Continuous-based Heuristics for Graph and Tree Isomorphisms, with Application to Computer Vision

Marcello Pelillo (pelillo@dsi.unive.it)
Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy

Kaleem Siddiqi (siddiqi@cim.mcgill.ca)
Center for Intelligent Machines, McGill University, Canada

Steven W. Zucker (zucker-steven@cs.yale.edu)
Center for Computational Vision and Control, Yale University, USA

Abstract

We present a new (continuous) quadratic programming approach for graph- and tree-isomorphism problems which is based on an equivalent maximum clique formulation. The approach is centered around a fundamental result proved by Motzkin and Straus in the mid-1960s, and recently expanded in various ways, which allows us to formulate the maximum clique problem in terms of a standard quadratic program. The attractive feature of this formulation is that a clear one-to-one correspondence exists between the solutions of the quadratic programs and those in the original, combinatorial problems. To approximately solve the program we use the so-called "replicator" equations, a class of straightforward continuous- and discrete-time dynamical systems developed in various branches of theoretical biology. We show how, despite their inherent inability to escape from local solutions, they nevertheless provide experimental results which are competitive with those obtained using more sophisticated mean-field annealing heuristics. Application of this approach to shape matching problems arising in computer vision and pattern recognition are also presented.

Keywords: Maximum clique, quadratic programming, replicator dynamics, shape recognition.

1 Introduction

The relationships between discrete and continuous mathematics have always been a subject of intensive study. Apart from the underlying philosophical implications, in fact, the interaction between the two domains can provide new insights into old problems and often allows techniques from one side to be profitably imported into the other. Entire branches of modern mathematics have been created with the specific motivation of exploring such connections, examples of which are singularity theory, combinatorial topology, and spectral graph theory. In more recent years, with the introduction of the ellipsoid and the interior point methods for linear programming, there has also been a tremendous interest in computer science and operations research in solving combinatorial optimization problems using continuous methods [17, 38].

Following the seminal works of Hopfield and Tank [22], and Durbin and Willshaw [11], the neural network community also became interested in using continuous approaches for combinatorial optimization. The basic idea consists in deriving a continuous "energy" function whose minimizers are in correspondence with the solutions of the discrete problem, and then minimizing it using continuous- or discrete-time dynamical systems, typically embedded in a parallel network of locally interacting processing elements. Almost invariably, the minimization algorithms developed so far incorporate techniques borrowed from statistical mechanics, in particular mean field theory, which allow one to escape from poor local solutions. Examples of problems attacked within this framework include the traveling salesman problem [11, 22], graph bipartitioning [12], the maximum clique problem [23, 40], the linear assignment problem [26], the knapsack problem [36] and the graph/subgraph isomorphism problems [16, 45].

Thus far, the focus has been on "flat" problems, in the sense that there is no partial ordering imposed on the data. In many practical problems, however, data are organized in a hierarchical manner, i.e., are *trees*, and the problem of *matching* such representations is of interest for pattern recognition. Applications in domains like computer vision [32, 47, 48, 52, 56], molecular biology [50] and natural language processing [35] abound, and many traditional, discrete algorithms have been developed [29, 31, 33, 46, 51]. On the other hand, no attempt has yet been made to approach such problems within a continuous framework, using analog continuous-time dynamics. The main difficulty is that it is not clear how to map the hierarchy embedded in the representations onto a "flat" optimization network.

In this paper, we consider isomorphism problems on graphs and trees. We present a new continuous framework for these problems which is based on the idea of reducing it to the maximum clique problem, another well-known combinatorial optimization problem [6]. Central to our approach is a powerful result originally proved by Motzkin and Straus [34], and recently extended in various ways [5, 14, 15, 42], which allows us to formulate the maximum clique problem in terms of an indefinite quadratic program. In the proposed formulation an elegant one-to-one correspondence exists between the solutions of the (continuous) quadratic program and those of the original (discrete) problem. We also present a class of straightforward continuous- and discrete-time dynamical systems known in mathematical biology as *replicator equations*, and show how, owing to their properties, they provide a natural and useful heuristic for solving the Motzkin-Straus program, and hence graph and tree isomorphism problems. Applications of this approach to shape recognition problems arising in computer vision and pattern recognition are presented.

We note that the graph isomorphism problem still resist any computational complexity characterization [13, 24]. Despite decades of active research, no polynomial-time algorithm for it has yet been found. At the same time, while clearly belonging to NP, no proof has been provided that it is NP-complete. Indeed, there is strong evidence that this cannot be the case, for otherwise the polynomial hierarchy would collapse [9, 49]. The current belief is that the problem lies strictly between the P and NP-complete classes. The subtree isomorphism problem, instead, can be solved in polynomial time [33, 46]. It may be argued therefore that trying to solve these problems by reducing it to the maximum clique problem is an altogether inappropriate choice. In contrast to them, in fact, finding just the cardinality of the maximum clique in a graph is known to be NP-complete and, according to recent theoretical results, so is the problem of approximating it within a certain tolerance [1, 19].[1] The experimental results presented in this paper, however, seem to contradict this claim. By using simple relaxation equations which are inherently unable to avoid local optima, we get results which compare favorably with those obtained using state-of-the-art sophisticated neural network algorithms which, by contrast, are explicitly designed to escape from local solutions. This suggests therefore that the proposed Motzkin-Straus formulation is a promising framework within which to develop powerful graph isomorphism heuristics.

Our continuous heuristics are attractive not only because they offer the advantage of biological plausibility, but also because they can motivate parallel, analog VLSI implementations. This is especially important in such applications as computer vision and robotics, where real-time response is a concern.

2 Graph and tree isomorphisms as clique search

2.1 Notations and definitions

Before going into the details of the proposed framework, we need to introduce some graph-theoretical notations and definitions. More details can be found in standard textbooks of graph theory, such as [18]. Let $G = (V, E)$ be a graph, where V is the set of nodes 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 *degree* of a vertex is the number of vertices 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

[1] It should be pointed out, however, that these are worst-case results, and there are certain classes of graphs for which the problem is solvable in polynomial time [17, 6].

is n. If $u_0 = u_n$ the path is called a *cycle*. A graph is said to be *connected* if any pair of nodes is 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 *tree*. A *rooted* tree is one which has a distinguished node, called the *root*. The *level* of a node u in a rooted tree, denoted by $\text{lev}(u)$, is the length of the path connecting the root to u. Note that there is an obvious equivalence between rooted trees and directed trees, where the edges are assumed to be oriented. We shall therefore use the same terminology typically used for directed trees to define the relation between two adjacent nodes. In particular, if $u \sim v$ and $\text{lev}(v) - \text{lev}(u) = +1$, we say that u is the *parent* of v and, conversely, v is a *child* of u. Trees have a number of interesting properties. One which turns out to be very useful for our characterization is that in a tree any two nodes are connected by a *unique* path.

Given a graph $G = (V, E)$, a subset of vertices C is called a *clique* if all its vertices are mutually adjacent, i.e., for all $u, v \in C$ we have $u \sim v$. A clique is said to be *maximal* if it is not contained in any larger clique, and *maximum* if it is the largest clique in the graph. The *clique number*, denoted by $\omega(G)$, is defined as the cardinality of the maximum clique.

