Automatica 44 (2008) 2817–2824

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A dynamical systems approach to weighted graph matching[☆]

Michael M. Zavlanos [*], George J. Pappas

Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA

ARTICLE INFO

Article history:
Received 26 April 2007
Received in revised form
28 September 2007
Accepted 8 April 2008
Available online 9 October 2008

Keywords:
Graph matching
Isomorphic graphs
Orthogonal matrices
Differential flows

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.

1. Introduction

Given two graphs on the same number of nodes 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 (Gold & Rangarajan, 1996; Rangarajan & Mjolsness, 1996; Umeyama, 1988). In distributed control and distributed robotics, graphs are recently emerging as a natural mathematical description for capturing interconnection topology (Cortes, Martinez, & Bullo, 2006; Jadbabaie, Lin, & Morse, 2003; Mesbahi, 2005; Muhammad & Egerstedt, 2005; Olfati-Saber & Murray, 2003; Tanner, Jadbabaie, & Pappas, 2007; Zavlanos & Pappas, 2007). Graph matching in this context appears in problems involving multi-agent target assignment or formation stabilization (Smith & Bullo, 2006; Zavlanos & Pappas, 2008, 2007). Finally, in facility allocation, graph matching is similar to the well known Quadratic Assignment Problem (Faye & Roupin, 2005; Wolkowicz, 2000).

In addition to its frequent appearance in various fields, weighted graph matching has also received a lot of attention due to its difficulty. Since, it includes as a special case the largest common subgraph problem (Gold & Rangarajan, 1996), which is NP-hard (Garey & Johnson, 1979), it is also NP-hard. 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 (Wolkowicz, 2000). Hence, many relaxations to the problem have been proposed (Faye & Roupin, 2005; Gold & Rangarajan, 1996; Rangarajan & Mjolsness, 1996; Umeyama, 1988; Wolkowicz, 2000). In Umeyama (1988) the authors propose a spectral approach to the optimal matching problem. They consider weighted graphs with the same number of nodes and employ an analytic approach by using the eigenstructure 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 Rangarajan and Mjolsness (1996) 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. Recently, semi-definite programming relaxations for the quadratic assignment problem have also been proposed (Faye & Roupin, 2005; Wolkowicz, 2000). Such approaches typically aim at generating bounds that can be used in exact algorithms, such as branch and bound.

☆ This work is partially supported by ARO MURI SWARMS Grant W911NF-05-1-0219 and the NSF ITR Grant 0324977. This paper was not presented at any IFAC meeting. A preliminary version of this work can be found in [Zavlanos, M. M., & Pappas, G. J. (2006). A dynamical systems approach to weighted graph matching. In *Proceedings of the 45th IEEE conference on decision and control* (pp. 3492–3497)]. This paper was recommended for publication in revised form by Associate Editor Dragan Nešić under the direction of Editor Hassan K. Khalil.

* Corresponding author. Tel.: +1 (215) 898 5814; fax: +1 (215) 573 2048.
E-mail addresses: zavlanos@grasp.upenn.edu (M.M. Zavlanos), pappasg@grasp.upenn.edu (G.J. Pappas).

0005-1098/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2008.04.009

Since permutation matrices live in the intersection of the non-convex space of orthogonal matrices and the space of element-wise non-negative matrices, relaxing the non-convex orthogonality constraint results in a natural approximation to the problem. This relaxation is employed by most of the above optimization-based approaches. 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 (Zavlanos & Pappas, 2006). 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 (Brockett, 1991, 1989) (see also Bloch, Brockett, and Crouch (1997), Chu (1992), Chu and Driessel (1990) and Helmke, Moore, and Brockett (1996)). 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 2, we develop the graph theoretic framework for our problem and illustrate the relaxation that motivates our dynamical systems approach. In Section 3, 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 4 illustrates our approach in large matching problems, and discusses initialization issues for our method.

2. Graph matching

2.1. 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, we define $a_{ii} = 0$ for all $i \in \{1, 2, \dots, n\}$. 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, further, the set $\mathcal{S}_n$ of all permutations of the positive integers $\{1, 2, \dots, n\}$. Then, 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 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 (Fig. 1)

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

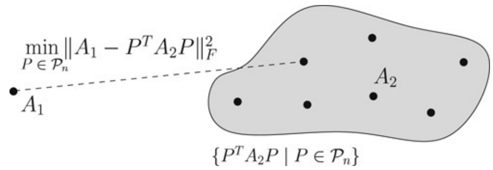

Fig. 1. Problem formulation.

where $\|\cdot\|_F$ denotes the Frobenius norm defined as, $\|X\|_F = \left(\mathbf{tr}(XX^{\mathrm{T}})\right)^{1/2}$, for $X \in \mathbb{R}^{n\times n}$. A class of graphs that often appear in the context of graph matching are the so called isomorphic graphs. More formally, let $v_i \sim v_j$ denote adjacent vertices in a graph $\mathcal{G}$, namely vertices such that the edge (v_i, v_j) belongs to the edge set of $\mathcal{G}$. Then, we have the following definition.

Definition 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 $v_i \sim v_j$ in $\mathcal{G}_1$ if and only if $\varphi(v_i) \sim \varphi(v_j)$ in $\mathcal{G}_2$.

Clearly, all isomorphic graphs have the same structure, since one results from another by a simple relabelling of vertices. The following lemma will help us connect the notion of isomorphic graphs to that of a permutation matrix.

