Matched Filters for Noisy Induced Subgraph Detection

Daniel L. Sussman
Boston University, Department of Mathematics and Statistics
Boston, MA, USA
sussman@bu.edu

Vince Lyzinski
University of Massachusetts Amherst, Department of Mathematics and Statistics
Amherst, MA, USA
lyzinski@math.umass.edu

Youngser Park
Carey E. Priebe
Johns Hopkins University, Department of Applied Mathematics and Statistics
Baltimore, MD, USA

ABSTRACT

We consider the problem of finding the correspondence between two graphs with different sizes where the small graph is still large. We propose using graph matching methodology and padding the smaller matrix in different ways. We show that under a statistical model for correlated pairs of graphs, the resulting optimizations problems can be guaranteed to perform well, though there are currently no fast algorithms to solve these problems. We also consider an algorithm that exploits a partially known correspondence and show via simulations and applications to the Drosophila connectome that in practice this algorithm can achieve good performance using random restarts.

CCS CONCEPTS

• **Mathematics of computing → Random graphs**; **Graph algorithms**; **Probability and statistics**;

KEYWORDS

multiple graph inference, subgraph detection, graph matching

ACM Reference Format:

Daniel L. Sussman, Vince Lyzinski, Youngser Park, and Carey E. Priebe. 2018. Matched Filters for Noisy Induced Subgraph Detection. In *Proceedings of GTA^3 2018: Workshop on Graph Techniques for Adversarial Activity Analytics (GTA^3 2018).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In many settings, we often want to quantify how multiple networks relate to each other in order to study how actors jointly use these networks. This may arise from multiple modalities, such as communications networks, delivery networks, financial networks, and social networks, or from a time dynamic setting. Similarly, in neuroscience or biology we may seek to compare brain networks or protein networks of different individuals or species. Often these networks are on different unmatched sets of vertices that are not the same size. This limits the set of available tools as the adjacency matrices can not be directly compared.

In a related fashion, we often want to detect and locate the presence of particular induced subgraphs that correspond to a given activity or structure of interest. When the induced graphs are small, with up to ≈10−20 vertices, there are a number of approaches including color coding [Zhao et al. 2010], backtracking algorithms [Kuramochi and Karypis 2001]. However, it may be the case that these subgraphs may contain > 20 vertices in which case many existing approaches will either fail to find the subgraphs of interest or be computationally intractable. Furthermore, we often expect the subgraphs may not appear exactly in the graph but rather only approximately, due to errors in one or both of the graphs, so finding an exact subgraph might not always be possible. A more challenging problem that we will not consider is to detect anomalous subgraphs within a collection of graphs [Akoglu et al. 2015].

Herein, we will use the machinery of graph matching to construct a matched filter to detect (potentially errorful) subgraphs of interest within a larger network. In the simplest setting, when the two graphs adjacency matrices $A, B \in \{0,1\}^{n\times n}$ are of equal size, the graph matching problem (GMP) seeks the permutation matrix

$$\mathrm{argmin}_{P\in\mathcal{P}} \|A - PBP^T\| = \mathrm{argmax}_{P\in\mathcal{P}} \mathrm{tr}(APBP^T). \qquad (1)$$

While this problem is NP-hard in general, there are numerous approaches in the literature that efficiently approximately solve the GMP; see [Conte et al. 2004; Emmert-Streib et al. 2016] for a review of the prescient literature. In particular, when prior knowledge about the correspondence between vertices the can be incorporated into the algorithms, the GMP can be approximately solved efficiently for graphs with more than 10^5 vertices [Lyzinski et al. 2015; Yartseva and Grossglauser 2013] without the need for sophisticated modern parallel computing to be brought to bear.

Our main goals in this manuscript are to investigate the theoretical limitations of noisy graph matching when the graphs may be of very different sizes; as is often the case in the noisy subgraph detection framework. To match graphs of radically different sizes in Eq. (1), we consider a number of padding schemes for the smaller matrix to render graph matching an appropriate tool for the problem [Fishkind et al. 2017]. Under a statistical model for noisily implanting an induced subgraph into a larger network, we show that the true induced subgraph will be found by an oracle GM algorithm using an appropriate padding scheme provided the correlations and probabilities satisfy certain mild model assumptions. We further demonstrate the effectiveness of these strategies when the vertex correspondence is partially known.