2.2 Graph isomorphism

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, an *isomorphism* between them is any bijection $\phi : V_1 \to V_2$ such that $u \sim v \Leftrightarrow \phi(u) \sim \phi(v)$, for all $u, v \in V_1$. Two graphs are said to be *isomorphic* if there exists an isomorphism between them. The graph isomorphism problem is therefore to decide whether two graphs are isomorphic and, in the affirmative, to find an isomorphism. The maximum common subgraph problem is more general and difficult [13], and includes the graph isomorphism problem as a special case. It consists of finding the largest isomorphic subgraphs of G_1 and G_2. A simpler version of this problem is to find a maximal common subgraph, i.e., an isomorphism between subgraphs which is not included in any larger subgraph isomorphism.

Barrow and Burstall [3], and also Kozen [27], introduced the notion of an *association graph* as a useful auxiliary graph structure for solving general graph/subgraph isomorphism problems.

Definition 1 *The association graph derived from graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the undirected graph $G = (V, E)$, where*

$$V = V_1 \times V_2$$

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

$$(u, w) \sim (v, z) \quad \textit{iff} \quad u \neq v, \; w \neq z, \; \textit{and} \; u \sim v \Leftrightarrow w \sim z$$

The following result, whose proof is obvious from the definition just given, establishes an equivalence between the graph isomorphism problem and the maximum clique problem.

Theorem 2 *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs of order n, and let G be the corresponding association graph. Then, G_1 and G_2 are isomorphic if and only if $\omega(G) = n$. In this case, any maximum clique of G induces an isomorphism between G_1 and G_2, and vice versa. In general, maximal/maximum cliques in G are in one-to-one correspondence with maximal/maximum common subgraph isomorphisms between G_1 and G_2, respectively.*

2.3 Subree isomorphism

In many computer vision problems relational structures are organized in a hierarchical manner, i.e., are *trees* (see, for example, [32, 47, 48, 52, 56]). Since in the standard association graph formulation the solutions are not constrained to preserve the required partial order, it is not clear how to apply the association graph framework in these cases. To illustrate the difficulties with the standard formulation, consider the problem of finding the largest subtree in the left tree of Figure 1 which is isomorphic to a subtree in the right tree. Up to permutations, the correct solution is clearly given by 3 $\rightarrow$ a, 4 $\rightarrow$ b, 5 $\rightarrow$ c, 6 $\rightarrow$ d, 7 $\rightarrow$ f, and 8 $\rightarrow$ g. In other words, the subtree rooted at node 3 is matched against that rooted at node a in the tree on the right. However, using the standard association graph formulation given before, it is easily verified that the solutions induced by the maximum cliques correspond (up to permutations) to the following: 2 $\rightarrow$ h, 3 $\rightarrow$ a, 4 $\rightarrow$ b, 5 $\rightarrow$ c, 6 $\rightarrow$ d, 7 $\rightarrow$ f, and 8 $\rightarrow$ g, which, while perfectly in accordance with the usual subgraph isomorphism constraints, *does* violate the requirement that the matched subgraphs be trees (note, in fact, that nodes 2 and h are isolated from the rest of the matched subtrees).

In this section, we introduce a solution to this problem by providing a novel way of deriving an association graph from two (rooted) trees, based on the graph-theoretic notions of connectivity and the distance matrix.

Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two rooted trees. Any bijection $\phi : H_1 \rightarrow H_2$, with $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$, is called a *subtree isomorphism* if it preserves the adjacency and hierarchical relationships between the nodes and, in addition, the induced subgraphs $T_1[H_1]$ and $T_2[H_2]$ are trees. The former condition amounts to stating that, given $u, v \in H_1$, we have $u \sim v$ if and only if $\phi(u) \sim \phi(v)$, and u is the parent of v if and only if $\phi(u)$ is the parent of $\phi(v)$. A subtree isomorphism is *maximal* if there is no other subtree isomorphism $\phi' : H_1' \rightarrow H_2'$ with H_1 a strict subset of H_1', and *maximum* if H_1 has largest cardinality. The maximal (maximum)

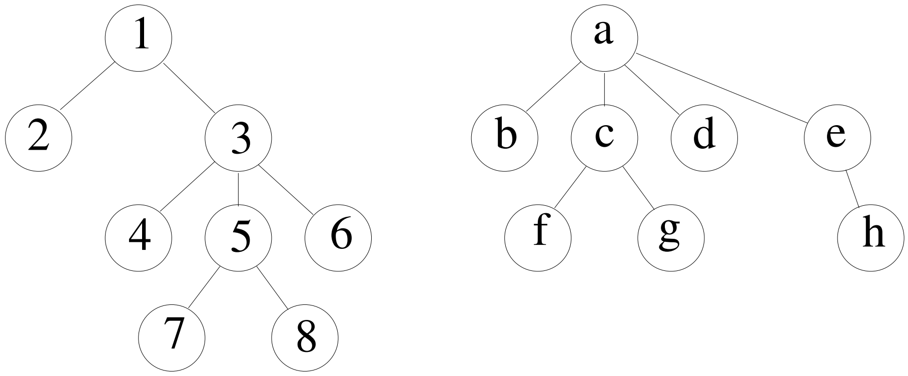


Figure 1: An example of matching two trees. In the standard formulation of the association graph, the maximum cliques do not preserve the hierarchical structure of the two trees (see text for details).

subtree isomorphism problem is to find a maximal (maximum) subtree isomorphism between two rooted trees.

We now introduce the notion of a *path-string*, which will be central to the subsequent development.

Definition 3 *Let u and v be two distinct nodes of a rooted tree T, and let $u = x_0 x_1 \dots x_n = v$ be the (unique) path joining them. The* path-string *of u and v, denoted by $\mathrm{str}(u, v)$, is the string $s_1 s_2 \dots s_n$ on the alphabet $\{-1, +1\}$ where, for all $i = 1 \dots n$, $s_i = \mathrm{lev}(x_i) - \mathrm{lev}(x_{i-1})$. By convention, when $u = v$ we define $\mathrm{str}(u, v) = \varepsilon$, where ε is the null string (i.e., the string having zero length).*

The path-string concept has a very intuitive meaning. Suppose that you stand on a particular node in a rooted tree and want to move to another adjacent node. Because of the orientation induced by the root, only two types of moves can be done, i.e., going down to one of the children (if one exists) or going up to the parent (if you are not on the root). Let us assign to the first move the label $+1$, and to the second the label -1. Now, suppose that you want to move from node u to v, following the unique path joining them. Then, the path-string of u and v is simply the string of elementary moves required to reach v, starting from u. It may be thought of as the degree of relationship between two relatives in a "family" tree. As an illustrative example, referring to Figure 1, we have $\mathrm{str}(2, 8) = -1 + 1 + 1 + 1$.