Lemma 3 (Godsil & Royle, 2001). *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 $A_2 = P^{\mathrm{T}} A_1 P$, where A_i denotes the adjacency matrix of the graph $\mathcal{G}_i$.*

Note that the existence of a permutation matrix P in Lemma 3, does not necessarily imply that it is also the unique orthogonal matrix satisfying the condition $A_2 = P^{\mathrm{T}} A_1 P$. 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 $A_2 = P^{\mathrm{T}} A_1 P$ is a similarity transformation, A_1 and A_2 have the same eigenvalues) and $A_1 = U \Lambda U^{\mathrm{T}}$ and $A_2 = V \Lambda V^{\mathrm{T}}$ be their corresponding eigendecompositions, with U and V orthogonal matrices. Then, $A_2 = P^{\mathrm{T}} A_1 P$ implies that $V \Lambda V^{\mathrm{T}} = P^{\mathrm{T}} U \Lambda U^{\mathrm{T}} P$ and hence, $V = P^{\mathrm{T}} U S$ or equivalently $P = U S V^{\mathrm{T}}$, where $S = \mathrm{diag}(\pm 1, \dots, \pm 1)$. Clearly, P is orthogonal, but not necessarily a permutation matrix.

2.2. Problem reformulation

In the spirit of Section 2.1, 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 $\|A_1 - P^{\mathrm{T}} A_2 P\|_F^2$ can attain.

Theorem 4 (Umeyama, 1988). *Let A_1 and A_2 be $n \times n$ symmetric matrices with eigenvalues $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$ and $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_n$ respectively. Then, $\|A_1 - A_2\|_F^2 \ge \sum_{i=1}^n (\lambda_i - \mu_i)^2$.*

Since A_2 and $P^{\mathrm{T}} A_2 P$ have the same eigenvalues for any orthogonal matrix P, Theorem 4 implies that, $\|A_1 - P^{\mathrm{T}} A_2 P\|_F^2 \ge \sum_{i=1}^n (\lambda_i - \mu_i)^2$. 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 element-wise non-negative matrices (Horn & Johnson, 1985). 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 5. *Let $\mathcal{O}_n$ denote the set of $n \times n$ orthogonal matrices and $\mathcal{N}_n$ denote the set of $n \times n$ element-wise 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 ($PP^{\mathrm{T}} = I$), for all $i \neq j$ we have $\sum_{k=1}^{n} p_{ik}p_{jk} = 0$. Moreover, since P is element-wise 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 ($k = 1, \ldots, n$) 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 of P also form vectors of unit magnitude, P must be a permutation matrix. Hence, $\mathcal{P}_n \supseteq \mathcal{O}_n \cap \mathcal{N}_n$ which completes the proof. □

Lemma 5 implies that if we restrict P to be orthogonal and element-wise non-negative, we get a permutation matrix. Using this result, as well as the fact that P has to be orthogonal (i.e., $P^{\mathrm{T}}P = PP^{\mathrm{T}} = I$), we get an equivalent representation for the graph matching problem (1),

$$\min \|PA_1 - A_2P\|_F^2 \quad \text{s.t. } P \in \mathcal{O}_n \cap \mathcal{N}_n. \tag{2}$$

Clearly, the objective function is convex, since $(PA_1 - A_2P)$ is affine in P and the Frobenius 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 cannot use the already available tools from convex optimization to solve this problem. Various approaches have been proposed in the literature that, most of the time, 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 (Wolkowicz, 2000). In this paper, we follow a different approach. In particular we are interested in the following problem.

Problem 6. 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 asymptotically converges to a matrix $\lim_{t\to\infty} P(t) = P_\infty$ 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.

3. Gradient flows on $\mathcal{O}_n$

In this section we construct two differential equations that respectively satisfy conditions 1 and 2 of Problem 6, 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 Bloch et al. (1997), Brockett (1991, 1989), Chu (1992) and Chu and Driessel (1990). In particular, we parameterize the neighborhood of the orthogonal matrix P as $P(\Omega) = P(I + \Omega + \Omega^2/2! + \cdots)$, where Ω is skew-symmetric, i.e., $\Omega^{\mathrm{T}} = -\Omega$. Then, the first order approximation of the cost function V in a neighborhood of P becomes

$$V(P(I+\Omega)) \approx V(P) + \mathbf{tr}\left(\mathcal{F}_V(P)^{\mathrm{T}}\Omega\right),$$

where $\mathcal{F}_V(P)$ is a skew-symmetric matrix function of P.[1] Define, further, the standard metric on $\mathcal{O}_n$ by the matrix inner product $\langle A, B\rangle = \mathbf{tr}(A^{\mathrm{T}}B)$ (on the special orthogonal group $SO(n)$, this is proportional to the Killing form). Then, the quantity $\langle \mathcal{F}_V(P)^{\mathrm{T}}, \cdot\rangle$ represents the negative gradient of the function V at P.[2] Using $\dot{P} = P\Omega$ we can express the gradient flow as

$$P^{\mathrm{T}}\dot{P} = \mathcal{F}_V(P)^{\mathrm{T}}.$$

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 7. *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 $P(0) \in \mathcal{O}_n$.*

Proof. We have, $\frac{\mathrm{d}}{\mathrm{d}t}\left(P(t)P(t)^{\mathrm{T}}\right) = \dot{P}(t)P(t)^{\mathrm{T}} + P(t)\dot{P}(t)^{\mathrm{T}} = P(t)\Omega(t)P(t)^{\mathrm{T}} - P(t)\Omega(t)P(t)^{\mathrm{T}} = 0$. Hence, $P(t)P(t)^{\mathrm{T}} =$ const. for all $t \geq 0$ and since $P(0)P(0)^{\mathrm{T}} = I$ we conclude that $P(t)P(t)^{\mathrm{T}} = I$ for all $t \geq 0$, i.e., $P(t) \in \mathcal{O}_n$ for all $t \geq 0$. □

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.

