A Dynamical Systems Approach to Weighted Graph Matching

Michael M. Zavlanos and George J. Pappas

Abstract— **Graph matching is a fundamental problem that arises frequently in the areas of distributed control, computer vision, and facility allocation. In this paper, we consider the optimal graph matching problem for weighted graphs, which is computationally challenging due the combinatorial nature of the set of permutations. Contrary to optimization-based relaxations to this problem, in this paper we develop a novel relaxation by constructing dynamical systems on the manifold of orthogonal matrices. In particular, since permutation matrices are orthogonal matrices with nonnegative elements, we define two gradient flows in the space of orthogonal matrices. The first minimizes the cost of weighted graph matching over orthogonal matrices, whereas the second minimizes the distance of an orthogonal matrix from the finite set of all permutations. The combination of the two dynamical systems converges to a permutation matrix which, provides a suboptimal solution to the weighted graph matching problem. Finally, our approach is shown to be promising by illustrating it on nontrivial problems.**

I. Introduction

Given two graphs with weights on edges, the weighted graph matching problem searches for an optimal permutation of nodes of one graph so that the difference between the edge weights is minimized. Graph matching problems arise frequently in computer vision, facility allocation problems, as well as distributed control.

In computer vision, matching structural descriptions of an object to those of a model is formulated as a graph matching problem [6], [7], [8]. In distributed control and distributed robotics, graphs are recently emerging as a natural mathematical description for capturing interconnection topology [11] – [17]. Graph matching problems in this context can be used by a team of robots to reach a particular formation or minimize a distance from a particular formation. Finally, in facility allocation, graph matching is similar to the well known Quadratic Assignment Problem [9], [10].

In addition to its frequent appearance in various fields, weighted graph matching has also received a lot of attention due to its hardness. Since, it includes as a special case the largest common subgraph problem [8], which is NP-complete [18], it is also NP-complete. In particular, by its similarity to the quadratic assignment problem, problems with 20-25 nodes are considered very hard, and problems with more than 30 nodes are practically intractable [10]. Hence, many relaxations to the problem have been proposed [6], [7], [8], [9], [10]. In [6] the authors propose a spectral approach to the optimal matching problem. They treat weighted graphs with the same number of nodes and employ an analytic approach by using the eigen-structure of adjacency matrices (undirected graph matching) or some Hermitian matrices derived from the adjacency matrices (directed graph matching). An almost optimal matching can be found when the graphs are sufficiently close to each other. In [7] the authors propose a Lagrangian Relaxation Network for the same problem. They formulate the permutation matrix constraints in the framework of deterministic annealing and achieve exact constraint satisfaction at each temperature within deterministic annealing. More recently, semi-definite programming relaxations for the quadratic assignment problem have been proposed in [9] and [10]. In particular, in [9] the authors propose a cutting planes algorithm that provides good solutions.

Since permutation matrices live in the intersection of the non-convex space of orthogonal matrices and the space of non-negative (element-wise) matrices, the above optimization-based approaches relax the non-convex orthogonality constraint. In this paper, we take the opposite approach, and relax the non-negativity constraint by defining dynamical systems that are, by construction, guaranteed to evolve on the manifold of orthogonal matrices. In particular, we construct two gradient flows, one that minimizes the cost of weighted graph matching over orthogonal matrices, and a second that minimizes the distance of an orthogonal matrix from the set of permutations. The combination of the two dynamical systems converges to a permutation matrix, which provides a suboptimal solution to the weighted graph matching problem. In the spirit of analog solutions to combinatorial problems, our approach is inspired by the so-called isospectral double-bracket dynamical system that sorts lists and solves various combinatorial problems [1], [2] (see also [3], [4], [5]). We illustrate our approach in examples involving more than 50 nodes, which are considered practically intractable, and also challenging for semi-definite relaxations using standardized optimization packages. This shows that our method is very promising. We also argue that, for applications where mobility is critical, such as distributed robotics, our approach is also more natural.

The paper is organized as follows: In Section II, we develop the graph theoretic framework for our problem and illustrate the relaxation that motivates our dynamical systems approach. In Section III, we derive in detail the two gradient flows, characterize their equilibrium points and discuss how to combine them in order to get a solution to the graph matching problem. Finally, Section IV illustrates our approach in large matching problems, and discusses initialization issues for our method.

This work is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977.

Michael M. Zavlanos and George J. Pappas are with GRASP Laboratory, Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {zavlanos,pappasg}@grasp.upenn.edu

II. Graph Matching

A. Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a weighted undirected graph with vertices $\mathcal{V} = \{v_1, \dots, v_n\}$ and edges in the set $\mathcal{E}$. We define the weighted adjacency matrix of the graph $\mathcal{G}$ to be the matrix $A = (a_{ij})$, such that $a_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Since we do not allow self-loops, for every $i \in \{1, 2, \dots, n\}$ we define $a_{ii} = 0$. Moreover, since $\mathcal{G}$ is an undirected graph, A is a symmetric matrix.