Definition 4 *The* tree association graph *(TAG) of two rooted trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ is the graph $G = (V, E)$ where*

$$V = V_1 \times V_2$$

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

$$(u, w) \sim (v, z) \quad \textit{iff} \quad \mathrm{str}(u, v) = \mathrm{str}(w, z) \ .$$

Intuitively, two nodes (u, w) and (v, z) are adjacent in the TAG, if and only if the relationship between u and v in T_1 is the same as that between w and z in T_2.

The following theorem establishes a one-to-one correspondence between the maximum subtree isomorphism problem and the maximum clique problem.

Theorem 5 *Any maximal (maximum) subtree isomorphism between two rooted trees induces a maximal (maximum) clique in the corresponding TAG, and vice versa.*

Proof (outline). Let $\phi : H_1 \to H_2$ be a maximal subtree isomorphism between rooted trees T_1 and T_2, and let $G = (V, E)$ denote the corresponding tree association graph. 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 $\mathrm{str}(u, v) = \mathrm{str}(\phi(u), \phi(v))$ 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 $i = 1 \dots n$. From the definition of a tree association graph 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 both the adjacency and the hierarchical 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 induced subgraphs $T_1[H_1]$ 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 $u_i = x_0 x_1 \dots x_m = u_j$ joining them in T_1. Let $x^* = x_k$, for some $k = 1 \dots m$, 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 $w_i = y_0 y_1 \dots y_m = w_j$ which joins w_i and w_j in T_2 (remember that $\mathrm{str}(u_i, u_j) = \mathrm{str}(w_i, w_j)$, and hence $d(w_i, w_j) = m$).

It is easy to show that the set $\{(x^*, y^*)\} \cup C \subseteq V$ is a clique, and this contradicts the hypothesis that C is a maximal clique. This can be proved by exploiting the obvious fact that if x is a node on the path joining any two nodes u and v, then $\mathrm{str}(u, v)$ can be obtained by concatenating $\mathrm{str}(u, x)$ and $\mathrm{str}(x, v)$.

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

The next proposition provides us with a straightforward criterion to construct the TAG.

Proposition 6 *Let $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ be two rooted trees, $u, v \in V_1$, and $w, z \in V_2$. Then,* $\mathrm{str}(u, v) = \mathrm{str}(w, z)$ *if and only if the following two conditions hold:*

(a) $d(u,v) = d(w,z)$

(b) $\mathrm{lev}(u) - \mathrm{lev}(v) = \mathrm{lev}(w) - \mathrm{lev}(z)$

This property allows us to efficiently derive the TAG by using a classical representation for graphs, i.e., the so-called *distance matrix* (see, e.g., [18]) 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.

3 Continuous formulation of the maximum clique problem

Let $G = (V, E)$ be an arbitrary undirected graph of order n, and let S_n denote the standard simplex of $\mathrm{I\!R}^n$:

$$S_n = \left\{ \mathbf{x} \in \mathrm{I\!R}^n \ : \ x_i \geq 0 \ \text{ for all } i = 1 \ldots n, \ \text{ and } \sum_{i=1}^{n} x_i = 1 \right\} .$$

Given a subset of vertices C of G we shall denote by $\mathbf{x}^c$ its *characteristic vector*, which is the point in S_n 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

$$f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \tag{1}$$

where $A = (a_{ij})$ is the adjacency matrix of G, i.e., the $n \times n$ symmetric matrix $A = (a_{ij})$ defined as

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

and "T" denotes transposition. A point $\mathbf{x}^* \in S_n$ is said to be a *global* maximizer of f in S_n if $f(\mathbf{x}^*) \geq f(\mathbf{x})$, for all $\mathbf{x} \in S_n$. It is said to be a *local* maximizer if there exists an $\epsilon > 0$ such that $f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all $\mathbf{x} \in S_n$ whose distance from $\mathbf{x}^*$ is less than ϵ, and if $f(\mathbf{x}^*) = f(\mathbf{x})$ implies $\mathbf{x}^* = \mathbf{x}$, then $\mathbf{x}^*$ is said to be a *strict* local maximizer.

The Motzkin-Straus theorem [34] establishes a remarkable connection between global (local) maximizers of the function f in S_n and maximum (maximal) cliques of

G. Specifically, it states that a subset of vertices C of a graph G is a maximum clique if and only if its characteristic vector $\mathbf{x}^c$ is a global maximizer of f on S_n. A similar relationship holds between (strict) local maximizers and maximal cliques [15, 42]. This result has an intriguing computational significance in that it allows us to shift from the discrete to the continuous domain in an elegant manner. The Motzkin-Straus theorem has served as the basis of many clique-finding procedures [8, 7, 14, 39, 40], and has also been used to determine theoretical bounds on the clique number [39, 55].

One drawback associated with the original Motzkin-Straus formulation relates to the existence of spurious solutions, i.e., maximizers of f which are not in the form of characteristic vectors. This was observed empirically by Pardalos and Phillips [39] and has more recently been formalized by Pelillo and Jagota [42]. In principle, spurious solutions represent a problem since, while providing information about the *cardinality* of the maximum clique, do not allow us to easily extract its vertices. Fortunately, there is straightforward solution to this problem which has recently been introduced and studied by Bomze [5]. Consider the following regularized version of function f:

$$\hat{f}(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{x} \tag{2}$$

which is obtained from (1) by substituting the adjacency matrix A of G with

$$\hat{A} = A + \frac{1}{2} I_n$$

where I_n is the $n \times n$ identity matrix.

The following is the spurious-free counterpart of the original Motzkin-Straus theorem (see [5] for proof).

Theorem 7 *Let C be a subset of vertices of a graph G, and let $\mathbf{x}^c$ be its characteristic vector. Then the following statements hold:*

(a) C is a maximum clique of G if and only if $\mathbf{x}^c$ is a global maximizer of the function $\hat{f}$ over the simplex S_n. In this case, $\omega(G) = 1/2(1 - \hat{f}(\mathbf{x}^c))$.

(b) C is a maximal clique of G if and only if $\mathbf{x}^c$ is a local maximizer of $\hat{f}$ in S_n.

(c) All local (and hence global) maximizers of $\hat{f}$ over S_n are strict.

Unlike the Motzkin-Straus formulation, the previous result guarantees us that *all* maximizers of $\hat{f}$ on S_n are strict, and are characteristic vectors of maximal/maximum cliques in the graph. In an exact sense, therefore, a one-to-one correspondence exists between maximal cliques and local maximizers of $\hat{f}$ in S_n on the one hand, and maximum cliques and global maximizers on the other hand. This solves the spurious solution problem in a definitive manner.

4 Replicator equations for graph and tree isomorphisms

Let W be a non-negative $n \times n$ matrix, and consider the following dynamical system:

$$\frac{d}{dt}x_i(t) = x_i(t)\left(\pi_i(t) - \sum_{j=1}^{n} x_j(t)\pi_j(t)\right) \ , \quad i = 1 \ldots n \tag{3}$$

where $\pi_i(t) = \sum_{j=1}^{n} w_{ij}x_j(t)$, $i = 1 \ldots n$, and its discrete-time counterpart:

$$x_i(t+1) = \frac{x_i(t)\pi_i(t)}{\sum_{j=1}^{n} x_j(t)\pi_j(t)} \ , \quad i = 1 \ldots n \ . \tag{4}$$

It is readily seen that the simplex S_n is invariant under these dynamics, which means that every trajectory starting in S_n will remain in S_n for all future times.