2 BACKGROUND FOR GRAPH MATCHING

In this section we provide a brief background on graph matching, some methods to incorporate prior information, and a statistical

GTA^3 2018, February 9, Marina Del Rey, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

model for correlated graphs. Throughout the remainder of this article we will use the following notation. Let $[n] = \{1, 2, \ldots, n\}$. Let $\mathcal{P}_n$ and $\mathcal{D}_n$ denote the set of $n \times n$ permutation matrices and doubly stochastic matrices, respectively. Let $\mathbf{J}_n$ and $\mathbf{0}_n$ denote the $n \times n$ all ones and all zeros matrices, respectively. Let $\mathcal{A}_n$ denote the set of adjacency matrices corresponding to simple undirected graphs. When clear from context, we may omit subscripts. Finally, let $\oplus$ denote the direct sum of two matrices.

2.1 Algorithms

Solving the GMP problem is very challenging but there are a number of approaches which have shown promise [Conte et al. 2004; Emmert-Streib et al. 2016]. Some approaches rely on tree based methods for exact branch and bound algorithms for integer programs. For larger graphs, the constraints for GMP are often relaxed so that continuous optimization machinery can be brought to bear on the graph matching problem; see, for example, [Fiori et al. 2013; Vogelstein et al. 2014; Zaslavskiy et al. 2009]. The relaxed solutions are then projected back onto $\mathcal{P}$ yielding an approximate solution to the graph matching problem. Some relaxations involve applying spectral methods which allows the application of fast linear algebra techniques [Egozi et al. 2013]. For the computational experiments in this manuscript we will rely on a principled indefinite relaxation of the GMP constraints [Lyzinski et al. 2016] to the set of doubly stochastic matrices $\mathcal{D}$, the convex hull of $\mathcal{P}$. Details are discussed in Section 3.2.

Frequently, these approaches can exploit seeds, a partial list of known correspondences between the two vertex sets. When sufficiently many seeds are present, these algorithms, which often have few guarantees, can be solved efficiently and the resulting match can be guaranteed to be correct [Fishkind et al. 2017; Lyzinski et al. 2014] asymptotically almost surely for relatively general random graph models. While the theory we discuss below does not require seeds, our algorithms will use seeds and subsequent algorithmic performance relies heavily on the number of seeds.

2.2 Statistical Models

In order to understand the applicability and limitations of a graph matching approach for subgraph detections, we will analyze the problem from a statistical viewpoint, situating our approach in the correlated heterogeneous Erdős-Rényi model [Lyzinski et al. 2016]. The following definition provides a distribution for pairs of random graphs with different sizes where there is a positive correlation between corresponding edge-pairs. Without loss of generality, we assume that the smaller matrix is A and that the corresponding vertices to A in B are the first $n_c \le n$ vertices.

Definition 2.1 (Correlated Erdős-Rényi). Suppose $\Lambda \in [0,1]^{n \times n}$ and $R \in [0,1]^{n_c \times n_c}$ for $0 < n_c \le n$. Denote by Λ^c, the order n_c principal submatrix of Λ. A pair of adjacency matrices $(A, B) \sim \mathrm{CorrER}(\Lambda, R)$ if $A \in \mathcal{A}_{n_c}$, $B \in \mathcal{A}_n$, for each $u < v$, B_{uv} are independent with $B_{uv} \sim \mathrm{Bernoulli}(\Lambda_{uv})$ and for $u < v \le n_c$, A_{uv} are independent with $A_{uv} \sim \mathrm{Bernoulli}(\Lambda_{uv})$. Additionally, the Bu, v's and $A_{u',v'}$'s are mutually independent except that for $u, v \in [n_c]$, $u < v$, it holds that the Pearson correlation $\mathrm{corr}(A_{uv}, B_{uv}) = R_{uv}$.

When $n_c = n$, it can be shown that the solution to the GMP will asymptotically almost surely yield the correct vertex correspondence, i.e., the only element in the argmin in 1 is the identity matrix I [Lyzinski et al. 2016, 2014].

3 PADDING APPROACHES

In order to match pairs of nodes with differing numbers of vertices we propose to pad the smaller matrix with enough rows and columns to match the size of larger matrix. We will consider a trio of padding schemes which will result in differing qualities for the resulting match [Fishkind et al. 2017].