Consider, now, two weighted undirected graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, as before, with $|\mathcal{V}_1| = |\mathcal{V}_2| = n$ and let $A_1 = \left(a_{ij}^{(1)}\right)$ and $A_2 = \left(a_{ij}^{(2)}\right)$ be their corresponding weighted adjacency matrices. Consider the set of positive integers $\{1, 2, \dots, n\}$, and let $\mathcal{S}_n$ be the set of permutations of $\{1, 2, \dots, n\}$. The graph matching problem consists of finding a permutation $\pi^\star \in S_n$ such that,

$$\pi^\star = \arg\min_{\pi} \sum_{i,j} \left(a_{\pi(i)\pi(j)}^{(1)} - a_{ij}^{(2)}\right)^2$$

We define a permutation matrix P as follows,

Definition 2.1 (Permutation Matrix): An $n \times n$ matrix $P = (p_{ij})$ is a permutation matrix if $p_{ij} \in \{0, 1\}$ and,

1. $\sum_{i=1}^{n} p_{ij} = 1$ for all $j = 1, \dots, n$.
2. $\sum_{j=1}^{n} p_{ij} = 1$ for all $i = 1, \dots, n$.

Let $\mathcal{P}_n$ denote the set of all permutation matrices of size $n \times n$. Since the sets $\mathcal{S}_n$ and $\mathcal{P}_n$ are into one-to-one correspondence, the graph matching problem can be reformulated as follows,

$$\begin{aligned} \min \quad & \|A_1 - P^T A_2 P\|_F^2 \\ \text{s.t.} \quad & P \in \mathcal{P}_n \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm defined as, $\|X\|_F = \left(\mathbf{tr}(XX^T)\right)^{1/2}$, for $X \in \mathbb{R}^{n \times n}$. Suppose there exists a permutation matrix $P \in \mathcal{P}_n$ that makes the objective value of this minimization problem equal to zero. Then, graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic. More formally,

Definition 2.2 (Isomorphic Graphs): Two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic if there exists a bijection φ from $\mathcal{V}_1$ to $\mathcal{V}_2$ such that $x \sim y$ in $\mathcal{G}_1$ if and only if $\varphi(x) \sim \varphi(y)$ in $\mathcal{G}_2$.

where $x \sim y$ implies that vertices x and y are adjacent in $\mathcal{G}_1$, or in other words, that the edge (x, y) belongs to the set of edges of the graph $\mathcal{G}_1$. Hence, all isomorphic graphs have the same structure, since one results from another by simple relabelling of the vertices. The following lemma will help us connect the notion of isomorphic graphs to that of a permutation matrix.

Lemma 2.3 ([19]): Let $\mathcal{G}_1, \mathcal{G}_2$ be graphs on the same vertex set. Then, they are isomorphic if and only if there is a permutation matrix P such that $A_2 = P^T A_1 P$, where A_i denotes the adjacency matrix of the graph $\mathcal{G}_i$.

Note that the existence of a permutation matrix P in Lemma 2.3, does not necessarily imply that it is also the unique orthogonal matrix satisfying the condition $A_2 = P^T A_1 P$. To see this, suppose that $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic. Let $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ be the eigenvalues of A_1 and A_2 (since $A_2 = P^T A_1 P$ is a similarity transformation, A_1 and A_2 have the same eigenvalues) and $A_1 = U \Lambda U^T$ and $A_2 = V \Lambda V^T$ be their corresponding eigendecompositions, with U and V orthogonal matrices. Then, $A_2 = P^T A_1 P$ would imply that $V \Lambda V^T = P^T U \Lambda U^T P$ and hence, $V = P^T U$ or equivalently $P = U V^T$. Clearly, P is orthogonal, however, Lemma 2.3 does not imply that it also a permutation matrix.

B. Problem Reformulation

In the spirit of subsection II-A, given any two, in general, not isomorphic graphs, our goal is to find a permutation matrix that minimizes the objective function in (1). The following result provides a lower bound on the value that $\|A_1 - P^T A_2 P\|_F^2$ can attain if the graphs are not isomorphic.

Theorem 2.4 ([6]): Let A_1 and A_2 be $n \times n$ symmetric matrices with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ respectively. Then, $\|A_1 - A_2\|_F^2 \geq \sum_{i=1}^{n} (\lambda_i - \mu_i)^2$

Since A_2 and $P^T A_2 P$ have the same eigenvalues for any orthogonal matrix P, Theorem 2.4 implies that, $\|A_1 - P^T A_2 P\|_F^2 \geq \sum_{i=1}^{n} (\lambda_i - \mu_i)^2$. This general form of the weighted graph matching problem does not have an analytic solution. Relaxations critically rely on the structure of permutation matrices being on the intersection of orthogonal matrices and elementwise non-negative matrices [20]. The following lemma provides this equivalent representation of the set of permutation matrices, and gives rise to the relaxation that we will adopt in our analysis.

Lemma 2.5: Let $\mathcal{O}_n$ denote the set of $n \times n$ orthogonal matrices and $\mathcal{N}_n$ denote the set of $n \times n$ elementwise non-negative matrices. Then, $\mathcal{P}_n = \mathcal{O}_n \cap \mathcal{N}_n$, where $\mathcal{P}_n$ is the set of $n \times n$ permutation matrices.