3.1. 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.

Proposition 8 (*Adopted from Brockett (1991)*). *Assuming the standard metric on the orthogonal group, the gradient flow of the function $V_1 : \mathcal{O}_n \to \mathbb{R}$ is given by,*

$$\dot{P} = P\left(P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P\right). \tag{4}$$

Proof. Using the first order approximation for the neighborhood of the orthogonal matrix P, $P(\Omega) \approx P(I+\Omega)$, where Ω is skew-symmetric, we can show that

$$\frac{1}{2}\|P(I+\Omega)A_1 - A_2P(I+\Omega)\|_F^2 \approx \frac{1}{2}\|PA_1 - A_2P\|_F^2 + \mathbf{tr}\left(P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P\right)\Omega,$$

where we have neglected terms of the order of Ω^2. Clearly, the quantity $\langle (P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P), \cdot\rangle$ represents the negative gradient of V_1 at P. Using $\dot{P} = P\Omega$ we can express the gradient flow as $P^{\mathrm{T}}\dot{P} = P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P$, which completes the proof. □

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

Theorem 9 (*Adopted from Brockett (1991)*). *For any $P(0) \in \mathcal{O}_n$, consider the gradient flow (4). Then, $\lim_{t\to\infty} P(t) = P_\infty$ exists and is a orthogonal matrix of the form $P_\infty = V\Pi SU^{\mathrm{T}}$ that locally minimizes the value of the objective function V_1, where U, V orthogonal, Π a permutation matrix and S a square root of the identity matrix, i.e., $S = \mathrm{diag}(\pm 1, \ldots, \pm 1)$.*

[1] For any matrix $\mathcal{F}_V(P)$, note that $\mathbf{tr}\left(\mathcal{F}_V(P)^{\mathrm{T}}\Omega\right) = \frac{1}{2}\mathbf{tr}\left((\mathcal{F}_V(P)^{\mathrm{T}} - \mathcal{F}_V(P))\Omega\right)$, where $(\mathcal{F}_V(P)^{\mathrm{T}} - \mathcal{F}_V(P))$ is skew-symmetric. This follows from the fact that skew-symmetric and symmetric matrices consist an orthogonal decomposition of the set of all matrices.

[2] Recall that $\mathcal{F}_V(P)$ is skew-symmetric.

Proof. Since $(P^TA_2PA_1 - A_1P^TA_2P)$ is skew-symmetric for all $t \geq 0$, $P(t)$ is orthogonal for all $t \geq 0$, by Lemma 7. Let $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ be a Lyapunov function candidate for our problem. Clearly, $V_1(P) \geq 0$ for all P with equality if and only if P is a permutation matrix and A_1, A_2 are isomorphic. Expanding $V_1(P)$ we get

$$\begin{aligned} V_1(P) &= \frac{1}{2}\mathbf{tr}(PA_1 - A_2P)(PA_1 - A_2P)^T \\ &= \frac{1}{2}\mathbf{tr}(A_1^2 + A_2^2) - \mathbf{tr}(PA_1P^TA_2). \end{aligned}$$

To simplify notation, let $X_1(P) = P^TA_2PA_1 - A_1P^TA_2P$ with $X_1^T = -X_1$. Then, $\dot{P} = PX_1$ and the time derivative of $V_1(P)$ becomes

$$\begin{aligned} \dot{V}_1(P) &= -\mathbf{tr}(\dot{P}A_1P^TA_2 + PA_1\dot{P}^TA_2) \\ &= -\mathbf{tr}\left(PX_1A_1P^TA_2 - PA_1X_1P^TA_2\right) \\ &= -\mathbf{tr}\left(A_1P^TA_2P - P^TA_2PA_1\right)X_1 \\ &= -\mathbf{tr}\, X_1^TX_1 = -\|X_1\|_F^2. \end{aligned}$$

Hence, $\dot{V}_1(P)$ is non-increasing, which implies that P asymptotically converges to the set $\mathcal{C} = \{P \mid \dot{V}_1(P) = 0\} = \{P \mid P^TA_2PA_1 = A_1P^TA_2P\}$ of critical points of the flow (4).

