Matching Free Trees, Maximal Cliques, and Monotone Game Dynamics *

Marcello Pelillo

Dipartimento di Informatica

Università Ca' Foscari di Venezia

Via Torino 155, 30172 Venezia Mestre, Italy

E-mail: pelillo@dsi.unive.it

January 23, 2002

Abstract

Motivated by our recent work on rooted tree matching, in this paper we provide a solution to the problem of matching two free (i.e., unrooted) trees by constructing an association graph whose maximal cliques are in one-to-one correspondence with maximal common subtrees. We then solve the problem using simple payoff-monotonic dynamics from evolutionary game theory. We illustrate the power of the approach by matching articulated and deformed shapes described by shape-axis trees. Experiments on hundreds of larger, uniformly random trees are also presented. The results are impressive: despite the inherent inability of these simple dynamics to escape from local optima, they always returned a globally optimal solution.

Index Terms—Graph matching, combinatorial optimization, quadratic programming, dynamical systems, evolutionary game theory, shape recognition.

IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)

1 Introduction

Graph matching is a classic problem in computer vision and pattern recognition, instances of which arise in areas as diverse as object recognition, motion and stereo analysis. A well-known approach to solving this problem consists of transforming it into the equivalent problem of finding a maximum clique in an auxiliary graph structure, known as the *association graph* [2]. This framework is attractive because it casts graph matching as a pure graph-theoretic problem, for which a solid theory and powerful algorithms have been developed. Note that, although the maximum clique problem is known to be NP-hard, powerful heuristics exist which efficiently find good approximate solutions, and there exist several classes of graphs for which the problem can be solved in polynomial time [5].

In many computer vision problems the graphs at hand have a peculiar structure: they are connected and acyclic, i.e. they are *free trees* (see, e.g., [3, 16, 18, 26]). Other application domains where free trees arise quite frequently are pattern recognition [9] and biochemistry [1]. Note that, unlike "rooted" trees, in free trees there is no distinguished node playing the role of the root, and hence no hierarchy is imposed on them. Since in the standard association graph formulation the solutions are not constrained to preserve connectedness, it is not clear how to apply the framework in these cases, and the extension of association graph techniques to free tree matching problems is therefore of considerable interest.

Motivated by our recent work on rooted tree matching [21], in this paper we propose a solution to this problem by providing a straightforward way of deriving an association graph from two free trees. We prove that in the new formulation there is a one-to-one correspondence between maximal (maximum) cliques in the derived association graph and maximal (maximum) subtree isomorphisms. As an obvious corollary, the computational complexity of finding a maximum clique in such graphs is therefore the same as the maximum common subtree problem, which is known to be polynomial in the number of nodes [11].

Following [20, 21], we use a recent generalization of the Motzkin-Straus theorem [19] to formulate the maximum clique problem as a quadratic programming problem. To (approximately) solve it we employ *payoff-monotonic dynamics*, a class of simple dynamical systems recently developed and studied in evolutionary game theory [14, 23]. Such continuous solutions to discrete problems are interesting as they can motivate analog and biological implementations. It is worth remarking that traditional energy-minimization graph matching algorithms such as [12] are not applicable to the tree matching problem (either rooted or unrooted) since they suffer from the very same problem as standard association graph

techniques, i.e., they fail to preserve connectedness.

We illustrate the power of our approach via several examples of matching articulated and deformed shapes described by *shape-axis* trees [18]. We also present experiments on hundreds of much larger uniformly random trees, and study the sensitivity of the method to structural perturbations. The results are impressive: despite the counter-intuitive maximum clique formulation of the tree matching problem, and the inherent inability of these simple dynamics to escape from local optima, they always found a globally optimal solution.

2 Subtree isomorphisms and maximal cliques

Let $G = (V, E)$ be a graph, where V is the set of nodes (or vertices) and E is the set of undirected edges. The *order* of G is the number of nodes in V, while its *size* is the number of edges. Two nodes $u, v \in V$ are said to be *adjacent* (denoted $u \sim v$) if they are connected by an edge. The *adjacency matrix* of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$ defined as

$$a_{ij} = \begin{cases} 1, & \text{if } v_i \sim v_j \\ 0, & \text{otherwise .} \end{cases}$$

The *degree* of a node u, denoted $\deg(u)$, is the number of nodes adjacent to it. A *path* is any sequence of distinct nodes $u_0 u_1 \ldots u_n$ such that for all $i = 1 \ldots n$, $u_{i-1} \sim u_i$; in this case, the *length* of the path is n. If $u_n \sim u_0$ the path is called a *cycle*. A graph is said to be *connected* if any two nodes are joined by a path. The *distance* between two nodes u and v, denoted by $d(u, v)$, is the length of the shortest path joining them (by convention $d(u, v) = \infty$, if there is no such path). Given a subset of nodes $C \subseteq V$, the *induced subgraph* $G[C]$ is the graph having C as its node set, and two nodes are adjacent in $G[C]$ if and only if they are adjacent in G. A connected graph with no cycles is called a *free tree*, or simply a *tree*.

