)}, AB^{\mathsf{T}}\rangle = \sum_{i\in[k]} \alpha_i \langle P^{(i)}, AB^{\mathsf{T}}\rangle$$
$$\le \sum_{i\in[k]} \alpha_i \langle P', AB^{\mathsf{T}}\rangle = \langle P', AB^{\mathsf{T}}\rangle \le \langle P, AB^{\mathsf{T}}\rangle,$$

because P' is feasible in rLAP. □

This relaxation motivates our approach to approximating QAP.

Acknowledgments

The authors would like to acknowledge two helpful reviewers as well as Lav Varshney for providing the data.

References

[1] D. Conte, P. Foggia, C. Sansone, and M. Vento, "THIRTY YEARS OF GRAPH MATCHING," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.

[2] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998. [Online]. Available: http://books.google.com/books?hl=en\&amp;lr=\&amp;id=u1RmDoJqkF4C\&amp;oi=fnd\&amp;pg=PR11\&amp;dq=Combinatorial+optimization:+algorithms+and+complexity\&amp;ots=S8xMKZ2Xyb\&amp;sig=_bloCXwzNDmQS7XEt3oYM9zVAHc

[3] R. E. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. SIAM, 2009. [Online]. Available: http://books.google.com/books?id=nHIzbApLOr0C\&pgis=1

[4] R. E. Burkard, S. E. Karisch, and F. Rendl, "QAPLIB A Quadratic Assignment Problem Library," *Journal of Global Optimization*, vol. 10, no. 4, pp. 391–403, 1997. [Online]. Available: http://www.springerlink.com/content/n5468q3g33184443/fulltext.pdf

[5] O. Sporns, G. Tononi, and R. Kotter, "The Human Connectome: A Structural Description of the Human Brain," *PLoS Computational Biology*, vol. 1, no. 4, p. e42, 2005.

[6] P. Hagmann, "From diffusion MRI to brain connectomics," Ph.D. dissertation, Institut de traitement des signaux, 2005.

[7] X.-N. Zuo, R. Ehmke, M. Mennes, D. Imperati, F. X. Castellanos, O. Sporns, and M. P. Milham, "Network Centrality in the Human Functional Connectome." *Cerebral cortex (New York, N.Y. : 1991)*, pp. bhr269–, Oct. 2011. [Online]. Available: http://cercor.oxfordjournals.org/cgi/content/abstract/bhr269v2

[8] J. G. Csernansky, L. Wang, S. C. Joshi, J. T. Ratnanather, and M. I. Miller, "Computational anatomy and neuropsychiatric disease: probabilistic assessment of variation and statistical inference of group difference, hemispheric asymmetry, and time-dependent change." *NeuroImage*, vol. 23 Suppl 1, pp. S56–68, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.neuroimage.2004.07.025

[9] M. Kubicki, R. McCarley, C.-F. Westin, H.-J. Park, S. Maier, R. Kikinis, F. A. Jolesz, and M. E. Shenton, "A review of diffusion tensor imaging studies in schizophrenia." *Journal of Psychiatric Research*, vol. 41, no. 1-2, pp. 15–30, 2007. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2768134\&tool=pmcentrez\&rendertype=abstract

[10] V. D. Calhoun, J. Sui, K. Kiehl, J. Turner, E. Allen, and G. Pearlson, "Exploring the psychosis functional connectome: aberrant intrinsic networks in schizophrenia and bipolar disorder." *Frontiers in psychiatry / Frontiers Research Foundation*, vol. 2, p. 75, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3254121\&tool=pmcentrez\&rendertype=abstract

[11] A. Fornito and E. T. Bullmore, "Connectomic intermediate phenotypes for psychiatric disorders." *Frontiers in psychiatry / Frontiers Research Foundation*, vol. 3, p. 32, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3329878\&tool=pmcentrez\&rendertype=abstract

[12] A. Fornito, A. Zalesky, C. Pantelis, and E. T. Bullmore, "Schizophrenia, neuroimaging and connectomics," *NeuroImage*, Feb. 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22387165

[13] J. T. Vogelstein and C. E. Priebe, "Shuffled Graph Classification: Theory and Connectome Applications," *Submitted to IEEE PAMI*, 2011.

[14] R. P. W. Duin, E. Pkalska, and E. Pkalskab, "The dissimilarity space: Bridging structural and statistical pattern recognition," *Pattern Recognition Letters*, vol. in press, no. April, May 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167865511001322http://www.sciencedirect.com/science/article/pii/S0167865511001322http://dx.doi.org/10.1016/j.patrec.2011.04.019

[15] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[16] J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville, "Decoding brain states from fMRI connectivity graphs." *NeuroImage*, Jun. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20541019

[17] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19,

no. 7, pp. 775–779, Jul. 1997. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=598235

[18] T. C. Koopmans and M. Beckmann, "Assignment Problems and the Location of Economic Activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, 1957.