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be the eigenvalues of A_1 and A_2 respectively and let $A_1 = U\Lambda U^T$ and $A_2 = VMV^T$ be their corresponding eigendecompositions, with U and V orthogonal matrices and $\Lambda = \text{diag}(\lambda_i)$ and $M = \text{diag}(\mu_i)$. We have

$$\begin{aligned} &P^TA_2PA_1 = A_1P^TA_2P \\ &P^TVMV^TPU\Lambda U^T = U\Lambda U^TP^TVMV^TP \\ &MV^TPU\Lambda U^TP^TV = V^TPU\Lambda U^TP^TVM \\ &M\Theta = \Theta M, \end{aligned}$$

where $\Theta = V^TPU\Lambda U^TP^TV$. Since the (i,j)th element of $M\Theta = \Theta M$ is $(\mu_i - \mu_j)\theta_{ij}$ we see that $M\Theta = \Theta M$ vanishes only when Θ is diagonal. Since V^TPU is orthogonal, Λ and Θ have the same eigenvalues and since they are both diagonal matrices, Θ should result from some permutation on the matrix Λ. In other words, V^TPU has to be a matrix of the form $V^TPU = \Pi S$, where Π is a permutation matrix and S is a square root of the identity matrix, i.e., $S = \text{diag}(\pm 1, \ldots, \pm 1)$. Hence, $P = V\Pi SU^T$, which further implies that the set of critical points $\mathcal{C}$ of Eq. (4) is a finite set consisting of $2^n n!$ elements. Hence, the limit P_∞ exists. □

3.2. Converging to a permutation matrix

By Lemma 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 element-wise non-negative. Hence, we need to define a cost function that penalizes negative entries in P. Inspired by the well-known *Big M* method (Hillier & Lieberman, 1995), where multiplication of certain variables in the cost function by large weights forces them to become either negative or positive in the final optimal solution, let $V_2 : \mathcal{O}_n \to \mathbb{R}$ be defined by

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

where $A \circ B$ denotes the *Hadamard* or element-wise 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 $P^TP = PP^T = I$ and so, $V_2(P) = \frac{2n}{3} - \frac{2}{3}\sum_{i,j=1}^n p_{ij}^3$. Due to multiplication of the entries p_{ij} of P in $V_2(P)$ by positive weights $p_{ij}^2 > 0$, minimizing $V_2(P)$ forces them to be as "positive" as possible, according to the *Big M* concept. In particular, we show that the gradient flow on the cost function defined in (5) indeed converges to a permutation matrix. The following proposition describes the gradient flow of $V_2(P)$.

Proposition 10. *Assuming the standard metric on the orthogonal group, the gradient flow of the function $V_2 : \mathcal{O}_n \to \mathbb{R}$ is given by*

$$\dot{P} = P\left(P^T(P \circ P) - (P \circ P)^TP\right). \tag{6}$$

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

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

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

$$X_1(P(I + \Omega)) \approx \mathbf{tr}(P \circ P)^TP + \mathbf{tr}\left((P \circ P)^TP - 2P^T(P \circ P)\right)\Omega, \tag{8}$$

where we have neglected terms of the order of Ω^2 and have made use of the identity $\mathbf{tr}(P^T \circ \Omega P^T)P = \sum_{i,k} p_{ki}\left(\sum_j \omega_{ij}p_{kj}\right)p_{ki} = \sum_{k,i} p_{ki}^2\left(\sum_j \omega_{ij}p_{kj}\right) = \mathbf{tr}(P \circ P)\Omega P^T$. Similarly,

$$X_2(P(I + \Omega)) \approx \mathbf{tr}\, P^T(P \circ P) + \mathbf{tr}\left(2(P \circ P)^TP - P^T(P \circ P)\right)\Omega, \tag{9}$$

where again we have neglected terms of the order of Ω^2 and have made use of the identity $\mathbf{tr}\, P^T(P \circ P\Omega) = \sum_{i,k} p_{ki}p_{ki}\left(\sum_j p_{kj}\omega_{ji}\right) = \sum_{i,k} p_{ki}^2\left(\sum_j p_{kj}\omega_{ji}\right) = \mathbf{tr}(P \circ P)^TP\Omega$. Substituting Eqs. (8) and (9) in (7) we get

$$V_2(P(I + \Omega)) \approx V_2(P) + \mathbf{tr}\left(P^T(P \circ P) - (P \circ P)^TP\right)\Omega.$$

As before, we conclude that the quantity $\langle(P^T(P \circ P) - (P \circ P)^TP), \cdot\rangle$ represents the negative gradient of V_2 at P. Using $\dot{P} = P\Omega$ we can express the gradient flow as $P^T\dot{P} = P^T(P \circ P) - (P \circ P)^TP$, which completes the proof. □

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 11. *Let $V_2(P) = \frac{2}{3}\mathbf{tr}\, P^T(P - (P \circ P))$. 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 12. *For any $P(0) \in \mathcal{O}_n$, consider the gradient flow (6) and, as before, let $V_2(P) = \frac{2}{3}\mathbf{tr}\, P^T(P - (P \circ P))$. 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 $P^T(P\circ P)-(P\circ P)^TP$ is skew-symmetric for all $t\geq 0$, $P(t)$ is orthogonal for all $t\geq 0$, by Lemma 7. From Eq. (7) we have

$$V_2(P)=\frac{2n}{3}-\frac{1}{3}\mathbf{tr}\left((P\circ P)^TP+P^T(P\circ P)\right).$$

To simplify notation, let $X_2(P)=P^T(P\circ P)-(P\circ P)^TP$ with $X_2^T=-X_2$. Then, $\dot{P}=PX_2$ and the time derivative of $V_2(P)$ becomes

$$\begin{aligned}\dot{V}_2(P)&=-\frac{1}{3}\mathbf{tr}\left(2(P\circ\dot{P})^TP+(P\circ P)^T\dot{P}+\dot{P}^T(P\circ P)+2P^T(P\circ\dot{P})\right)\\&=\mathbf{tr}\left(P^T(P\circ P)-(P\circ P)^TP\right)X_2\\&=\mathbf{tr}\,X_2^2=-\mathbf{tr}\,X_2X_2^T=-\|X_2\|_F^2,\end{aligned}$$

where we have also used the identities $\mathbf{tr}(P^T\circ X_2P^T)P=\mathbf{tr}(P\circ P)X_2P^T$ and $\mathbf{tr}\,P^T(P\circ PX_2)=\mathbf{tr}(P\circ P)^TPX_2$. Hence, $\dot{V}_2(P)\leq 0$ for all $t\geq 0$ with $\dot{V}_2(P)=0$ if and only if $(P\circ P)^TP=P^T(P\circ P)$. In the rest of this proof we will explicitly describe the orthogonal matrices P that satisfy $(P\circ P)^TP=P^T(P\circ P)$.