Trees have a number of interesting properties. One which turns out to be very useful is that in a tree any two nodes are connected by a unique path. Another useful property is that, given three distinct vertices u, v, z of a tree, there exists a unique vertex x that lies on the uv-path, on the uz-path and on the vz-path (see Figure 1). The vertex x is sometimes referred to as the *median* of u, v, z. Note that the median of u, v, z can well be one of the vertices u, v, z.

Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two trees. Any bijection $\phi : H_1 \to H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is called a *subtree isomorphism* if it preserves both the adjacency relationships between the nodes and the connectedness of the matched subgraphs. Formally,

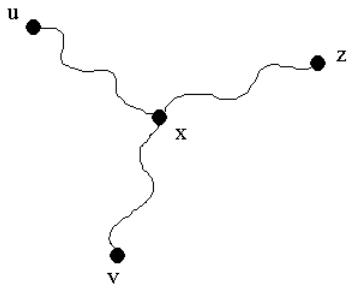

Figure 1: The median of three vertices.

this means that, given $u, v \in H_1$, we have $u \sim v$ if and only if $\phi(u) \sim \phi(v)$ and, in addition, the induced subgraphs $T_1[H_1]$ and $T_2[H_2]$ are connected. A subtree isomorphism is *maximal* if there is no other subtree isomorphism $\phi' : H_1' \to H_2'$ with H_1 a strict subset of H_1', and *maximum* if H_1 has largest cardinality. The maximal (maximum) subtree isomorphism problem is to find a maximal (maximum) subtree isomorphism between two trees. A word of caution about terminology is in order here. Despite name similarity, we are not addressing the so-called subtree isomorphism problem, which consists of determining whether a given tree is isomorphic to a subtree of a larger one. In fact, we are dealing with a generalization thereof, the maximum common subtree problem, which consists of determining the largest isomorphic subtrees of two given trees. We shall continue to use our own terminology, however, as it emphasizes the role of the isomorphism ϕ.

The *free tree association graph* (FTAG) of two trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ is the graph $G = (V, E)$ where

$$V = V_1 \times V_2 \tag{1}$$

and, for any two nodes (u, w) and (v, z) in V, we have

$$(u, w) \sim (v, z) \Leftrightarrow d(u, v) = d(w, z) \ . \tag{2}$$

Note that this definition of the association graph is stronger than the standard one used for matching arbitrary relational structures [2].

A subset of vertices of a graph G is said to be a *clique* if all its nodes are mutually adjacent. A *maximal* clique is one which is not contained in any larger clique, while a *maximum* clique is a clique having largest cardinality. The maximum clique problem is to find a maximum clique of G [5]. The main result of this section establishes a one-to-one

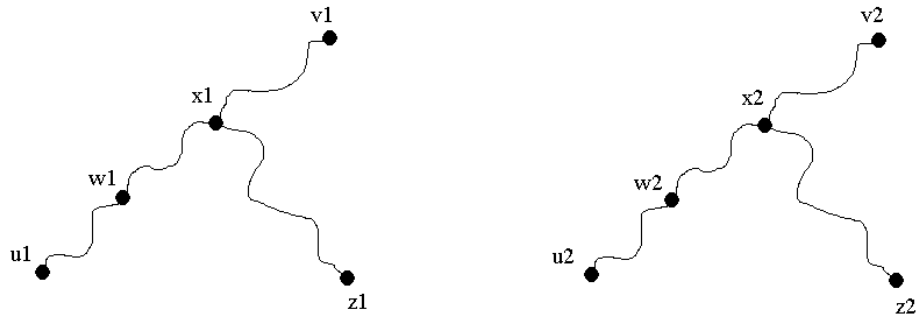

Figure 2: Illustration of the hypotheses of Lemma 1 (see text for explanation).

correspondence between maximal/maximum subtree isomorphisms and maximal/maximum cliques in the FTAG. To prove it we first need the following lemma.

Lemma 1 *Let $u_1, v_1, w_1, z_1 \in V_1$ and $u_2, v_2, w_2, z_2 \in V_2$ be distinct nodes of trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$, and suppose that the following conditions hold:*

1. *w_1 is on the u_1v_1-path and w_2 is on the u_2v_2-path*
2. *$d(u_1, w_1) = d(u_2, w_2)$*
3. *$d(w_1, v_1) = d(w_2, v_2)$*
4. *$d(u_1, z_1) = d(u_2, z_2)$*
5. *$d(v_1, z_1) = d(v_2, z_2)$*

Then, $d(w_1, z_1) = d(w_2, z_2)$.