Naive Padding The naive padding scheme is to let $\tilde{A} = A \oplus \mathbf{0}_{n_j}$ and to match $\tilde{A}$ and B.

Centered Padding The centered padding scheme is to let $\tilde{A} = (2A - \mathbf{J}) \oplus \mathbf{0}_{n_j}$, let $\tilde{B} = 2B - \mathbf{J}$, and match $\tilde{A}$ and $\tilde{B}$.

Oracle Padding The oracle padding scheme is to let $\tilde{A} = (A - \Lambda_{n_c}) \oplus \mathbf{0}$, let $\tilde{B} = B - \Lambda$, and match $\tilde{A}$ and $\tilde{B}$.

As we will see in the next section, the naive padding scheme—which finds the best fitting subgraph of B to match with A—will not be guaranteed to find the true correspondence between nodes, while the other padding schemes, the centered padding—which finds the best fitting induced subgraph of B to match with A—and the oracle padding scheme are guaranteed to succeed under mild model conditions, even in the presence of an exponentially small (in terms of the size of B) subgraph A. In general the oracle padding scheme will be inaccessible as Λ is unknown, but using various methods to estimate Λ [Chatterjee 2014; Davenport et al. 2014], we can approximate Λ in ways that can improve matching performance.

3.1 Theory

For each of the padding scenarios, we will consider n_c and n as tending to ∞ in order to understand the ability of these optimization programs to detect progressively larger subgraphs. Note that we will require that the number of vertices n_c is growing with n, as if n_c is fixed and n grows then eventually every subgraph of size n_c will appear as an induced subgraph in B multiple times just by chance. For each padding scenario let $P^* \in \{0,1\}^{n_c \times n_c}$ denote the order n_c principal submatrix of the solution to the corresponding graph matching problem. The proofs of these theorems can be found in [Lyzinski and Sussman 2017].

The first theorem we present is a negative result that shows that under weak assumptions on Λ^c, one can construct a Λ under which the naive padding scheme is almost surely guaranteed to not detect the errorful version of A in B.

Theorem 3.1. *Let $A, B \sim \mathrm{CorrER}(\Lambda, R)$ with $R \in [0,1]^{n_c \times n_c}$ and $\Lambda \in [0,1]^{n \times n}$. If $2n_c < n$ and there exists constants $\rho, \beta > 0$ such that (entry-wise) $\Lambda^c \le \beta < 1$, $\Lambda^c = \omega(n_c^{-1} \log n_c)$, and $R \le \rho < 1$, then there exists a choice of Λ such that using the naive padding scheme,*

$$\mathbb{P}[P^* \ne I] = 1 - o(1). \tag{2}$$

This occurs due to the fact that the naive padding scheme finds the best matching subgraph, rather than the best matching induced subgraph, since there is no penalty for matching non-edges in A to edges in B. Hence, if B has a dense substructure of size n_c which does not correspond to A then the naive padding scheme will match A to that dense structure, regardless of the underlying correlation.

On the other hand, the centered padding scheme can be guaranteed to result in the correct detection of the subgraph even when the number of vertices in B is exponentially larger than the number of vertices in A provided $R > 1/2 + \epsilon$.

Theorem 3.2. *Suppose that $A, B \sim \mathrm{CorrER}(\Lambda, R)$ with $R_{uv} \in [1/2 + \epsilon, 1]$ and $\Lambda_{uv} \in [\alpha, 1 - \alpha]$ for $\alpha, \epsilon \in (0, 1/2)$. It holds that $\frac{\log(n)}{\epsilon^2 n_c \alpha(1-\alpha)^2} = o(1)$ implies that using the centered padding scheme*

$$\mathbb{P}[P^* \neq I] \leq 2 \exp\left\{-\Theta(\epsilon^2 n_c \alpha(1 - \alpha)^2)\right\}.$$

Hence, for this theorem we only require that the correlations are sufficiently large to guarantee that large subgraphs of logarithmic size can be found via an oracle GM algorithm.

Finally, while the oracle padding is inaccessible for general Λ, it represents the optimal padding scheme as it eliminates any empirical correlations introduced by Λ leaving only the theoretical correlations from R.