Both (3) and (4) are called *replicator equations* in theoretical biology, since they are used to model evolution over time of relative frequencies of interacting, self-replicating entities [21]. The discrete-time dynamical equations turn also out to be a special case of a general class of dynamical systems introduced by Baum and Eagon [4] in the context of Markov chain theory.

Theorem 8 *If W is symmetric, then the quadratic polynomial $F(\mathbf{x}) = \mathbf{x}^T W \mathbf{x}$ is strictly increasing along any non-constant trajectory of both continuous-time (3) and discrete-time (4) replicator equations. Furthermore, any such trajectory converges to a (unique) stationary point. Finally, a vector $\mathbf{x} \in S_n$ is asymptotically stable under (3) and (4) if and only if $\mathbf{x}$ is a strict local maximizer of F on S_n.*

The previous result is known in mathematical biology as the Fundamental Theorem of Natural Selection [21, 54]. As far as the discrete-time model is concerned, it can be regarded as a straightforward implication of the more general Baum-Eagon theorem [4]. The fact that all trajectories of the replicator dynamics converge to a stationary point is proven in [30].

Recently, there has been much interest in evolutionary game theory around the following exponential version of replicator equations, which arises as a model of evolution guided by imitation [20, 54]:

$$\frac{d}{dt}x_i(t) = x_i(t)\left(\frac{e^{\kappa\pi_i(t)}}{\sum_{j=1}^{n} x_j(t)e^{\kappa\pi_j(t)}} - 1\right) \ , \quad i = 1 \ldots n \tag{5}$$

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 (3), slowed down by the factor κ. Hofbauer [20] has recently proven that when the matrix W is symmetric, the quadratic polynomial F defined in Theorem 8 is also strictly increasing, as in the first-order case. After discussing various properties of this, and more general dynamics, he concluded that the model behaves essentially in the same way as the standard replicator equations, the only difference being the size of the basins of attraction around stable equilibria. A customary way of discretizating equation (5) is given by the following difference equations:

$$x_i(t+1) = \frac{x_i(t)e^{\kappa\pi_i(t)}}{\sum_{j=1}^{n} x_j(t)e^{\kappa\pi_j(t)}}, \quad i = 1 \dots n \tag{6}$$

which enjoys many of the properties of the first-order system (4), e.g., they have the same set of equilibria.

The properties discussed above naturally suggest using replicator equations as a useful heuristic for graph and tree isomorphism problems. Considering graph isomorphism, let G_1 and G_2 be two graphs of order n, and let A denote the adjacency matrix of the corresponding N-vertex association graph G. By letting

$$W = A + \frac{1}{2}I_N$$

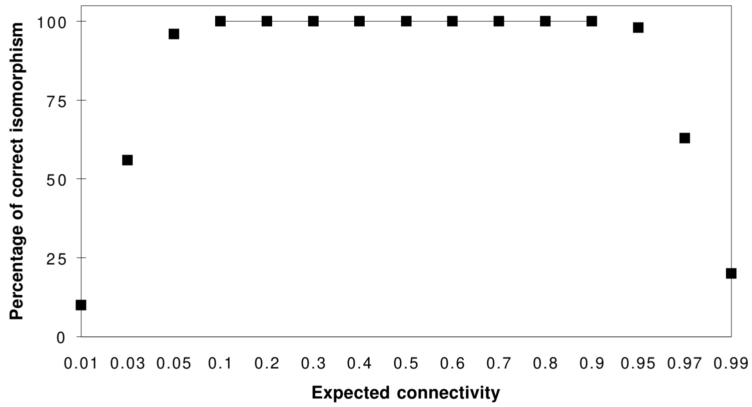
we know that the replicator dynamical systems, starting from an arbitrary initial state, will iteratively maximize the function $\hat{f}(\mathbf{x}) = \mathbf{x}^T(A + \frac{1}{2}I_N)\mathbf{x}$ in S_N, and will eventually converge to a strict local maximizer which, by virtue of Theorem 7 will then correspond to the characteristic vector of a maximal clique in the association graph. This will in turn induce an isomorphism between two subgraphs of G_1 and G_2 which is "maximal," in the sense that there is no other isomorphism between subgraphs of G_1 and G_2 which includes the one found. For subtree isomorphism, we proceed in the same way, that is by making replicator equations search for a maximal clique in the corresponding TAG. The converged solutions will induce a maximal subtree isomorphism.

Clearly, in theory there is no guarantee that the converged solution will be a *global* maximizer of $\hat{f}$, and therefore that it will induce a "maximum" isomorphism between the two original graphs. Previous work done on the maximum clique problem [8, 40], and also the results presented in this paper, however, suggest that the basins of attraction of global maximizers are quite large, and very frequently the algorithm converges to one of them.

5 Experiments and results

5.1 Graph isomorphism

In order to assess the effectiveness of the proposed approach, extensive simulations were performed over randomly generated graphs of various connectivities. Random graphs 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



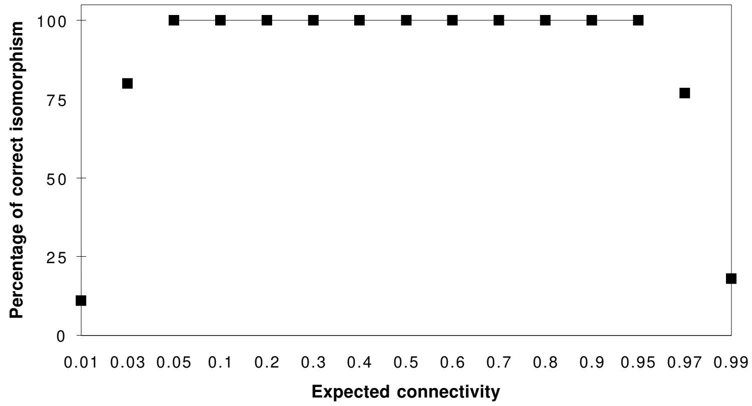


Figure 2: Percentage of correct isomorphisms obtained using the first-order (TOP) and the exponential (BOTTOM) replicator equations, as a function of the expected connectivity.

hence to make comparisons with other algorithms. Before going into the details of the experiments, however, we need to enter a preliminary caveat.

It is often said that random graph isomorphism is trivial. Essentially, this claim is based on a result due to Babai, Erdös and Selkow [2] which shows that a straightforward, linear-time graph isomorphism algorithm does work for *almost all* random graphs.[2] It should be pointed out, however, that there are various probability models for random graphs [37]. The one adopted by Babai considers random graphs as uniformly distributed random variables, i.e., they assume that the probability of generating any n-vertex graph equals $2^{-\binom{n}{2}}$. By contrast, the customary way in which random graphs are generated leads to a distribution which only in a special case is uniform. Specifically, given a parameter p $(0 < p < 1)$ which represents the expected connectivity, a graph of order n is generated by randomly entering edges between

[2]A property is said to hold for *almost all* graphs, if the probability that the property holds tends to 1 as the order of the graph approaches infinity.