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 $\Sigma=\mathrm{diag}(\sigma_i)$. Since, $(P\circ P)^TP=P^T(P\circ P)$ we have that

$$\begin{aligned}(P\circ P)^T(P\circ P)&=(P\circ P)^TPP^T(P\circ P)\\&=P^T(P\circ P)(P\circ P)^TP\\&=P^TU\Sigma V^TV\Sigma U^TP=P^TU\Sigma^2U^TP.\end{aligned}$$

Moreover, $(P\circ P)^T(P\circ P)=V\Sigma U^TU\Sigma V^T=V\Sigma^2V^T$ and hence, $V\Sigma^2V^T=P^TU\Sigma^2U^TP$, which implies that $\Sigma^2=V^TP^TU\Sigma^2U^TPV$ or equivalently, that $U^TPV=S$ and so, $P=USV^T$ for some square root of the identity matrix S. In other words, 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 Eq. (10), by Corollary 17 (see Appendix). Moreover, by uniqueness of the singular values, $\Sigma=(S\circ S)=I$ is the unique solution to Eq. (10). Hence, by Corollary 17 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.[3] Hence, Eq. (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 13. *Consider the gradient flow (6) and let $\mathcal{C}=\{P\mid\dot{V}_2(P)=0\}$ be its set of critical points. Then, the only stable critical points are the permutation matrices.*

Proof. In order to characterize the critical points of $\dot{P}=P\left(P^T(P\circ P)-(P\circ P)^TP\right)$ we will study the linearization of the system in a neighborhood of a critical point. Using orthogonality of P we have $\dot{P}=(P\circ P)-P(P\circ P)^TP$, which expressed elementwise becomes

$$\dot{p}_{ij}=p_{ij}^2-\sum_{k=1}^n p_{ik}\sum_{m=1}^n p_{mk}^2p_{mj}.$$

Let $\bar{p}=[p_{11}\ p_{12}\ \cdots\ p_{(n-1)n}\ p_{nn}]^T$ be the $n^2\times 1$ vector resulting from stacking the columns of P in a single column vector and define

$$f_{ij}(\bar{p})\triangleq p_{ij}^2-\sum_{k=1}^n p_{ik}\sum_{m=1}^n p_{mk}^2p_{mj}.$$

Let $\bar{f}(\bar{p})=[f_{11}(\bar{p})\ f_{12}(\bar{p})\ \cdots\ f_{(n-1)n}(\bar{p})\ f_{nn}(\bar{p})]^T$ be the $n^2\times 1$ vector resulting from stacking all the functions $f_{ij}(\bar{p})$ in a single vector. Then, the linearization of the system in a neighborhood of a critical point becomes

$$\dot{\bar{p}}=\left.\frac{\partial\bar{f}(\bar{p})}{\partial\bar{p}}\right|_{\mathcal{C}}\bar{p}. \tag{11}$$

Hence, we need to compute the partial derivative of $f_{ij}(\bar{p})$ with respect to p_{st} and evaluate it at a critical point. Let

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

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

$$\begin{aligned}\frac{\partial f_{ij}(\bar{p})}{\partial p_{st}}&=\frac{\partial p_{ij}^2}{\partial p_{st}}-\sum_{k,m=1}^n\frac{\partial}{\partial p_{st}}(p_{ik}p_{mk}^2p_{mj})\\&=2\delta_{ij,st}p_{ij}-\sum_{k,m=1}^n\big(\delta_{ik,st}p_{mk}^2p_{mj}\\&\quad+2\delta_{mk,st}p_{ik}p_{mk}p_{mj}+\delta_{mj,st}p_{ik}p_{mk}^2\big)\\&=2\delta_{ij,st}p_{ij}-\sum_{m=1}^n\delta_{i,s}p_{mt}^2p_{mj}-2p_{it}p_{st}p_{sj}-\sum_{k=1}^n\delta_{j,t}p_{ik}p_{sk}^2.\end{aligned}$$

From Lemma 12, the set of critical points is $\mathcal{C}=\{P\mid\dot{V}_2(P)=0\}=\{P\mid P=S\Pi\}$, where $S=\mathrm{diag}(s_i)$ is a square root of the identity matrix and $\Pi=(\pi_{ij})$ is a permutation matrix. Hence, at a critical point we have $p_{ij}=s_i\pi_{ij}$ and evaluating the partial derivative $\frac{\partial f_{ij}(\bar{p})}{\partial p_{st}}$ at that point we get

$$\begin{aligned}\left.\frac{\partial f_{ij}(\bar{p})}{\partial p_{st}}\right|_{\mathcal{C}}&=2\delta_{ij,st}s_i\pi_{ij}-\sum_{m=1}^n\delta_{i,s}\delta_{j,t}(s_m\pi_{mj})^3\\&\quad-2\delta_{ij,st}(s_i\pi_{ij})^3-\sum_{k=1}^n\delta_{j,t}\delta_{i,s}(s_i\pi_{ik})^3\\&=-\delta_{ij,st}\left(s_i+\sum_{m=1}^n s_m\pi_{mj}\right).\end{aligned}$$

So, the linearization (11) becomes

$$\dot{p}_{ij}=-\left(s_i+\sum_{m=1}^n s_m\pi_{mj}\right)p_{ij}\quad\text{for all } i,j. \tag{12}$$

Suppose that $s_i=1$ for all i. Then, $\dot{p}_{ij}=-2p_{ij}$ for all i,j and hence, the system (12) 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, which completes the proof. □

We can summarize the results of Lemmas 11–13 in the following theorem.

