A game–theoretic approach to partial clique enumeration

Samuel Rota Bulò *, Andrea Torsello, Marcello Pelillo

Dipartimento di Informatica, Università "Ca Foscari" di Venezia, via Torino 155, 30172 Venezia Mestre, Italy

ARTICLE INFO

ABSTRACT

In many computer vision and pattern recognition applications using graph-based representations, it is of great interest to be able to extract the k largest cliques in a graph. However, most methods are geared either towards extracting a single clique of maximum size, or enumerating all cliques, without following any particular order. In this paper, we present a novel approach for partial clique enumeration, which is the problem of extracting the k largest cliques of a graph. Our approach is based on a continuous formulation of the clique problem developed in the 1960s by Motzkin and Straus, and is able to avoid extracting the same clique multiple times. This is done by casting the problem into a game–theoretic framework, where stable strategies are in correspondence with maximal cliques, and by iteratively rendering the extracted solutions unstable. The approach has been tested on the maximum clique problem and compared against several state-of-the-art algorithms both on random as well as DIMACS benchmark graphs. Further, we applied our enumerative heuristic to the matching of shapes using the shock-graph representation. The results confirm the effectiveness of the approach.

1. Introduction

Many problems in computer vision and pattern recognition can be formulated in terms of finding a completely connected subgraph (i.e., a *clique*) of a given graph, having largest cardinality. This is called the Maximum Clique Problem (MCP). One popular approach to object recognition, for example, involves matching an input scene against a stored model, each being abstracted in terms of a relational structure [3,17,43,47], and this problem, in turn, can be conveniently transformed into the equivalent problem of finding a maximum clique of the corresponding *association graph*. This idea was pioneered by Ambler et al. [1] and was later developed by Bolles and Cain [7] as part of their local-feature-focus method. Now, it has become a standard technique in computer vision, and has been employing in such diverse applications as stereo correspondence [25], point pattern matching [35], and image sequence analysis [44]. Other interesting applications of the maximum clique problem arise in the context of cluster analysis, where graph-theoretical methods have long proven to be especially effective [2,28,37], and in the context of category learning and knowledge discovery [18,34]. Furthermore, clique finding is also linked with the learning of graphical structure by the Hammersley–Clifford theorem [21].

From a computational point of view, the MCP belongs to the class of NP-Complete problems, whose intractability forces us to fall back on approximation methods. Unfortunately, even approximat-

ing the MCP is intractable [23]. Due to this pessimistic state of affairs, much attention has gone into developing efficient heuristics for the MCP, for which no formal guarantee of performance may be provided, but are nevertheless useful in practical applications. We refer to Bomze et al. [9] for a survey concerning algorithms, applications, and complexity issues of this important problem.

In a recent series of papers [8,12,16,20,39,42], a novel approach has emerged, which is based around a classical result from graph theory due to Motzkin and Straus [33] that allows us to formulate the MCP as a continuous quadratic optimization problem with simplex constraints. This program is typically solved using the replicator dynamics, a well-known class of continuous- and discrete-time dynamical systems developed and studied in the field of evolutionary game theory. Although previous work has focused mainly on extracting a single maximum clique, in several contexts it is of great interest to be able to extract multiple (say k) large cliques. For example, in knowledge discovery, where categories are abstracted in terms of cliques, each element can belong to multiple categories, which prompts us to look for more than one category [18,34,36]. In computational biology Shi et al. [45] equate the problem of predicting the loop of protein 3D structure to a problem of finding the maximal cliques with the best weight, while in the field of computer vision, Horaud and Skordas [25] use the largest cliques to find stereo correspondences in image pairs. In the latter work, the authors extract straight lines from each image and enrich them with attributes and neighborhood relationships. The obtained structural descriptions are represented with relational graphs and stereo correspondences between images are extracted by matching them. Horaud and Skordas however noticed that among

---

* Corresponding author. Tel.: +39 041 2348496; fax: +39 041 2348419.
E-mail addresses: srotabul@dsi.unive.it (S. Rota Bulò), torsello@dsi.unive.it (A. Torsello), pelillo@dsi.unive.it (M. Pelillo).

the many maximal cliques detected, the largest one did not always provide for the best solution. Therefore they enumerate several large cliques and take the match, which maximizes an objective function, that could not be formulated in terms of the weight of the clique.

The clique enumeration literature mainly deals with exhaustive searches and hence, enumerate, either implicitly or explicitly, all cliques (in particular all maximal cliques) of an undirected graph. The first enumerative algorithm is probably due to Harary and Ross [22], followed by several others [13,15,29,51]. However, these algorithms can only guarantee that the cliques are extracted in lexicographical order, and no guarantee is given about the size of the cliques extracted at each step.