Proof. Let x_i denote the median of u_i, v_i and z_i $(i = 1, 2)$ and suppose, without loss of generality, that x_1 lies on the w_1v_1-path, i.e. $d(u_1, w_1) \leq d(u_1, x_1)$ or, equivalently, $d(v_1, w_1) \geq d(v_1, x_1)$. It is easy to show that x_2 is on the w_2v_2-path (see Figure 2). Indeed, suppose to the contrary that this is not the case, i.e. $d(u_2, x_2) < d(u_2, w_2)$ or, equivalently, $d(v_2, x_2) > d(v_2, w_2)$. Then, from 4) and 2) we get:

$$\begin{aligned} d(x_1, z_1) - d(x_2, z_2) &= d(u_2, x_2) - d(u_1, x_1) \\ &< d(u_2, w_2) - d(u_1, x_1) \\ &= d(u_1, w_1) - d(u_1, x_1) \\ &\leq 0 \ . \end{aligned}$$

Hence, $d(x_1, z_1) < d(x_2, z_2)$. On the other hand, hypotheses 5) and 3) yield:

$$\begin{aligned} d(x_1, z_1) - d(x_2, z_2) &= d(v_2, x_2) - d(v_1, x_1) \\ &\geq d(v_2, x_2) - d(v_1, w_1) \\ &= d(v_2, x_2) - d(v_2, w_2) \\ &> 0 \end{aligned}$$

which implies $d(x_1, z_1) > d(x_2, z_2)$, a contradiction.

Therefore, w_i is on the $u_i z_i$-path ($i = 1, 2$) and hence:

$$\begin{aligned} d(w_1, z_1) &= d(u_1, z_1) - d(u_1, w_1) \\ &= d(u_2, z_2) - d(u_2, w_2) \\ &= d(w_2, z_2) \end{aligned}$$

which proves the lemma. □

Theorem 2 *Any maximal (maximum) subtree isomorphism between two trees induces a maximal (maximum) clique in the corresponding FTAG, and vice versa.*

Proof. Let $\phi : H_1 \to H_2$ be a maximal subtree isomorphism between trees T_1 and T_2, and let $G = (V, E)$ denote the corresponding FTAG. Let $C_\phi \subseteq V$ be defined as $C_\phi = \{(u, \phi(u)) : u \in H_1\}$. From the definition of a subtree isomorphism it follows that ϕ maps the path between any two nodes $u, v \in H_1$ onto the path joining $\phi(u)$ and $\phi(v)$. This clearly implies that $d(u, v) = d(\phi(u), \phi(v))$ for all $u \in H_1$, and therefore C_ϕ is a clique. Trivially, C_ϕ is a maximal clique because ϕ is maximal, and this proves the first part of the theorem.

Suppose now that $C = \{(u_1, w_1), \cdots, (u_n, w_n)\}$ is a maximal clique of G, and let $H_1 = \{u_1, \cdots, u_n\} \subseteq V_1$ and $H_2 = \{w_1, \cdots, w_n\} \subseteq V_2$. Define $\phi : H_1 \to H_2$ as $\phi(u_i) = w_i$, for all $i = 1 \ldots n$. From the definition of the FTAG and the hypothesis that C is a clique, it is simple to see that ϕ is a one-to-one and onto correspondence between H_1 and H_2, which trivially preserves the adjacency relationships between nodes. The fact that ϕ is a maximal isomorphism is a straightforward consequence of the maximality of C.

To conclude the proof we have to show that the subgraphs that we obtain when we restrict ourselves to H_1 and H_2, i.e. $T_1[H_1]$ and $T_2[H_2]$, are trees, and this is equivalent to showing that they are connected. Suppose by contradiction that this is not the case, and let $u_i, u_j \in H_1$ be two nodes which are not joined by a path in $T_1[H_1]$. Since both u_i and u_j are nodes of T_1, however, there must exist a path $u_i = x_0 x_1 \ldots x_m = u_j$ joining them in

T_1. Let $x^* = x_k$, for some $k = 1 \ldots m$, be a node on this path which is not in H_1. Moreover, let $y^* = y_k$ be the k-th node on the path $w_i = y_0 y_1 \ldots y_m = w_j$ which joins w_i and w_j in T_2 (remember that $d(u_i, u_j) = d(w_i, w_j)$, and hence $d(w_i, w_j) = m$). We now show that the set $\{(x^*, y^*)\} \cup C \subseteq V$ is a clique. To this end, let $(u, w) \in C$. Since (u_i, w_i) and (u_j, w_j) are also nodes in C we have $d(u_i, u) = d(w_i, w)$, and $d(u_j, u) = d(w_j, w)$. Furthermore, we have that x^* and y^* are on the $u_i u_j$- and $w_i w_j$-paths, respectively, and clearly $d(u_i, x^*) = d(w_i, y^*)$ and $d(x^*, u_j) = d(y^*, w_j)$. Therefore, all the hypotheses of Lemma 1 are satisfied, and this implies that $d(x^*, u) = d(y^*, w)$, which amounts to stating that node (x^*, y^*) is adjacent to (u, w), for all $(u, w) \in C$. This means that $\{(x^*, y^*)\} \cup C$ is a clique, thereby contradicting the hypothesis that C is a maximal clique and proving the second part of theorem.