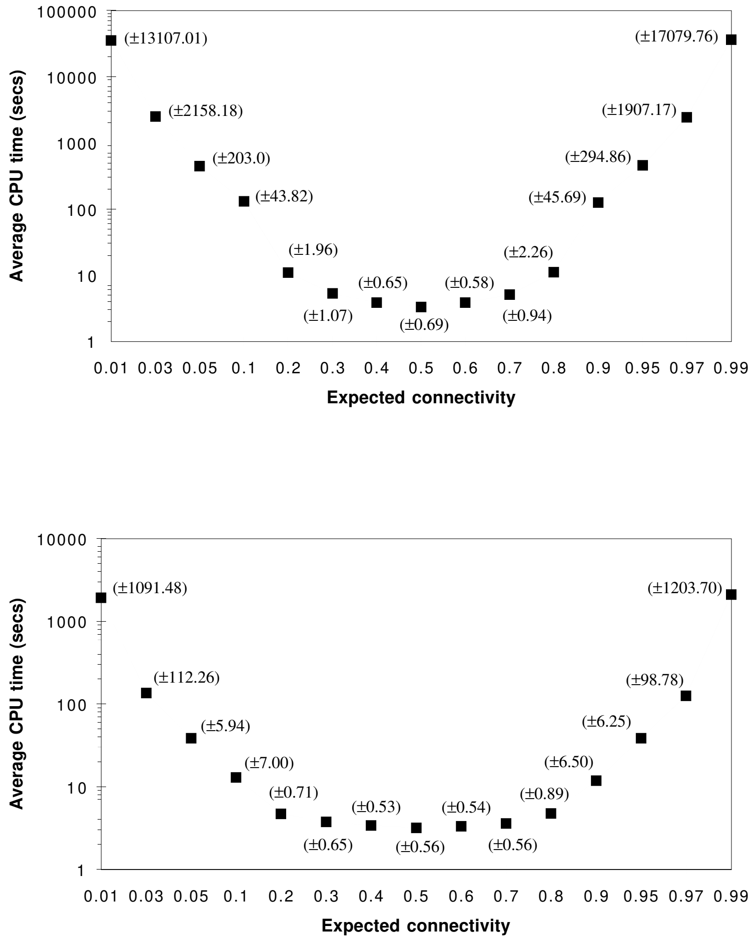


Figure 3: Average computational time taken by the first-order (TOP) and the exponential (BOTTOM) replicator equations, as a function of the expected connectivity. The vertical axes are in logarithmic scale, and the numbers in parentheses represent the standard deviation.

the vertices with probability p. Note that p is related to the expected size of the resulting graph, which indeed is $\binom{n}{2}p$. It is straightforward to see that in so doing the probability that a graph of order n and size s be generated is given by $p^s(1-p)^{\binom{n}{2}-s}$, which only in the case $p = \frac{1}{2}$ equals Babai's uniform distribution. Now, the results presented in [2] are based on the observation that, using a uniform probability model, the degrees of the vertices have large variability and this is in fact the key to their algorithm. In the non-uniform probability model, the degree random variable has variance $(n-1)p(1-p)$ and it is no accident that attains its largest value exactly at $p = \frac{1}{2}$. However, as p moves away from $\frac{1}{2}$ the variance becomes smaller and smaller, tending to 0 as p approaches 0 or 1. As a result, Babai's arguments are no longer applicable. It seems therefore that, using the customary graph generation model, random graph isomorphism is not as trivial as is generally believed, especially for very sparse and very dense graphs. As a matter of fact, the experience reported in [44, 45, 53] and also the results presented below, provide support to this claim.

In the experiments reported here, the discrete-time replicator equation (4) and its exponential counterpart (6) with $\kappa = 10$ were used. The algorithms were started from the barycenter of the simplex and they were stopped when either a maximal clique was found or the distance between two successive points was smaller than a fixed threshold, which was set to 10^{-17}. In the latter case the converged vector was randomly perturbed, and the algorithm 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. All the experiments were run on a Sparc20.

Undirected 100-vertex random graphs were generated with expected connectivities ranging from 1% to 99%. For each connectivity value, 100 graphs were produced and each of them had its vertices randomly permuted so as to obtain a pair of isomorphic graphs. Overall, therefore, 1500 pairs of isomorphic graphs were used. Each pair was given as input to the replicator models and, after convergence, a success was recorded when the cardinality of the returned clique was equal to the order of the graphs given as input (i.e., 100).[3] Because of the stopping criterion employed, this guarantees that a maximum clique, and therefore a correct isomorphism, was found. The proportion of successes as a function of the expected connectivities for both replicator models is plotted in Fig. 1, whereas Fig. 2 shows the average CPU time taken by the two algorithms to converge (in logarithmic scale). Notice how the exponential replicator system (6) is dramatically faster and also performs better than the first-order model (4).

These results are significantly superior to those reported by Simić [53] who obtained poor results at connectivities less than 40% even on smaller graphs (i.e., up to 75 vertices). They also compare favorably with the results obtained more recently by Rangarajan *et al.* [44] on 100-vertex random graphs for connectivities up to 50%. Specifically, at 1% and 3% connectivities they report a percentage of correct isomorphisms of about 30% and 0%, respectively. Using our approach we obtained, on the same kind of graphs, a percentage of success of 80% and 11%, respectively. Rangarajan and Mjolsness [45] also ran experiments on 100-vertex random graphs with various connectivities, using a powerful Lagrangian relaxation network. Except for a few instances, they always obtained a correct solution. The computational time required by their model, however, turns out to largely exceed ours. As an example, the average time taken by their algorithm to match two 100-vertex 50%-connectivity graphs was about 30 minutes on an SGI workstation. As shown in Fig. 2, we obtained identical results in about 3 seconds.

It should be emphasized that all the algorithms mentioned above do incorporate sophisticated annealing mechanisms to escape from poor local minima. By contrast,

[3]Due to the high computational time required, in the 1% and 99% cases the first-order replicator algorithm (4) was tested only on 10 pairs, instead of 100.

in the presented work no attempt was made to prevent the algorithm from converging to such solutions. It seems that, as far as the graph isomorphism problem is concerned, global maximizers of the Motzkin-Straus objective have large basins of attraction. A similar observation was also made in connection to earlier experiments concerning the maximum clique problem [8, 40].

5.2 Matching "shock" trees for visual shape recognition

We now illustrate our framework for matching hierarchical structures with numerical examples of shape matching. Because of the subtleties associated with generating random trees of relevance to applications in computer vision and pattern recognition, we use a class of example trees derived from a real system. Our representation for shape is based on an abstraction of the *shocks* (or singularities) of a curve evolution process, acting on a simple closed curve in the plane, into a shock tree. We begin by providing some background on the representation (for details see [25, 52]) and then present experimental results on matching shock trees.