Theorem 3.3. *Suppose that $A, B \sim \mathrm{CorrER}(\Lambda, R)$ with $R_{uv} \in [\rho, 1]$ and $\Lambda_{uv} \in [\alpha, 1 - \alpha]$, for some $\alpha, \rho \in (0, 1)$. It holds that $\frac{\log(n)}{n_c \rho \alpha(1-\alpha)^2} = o(1)$ implies that using the oracle padding scheme*

$$\mathbb{P}[P^* \neq I] \leq 2 \exp\left\{-\Theta(n_c \rho^2 \alpha(1 - \alpha)^2)\right\}.$$

3.2 Computation

Our approach to solve the graph matching problem in this setting will be analogous to the approach describe for graphs of equal sizes. In particular, we will relax the constraints GMP from $\mathcal{P}$ to $\mathcal{D}$ and use gradient ascent starting at a given D_0. We will also incorporate seeds, which without loss of generality we will assume are the first s nodes. This gradient ascent approach is then given by Algorithm 1.

Algorithm 1: FAQ Algorithm [Vogelstein et al. 2014]

Data: $A, B \in \mathcal{A}$, $D^0 \in \mathcal{D}$, $k = 0$
while *not converged* **do**
1 $P_k \leftarrow \mathrm{argmax}_{P \in \mathcal{P}} \mathrm{tr}(\tilde{A}_{\mathfrak{nn}} D_k \tilde{B}_{\mathfrak{nn}} P) + 2\mathrm{tr}(A_{\mathfrak{sn}} P B_{\mathfrak{ns}})$;
2 $\alpha_k \leftarrow \mathrm{argmax}_{\alpha \in [0,1]} \mathrm{tr}(\tilde{A}_{\mathfrak{nn}} D_\alpha \tilde{B}_{\mathfrak{nn}} D_\alpha) + 2\mathrm{tr}(A_{\mathfrak{sn}} D_\alpha B_{\mathfrak{ns}})$, where $D_\alpha = \alpha D_k + (1 - \alpha) P_k$;
3 $D_{k+1} \leftarrow D_\alpha$ and $k \leftarrow k + 1$;
end
4 Project D_k onto $\mathcal{P}$;

Note that using any of the padding schemes, we do not need to store or compute the entire matrices D_k or P_k as we only need know their first n_c rows in order to compute the objectives described above. Hence, lines 1 and 4 can be simplified and accomplished by searching over the set $n_c \times n$ matrices corresponding to injections from $[n_c]$ to $[n]$, or equivalently the first n_c rows of permutation matrices. In this way, lines 1 and 4 can be solved effectively by variants of the Hungarian algorithm for non-square matrices [Munkres 1957]. Line 2 is a quadratic equation in α and is easily solved.

Note the convergence criterion is generally easy to check as the optimal doubly stochastic matrix is frequently itself a permutation

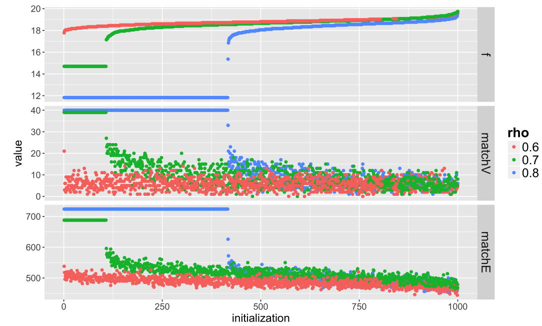

Figure 1: Subgraph detection in the $\mathrm{CorrER}(\lambda \mathbf{J}_n, \rho \mathbf{J}_{n_c})$ with $\lambda = 0.5$ and $\rho = 0.6, 0.7, 0.8$. We consider $n = 500$, $n_c = 40$, and $M = 1000$. In the top panel of the figure we plot the GMP objective function for each initialization (note that we ordered the random restarts in the figure to be monotone in f); in the middle we plot the number of vertices correctly matched; in the bottom panel we plot the number of edges correctly matched. In red we plot $\rho = 0.5$, in green $\rho = 0.7$, and in blue $\rho = 0.8$. In all cases, we consider $s = 7$ seeds to initialize the FAQ algorithm.

matrix, which also means the final projection step can be omitted. While this algorithm is not guaranteed to converge to a global optimum, if there are enough seeds or if the matrix D^0 is sufficiently close to the identity, the local maximum which this procedure converges to will be the identity.