Theorem 14. *Consider the gradient flow $\dot{P}=P(P^T(P\circ P)-(P\circ P)^TP)$, defined in (6). Then $\lim_{t\to\infty}P(t)=P_\infty$ exists and, for almost all initial conditions $P(0)\in\mathcal{O}_n$, is a permutation matrix.*

3.3. Superposition of the gradient flows

Up to this point we have defined two gradient flows on the space of orthogonal matrices that minimize their associated cost functions, while the second one also converges to a permutation

[3] Note that, by Lemma 16 (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 Eq. (10), this would imply that $\Sigma=S$ which contradicts the requirement that $\Sigma\geq 0$.

matrix. It is reasonable, thus, to expect that by combining these two gradient flows we can achieve the desired objective in Problem 6. 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 initially ignores the non-negativity requirement and switches to the permutation gradient flow when the objective has been sufficiently minimized.

Theorem 15. *Consider the convex combination of the gradient flows* (4) and (6)

$$\dot{P} = (1-k)P\left(P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P\right) + kP\left(P^{\mathrm{T}}(P\circ P) - (P\circ P)^{\mathrm{T}}P\right), \quad (13)$$

where $0 < k < 1$*. Then, for* k *sufficiently close to 1,* $\lim_{t\to\infty} P(t) = P_\infty$ *exists and, for almost all initial conditions* $P(0) \in \mathcal{O}_n$*, approximates a permutation matrix that locally minimizes the cost* $V_1(P)$*.*

Proof. To simplify notation, let $X_1(P) = P^{\mathrm{T}}A_2PA_1 - A_1P^{\mathrm{T}}A_2P$ and $X_2(P) = P^{\mathrm{T}}(P\circ P) - (P\circ P)^{\mathrm{T}}P$, where $X_1(P)$ and $X_2(P)$ are skew-symmetric matrices. Then, the system dynamics (13) become

$$\dot{P} = (1-k)PX_1 + kPX_2 = P((1-k)X_1 + kX_2).$$

Let $V : \mathcal{O}_n \to \mathbb{R}$, such that $V(P) = (1-k)V_1(P) + kV_2(P)$ be a Lyapunov function candidate for the system, where $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ and $V_2(P) = \frac{2}{3}\mathbf{tr}\, P^{\mathrm{T}}(P - (P\circ P))$, as they were defined earlier. Since, $V_1(P) \geq 0$ and $V_2(P) \geq 0$ for all $P \in \mathcal{O}_n$, we also have that $V(P) \geq 0$ for all $P \in \mathcal{O}_n$ and $V(P) = 0$ if and only if the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic (in which case P is exactly a permutation matrix). The time derivative of $V(P)$ is $\dot{V}(P) = (1-k)\dot{V}_1(P) + k\dot{V}_2(P)$ and, as before we can show that $\dot{V}_1(P) = -\mathbf{tr}\, X_1((1-k)X_1 + kX_2)^{\mathrm{T}}$ and $\dot{V}_2(P) = -\mathbf{tr}\, X_2((1-k)X_1 + kX_2)^{\mathrm{T}}$. Hence,

$$\begin{aligned}\dot{V}(P) &= -\mathbf{tr}((1-k)X_1 + kX_2)((1-k)X_1 + kX_2)^{\mathrm{T}} \\ &= -\|(1-k)X_1 + kX_2\|_F^2,\end{aligned}$$

which implies that $\dot{V}(P)$ is non-increasing and so P will converge to a local minimum, P_∞. The set of critical points of the flow (13) is $\mathcal{C} = \{P \mid (1-k)X_1(P) = -kX_2(P),\ 0 < k < 1\}$. Clearly, if $X_2(P_\infty) = 0$, then $P_\infty = S\Pi$, with $S = \mathrm{diag}(\pm 1, \ldots, \pm 1)$ and Π a permutation matrix (Lemma 12). Hence, as in Lemma 13, linearizing (13) around $\mathcal{C}$, we can show that for k sufficiently close to 1, 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 $\|X_2(P_\infty)\|_2 < \epsilon$. Since, $P_\infty \in \mathcal{C}$ the equation $(1-k)X_1(P_\infty) = -kX_2(P_\infty)$ implies that $(1-k)\|X_1(P_\infty)\|_2 = k\|X_2(P_\infty)\|_2$. Hence, $\|X_2(P_\infty)\|_2 < \epsilon$ if and only if $\frac{(1-k)}{k}\|X_1(P_\infty)\|_2 < \epsilon$. So, by choosing, $k > \frac{\frac{1}{\epsilon}\max_{P\in\mathcal{O}_n}\|X_1(P)\|_2}{1+\frac{1}{\epsilon}\max_{P\in\mathcal{O}_n}\|X_1(P)\|_2}$ we can guarantee that P_∞ will be as close as we want to a permutation matrix. Clearly, the closer k is to 1, the smaller ϵ can be. Note at this point that $\max_{P\in\mathcal{O}_n}\|X_1(P)\|_2$ is bounded since $\|X_1(P)\|_2$ 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 15 that if we want P to converge exactly to a permutation matrix, we should choose $k \to 1$. 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 trade-off 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, 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