5.2.1 The shock tree

In [25] the following evolution equation was proposed for visual shape analysis:

$$\begin{aligned} \mathcal{C}_t &= (1+\alpha\kappa)\mathcal{N} \\ \mathcal{C}(p,0) &= \mathcal{C}_0(p). \end{aligned} \tag{7}$$

Here $\mathcal{C}(p,t)$ is the vector of curve coordinates, $\mathcal{N}(p,t)$ is the inward normal, p is the curve parameter, and t is the evolutionary time of the deformation. The constant $\alpha \geq 0$ controls the regularizing effects of curvature κ. When α is large, the equation becomes a geometric heat equation; when $\alpha = 0$, the equation is hyperbolic and *shocks* [28], or entropy-satisfying singularities, can form. In the latter case the locus of points through which the shocks migrate is related to Blum's grassfire transformation [10, 25], although significantly more information is available via a "coloring" of these positions. Four types can arise, according to the local variation of the radius function along the medial axis (Figure 4). Intuitively, the radius function varies monotonically at a type 1, reaches a strict local minimum at a type 2, is constant at a type 3 and reaches a strict local maximum at a type 4. The classification of shock positions according to their colors, and an enumeration of the possible local neighborhoods around each shock type is at the heart of the representation.

Shocks of the same type that form a connected component are grouped together to comprise the nodes of a *shock graph*, with the 1-shock groups separated at branch-points of the skeleton. Directed edges in the graph are placed between shock groups that touch one another, such that each parent node contains no shocks that formed prior to any shocks in a child node. This corresponds to a "reversal" in time of the curve evolution process to obtain a hierarchy of connected components. The graph

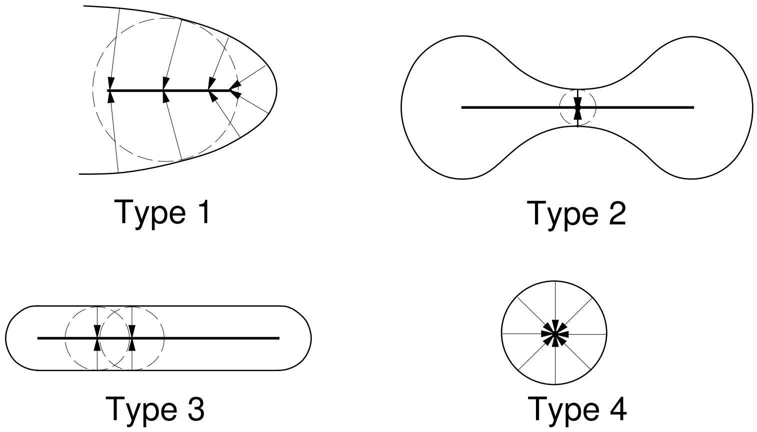


Figure 4: A coloring of shocks into four types [25]. A 1-shock derives from a *protrusion*, and traces out a curve segment of adjacent 1-shocks, along which the radius function varies monotonically. A 2-shock arises at a *neck*, where the radius function attains a strict local minimum, and is immediately followed by two 1-shocks flowing away from it in opposite directions. 3-shocks correspond to an annihilation into a curve segment due to a *bend*, along which the radius function is constant, and a 4-shock an annihilation into a point or a *seed*, where the radius function attains a strict local maximum. The loci of these shocks gives Blum's medial axis.

is rooted at a unique vertex #, the children of which are the last shock groups to form, e.g., the palm of the hand shape in Figure 5, with associated nodes 3-004a and 3-004b in the graph. (The letters 'a' and 'b' denote different sides of the same shock group). A key property of the shock graph is that its topological structure is highly constrained because the events that govern the birth, combination and death of shock groups are completely characterized by a shock grammar, with a small number of rewrite rules [52]. In particular, each shock graph can be reduced to a unique rooted shock tree.

5.2.2 Results

We now illustrate the power of the hierarchical structure matching algorithm on shock trees. Whereas each node has geometric attributes, related to properties of the shocks in each group, for now we shall consider only the shock tree topologies. (We shall consider the geometry shortly). We selected 25 silhouttes representing eight different object classes (Table 1, first column); the tool shapes were taken from the Rutgers Tools database. Each shape was then matched against *all* entries in the database. Figure 6 shows the maximal subtree isomorphisms found by the algorithm for three examples. The top 8 matches for each query shape, along with the associated scores, are shown in Table 1. The scores indicate the average of n/n_1 and n/n_2, where n is the size of the maximal clique found, and n_1 and n_2 are the number of nodes in each tree. We observed that in all our experiments the maximal cliques found were also

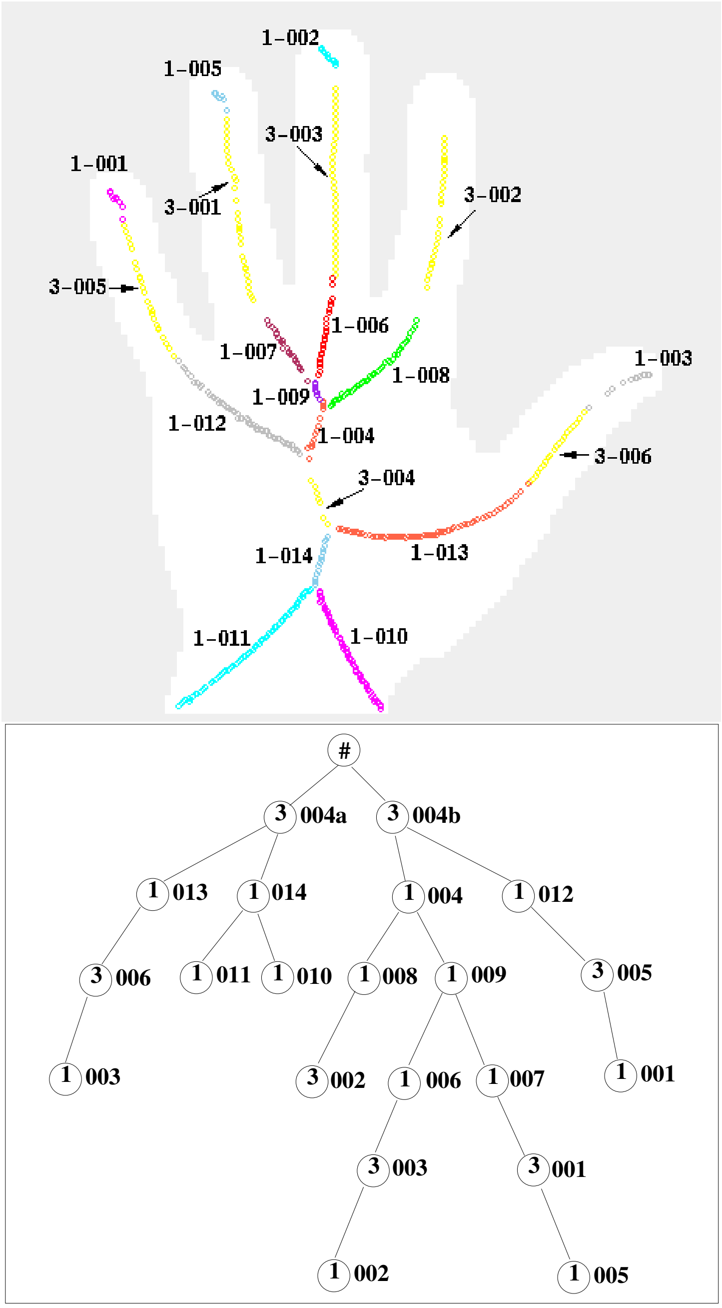


Figure 5: An illustrative example of the shocks obtained from curve evolution (from [52]). TOP: The notation associated with the locus of shock points is of the form shock_type-identifier. BOTTOM: The tree has the shock_type on each node, and the identifier is adjacent. The last shocks to form during the curve evolution process appear under the root node labeled #.