In this paper, we develop an approach which uses a game–theoretic formulation to enumerate a user-defined number of large cliques. Ideally, we would like to obtain the k largest maximal cliques after a small number of enumerations. The proposed approach is based on the fact that under a certain family of quadratic problems, there is a one-to-one correspondence between asymptotically stable points of the replicator dynamics and maximal cliques. Once we have extracted a maximal clique, we would like to force the dynamics not to converge to the same clique and this can be done by rendering the associated rest point unstable. To this end, we introduce a graph extension process which, given a maximal clique that we want to avoid, produces a directed graph which guarantees a one-to-one correspondence between the evolutionary stable points of the associated replicator dynamics and the remaining maximal cliques of the original graph.

The idea of rendering unstable some solutions under the replicator dynamics in order to improve the size of the extracted cliques was already used by Pelillo and co-workers in [12,42], where it is shown that by adding to each vertex a self-loop with a properly chosen negative weight α, a class of unwanted solutions become unstable. The main disadvantage associated with this approach is, however, the emergence of new spurious stable solutions, which do not correspond to cliques. To overcome the problem, instead of keeping the value of α fixed, the authors start with a sufficiently large negative value and adaptively increase it during the optimization process, in much the same spirit as simulated or mean field annealing procedures. Clearly, our solution differs from this one, since we are able to render unstable only selected unwanted solutions, without affecting other solutions.

To test the effectiveness of the proposed approach at extracting large cliques first, we conducted experiments on the maximum clique problem and compared it against several state-of-the-art algorithms both on random as well as DIMACS benchmark graphs. Further, we applied our method to a shape recognition problem; namely, the matching of shock-graphs.

The paper is organized as follows: in Section 2 we present a continuous characterization of the maximum clique problem in terms of solutions of a standard quadratic problem. In Section 3 we discuss evolutionary game theory and show a characterization of cliques in terms of classical game–theoretic equilibria. This generalizes the results presented in Section 2 and allows us to extend the notion of maximal clique to directed graphs. In Section 4 we introduce the replicator dynamics as a tool for finding maximal cliques and Section 5 presents our game–theoretic approach to maximal clique enumeration. Section 6 is dedicated to the experiments on the maximum clique problem and on shape matching and, finally, in Section 7 we draw some conclusions.

2. A family of quadratic programs for maximum clique

Let $G = (V, E)$ be an undirected graph without self-loops, where $V = \{1, 2, \ldots, n\}$ is the set of vertices and $E \subseteq V \times V$ the set of edges.

We define the *order* of a graph G as the cardinality of V. Two vertices $u, v \in V$ are *adjacent* if $(u, v) \in E$. A subset C of vertices in G is called a *clique* if all its vertices are mutually adjacent. It is a *maximal clique* if it is not a subset of other cliques in G. It is a *maximum clique* if it has maximum cardinality. The cardinality of a maximum clique of G is also called *clique number* and denoted by $\omega(G)$.

The *adjacency matrix* of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$, where $a_{ij} = 1$ if $(i, j) \in E$, $a_{ij} = 0$ otherwise.

Now, consider the following constrained quadratic program.

$$\begin{aligned} \text{maximize} \quad & f_\alpha(\mathbf{x}) = \mathbf{x}'(A_G + \alpha I)\mathbf{x}, \\ \text{subject to} \quad & \mathbf{x} \in \Delta \subset \mathbb{R}^n \end{aligned} \quad , \tag{2.1}$$

where n is the order of G, I the identity matrix, α is a real parameter and where Δ is the *standard simplex* of the n-dimensional Euclidean space given by

$$\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{e}'\mathbf{x} = 1, \quad \mathbf{x} \geqslant 0\},$$

where $\mathbf{e}$ is the vector with all components equal to 1.

In 1965 Motzkin and Straus [33] established a connection between the maximum clique problem and the program in (2.1) with $\alpha = 0$; they related the clique number of G to global solutions $\mathbf{x}^*$ of the program through the formula $\omega(G) = (1 - f_0(\mathbf{x}^*))^{-1}$, and showed that a subset of vertices C with cardinality $|C|$ is a maximum clique of G if and only if its *characteristic vector* $\mathbf{x}^C \in \Delta$, where

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

is a global maximizer of f_0 on Δ.[1] Pelillo and Jagota [41], extended the Motzkin–Straus theorem by providing a characterization of maximal cliques in terms of local maximizers of f_0 in Δ.