4 EXPERIMENTS

In this section we demonstrate the effectiveness of a graph matching matched filter for errorful subgraph detection in both synthetic and real data settings. Our matched filter algorithm proceeds as follow: We initialize the FAQ algorithm at a random start point which initially aligns the seeded vertices across graphs. Repeating this process M times, we output M potential matches for the subgraph A in B.

Algorithm 2: GMMF

Data: Template A, network B, seeded vertex sets $\{S_1, S_2, \cdots, S_M\}$, number of MC replicates M
Result: Matchings $\{B_1, B_2, \cdots, B_M\}$
for $i \leftarrow 1$ **to** M **do**
1: Generate a random doubly stochastic matrix D;
2: Initialize FAQ at D with soft seed set S_i; match A to B;
3: FAQ output is B_i, subgraph of B matched to A
end

4.1 Correlated Erdős-Rényi graphs

For our synthetic data example, we will consider subgraph detection in the $\mathrm{CorrER}(\Lambda, R)$ with $\Lambda = 0.5$ (i.e., the maximum entropy ER model) and $R = \rho$ for $\rho = 0.6, 0.7, 0.8$. We consider $n = 500$, $n_c = 40$, and $M = 100$. Results are plotted in Figure 1. In the top panel of the figure we plot f, the GMP objective function, for each of the $M = 100$ initializations (note that we ordered the random restarts

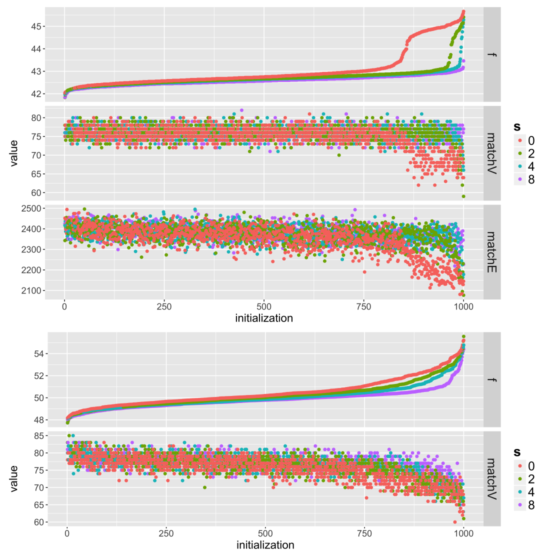

Figure 2: Detecting KC cells in the *Drosophila* mushroom body. In the top panel, in the first row, we plot the GMP objective function for each initialization (note that we ordered the random restarts in the figure to be monotone in f); in the middle row, we plot the number of vertices correctly matched; in the bottom row we plot the number of edges correctly matched. In the top panel of Figure 2, we plot the performance of our filter with the centered padding scheme, and in the bottom panel we use an approximate to the oracle padding scheme, centering by the least-squares optimal rank 1 approximations of Λ and Λ_c. The number of edge errors are not reported for the oracle as the objective function does not correspond to edge errors.

in the figure to be monotone in f); in the middle panel we plot the number of vertices correctly matched across the $M = 100$ initializations; in the bottom panel we plot the number of edges correctly matched across the $M = 100$ initializations. In red we plot $\rho = 0.5$, in green $\rho = 0.7$, and in blue $\rho = 0.8$. In all cases, we consider $s = 7$ seeds to initialize the FAQ algorithm. Note two key observations from the figure: first, performance is monotonic in ρ; second, there is an objective function gap between perfect performance and imperfect performance. This is a key feature, as it allows for an online algorithm to dynamically choose the number of MC restarts needed to achieve the correct result; indeed, stop after observing the gap!

4.2 Finding KC cells in a *Drosophila connectome*

For our real data example, we consider using the matched filter to locate the induced subgraph of the Kenyon cells (KC) in the fully reconstructed *Drosophila* mushroom body of [Eichler et al. 2017]. Using the induced subgraph of the KC cells in the left hemisphere of the mushroom body (i.e., as A), we seek to find the KC cells on the right hemisphere. Although in this example, the KC cells are identified across both hemispheres, this was achieved only with great effort and expenditure. Being able to use one hemisphere to locate structure in the other hemisphere could potentially allow for faster, cheaper neuron identification in future connectomes. After initial data preprocessing, there are $n_c = 100$ KC cells in each hemisphere and $n = 213$ vertices total in the right hemisphere both. We consider $s = 0, 2, 4, 8$ seeds and $M = 1000$ random restarts for our matched filter.