The "maximum" part of the statement is proved similarly. □

The FTAG is readily derived by using a classical representation for graphs, i.e., the so-called *distance matrix* which, for an arbitrary graph $G = (V, E)$ of order n, is the $n \times n$ matrix $D = (d_{ij})$ where $d_{ij} = d(u_i, u_j)$, the distance between nodes u_i and u_j. Efficient, classical algorithms are available for obtaining such a matrix [8].

3 A continuous characterization of maximal cliques

After formulating the free tree matching problem as a maximum clique problem, we now proceed (following [20, 21]) by mapping the latter onto a continuous quadratic programming problem. Let $G = (V, E)$ be an arbitrary graph of order n, and let Δ denote the standard simplex of $\mathbb{R}^n$:

$$\Delta = \{ \mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{e}'\mathbf{x} = 1 \text{ and } x_i \geq 0, \ i = 1 \ldots n \}$$

where $\mathbf{e}$ is the vector whose components equal 1, and a prime denotes transposition. Given a subset of vertices C of G, we will denote by $\mathbf{x}^c$ its *characteristic vector* which is the point in Δ defined as

$$x_i^c = \begin{cases} 1/|C|, & \text{if } i \in C \\ 0, & \text{otherwise} \end{cases}$$

where $|C|$ denotes the cardinality of C.

Now, consider the following quadratic function

$$f_G(\mathbf{x}) = \mathbf{x}' A_G \mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{x} \tag{3}$$

where $A_G = (a_{ij})$ is the adjacency matrix of G. The following theorem, recently proved by Bomze [4], expands on the Motzkin-Straus theorem [19], a remarkable result which establishes a connection between the maximum clique problem and certain standard quadratic programs. This has an intriguing computational significance in that it allows us to shift from the discrete to the continuous domain in an elegant manner.

Theorem 3 *Let C be a subset of vertices of a graph G, and let $\mathbf{x}^c$ be its characteristic vector. Then, C is a maximal (maximum) clique of G if and only if $\mathbf{x}^c$ is a local (global) maximizer f_G in Δ. Moreover, all local (and hence global) maximizers of f_G in Δ are strict and are characteristic vectors of maximal cliques of G.*

In a formal sense, therefore, a one-to-one correspondence exists between maximal cliques and local maximizers of f_G in Δ on the one hand, and maximum cliques and global maximizers on the other hand.

4 Matching free trees with monotone game dynamics

Evolutionary game theory considers an idealized scenario whereby in a large population pairs of individuals are repeatedly drawn at random to play a symmetric two-player game. In contrast to traditional game theoretic models, players are not supposed to behave rationally or to have complete knowledge of the details of the game. They act instead according to a preprogrammed behavior pattern, or pure strategy, and it is supposed that some evolutionary selection process operates over time on the distribution of behaviors. We refer the reader to [14, 23] for excellent introductions to this rapidly expanding field.

Let $J = \{1, \cdots, n\}$ be the set of available pure strategies and, for all $i \in J$, let $x_i(t)$ be the proportion of population members playing strategy i, at time t. The state of the population at a given instant is the vector $\mathbf{x} = (x_1, \cdots, x_n)'$. Clearly, population states are constrained to lie in the standard simplex Δ. For a given population state $\mathbf{x} \in \Delta$, we shall denote by $\sigma(\mathbf{x})$ the *support* of $\mathbf{x}$, i.e. the set of non-extinct strategies:

$$\sigma(\mathbf{x}) = \{i \in J \ : \ x_i > 0\} \ .$$

Let $A = (a_{ij})$ be the $n \times n$ payoff matrix. Specifically, for each pair of strategies $i, j \in J$, a_{ij} represents the payoff of an individual playing strategy i against an opponent playing strategy j. If the population is in state $\mathbf{x}$, the expected payoff earned by an i-strategist is:

$$\pi_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} x_j = (A\mathbf{x})_i \tag{4}$$

while the mean payoff over the entire population is

$$\pi(\mathbf{x}) = \sum_{i=1}^{n} x_i \pi_i(\mathbf{x}) = \mathbf{x}' A \mathbf{x} \ . \tag{5}$$

In evolutionary game theory the assumption is made that the game is played over and over, generation after generation, and that the action of natural selection will result in the evolution of the fittest strategies. A general class of evolution equations is given by the following set of ordinary differential equations:

$$\dot{x}_i = x_i g_i(\mathbf{x}) \tag{6}$$

where a dot signifies derivative with respect to time, and $g = (g_1, \ldots, g_n)$ is a function with open domain containing Δ. Here, the function g_i $(i \in J)$ specifies the rate at which pure strategy i replicates. It is usually required that the growth function g is *regular* [23], which means that it is Lipschitz continuous and that $g(\mathbf{x}) \cdot \mathbf{x} = 0$ for all $\mathbf{x} \in \Delta$. The former condition guarantees us that the system of differential equations (6) has a unique solution through any initial population state. The condition $g(\mathbf{x}) \cdot \mathbf{x} = 0$, instead, ensures that the simplex Δ is invariant under (6), namely any trajectory starting in Δ will remain in Δ.