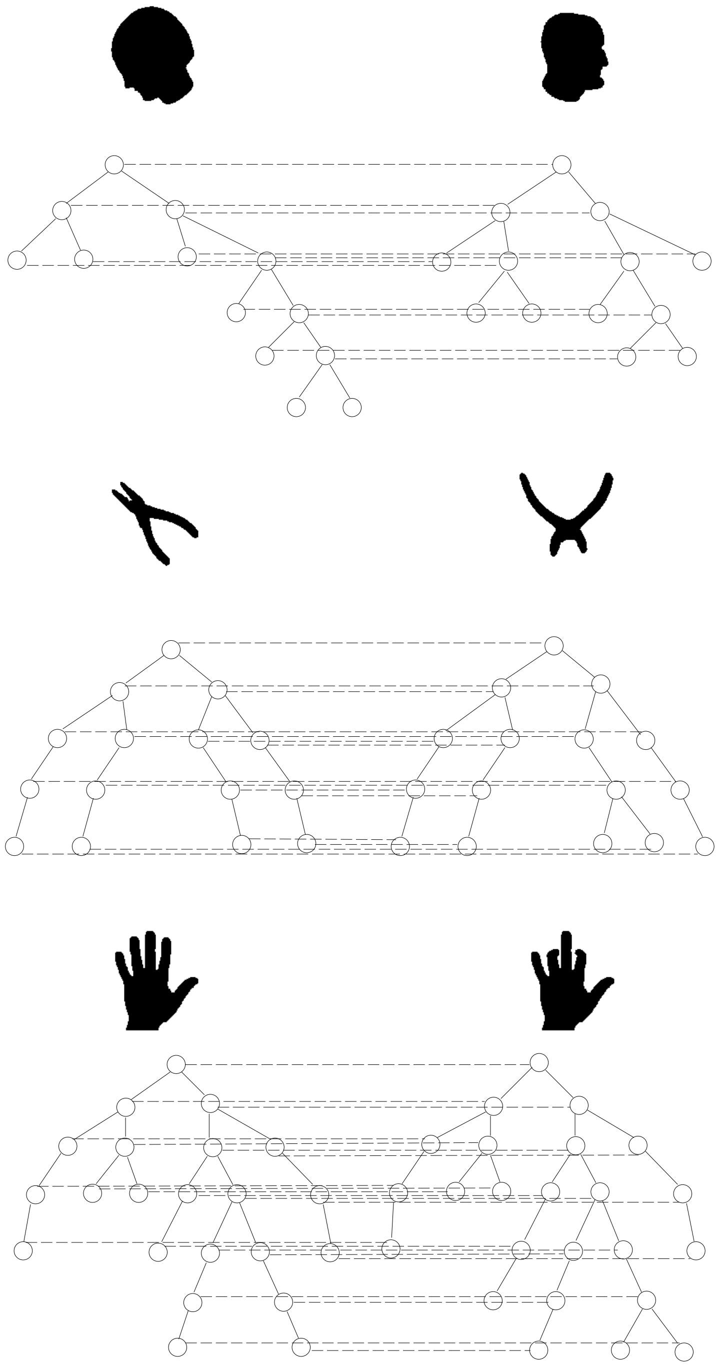


Figure 6: Maximal subtree isomorphisms found for three illustrative examples.

Query Shape	Top 8 Topological Matches							
	1	2	3	4	5	6	7	8
	1.00	1.00	.916	.900	.825	.771	.750	.750
	1.00	1.00	.916	.900	.825	.771	.750	.750
	1.00	.900	.900	.833	.833	.833	.833	.807
	1.00	.958	.875	.825	.729	.666	.641	.641
	1.00	.909	.875	.826	.738	.738	.738	.711
	1.00	.958	.909	.859	.755	.668	.668	.609
	1.00	1.00	.966	.966	.900	.800	.784	.771
	1.00	.937	.937	.904	.875	.750	.731	.731
	1.00	1.00	.966	.966	.937	.928	.773	.771
	1.00	.928	.928	.900	.900	.875	.833	.825
	1.00	1.00	.966	.966	.937	.928	.773	.771
	1.00	1.00	.966	.966	.904	.900	.800	.784
	1.00	1.00	.801	.750	.733	.733	.720	.708
	1.00	1.00	.801	.750	.733	.733	.720	.708
	1.00	.807	.801	.801	.801	.801	.722	.721
	1.00	.900	.826	.755	.750	.729	.675	.675
	1.00	.900	.859	.825	.738	.675	.600	.600
	1.00	.977	.956	.800	.800	.785	.785	.773
	1.00	.977	.933	.784	.784	.772	.772	.759
	1.00	.956	.933	.771	.771	.760	.760	.746
	1.00	.916	.916	.833	.833	.833	.785	.772
	1.00	1.00	.833	.833	.801	.733	.733	.720
	1.00	1.00	.833	.833	.801	.733	.733	.720
	1.00	.687	.675	.675	.656	.612	.600	.600
	1.00	.693	.692	.692	.687	.673	.673	.661

Table 1: A tabulation of the top 8 topological matches for each query. The scores indicate the average of the fraction of nodes matched in each of the two trees (see text). Note that *only* the topology of the shock trees was used; the addition of geometric information permits finer discrimination.

maximum cliques, thereby confirming previous observations concerning the basins of attraction of the Motzkin-Straus quadratic program for maximum clique [40, 41, 8]. The matching algorithm generally takes only two to three seconds to converge on a Sparc 10.

Note that despite the fact that metric/label information associated with nodes in the shock trees was discounted altogether, all exemplars in the same class as the query shape are typically within the top 5 matches, illustrating the potential of a topological matching process for indexing into a database of shapes. Nevertheless, there are shapes with similar shock tree topologies but dissimilar geometries, e.g., the profile and the brushes in the first two rows. In [43] we extended our framework to incorporate geometric information contained in each shock sequence (the location, time of formation, speed and direction of each shock) as attributes on the nodes. This leads to better discrimination between shapes than that provided by shock tree topologies alone.