In the top panel of Figure 2, we plot the performance of our filter with the centered padding scheme, and in the bottom panel we use an approximate to the oracle padding scheme, centering by the least-squares optimal rank 1 approximations of Λ and Λ_c. In the figure, we see that more seeds produces better performance, and that the approximate oracle centering provides better performance (finding ≈ 85 of the 100 KC cells in the right hemisphere) than the data-independent centered padding scheme. We expect progressively more accurate estimations of the Λ matrices in the oracle centering to produce even better results, ideally resulting in an optimization gap as in the synthetic setting.

REFERENCES

Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* 29, 3 (May 2015), 626–688.

S. Chatterjee. 2014. Matrix estimation by universal singular value thresholding. *The Annals of Statistics* 43, 1 (2014), 177–214.

D. Conte, P. Foggia, C. Sansone, and M. Vento. 2004. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence* 18, 03 (2004), 265–298.

Mark A Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 2014. 1-Bit matrix completion. *Information and Inference: A Journal of the IMA* 3, 3 (Sept. 2014), 189–223.

Amir Egozi, Yosi Keller, and Hugo Guterman. 2013. A probabilistic approach to spectral graph matching. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (Jan. 2013), 18–27.

K. Eichler, F. Li, A. Litwin-Kumar, Y. Park, I. Andrade, C. M. Schneider-Mizell, T. Saumweber, A. Huser, C. Eschbach, B. Gerber, et al. 2017. The complete connectome of a learning and memory centre in an insect brain. *Nature* 548, 7666 (2017).

F. Emmert-Streib, M. Dehmer, and Y. Shi. 2016. Fifty years of graph matching, network alignment and network comparison. *Information sciences* 346–347 (2016), 180–197.

M. Fiori, P. Sprechmann, J. Vogelstein, P. MusÃĪ, and G. Sapiro. 2013. Robust Multi-modal Graph Matching: Sparse Coding Meets Graph Matching. *Advances in Neural Information Processing Systems* (2013), 127–135.

D.E. Fishkind, S. Adali, H. G. Patsolic, L. Meng, V. Lyzinski, and C.E. Priebe. 2017. Seeded Graph Matching. *arXiv preprint arXiv:1209.0367* (2017).

M Kuramochi and G Karypis. 2001. Frequent subgraph discovery. In *Proceedings 2001 IEEE International Conference on Data Mining*. 313–320.

V. Lyzinski, D. E. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe, and G. Sapiro. 2016. Graph matching: Relax at your own risk. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 1 (2016), 60–73.

V. Lyzinski, D. E. Fishkind, and C. E. Priebe. 2014. Seeded Graph Matching for Correlated Erdos-Renyi Graphs. *Journal of Machine Learning Research* 15 (2014), 3513–3540. http://jmlr.org/papers/v15/lyzinski14a.html

V. Lyzinski and D. L. Sussman. 2017. Graph matching the matchable nodes when some nodes are unmatchable. *arXiv preprint arXiv:1705.02294* (2017).

V. Lyzinski, D. L. Sussman, D. E. Fishkind, H. Pao, L. Chen, J. T. Vogelstein, Y. Park, and C. E. Priebe. 2015. Spectral clustering for divide-and-conquer graph matching. *Parallel Comput.* 47 (2015), 70–87.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.* 5, 1 (1957), 32–38.

J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. 2014. Fast Approximate Quadratic Programming for Graph Matching. *PLoS ONE* 10, 04 (2014).

L. Yartseva and M. Grossglauser. 2013. On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*. ACM, 119–130.

M. Zaslavskiy, F. Bach, and J.-P. Vert. 2009. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 12 (2009), 2227–2242.

Z Zhao, M Khan, V S A Kumar, and M V Marathe. 2010. Subgraph Enumeration in Large Social Contact Networks Using Parallel Color Coding and Streaming. In *2010 39th International Conference on Parallel Processing*. 594–603.