[19] S. Fortin, "The Graph Isomorphism Problem," *Technical Report, University of Alberta, Dept of CS*, 1996. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.9419

[20] L. Babai, P. Erds, and S. M. Selkow, "RANDOM GRAPH ISOMORPHISM * 1 . A straightforward algorithm . The problem of testing graphs for isomorphism belongs to those combinatorial search problems for which no polynomial-time algorithm is available as yet . It is , however , striking , that even t," *Society*, no. August, 1980.

[21] L. Babali, "Moderately Exponential Bound for Graph Isomorphism," pp. 34–50, Aug. 1981. [Online]. Available: http://portal.acm.org/citation.cfm?id=647890.739270

[22] J. Chen, "A Linear-Time Algorithm for Isomorphism of Graphs of Bounded Average Genus," *SIAM Journal on Discrete Mathematics*, vol. 7, no. 4, p. 614, Nov. 1994. [Online]. Available: http://portal.acm.org/citation.cfm?id=197033.197062

[23] M. Frank and P. Wolfe, "An Algorithm for Quadratic Programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956. [Online]. Available: http://www.ams.org/mathscinet/search/publications.html?pg1=MR\&s1=MR0089102

[24] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied Mathematical Programming*. Addison-Wesley, 1977. [Online]. Available: http://www.amazon.com/Applied-Mathematical-Programming-Stephen-Bradley/dp/020100464X

[25] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955. [Online]. Available: http://doi.wiley.com/10.1002/nav.3800020109

[26] D. Knig, "Gráfok és Mátrixok," *Matematikai és Fizikai Lapok*, vol. 38, pp. 116–119, 1931.

[27] J. Egeváry, "Matrixok kombinatorius tulajdonságairól," *Matematikai és Fizikai Lapok*, no. 38, pp. 16–28, 1931.

[28] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987. [Online]. Available: http://www.springerlink.com/index/7003M6N54004M004.pdf

[29] M. Zaslavskiy, F. R. Bach, and J.-p. P. Vert, "A path following algorithm for the graph matching problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2227–2242, 2009. [Online]. Available: http://eprints.pascal-network.org/archive/00004435/

[30] C. Schellewald, S. Roth, and C. Schnörr, "Evaluation of Convex Optimization Techniques for the Weighted Graph-Matching Problem in Computer Vision," in *Proceedings of the 23rd DAGMSymposium on Pattern Recognition*. Springer-Verlag, 2001, pp. 361–368.

[31] K. M. Anstreicher and N. W. Brixius, "A new bound for the quadratic assignment problem based on convex quadratic programming," *Mathematical Programming*, vol. 89, no. 3, pp. 341–357, Feb. 2001. [Online]. Available: http://www.springerlink.com/content/uvavn88ke7818djg/

[32] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377–388, Apr. 1996. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=491619

[33] S. Umeyama, "An Eigendecomposition Approach to Weighted Graph Matching Problems," *Analysis*, vol. I, no. 5, 1988.

[34] K. M. Anstreicher, "Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming," *October*, pp. 471–484, 2009.

[35] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, 1964. [Online]. Available: http://www.jstor.org/stable/2238545

[36] J. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans." *Philosophical Transactions of Royal Society London. Series B, Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986.

[37] L. R. Varshney, B. L. Chen, E. Paniagua, D. H. Hall, D. B. Chklovskii, C. Spring, and J. Farm, "Structural Properties of the Caenorhabditis elegans Neuronal Network," *World Wide Web Internet And Web Information Systems*, vol. 7, no. 2, pp. 1–41, Feb. 2011.

[38] M. Zaslavskiy, F. R. Bach, and J.-p. Vert, "Many-to-Many Graph Matching :," pp. 515–530, 2010.

[39] F. Escolano, E. Hancock, and M. Lozano, "Graph Matching through Entropic Manifold Alignment," 1980.

[40] C. E. Priebe, J. T. Vogelstein, and D. D. Bock, "Optimizing the quantity/quality trade-off in connectome inference," *Communications in Statistics Theory and Methods*, p. 7, 2011. [Online]. Available: http://arxiv.org/abs/1108.6271

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Foundations and Trends in Machine Learning," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: http://www.nowpublishers.com/product.aspx?product=MAL\&doi=2200000016