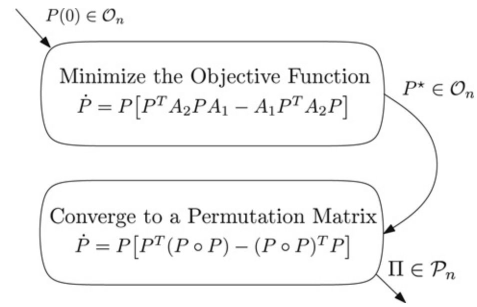

Fig. 2. Switched scheme.

matrix. In other words, initialization of the problem is important. In order to take advantage of this fact, we can think of the following switched scheme for our problem (Fig. 2).

The idea is to first apply the gradient flow (4) to minimize the objective function V_1, and use its solution $P^\star$ as a "good" initial condition for the gradient flow (6) in order to converge to a permutation matrix Π. In the following section we implement our dynamical systems approach to various non-trivial weighted graph matching problems.

4. Simulation results

In the previous sections we developed two provably correct gradient flows on the space of orthogonal matrices and discussed how they can be combined to solve 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 (with k sufficiently close to 1) in order to converge to a permutation matrix Π.

We implemented our algorithm using random weighted adjacency matrices A_1 and A_2, and initializing the first phase of our hybrid scheme with random orthogonal matrices (we sampled both connected components of $\mathcal{O}_n$). Below we present 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.

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

The following table shows how the objective function $V_1(P) = \frac{1}{2}\|PA_1 - A_2P\|_F^2$ varies after each one of the two phases of the proposed hybrid scheme, i.e., for $P = P^\star$ and $P = \Pi$, applied to an instance of the previous problem.

Det	Initialization	Phase I	Phase II
−1	34.8605	0.1527	5.9719
1	36.5412	0.1527	5.9672

Observe 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 $10! = 3628800$ 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.

Det	Initialization	Phase I	Phase II
−1	800.6253	0.3473	162.0595
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 2 GHz with 1 Gb memory laptop, using Matlab 6 and low level programming (Runge–Kutta 4 with constant step size), it took on average 30 min for the first phase to converge 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.

5. Conclusions

In this paper, we considered the problem of finding the optimal relabeling 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 using an equivalent representation for the set of permutation matrices as the intersection of the space of orthogonal matrices with the set of element-wise 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 for the weighted graph matching problem, as well as initialization issues that lead to a hybrid scheme consisting of sequentially combining the two flows. 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.

Appendix. Factoring properties of Hadamard products

Lemma 16. *Let $V \in \mathcal{O}_n$. Then, $(UV^{\mathrm{T}} \circ UV^{\mathrm{T}}) = (U \circ U)V^{\mathrm{T}}$ for all $U \in \mathcal{O}_n$ if and only if V is a permutation matrix.*

Proof. Let $V = (v_{ij})$ and $U = (u_{ij})$. Clearly, if $V \in \mathcal{P}_n$, the equality $(UV^{\mathrm{T}} \circ UV^{\mathrm{T}}) = (U \circ U)V^{\mathrm{T}}$ expressed elementwise gives $\left(\sum_{k=1}^n u_{ik}v_{jk}\right)^2 = \sum_{k=1}^n u_{ik}^2 v_{jk}$, for all i, j. Since V is a permutation matrix, there exists exactly one index m such that $v_{jm} = 1$ and $v_{jk} = 0$ for all $k \neq m$. Hence, the previous equality is identically true for all $U \in \mathcal{O}_n$. Now, suppose that $V \in \mathcal{O}_n$ but $V \notin \mathcal{P}_n$. By Lemma 5 there exists indices j and m such that $v_{jm} < 0$. We need to show that there exists a $U \in \mathcal{O}_n$ such that the previous equality is not true. In particular, let $U \in \mathcal{P}_n$ such that $u_{im} = 1$. Then, the equality $(UV^{\mathrm{T}} \circ UV^{\mathrm{T}}) = (U \circ U)V^{\mathrm{T}}$ expressed element-wise gives $v_{jm}^2 = v_{jm}$ which contradicts the assumption that $v_{jm} < 0$. Hence, V has to be a permutation matrix, which completes the proof. □

Corollary 17. *Let S be a square root of the identity matrix, i.e., $S = \mathrm{diag}(\pm 1, \ldots, \pm 1)$. Then, $(USV^{\mathrm{T}} \circ USV^{\mathrm{T}}) = U(S \circ S)V^{\mathrm{T}}$ for all S if and only if U and V are permutation matrices.*

Proof. By Lemma 16, $(USV^{\mathrm{T}} \circ USV^{\mathrm{T}}) = (US \circ US)V^{\mathrm{T}}$ for all $US \in \mathcal{O}_n$ if and only if V is a permutation matrix. Similarly, $(US \circ US) = U(S \circ S)$ if and only if U is a permutation matrix. Hence, $(USV^{\mathrm{T}} \circ USV^{\mathrm{T}}) = U(S \circ S)V^{\mathrm{T}}$ for all S if and only if U and V are permutation matrices. □

References

Bloch, A. M., Brockett, R. W., & Crouch, P. E. (1997). Double bracket equations and geodesic flows on symmetric spaces. *Communications on Mathematical Physics*, *187*, 357–373.