Proof: Let $P = (p_{ij})$ be such that $P \in \mathcal{P}_n$. Then, clearly P is orthogonal and its elements are non-negative. Hence, $P \in \mathcal{O}_n \cap \mathcal{N}_n$ which implies that $\mathcal{P}_n \subseteq \mathcal{O}_n \cap \mathcal{N}_n$. Now, let $P \in \mathcal{O}_n \cap \mathcal{N}_n$. Since P is orthogonal ($PP^T = I$), for all $i \neq j$ we have, $\sum_{k=1}^{n} p_{ik} p_{jk} = 0$. Moreover, since P is elementwise non-negative, we have that $p_{ik} p_{jk} = 0$ for all $i < j$. Let m be the first index such that $p_{mk} > 0$. Then, $p_{jk} = 0$ for all $j > m$. Since $\sum_{i=1}^{n} p_{ik}^2 = 1$ we conclude that $p_{mk} = 1$ and $p_{jk} = 0$ for all $j \neq m$. Repeating the same procedure for all the columns of P ($k = 1, \dots, n$) we get that every column of P has exactly one entry equal to 1 and the rest $n - 1$ entries equal to 0. Since, the rows P form vectors of unit magnitude as well, P must be a permutation matrix. Hence, $\mathcal{P}_n \supseteq \mathcal{O}_n \cap \mathcal{N}_n$ which completes the proof. ■

Lemma 2.5 implies that if we restrict P to be orthogonal and elementwise non-negative, we get a permutation matrix. Using this result as well as the fact that P has to be orthogonal (i.e., $P^T P = PP^T = I$), we get an equivalent representation for the graph matching problem in (1),

$$\begin{aligned} \min \quad & \|PA_1 - A_2 P\|_F^2 \\ \text{s.t.} \quad & P \in \mathcal{O}_n \cap \mathcal{N}_n \end{aligned} \tag{2}$$

Clearly, the objective function is convex, since $(PA_1 - A_2 P)$ is affine in P and the Frobenious norm $\|\cdot\|_F^2$ is convex. Moreover, the set $\mathcal{N}_n$ is also convex. However, the set of

orthogonal matrices $\mathcal{O}_n$ is not convex and so we can not use the already available tools from convex optimization to solve this problem. Various approaches have been proposed in the literature that, most of the times, relax the non-convex constraint that $P \in \mathcal{O}_n$ and, hence, solve a convex problem to get an approximate solution from which a permutation matrix is finally extracted [10]. In this paper, we follow a different approach. In particular we are interested in the following problem,

Problem 1: Derive a matrix differential equation $\dot{P}(t) = f(A_1, A_2, P(t))$, with $P(t) \in \mathcal{O}_n$ for all $t \geq 0$, that converges to a limit $\lim_{t\to\infty} P(t) = P_\infty$ such that,

1. P_∞ minimizes the objective function.
2. $P_\infty \in \mathcal{O}_n \cap \mathcal{N}_n$.

It is clear from the above problem formulation that $P(t)$ does not need to belong to the set $\mathcal{O}_n \cap \mathcal{N}_n$ for all time. However, the limit P_∞ has to satisfy both conditions of the problem. Such an approach is more flexible and we will show that it also gives good numerical results.

III. Gradient Flows on $\mathcal{O}_n$

In this section we construct two differential equations that respectively satisfy conditions 1 and 2 of Problem 1, and show how to combine them in order to get the sought behavior. We construct these differential equations by defining a gradient flow on the space of orthogonal matrices for an appropriately chosen cost function $V : \mathcal{O}_n \to \mathbb{R}$, as in [1], [2], [3], [4] and [5]. In particular, we parametrize the neighborhood of the orthogonal matrix P as, $P(\Omega) = P(I+\Omega+\Omega^2/2!+\dots)$, where Ω is skew-symmetric. Define the matrix inner product as $\langle A, B\rangle = \mathbf{tr}(A^T B)$. Then, the quantity $\langle (\nabla_P V(P))^T, \cdot\rangle$ represents the gradient of the function V at P. Using $\dot{P} = P\Omega$ we can express the gradient flow as,

$$P^T \dot{P} = (\nabla_P V(P))^T$$

The following, well known, result guarantees that any $P(t)$ that satisfies the previous matrix differential equation, will be orthogonal for all $t \geq 0$.

Lemma 3.1: Let $\Omega(t)$ be skew-symmetric for all $t \geq 0$ and define the matrix differential equation $\dot{P}(t) = P(t)\Omega(t)$. Then, $P(t) \in \mathcal{O}_n$ for all $t \geq 0$ if $P(0) \in \mathcal{O}_n$.

In the rest of this section we provide the gradient flow for the objective function and for a cost function we introduce in order to penalize negative entries in the orthogonal matrix P. Finally, we show that by superimposing these gradient flows we get a solution to the graph matching problem that is as close as we want to a permutation matrix.

A. Minimizing the Objective Function

Let $V_1 : \mathcal{O}_n \to \mathbb{R}$ be defined by,

$$V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2 \tag{3}$$

The following proposition describes an algorithm that minimizes this function.[1]

[1]Due to space limitations we omit the proofs in Subsection III-A.