Payoff-monotonic game dynamics represent a wide class of regular selection dynamics for which useful properties hold. Intuitively, for a payoff-monotonic dynamics the strategies associated to higher payoffs will increase at higher rate. Formally, a regular selection dynamics (6) is said to be payoff-monotonic if:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \ \Leftrightarrow \ \pi_i(\mathbf{x}) > \pi_j(\mathbf{x}) \tag{7}$$

for all $\mathbf{x} \in \Delta$.

In an unpublished paper [13], Hofbauer shows that the average population payoff is strictly increasing along the trajectories of any monotone game dynamics, provided that payoffs are symmetric. This result generalizes the celebrated fundamental theorem of natural selection [14, 23]. Here, we provide a different proof adapting a technique from [10].

Theorem 4 *If the payoff matrix A is symmetric, then $\pi(\mathbf{x}) = \mathbf{x}' A \mathbf{x}$ is strictly increasing along any non-constant trajectory of any payoff-monotonic dynamics. In other words, $\dot{\pi}(\mathbf{x}(t)) \geq 0$ for all t, with equality if and only if $\mathbf{x} = \mathbf{x}(t)$ is a stationary point.*

Proof. For $\mathbf{x} \in \Delta$, let:

$$\sigma_+(\mathbf{x}) = \{ i \in \sigma(\mathbf{x}) \ : \ g_i(\mathbf{x}) \geq 0 \}$$

and

$$\sigma_-(\mathbf{x}) = \{i \in \sigma(\mathbf{x}) \ : \ g_i(\mathbf{x}) < 0\} \ .$$

Clearly, $\sigma_+(\mathbf{x}) \cup \sigma_-(\mathbf{x}) = \sigma(\mathbf{x})$. Moreover, let:

$$\overline{\pi}(\mathbf{x}) = \min\{\pi_i(\mathbf{x}) \ : \ i \in \sigma_+(\mathbf{x})\}$$

and

$$\underline{\pi}(\mathbf{x}) = \max\{\pi_i(\mathbf{x}) \ : \ i \in \sigma_-(\mathbf{x})\}$$

Because of payoff-monotonicity, note that $\overline{\pi}(\mathbf{x}) \geq \underline{\pi}(\mathbf{x})$. Note also that $\dot{x}_i \geq 0$ if and only if $i \in \sigma_+(\mathbf{x})$.

Now, since A is symmetric, we have:

$$\begin{aligned}
\frac{\dot{\pi}(\mathbf{x})}{2} &= \sum_{i \in \sigma(\mathbf{x})} \dot{x}_i \pi_i(\mathbf{x}) \\
&= \sum_{i \in \sigma_+(\mathbf{x})} \dot{x}_i \pi_i(\mathbf{x}) + \sum_{i \in \sigma_-(\mathbf{x})} \dot{x}_i \pi_i(\mathbf{x}) \\
&\geq \overline{\pi}(\mathbf{x}) \sum_{i \in \sigma_+(\mathbf{x})} \dot{x}_i + \underline{\pi}(\mathbf{x}) \sum_{i \in \sigma_-(\mathbf{x})} \dot{x}_i \\
&= (\overline{\pi}(\mathbf{x}) - \underline{\pi}(\mathbf{x})) \sum_{i \in \sigma_+(\mathbf{x})} \dot{x}_i \\
&\geq 0
\end{aligned}$$

where the last equality follows from $\sum \dot{x}_i = 0$. Finally, note that $\dot{\pi}(\mathbf{x}) = 0$ if and only if $\pi_i(\mathbf{x})$ is constant for all $i \in \sigma(\mathbf{x})$ which amounts to saying that $\mathbf{x}$ is stationary [23, Proposition 4.7]. □

A well-known subclass of payoff-monotonic game dynamics is given by:

$$\dot{x}_i = x_i \left(f(\pi_i(\mathbf{x})) - \sum_{j=1}^{n} x_j f(\pi_j(\mathbf{x})) \right) \tag{8}$$

where $f(u)$ is an increasing function of u. These models arise in modeling the evolution of behavior by way of imitation processes, where players are occasionally given the opportunity to change their own strategies [13, 23].