Brockett, R. W. (1991). Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. *Linear Algebra and Its Applications*, *146*, 79–91.

Brockett, R. W. (1989). Least squares matching problems. *Linear Algebra and its Applications*, *122 124*, 761–777.

Chu, M. T. (1992). Matrix differential equations: A continuous realization process for linear algebra problems. *Nonlinear Analysis*, *18*, 1125–1146.

Chu, M. T., & Driessel, K. R. (1990). The projected gradient method for least squares matrix approximations with spectral constraints. *SIAM Journal on Numerical Analysis*, *27*, 1050–1060.

Cortes, J., Martinez, S., & Bullo, F. (2006). Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, *51*(8), 1289–1298.

Faye, A., & Roupin, F. (2005). A cutting planes algorithm based upon a semidefinite relaxation for the quadratic assignment problem. *Lecture Notes in Computer Science*, *3669*, 850–861.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of np-completeness*. New York, NY: W. H. Freeman & Co.

Godsil, C., & Royle, G. (2001). *Algebraic graph theory*. New York, NY: Springer-Verlag.

Gold, S., & Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(4), 377–388.

Helmke, U., Moore, J. B., & Brockett, R. W. (1996). *Optimization and dynamical systems*. New York, NY: Springer-Verlag.

Hillier, F. S., & Lieberman, G. J. (1995). *Introduction to mathematical programming*. New York, NY: McGraw-Hill.

Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge University Press.

Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, *48*(6), 988–1001.

Mesbahi, M. (2005). On state-dependent dynamic graphs and their controllability properties. *IEEE Transactions on Automatic Control*, *50*(3), 387–392.

Muhammad, A., & Egerstedt, M. (2005). Connectivity graphs as models of local interactions. *Journal of Applied Mathematics and Computation*, *168*(1), 243–269.

Olfati-Saber, R., & Murray, R. M. (2003). Agreement problems in networks with directed graphs and switching topology. In *Proceedings of the 42nd IEEE conference on decision and control* (pp. 4126–4132).

Rangarajan, A., & Mjolsness, E. (1996). A lagrangian relaxation network for graph matching. *IEEE Transactions on Neural Networks*, *7*(6), 1365–1381.

Smith, S. L., & Bullo, F. (2006). Target assignment for robotic networks: Asymptotic performance under limited communication. In *Proceedings of the American control conference* (pp. 1155–1160).

Tanner, H. G., Jadbabaie, A., & Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, *52*(5), 863–868.

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(5), 695–703.

Wolkowicz, H. (2000). Semidefinite programming approaches to the quadratic assignment problem. In *Nonlinear assignment problems: Algorithms and applications (combinatorial optimization)* (pp. 143–174). Norwell, MA: Kluwer.

Zavlanos, M. M., & Pappas, G. J. (2006). A dynamical systems approach to weighted graph matching. In *Proceedings of the 45th IEEE conference on decision and control* (pp. 3492–3497).
Zavlanos, M. M., & Pappas, G. J. (2007). Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics*, *23*(4), 812–816.
Zavlanos, M. M., & Pappas, G. J. (2008). Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, *24*(1), 232–242.
Zavlanos, M. M., & Pappas, G. J. (2007). Distributed formation control with permutation symmetries. In *Proceedings of the 46th IEEE conference on decision and control* (pp. 2894–2899).

Michael M. Zavlanos (S'05) received the Diploma in mechanical engineering from the National Technical University of Athens, Athens, Greece, and the M.S.E. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, in 2005 and 2008, respectively. Currently, he is a postdoctoral researcher in the Department of Electrical and Systems Engineering, University of Pennsylvania. His current research interests include the areas of distributed control systems, networked systems, and hybrid dynamical systems with applications to robotics, sensor networks and biomolecular networks. Mr. Zavlanos was a finalist of the Best Student Paper Award at the 45th IEEE Conference on Decision and Control, 2006.

George J. Pappas (S'91-M'98-SM'04) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, in 1998. He is currently a Professor at the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia. He also holds secondary appointments in the Departments of Computer and Information Sciences, and Mechanical Engineering and Applied Mechanics, and is a member and a former Director of the General Robotics and Active Sensory Perception (GRASP) Laboratory. He currently serves as the Deputy Dean in the School of Engineering and Applied Science. He is the Co-Editor of *Hybrid Systems: Computation and Control* (Springer-Verlag, 2004). His current research interests include the areas of hybrid and embedded systems, hierarchical control systems, distributed control systems, nonlinear control systems, and geometric control theory, with applications to robotics, unmanned aerial vehicles, and biomolecular networks.

Prof. Pappas was the recipient of numerous awards including the National Science Foundation (NSF) Career Award in 2002, the NSF Presidential Early Career Award for Scientists and Engineers (PECASE) in 2002, and the Eliahu Jury Award for Excellence in Systems Research from the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley in 1999. His and his students' papers were finalists for the Best Student Paper Award at the IEEE Conference on Decision and Control in 1998, 2001, 2004, and 2006, the American Control Conference in 2001 and 2004, and the IEEE Conference on Robotics and Automation in 2007.