Proposition 3.2 (Adopted from [1]): Assuming the standard metric on the orthogonal group, the gradient flow of the function $V_1 : \mathcal{O}_n \to \mathbb{R}$ defined by $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ is given by,

$$\dot{P} = P\big[P^T A_2 P A_1 - A_1 P^T A_2 P\big] \tag{4}$$

The following result guarantees that the gradient flow defined in equation (4) locally minimizes the cost function V_1. Moreover, it characterizes the critical points of this gradient flow.

Theorem 3.3 (Adopted from [1]): If $P(0) \in \mathcal{O}_n$ and $\dot{P} = P(P^T A_2 P A_1 - A_1 P^T A_2 P)$, then $\lim_{t\to\infty} P(t) = P_\infty$ exists and is a orthogonal matrix of the form $P = V\Pi S U^T$, with U, V orthogonal, Π a permutation matrix and S a square root of the identity matrix, i.e., $S = \mathrm{diag}(\pm 1, \dots, \pm 1)$, that minimizes the value of the objective function V_1.

B. Converging to a Permutation matrix

By Lemma 2.5, we can guarantee that P will converge to a permutation matrix, as long as it flows in the space of orthogonal matrices, and in the limit, it is elementwise non-negative. Hence, we need to define a cost function that penalizes negative entries in P. Inspired by the *Big M* method often used in optimization problems to force some variables be either negative or positive, let $V_2 : \mathcal{O}_n \to \mathbb{R}$ be defined by,

$$V_2(P) = \frac{2}{3}\mathbf{tr}P^T\big(P - (P \circ P)\big) \tag{5}$$

where $A \circ B$ denotes the *Hadamard* or elementwise product of the matrices $A = (a_{ij})$ and $B = (b_{ij})$, i.e., $A \circ B = (a_{ij}b_{ij})$. Since $P \in \mathcal{O}_n$ we have $P^T P = PP^T = I$ and so, $V_2(P) = \frac{2n}{3} - \frac{2}{3}\sum_{i,j=1}^n p_{ij}^3$, and the connection with the *Big M* method becomes clear. Thus, minimizing $V_2(P)$ forces the entries of P to be as "positive" as possible. In particular, we show that the gradient flow for the cost function defined in (5) does indeed converge to a permutation matrix. The following proposition describes the gradient flow of $V_2(P)$.

Proposition 3.4: Assuming the standard metric on the orthogonal group, the gradient flow of the function $V_2 : \mathcal{O}_n \to \mathbb{R}$ defined by $V_2(P) = \frac{2}{3}\mathbf{tr}P^T\big(P - (P \circ P)\big)$ is given by

$$\dot{P} = -P\big[(P \circ P)^T P - P^T (P \circ P)\big] \tag{6}$$

Proof: Observe that, $\mathbf{tr}P^T(P\circ P) = \frac{1}{2}\big(\mathbf{tr}(P\circ P)^T P + \mathbf{tr}P^T(P \circ P)\big)$. Hence,

$$V_2(P) = \frac{2n}{3} - \frac{1}{3}\Big(\underbrace{\mathbf{tr}(P \circ P)^T P}_{X_1(P)} + \underbrace{\mathbf{tr}P^T(P \circ P)}_{X_2(P)}\Big) \tag{7}$$

Using the first order approximation for the neighborhood of the orthogonal matrix P, $P(\Omega) = P(I + \Omega)$, where Ω is skew-symmetric, we get,

$$X_1(P(I+\Omega)) = \mathbf{tr}(P \circ P)^T P + \mathbf{tr}\big[(P \circ P)^T P - 2P^T(P \circ P)\big]\Omega \tag{8}$$

where we have neglected terms of the order of Ω^2 and have made use of the relation, $\mathbf{tr}(P^T \circ \Omega P^T)P = \mathbf{tr}(P \circ P)\Omega P^T$. Similarly,

$$X_2(P(I+\Omega)) = \mathbf{tr}P^T(P \circ P) + \mathbf{tr}\big[2(P \circ P)^T P - P^T(P \circ P)\big]\Omega \tag{9}$$

where again we have neglected terms of the order of Ω^2 and have made use of the relation, $\mathbf{tr}P^T(P \circ P\Omega) = \mathbf{tr}(P \circ P)^T P\Omega$. Substituting equations (8) and (9) in (7) we get, $V_2(P(I+\Omega)) = V_2(P) - \mathbf{tr}\big[(P \circ P)^T P - P^T(P \circ P)\big]\Omega$. As before, we may conclude that the quantity $\langle \big[(P\circ P)^T P - P^T(P \circ P)\big], \cdot \rangle$ represents the gradient of $V_2(P)$ at P. Using $\dot{P} = P\Omega$ we can express the gradient flow as $P^T\dot{P} = -\big[(P\circ P)^T P - P^T(P \circ P)\big]$. ■

In the rest of this section we show that the gradient flow defined in (6) decreases the value of the cost function $V_2(P)$ and in the limit, forces the entries of P to become non-negative. In particular, the following three results establish that $V_2(P)$ is a Lyapunov function for the system (6) and characterize its critical points.