When f is the identity function, i.e., $f(u) = u$, we obtain the standard replicator equations:

$$\dot{x}_i = x_i \left(\pi_i(\mathbf{x}) - \sum_{j=1}^{n} x_j \pi_j(\mathbf{x}) \right) \tag{9}$$

whose basic idea is that the average rate of increase $\dot{x}_i/x_i$ equals the difference between the average payoff of strategy i and the mean payoff over the entire population.

Another popular model arises when $f(u) = e^{\kappa u}$ which yields:

$$\dot{x}_i = x_i \left(e^{\kappa \pi_i(\mathbf{x})} - \sum_{j=1}^{n} x_j e^{\kappa \pi_j(\mathbf{x})} \right) \tag{10}$$

where κ is a positive constant. As κ tends to 0, the orbits of this dynamics approach those of the standard, first-order replicator model (9), slowed down by the factor κ; moreover, for large values of κ the model approximates the so-called "best-reply" dynamics [13, 14].

In light of their dynamical properties, payoff-monotonic dynamics naturally suggest themselves as simple heuristics for solving the maximal subtree isomorphism problem. Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two free trees, and let A_G denote the adjacency matrix of their FTAG G. By putting

$$A = A_G + \frac{1}{2} I \tag{11}$$

where I is the identity matrix, we know from Theorem 4 that any payoff-monotonic dynamics, starting from an arbitrary initial state, will iteratively maximize the function f_G defined in (3) over the simplex and will converge with probability 1 to a strict local maximizer which, by virtue of Theorem 3, will then correspond to the characteristic vector of a maximal clique in the FTAG. As stated in Theorem 2, this will in turn induce a maximal subtree isomorphism between T_1 and T_2.

5 Experimental results

In this section we present experiments of applying payoff-monotonic dynamics to the free tree matching problem. In our simulations, we used the following discrete-time models:

$$x_i(t+1) = \frac{x_i(t)\pi_i(t)}{\sum_{j=1}^{n} x_j(t)\pi_j(t)} \tag{12}$$

and

$$x_i(t+1) = \frac{x_i(t)e^{\kappa \pi_i(t)}}{\sum_{j=1}^{n} x_j(t)e^{\kappa \pi_j(t)}} \tag{13}$$

which correspond to well-known discretizations of equations (9) and (10), respectively [14, 23]. For the latter dynamics the value $\kappa = 10$ was used.

Both the first-order and the exponential processes were started from the simplex barycenter and stopped when either a maximal clique (i.e., a local maximizer of f_G) was found or

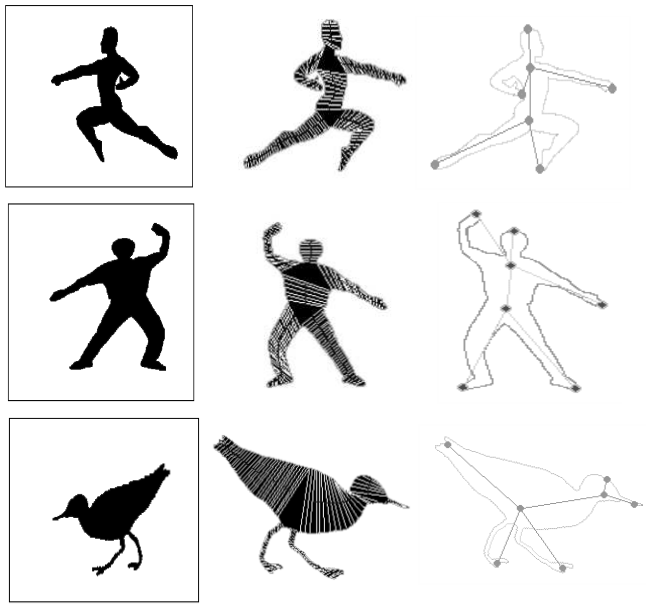

Figure 3: Illustration of the SA-tree construction. Three shapes (left), their shape-axis model (middle), and the corresponding SA-trees (right).

the distance between two successive points was smaller than a fixed threshold. In the latter case the converged vector was randomly perturbed, and the algorithms restarted from the perturbed point. Because of the one-to-one correspondence between local maximizers and maximal cliques, this situation corresponds to convergence to a saddle point.

5.1 Matching shape-axis trees

Recently, Liu *et al.* [18] introduced a new representation for shape based on the idea of self-similarity. Intuitively, given a closed planar shape, they consider two different parameterizations of its contour, namely one oriented counterclockwise, $\Gamma(s) = \{x(s) : 0 \leq s \leq 1\}$, and the other clockwise, $\hat{\Gamma}(t) = \{\hat{x}(t) = x(1-t) : 0 \leq t \leq 1\}$. By minimizing an appropriate cost functional they find a "good" match between Γ and $\hat{\Gamma}$, and then define the shape axis (SA) as the loci of middle points between the matched contour points. From a given SA, it is possible to construct a unique free tree, called the SA-tree, by grouping the discontinuities contained in the SA. In Figure 3 the SA-tree construction process for a few example shapes is illustrated.