6 Conclusions

In this paper, we have developed a continuous quadratic programming approach for graph- and tree-isomorphism problems which is based on an equivalent maximum clique formulation. The approach is centered around a fundamental result proved by Motzkin and Straus in the mid-1960s, and recently expanded in various ways, which allows us to formulate the maximum clique problem in terms of a standard quadratic program. To approximately solve the program we use the so-called "replicator" equations, a class of straightforward continuous- and discrete-time dynamical systems developed in various branches of theoretical biology. We show how, despite their inherent inability to escape from local solutions, they nevertheless provide experimental results which are competitive with those obtained using more sophisticated mean-field annealing heuristics. Application of this approach to shape matching problems arising in computer vision and pattern recognition have been presented.

In [43], we extend the framework to handle the matching of attributed trees, by casting the problem as that of finding a maximum weight clique in a weighted association graph. A recent generalization of the Motzkin-Straus theorem applies [15], allowing the use of the same replicator dynamics as in the unweighted case. A generalization to many-to-one mappings is also under investigation.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. In *Proc. 33rd Ann. Symp. Found. Comput. Sci.*, pages 14–23. Pittsburgh, PA, 1992.

[2] L. Babai, P. Erdös, and S. M. Selkow. Random graph isomorphism. *SIAM J. Comput.*, 9(3):628–635, 1980.

[3] H. G. Barrow and R. M. Burstall. Subgraph isomorphism, matching relational structures, and maximal cliques. *Inform. Process. Lett.*, 4(4):83–84, 1976.

[4] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

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

[6] 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*, volume 4. Kluwer Academic Publishers, Boston, MA, 1999.

[7] I. M. Bomze, M. Budinich, M. Pelillo, and C. Rossi. Annealed replication: A new heuristic for the maximum clique problem. Technical Report CS98-10, Università di Venezia, Dipartimento di Informatica, 1998. submitted for publication.

[8] I. M. Bomze, M. Pelillo, and R. Giacomini. Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale. In I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, editors, *Developments in Global Optimization*, pages 95–108. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[9] R. B. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inform. Process. Lett.*, 25:127–132, 1987.

[10] R. Brockett and P. Maragos. Evolution equations for continuous-scale morphology. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA, March 1992.

[11] R. Durbin and D. Willshaw. An analog approach to the travelling salesman problem using an elastic net method. *Nature*, 326:689–691, 1987.

[12] Y. Fu and P. W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimization. *J. Phys. A*, 19:1605–1620, 1986.

[13] 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.

[14] L. E. Gibbons, D. W. Hearn, and P. M. Pardalos. A continuous based heuristic for the maximum clique problem. In D. S. Johnson and M. Trick, editors, *Cliques, Coloring, and Satisfiability—Second DIMACS Implementation Challenge*, pages 103–124. American Mathematical Society, 1996.

[15] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana. Continuous characterizations of the maximum clique problem. *Math. Oper. Res.*, 22:754–768, 1997.

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

[17] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988.

[18] F. Harary. *Graph Theory.* Addison-Wesley, Reading, MA, 1969.

[19] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proc. 37th Ann. Symp. Found. Comput. Sci.*, pages 627–636, 1996.

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

[21] J. Hofbauer and K. Sigmund. *The Theory of Evolution and Dynamical Systems.* Cambridge University Press, Cambridge, UK, 1988.

[22] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biol. Cybern.*, 52:141–152, 1985.

[23] A. Jagota. Approximating maximum clique with a Hopfield network. *IEEE Trans. Neural Networks*, 6:724–735, 1995.

[24] D. S. Johnson. The NP-completeness column: An ongoing guide. *J. Algorithms*, 9:426–444, 1988.

[25] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *Int. J. Comp. Vision*, 15:189–224, 1995.

[26] J. J. Kosowsky and A. L. Yuille. The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural Networks*, 7:477–490, 1994.

[27] D. Kozen. A clique problem equivalent to graph isomorphism. *SIGACT News*, pages 50–52, Summer 1978.

[28] P. D. Lax. Shock waves and entropy. In E. H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 603–634, New York, 1971. Acad. Press.

[29] J. T. Li, K. Zhang, K. Jeong, and D. Shasha. A system for approximate tree matching. *IEEE Trans. Knowledge Data Eng.*, 6:559–571, 1994.

[30] V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *J. Math. Biol.*, 17:241–251, 1983.

[31] S. Y. Lu. A tree-matching algorithm based on node splitting and merging. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:249–256, 1984.

[32] D. Marr and K. H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. R. Soc. Lond. B*, 200:269–294, 1978.

[33] D. W. Matula. An algorithm for subtree identification. *SIAM Rev.*, 10:273–274, 1968.

[34] 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.

[35] M. Neff, R. Byrd, and O. Rizk. Creating and querying hierarchical lexical databases. In *Proc. 2nd Conf. Appl. Natural Language Process.*, pages 84–93, 1988.

[36] M. Ohlsson, C. Peterson, and B. Söderberg. Neural networks for optimization problems with inequality constraints: The knapsack problem. *Neural Computation*, 5:331–339, 1993.

[37] E. M. Palmer. *Graphical Evolution: An Introduction to the Theory of Random Graphs.* John Wiley & Sons, New York, 1985.

[38] P. M. Pardalos. Continuous approaches to discrete optimization problems. In G. D. Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, pages 313–328. Plenum Press, 1996.

[39] P. M. Pardalos and A. T. Phillips. A global optimization approach for solving the maximum clique problem. *Int. J. Comput. Math.*, 33:209–216, 1990.

[40] M. Pelillo. Relaxation labeling networks for the maximum clique problem. *J. Artif. Neural Networks*, 2:313–328, 1995.

[41] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11, 1999. in press.

[42] M. Pelillo and A. Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Networks*, 2:411–420, 1995.

[43] M. Pelillo, K. Siddiqi, and S. W. Zucker. Attributed tree matching and maximum weight cliques. submitted to *ICIAP'99—10th Int. Conf. on Image Analysis and Processing*, Venice, Italy, Sept. 1999.

[44] A. Rangarajan, S. Gold, and E. Mjolsness. A novel optimizing network architecture with applications. *Neural Computation*, 8:1041–1060, 1996.

[45] A. Rangarajan and E. Mjolsness. A lagrangian relaxation network for graph matching. *IEEE Trans. Neural Networks*, 7(6):1365–1381, 1996.

[46] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, 6:730–732, 1977.

[47] H. Rom and G. Medioni. Hierarchical decomposition and axial shape description. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(10):973–981, 1993.

[48] H. Samet. *Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.

[49] U. Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. Syst. Sci.*, 37:312–323, 1988.

[50] B. A. Shapiro and K. Zhang. Comparing multiple RNA secondary structures using tree comparisons. *Comput. Appl. Biosci.*, 6:309–318, 1990.

[51] D. Shasha, J. T. L. Wang, K. Zhang, and F. Y. Shih. Exact and approximate algorithms for unordered tree matching. *IEEE Trans. Syst. Man Cybern.*, 24:668–678, 1994.

[52] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. Comp. Vision*, to appear, 1998.

[53] P. D. Simić. Constrained nets for graph matching and other quadratic assignment problems. *Neural Computation*, 3:268–281, 1991.

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

[55] H. S. Wilf. Spectral bounds for the clique and independence numbers of graphs. *J. Combin. Theory, Ser. B*, 40:113–117, 1986.

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