Lemma 3.5: Let $V_2(P) = \frac{2}{3}\mathbf{tr}P^T\big(P - (P \circ P)\big)$. Then, $V_2(P) \geq 0$ for all $P \in \mathcal{O}_n$ with equality if and only if P is a permutation matrix.

Proof: Since, the rows and columns of P form vectors of unit magnitude, $|p_{ij}| \leq 1$ for all i, j. Hence, $|p_{ij}^3| \leq p_{ij}^2$ for all i, j with equality if and only if p_{ij} equals 0 or ± 1. Summing over j we get, $-1 \leq \sum_{j=1}^n p_{ij}^3 \leq 1$, since $\sum_{j=1}^n p_{ij}^2 = 1$ for all i by orthogonality of P. Clearly, the right-hand side inequality becomes equality if and only if $p_{ij} = 1$ for exactly one j and $p_{ik} = 0$ for $k \neq j$. In the same way, the left-hand side inequality becomes equality if and only if $p_{ij} = -1$ for exactly one j and $p_{ik} = 0$ for $k \neq j$. Now, summing over i we get, $-n \leq \mathbf{tr}P^T(P \circ P) \leq n$, which clearly implies that $V_2(P) \geq 0$ for all $P \in \mathcal{O}_n$ with equality if and only if $\mathbf{tr}P^T(P \circ P) = n$. Following the previous argument, $\mathbf{tr}P^T(P \circ P) = n$ is true if and only if each row of P has exactly one entry equal to 1 and the rest of the entries equal to 0. By orthogonality of P, this implies that such a P is a permutation matrix. ■

Lemma 3.6: Let $P(t)$ satisfy the matrix differential equation $\dot{P} = -P\big[(P \circ P)^T P - P^T(P \circ P)\big]$ with $P(0) \in \mathcal{O}_n$ for all $t \geq 0$, and define the function $V_2(P) = \frac{2}{3}\mathbf{tr}P^T\big(P - (P\circ P)\big)$. Then, $\dot{V}_2(P) \leq 0$ for all $t \geq 0$ with equality if and only if $P(t) = S\Pi$, where S is a square root of the identity matrix, i.e., $S = \text{diag}(\pm 1, \ldots, \pm 1)$, and Π is a permutation matrix.

Proof: Since $(P\circ P)^T P - P^T(P\circ P)$ is skew-symmetric for all $t \geq 0$, $P(t)$ is orthogonal for all $t \geq 0$, by Lemma 3.1. Using equation (7), it can be shown that, $\dot{V}_2(P) = -\|(P \circ P)^T P - P^T(P\circ P)\|_F^2$. Hence, $\dot{V}_2(P) \leq 0$ for all $t \geq 0$ with $\dot{V}_2(P) = 0$ if and only if $(P \circ P)^T P = P^T(P \circ P)$. In the rest of this proof we will explicitly describe the orthogonal matrices P that satisfy $(P \circ P)^T P = P^T(P \circ P)$.

Let $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ be the singular values of $(P \circ P)$ and let $(P\circ P) = U\Sigma V^T$ be its singular value decomposition, with U and V orthogonal matrices and $\Sigma = \text{diag}(\sigma_i)$. We can show that, $P^T U\Sigma^2 U^T P = (P\circ P)^T(P\circ P) = V\Sigma^2 V^T$, which implies that $\Sigma^2 = V^T P^T U\Sigma^2 U^T PV$ or equivalently, that $U^T PV = S$ and hence, $P = USV^T$ for some square root of the identity matrix S. Thus, there exists an S such that every critical point P can be written as $P = USV^T$. Since, $(P \circ P) = U\Sigma V^T$ we have that,

$$(USV^T \circ USV^T) = U\Sigma V^T \tag{10}$$

Clearly, $\Sigma = (S \circ S) = I$ is a solution to equation (10), by Corollary 6.2 (see appendix). Moreover, by uniqueness of the singular values, $\Sigma = (S \circ S) = I$ is the unique solution to equation (10). Hence, by Corollary 6.2 on factoring properties of Hadamard products, the only way that U and V^T can be factored out from the expression $(USV^T \circ USV^T)$ is when they both are permutation matrices.[2] Hence, equation (10) can only be true if U and V are permutation matrices and $\Sigma = (S \circ S) = I$. We conclude that every critical point P has to be of the form $P = S\Pi$, which completes the proof. ■

Lemma 3.7: Let $\mathcal{C} = \{P \mid \dot{V}_2(P) = 0\}$ be the set of critical points of the matrix differential equation $\dot{P} = -P\big[(P\circ P)^T P - P^T(P\circ P)\big]$. Then, the only stable critical points are the permutation matrices.