The proposed matching algorithms were tested on a selection of 17 shapes (SA-trees) representing six different object classes (horse, human, bird, dog, sheep, and rhino). We matched each shape against each other (and itself) and in *all* the 289 trials both algorithms returned the *maximum* isomorphism, i.e. a maximum clique in the FTAG. This is a remark-

able fact, considering that replicator dynamics are unable to escape from local solutions. Similar findings on related problems are discussed in [20, 21]. As far as the computational time is concerned, both dynamics took only a few seconds to converge on a 350MHz AMDK6-2 processor, the exponential one being slightly faster than the linear one (but see below for a rather different picture).

5.2 Matching larger trees

Encouraged by the results reported above we proceeded by testing our algorithms over much larger (random) trees, with the aim of studying their sensitivity to structural perturbations. Random structures represent a useful benchmark not only because they are not constrained to any particular application, but also because it is simple to replicate experiments and hence to make comparisons with other algorithms.

In this series of experiments, the following protocol was used. A hundred 100-node free trees were generated uniformly at random using a procedure described by Wilf in [24]. Then, each such tree was subject to a corruption process which consisted of randomly deleting a fraction of its terminal nodes, thereby obtaining a tree isomorphic to a proper subtree of the original one. Various levels of corruption (i.e., percentage of node deletion) were used, namely 2%, 10%, 20%, 30% and 40%. This means that the order of the pruned trees ranged from 98 to 60. Overall, therefore, 500 pairs of trees were obtained, for each of which the corresponding FTAG was constructed as described in Section 2. To keep the order of the association graph as low as possible, its vertex set was constructed as $V = \{(u, w) \in V' \times V'' : \deg(u) \leq \deg(w)\}$, assuming $|V'| \leq |V''|$, the edge set E being defined as in (2). It is straightforward to see that when the first tree is isomorphic to a subtree of the second, Theorem 2 continues to hold. This simple heuristic may significantly reduce the dimensionality of the search space. We also made experiments with unpruned FTAG's but no significant difference in performance was noticed apart, of course, heavier memory requirements.

As in the previous series of experiments, both the linear and the exponential dynamics were used, with identical parameters and stopping criterion. After convergence, we calculated the proportion of matched nodes, i.e., the ratio between the cardinality of the clique found and the order of the smaller subtree, and then we averaged. Figure 4(a) shows the results obtained using the linear dynamics (12) as a function of the corruption level. As can be seen, the algorithm was *always* able to find a correct maximum isomorphism, i.e. a maximum clique in the FTAG. Figure 4(b) plots the corresponding (average) CPU time taken by the

Figure 4: Results obtained over 100-node random trees with various levels of corruption, using the first-order dynamics (12). Top: Percentage of correct matches. Bottom: Average computational time taken by the replicator equations.

processes, with corresponding error bars (simulations were performed on the same machine used for the shape-axis experiments).

In Figure 5, the results pertaining to the exponential dynamics (13) are shown. In terms of solution's quality the algorithm performed exactly as its linear counterpart, but this time it was dramatically faster. This confirms earlier results reported in [20].

6 Conclusions

We have developed a formal approach for matching connected and acyclic relational structures, i.e. free trees, by constructing an association graph whose maximal cliques are in one-to-one correspondence with maximal subtree isomorphisms (i.e., maximal common sub-

Figure 5: Results obtained over 100-node random trees with various levels of corruption, using the exponential dynamics (13). Top: Percentage of correct matches. Bottom: Average computational time taken by the replicator equations.

trees). The framework is general and can be applied in a variety of computer vision problems as well as in other domains: we have demonstrated its potential for shape matching. The solution is found by using payoff-monotonic dynamical systems, which make them amenable to hardware implementation and offer the advantage of biological plausibility. Since it is difficult to establish theoretical bounds on the quality of the solutions found, extensive experiments on hundreds of uniformly random trees have been conducted, focusing on the issue of sensitivity to structural errors. As in previous work on graph isomorphism [20] and rooted tree matching [21], the results are impressive: despite the counter-intuitive maximum clique formulation of the tree matching problem, and the inherent inability of these simple dynamics to escape from local optima, they nevertheless were always able to find a globally optimal solution. This apparently non-accidental regularity raises intriguing questions about

the connections between standard notions of computational complexity and the "elusiveness" of global optima in a continuous setting.