Proof: Observe that, $\dot{P} = -P\big[(P\circ P)^T P - P^T(P \circ P)\big] = -P(P \circ P)^T P + (P \circ P)$, which expressed elementwise becomes, $\dot{p}_{ij} = -\sum_{k=1}^n p_{ik}\sum_{m=1}^n p_{mk}^2 p_{mj} + p_{ij}^2$. It can be shown that the linearization of the system in a neighborhood of the critical points $\mathcal{C} = \{P \mid \dot{V}_2(P) = 0\} = \{P \mid P = S\Pi\}$ is given by,

$$\dot{p}_{ij} = -\Big(s_i + \sum_{m=1}^n s_m \pi_{mj}\Big)p_{ij} \quad \text{for all } i, j \tag{11}$$

where $\Pi = (\pi_{ij})$, $S = \text{diag}(s_i)$ and,

$$\delta_{ij,st} = \begin{cases} 1 & \text{if } i = s \text{ and } j = t \\ 0 & \text{otherwise} \end{cases}$$

denotes the *Kronecker Delta* function. Similarly, we can define $\delta_{i,s}$ to equal 1 only if $i = s$ and 0 otherwise.

Suppose that $s_i = 1$ for all i. Then, $\dot{p}_{ij} = -2p_{ij}$ for all i, j and hence, the system is stable. Suppose now that there exists at least one index k such that $s_k = -1$. Then, since every row of Π has exactly one entry equal to 1, there exists an index l such that $\pi_{kl} = 1$ and so $\dot{p}_{kl} = 2p_{kl}$, i.e., there exists at least one unstable state. Hence, the only stable critical points are the permutation matrices Π and the proof is complete. ■

We can summarize the results of Lemmas 3.5, 3.6 and 3.7 in the following theorem.

Theorem 3.8: Let $P(0) \in \mathcal{O}_n$ and suppose that $P(t)$ satisfies the matrix differential equation, $\dot{P} = -P\big[(P \circ P)^T P - P^T(P\circ P)\big]$, for all $t \geq 0$. Then $\lim_{t\to\infty} P(t) = P_\infty$ exists and is a permutation matrix.

C. Superposition of the Gradient Flows

Up to this point we have defined two gradient flows on the space of orthogonal matrices that respectively minimize their cost functions, while the second one also converges to a permutation matrix. It is reasonable, thus, to expect that

[2]Note that, by Lemma 6.1 (see appendix), since USV^T is orthogonal, we can also have the factorization $(USV^T \circ USV^T) = USV^T$ with USV^T a permutation matrix. However, by equation (10), this would imply that $\Sigma = S$ which contradicts the requirement that $\Sigma \geq 0$.

by combining these two gradient flows we can achieve the desired behavior of Problem 1. In particular, we consider two ways of combining the gradient flows. The first approach superimposes the gradient flows by adding them, whereas the second approach ignores the nonnegativity requirement and switches to permutation gradient flow when the objective has been sufficiently minimized.

Theorem 3.9: Assume that A_1 and A_2 are weighted adjacency matrices corresponding to the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. Assume further that $P(0) \in \mathcal{O}_n$ and suppose that $P(t)$ satisfies the matrix differential equation,

$$\begin{aligned}\dot{P} &= P\big[P^T A_2 P A_1 - A_1 P^T A_2 P\big] - \\ &\quad -kP\big[(P\circ P)^T P - P^T(P\circ P)\big] \qquad (12)\end{aligned}$$

for all $t \geq 0$, where k is a positive constant. Then, for sufficiently large k, $\lim_{t\to\infty} P(t) = P_\infty$ exists and approximates a permutation matrix that also minimizes the distance $\|PA_1 - A_2P\|_F^2$. The larger k is, the better P_∞ approximates a permutation matrix.

Proof: Consider the function $V : \mathcal{O}_n \to \mathbb{R}$, defined by, $V(P) = V_1(P) + kV_2(P)$, with $V_1(P)$ and $V_2(P)$ as in equations (3) and (5) resepectively, to be a Lyapunov function candidate for the system. Clearly, $V(P) \geq 0$ for all $P \in \mathcal{O}_n$ and $V(P) = 0$ if and only if $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic (in which case P is a permutation matrix). It can be shown that, $\dot{V}(P) = -\|X_1 - kX_2\|_F^2$, where $X_1(P) = P^T A_2 P A_1 - A_1 P^T A_2 P$ and $X_2(P) = (P\circ P)^T P - P^T(P\circ P)$. Hence, $\dot{V}(P)$ is non-increasing which means that P will converge to a local minimum, P_∞. The set of critical points of $\dot{V}(P)$ is $\mathcal{C} = \{P \mid X_1(P) = kX_2(P),\ k \geq 0\}$. Clearly, if $X_2(P_\infty) = 0$, then $P_\infty = S\Pi$, with $S = \text{diag}(\pm 1, \ldots, \pm 1)$ and Π a permutation matrix (Lemma 3.6). Hence, as in Lemma 3.7, linearizing (12) around $\mathcal{C}$, we can show that for sufficiently large k, the only stable critical points are permutation matrices.

Assume that we want P_∞ to be in an ϵ neighborhood of a permutation matrix, for some $\epsilon > 0$, i.e., we want P_∞ to satisfy a condition of the form $\|X_2(P_\infty)\| < \epsilon$. Since, $P_\infty \in \mathcal{C}$ the equation $X_1(P_\infty) = kX_2(P_\infty)$ implies that $\|X_1(P_\infty)\| = k\|X_2(P_\infty)\|$. Hence, $\|X_2(P_\infty)\| < \epsilon$ if and only if $\frac{1}{k}\|X_1(P_\infty)\| < \epsilon$. So, by choosing, $k > \frac{1}{\epsilon}\max_{P\in\mathcal{O}_n}\|X_1(P)\|$ we can guarantee that P_∞ will be as close as we want to a permutation matrix. Clearly, the larger k is, the smaller ϵ can be. Note at this point that $\max_{P\in\mathcal{O}_n}\|X_1(P)\|$ is bounded since $\|X_1(P)\|$ is a continuous function of P on the compact space of orthogonal matrices $\mathcal{O}_n$. Hence, k is well defined. ■

It is clear from Theorem 3.9 that if we want P to converge exactly to a permutation matrix, we should choose a very large value for k. In this way, however, we lose track of the other objective which is to find the permutation matrix that minimizes the distance between the weighted graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. Hence, there is a tradeoff between how close the final solution is to a permutation matrix and how well it serves as a minimizer of the objective function (3). Intuitively, however, the closer P is to the optimal permutation matrix, the less k affects the performance of the algorithm, since in this case, it only affects the speed of convergence to that permutation matrix. In other words, initialization of the problem is important. In order to take advantage of this fact, we can think of the following hybrid scheme for our problem.

$$P(0) \in \mathcal{O}_n \quad \overset{PhaseI}{\longrightarrow} \quad P^\star \in \mathcal{O}_n \quad \overset{PhaseII}{\longrightarrow} \quad \Pi \in \mathcal{P}_n$$

In *Phase I* we apply the gradient flow (4) to minimize the objective function V_1, and use its solution $P^\star$ as a "good" initial condition for *Phase II* of our hybrid scheme, where we apply the gradient flow (12) with sufficiently large k in order to converge to a permutation matrix Π. In the following section we implement our dynamical systems approach to weighted graph matching problems.

IV. Simulation Results

In the previous sections we developed two provably correct gradient flows on the space of orthogonal matrices and discussed how we can combine them for the case of the weighted graph matching problem. In this section we discuss some heuristics and show that our algorithm gives very good results in practice.

Note first, that the space of orthogonal matrices $\mathcal{O}_n$ consists of two connected components, one with elements having determinant 1 and one with elements having determinant -1. Since, the gradient flow we defined on $\mathcal{O}_n$ remains, for all time, in the connected component in which it was initialized, we need to sample both connected components, at least once each, and take the best solution. Moreover, since initialization of the problem is important, we choose to employ the hybrid scheme we discussed earlier, where we first apply dynamical system (4) to minimize the objective function V_1 and get a "good" solution $P^\star$ (that is orthogonal but not a permutation matrix), which we then use as a "good" initial condition for the combined dynamical system in order to converge to a permutation matrix Π.

We implemented our algorithm with random weighted adjacency matrices A_1 and A_2, and initialized the first phase of our hybrid scheme with random orthogonal matrices (we sampled both connected components of $\mathcal{O}_n$). Bellow one can see an instance of the final permutation matrix Π that our algorithm gave in the case of a 10×10 graph matching problem with weighted adjacency matrices randomly generated from the uniform distribution.

$$\Pi_{4:10,3:6} = \begin{bmatrix} 0.0022 & 0.0031 & 1.0000 & -0.0020 \\ -0.0004 & 0.0001 & 0.0019 & 1.0000 \\ -0.0002 & 0.0005 & -0.0013 & -0.0030 \\ 0.0005 & 0.0031 & -0.0027 & -0.0024 \\ -0.0001 & 0.0024 & -0.0011 & -0.0015 \\ 0.0012 & 1.0000 & -0.0031 & -0.0001 \\ 1.0000 & -0.0012 & -0.0022 & 0.0004 \end{bmatrix}$$

One can also get an idea about how much the objective function is minimized in each phase. In the following table we present the value of the objective function $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ at the end of the two phases of the algorithm for the previous problem, i.e., for $P = P^\star$ and $P = \Pi$.

	Initialization	Phase I	Phase II
$det = -1$	34.8605	0.1527	5.9719
$det = 1$	36.5412	0.1527	5.9672

It can be seen that the final solution results in an objective value approximately six times smaller than the initial one. This considerable decrease in the objective function gives rise to the question of how close the final solution is to the optimal one. Since, we do not have any global results, but only guarantees that the algorithm will converge to a local minimum, we compared the best solution, i.e., the solution Π such that $V_1(\Pi) = 5.9672$, with a sample of 10^6 randomly generated 10×10 permutation matrices P (there are $10! = 3,628,800$ such permutation matrices in total). We observed that $V_1(\Pi) \leq V_1(P)$ for approximately 96% of the samples P. Hence, roughly speaking, we may conclude that our algorithm does indeed provide a very good solution to the weighted graph matching problem.

Finally, we implemented our method for problems of size 50×50. In the following table we present the value of the objective function $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ at the end of the two phases of the algorithm.

	Initialization	Phase I	Phase II
$det = -1$	800.6253	0.3473	162.0595
$det = 1$	807.1942	0.3416	169.3697

Again we notice a significant decrease in the value of the objective function. More important, however, is the running time of the algorithm, which makes us believe that it could be a promising idea for further research. In particular, on an Intel Centrino 2GHz with 1Gb memory laptop, using Matlab 6 and low level programming (Runge-Kutta 4 with constant step size), it took on average 30 mins for the first phase to terminate and 3 mins for the second phase. It is worth noting that on the same laptop and using SeDuMi, we failed to solve semidefinite programming relaxations (of the non-convex orthogonality constraint) of the same size.