Before concluding, we note that there exists a different approach to comparing graphs (and, in particular, trees) which, instead of looking for maximal common substructures as we do, is based on the idea of computing their edit-distance, namely, the minimum cost to transform one graph into another by elementary edit operations. This idea is attractive especially when the structures being matched are subject to significant structural distortions. Unfortunately, it turns out that computing the edit-distance on free trees is NP-complete [25], although it is solvable in polynomial time by restricting ourselves to ordered trees, where each node is assigned a cyclic ordering of its incidente edges [15]. Moreover, determining the set of elementary edit operations and the associated costs depends heavily on the application domain and can be problematic (see [16, 17] for some examples of edit operations motivated by shape matching problems). This choice is in fact crucial as two graphs that are similar under one cost function may be quite dissimilar using another one, and the optimal node correspondences may vary considerably. It is worth mentioning that Bunke [7] has shown that on generic graphs, under certain assumptions concerning the edit-costs, determining the maximum common subgraph is equivalent to computing the graph edit-distance. Moreover, new graph-distance metrics have recently been proposed based on the maximum common subgraph [6].

In order to deal with graphs arising from realistic computer vision problems, the framework presented here can be extended along several lines. For instance, in many applications the underlying graphs have nodes with an associated vector of symbolic and/or numeric attributes, and one should incorporate into the matching algorithm a mechanism to deal with attribute perturbations. Following our previous work on rooted tree matching [21], this can be done in a straightforward and elegant way by placing weights on the nodes of the FTAG which quantify the similarity of attribute vectors. The matching problem is then transformed into that of finding a clique of largest weight rather than cardinality, which corresponds to a subtree isomorphism of maximal similarity. Again, this can be done efficiently using monotone game dynamics. Moreover, in practical applications the matching algorithm should also be robust under structural perturbations. For example, the medial axis representation is known to be prone to erroneous branches (see, e.g., [26] for a discussion on shape representation and matching). The approach described here can indeed cope with certain types of perturbations such as the addition or deletion of nodes in both trees, for we aim at finding a maximal common substructure rather than looking at perfect isomorphisms,

and the experimental results presented in the paper do confirm this. However, it would not work satisfactorily in the presence of more complex distortions like, e.g., the merging of two nodes. To cope with such problems, it is simple to formulate error-tolerant versions of our matching framework following the lines suggested in [22] for rooted attributed trees, where many-to-many node correspondences are allowed. All this will be the subject of future investigations.

Acknowledgments. The author would like to thank M. Zuin for his support in performing the experiments, T.-L. Liu for kindly providing us with the shape database, and the anonymous reviewers for their comments.

References

[1] J. E. Ash, P. A. Chubb, S. E. Ward, S. M. Welford, and P. Willett. *Communication, Storage and Retrieval of Chemical Information.* Ellis Horwood, Chichester, UK, 1985.

[2] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[4] I. M. Bomze. Evolution towards the maximum clique. *J. Glob. Optim.*, 10:143–164, 1997.

[5] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization (Suppl. Vol. A)*, pages 1–74. Kluwer, Boston, MA, 1999.

[6] H. Bunke and K. Shearer. A graph-distance metric based on the maximal common subgraph. *Pattern Recognition Lett.*, 19:255–259, 1998.

[7] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(9):917–922, 1999.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 1990.

[9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, New York, 2nd edition, 2001.

[10] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games.* MIT Press, Cambridge, MA, 1998.

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

[12] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(4):377–388, 1996.

[13] J. Hofbauer. Imitation dynamics for games. Collegium Budapest, preprint, 1995.

[14] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK, 1998.

[15] P. Klein. Computing the edit-distance between unrooted ordered trees. In G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, editors, *Algorithms—ESA'98, 6th Annual European Symposium*, pages 91–102. Springer, Berlin, 1998.

[16] P. Klein, S. Tirthapura, D. Sharvit, and B. Kimia. A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proc. 10th Ann. ACM–SIAM Symp. Discr. Algorithms (SODA)*, pages 696–704, San Francisco, CA, 2000.

[17] T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *Proc. ICCV'99—7th Int. Conf. Computer Vision*, pages 456–462, Kerkyra, Greece, 1999.

[18] T.-L. Liu, D. Geiger, and R. V. Kohn. Representation and self-similarity of shapes. In *Proc. ICCV'98—6th Int. Conf. Computer Vision*, pages 1129–1135, Bombay, India, 1998.

[19] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canad. J. Math.*, 17:533–540, 1965.

[20] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):2023–2045, 1999.

[21] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Anal. Machince Intell.*, 21(11):1105–1120, 1999.

[22] M. Pelillo, K. Siddiqi, and S. W. Zucker. Many-to-many matching of attributed trees using association graphs and game dynamics. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Visual Form 2001*, pages 583–593. Springer, Berlin, 2001.

[23] J. W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, MA, 1995.

[24] H. Wilf. The uniform selection of free trees. *J. Algorithms*, 2:204–207, 1981.

[25] K. Zhang, J. T. L. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs. *Int. J. Found. Computer Sci.*, 7(1):43–57, 1996.

[26] S. C. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modeling system. *Int. J. Computer Vision*, 20(3):187–212, 1996.