V. Conclusions

In this paper, we considered the problem of finding the optimal relabelling of the vertices of a graph so that its distance from some reference graph is minimized in the Frobenious norm sense. We relaxed the combinatorial nature of the problem by giving an equivalent representation for the set of permutation matrices as the intersection of the space of orthogonal matrices with the set of elementwise non-negative matrices. This representation gave rise to defining two gradient flows on the space of orthogonal matrices, such that one minimizes the distance of the two graphs and the second converges to a permutation matrix. We discussed superimposing the two gradient flows to apply them to the weighted graph matching problem, as well as initialization and a hybrid scheme for combining the two dynamical systems. Our algorithm is provably correct and the simulations illustrate our theoretical results, as well as the high performance achieved when combined with the proposed heuristics.

References

[1] R.W. Brockett. *Dynamical Systems that Sort Lists, Diagonalize Matrices and Solve Linear Programming Problems*, Linear Algebra and Its Applications, 146 (1991), pp. 79-91

[2] R. W. Brockett. *Least Squares Matching Problems*, Linear Algebra and its Applications, 122 - 124 (1989), pp. 761-777

[3] Moody T. Chu and K. R. Driessel. *The Projected Gradient Method for Least Squares Matrix Approximations with Spectral Constraints*, SIAM Journal on Numerical Analysis, 27 (1990), pp. 1050-1060

[4] Moody T. Chu. *Matrix Differential Equations: A Continuous Realization Process for Linear Algebra Problems*, Nonlinear Analysis, 18 (1992), pp. 1125-1146

[5] A. M. Bloch, R.W. Brockett and P.E. Crouch. *Double Bracket Equations and Geodesic Flows on Symmetric Spaces*, Communications on Mathematical Physics 187 (1997), pp. 357-373

[6] Shinji Umeyama. *An Eigendecomposition Approach to Weighted Graph Matching Problems*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, September 1988.

[7] Anand Rangarajan and Eric Mjolsness. *A Lagrangian Relaxation Network for Graph matching*, IEEE Transactions on Neural Networks, Vol. 7, No. 6, November 1996.

[8] Steven Gold and Anand Rangarajan. *A Graduated Assignment Algorithm for Graph Matching*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 (4), 1996.

[9] Alain Faye and Frederic Roupin. *A Cutting Planes Algorithm Based Upon a Semidefinite Relaxation for the Quadratic Assignment Problem*, In ESA 2005, 3-6 Octobre, Majorque, Espagne. LNCS 3669, pp. 850-861, 2005. (ref. CEDRIC 835).

[10] Henry Wolkowicz. *Semidefinite Programming Approaches to the Quadratic Assignment Problem*, Nonlinear Assignment Problems: Algorithms and Applications, Combinatorial Optimization Series, 7, 143-174, Kluwer Academic Publishers, 2000.

[11] Mehran Mesbahi. *On State-dependent Dynamic Graphs and their Controllability Properties*, IEEE Transactions on Automatic Control (50) 3: 387- 392, 2005.

[12] H. Tanner, A Jadbabaie and G. Pappas. Flocking in Fixed and Switching Networks, IEEE Transactions on Automatic Control, April 2005. Submitted.

[13] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. *On Synchronous Robotic Networks - Part I: Models, Tasks and Complexity Notions*, IEEE Transactions on Automatic Control, April 2005, to appear.

[14] A. Jadbabaie, J. Lin and A. S. Morse. *Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules*, IEEE Transactions on Automatic Control, (48) 6: 988-1001, 2003.

[15] R. Olfati-Saber and R. M. Murray. *Agreement Problems in Networks with Directed Graphs and Switching Topology*, Proceedings of the 42nd IEEE Conference on Decision and Control, December 2003.

[16] A. Muhammad and M. Egerstedt. *Connectivity Graphs as Models of Local Interactions*, Journal of Applied Mathematics and Computation, 168 (1), pp. 243-269, 2005.

[17] J. Lin, A. S. Morse, and B. D. O. Anderson. *The Multi-Agent Rendezvous Problem*, Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 1508-1513, December 2003.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractabiltiy: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA, 1979.

[19] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*, Springer Verlag New York, 2001.

[20] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*, Cambridge University Press, 1985.

[21] Uwe Helmke and John B. Moore . *Optimization and Dynamical Systems*, Springer-Verlag, London, New York, 1994.

VI. Appendix

Factoring properties of the Hadamard product (stated without proof due to space limitations).

Lemma 6.1: Let $U, V \in \mathcal{O}_n$. Then, $(UV^T \circ UV^T) = (U \circ U)V^T$ if and only if V is a permutation matrix.

Corollary 6.2: Let $S = \text{diag}(\pm 1, \ldots, \pm 1)$ and $U, V \in \mathcal{O}_n$. Then, $(USV^T \circ USV^T) = U(S \circ S)V^T$ if and only if U and V are permutation matrices.