A drawback of the original Motzkin–Straus formulation is the existence of "spurious" solutions, i.e., maximizers of f_0 that are not in the form of characteristic vectors [41]. Bomze et al. [10] proved that for $0 < \alpha < 1$ all local maximizer of (2.1) are strict and are in one-to-one relation with the characteristic vectors of the maximal cliques of G, thus, overcoming the problem.

Following [39,42], the problem of finding the maximizers of (2.1) is cast in a game–theoretic setting, where the replicator dynamics, a well-known formalization of the selection process, are used as a growth transformation. In the next section we will review some concepts from evolutionary game theory that will be useful throughout the paper and provide the link between game theory and maximal cliques.

3. A game–theoretic perspective

Let $O = \{1, 2, \ldots, n\}$ be the set of *pure strategies* available to the players and $A = (a_{ij})$ the $n \times n$ payoff or utility matrix [53] where a_{ij} is the payoff that a player gains when playing the strategy i against an opponent playing strategy j. In biological contexts, payoffs are typically measured in terms of Darwinian fitness or reproductive success whereas in economics applications, they usually represent firms' profits or consumers' utilities.

A *mixed strategy* is a probability distribution $\mathbf{x} = (x_1, x_2, \ldots, x_n)'$ over the available strategies in O. Mixed strategies clearly lie in the standard simplex Δ of the n-dimensional Euclidean space. The *support* of a mixed strategy $\mathbf{x} \in \Delta$, denoted by $\sigma(\mathbf{x})$, defines the set of elements with non-zero probability: $\sigma(\mathbf{x}) = \{i \in O : x_i > 0\}$.

The expected payoff that a player obtains by playing the element i against an opponent playing a mixed strategy $\mathbf{x}$ is

---

[1] In the original paper, Motzkin and Straus proved the "only-if" part of this theorem. The converse however is a straightforward consequence of their result (Pelillo and Jagota [41]).

$(A\boldsymbol{x})_i = \sum_j a_{ij}x_j$. Hence, the expected payoff received by adopting a mixed strategy $\boldsymbol{y}$ is $\boldsymbol{y}'A\boldsymbol{x}$.

Evolutionary game theory considers an idealized scenario wherein pairs of individuals are repeatedly drawn from a large population to play a two-player symmetric game. Each player is not supposed to behave rationally or have a complete knowledge of the details of the game, but he acts according to a pre-programmed pure strategy. This dynamic activates some selection process that results in the evolution of the fittest strategies. A well-known formalization of the selection process is given by the replicator equations (see Section 4).

The *best replies* against a mixed strategy $\boldsymbol{x}$ is the set of mixed strategies $\beta(\boldsymbol{x}) = \{\boldsymbol{y} \in \Delta : \boldsymbol{y}'A\boldsymbol{x} = \max_z \boldsymbol{z}'A\boldsymbol{x}\}$.

A mixed strategy $\boldsymbol{x}$ is a *Nash equilibrium* if it is a best reply to itself, i.e., $\forall \boldsymbol{y} \in \Delta$, $\boldsymbol{y}'A\boldsymbol{x} \leqslant \boldsymbol{x}'A\boldsymbol{x}$. This implies that for all $i \in \sigma(\boldsymbol{x})$, $(A\boldsymbol{x})_i = \boldsymbol{x}'A\boldsymbol{x}$, hence the payoff of every strategy in the support of $\boldsymbol{x}$ is constant, while all strategies outside the support of $\boldsymbol{x}$ earn a payoff that is less than or equal to $\boldsymbol{x}'A\boldsymbol{x}$.

A strategy $\boldsymbol{x}$ is said to be an *Evolutionary Stable Strategy* (ESS) if it is a Nash equilibrium and for each best reply $\boldsymbol{y}$ to $\boldsymbol{x}$, i.e., such that $\boldsymbol{y}'A\boldsymbol{x} = \boldsymbol{x}'A\boldsymbol{x}$, we have $\boldsymbol{x}'A\boldsymbol{y} > \boldsymbol{y}'A\boldsymbol{y}$. This second condition can also be rephrased as having $\varepsilon'A\varepsilon < 0$ for all $\varepsilon \in M(\boldsymbol{x})$ where

$$M(\boldsymbol{x}) = \{\varepsilon \in \mathbb{R}^n : \varepsilon'\boldsymbol{e} = 0 \text{ and } \varepsilon_i = 0, \quad \forall i \cdot (A\boldsymbol{x})_i < \boldsymbol{x}'A\boldsymbol{x}\}. \tag{3.1}$$

Intuitively, *ESS*s are strategies such that any small deviation will be reabsorbed as it will lead to an inferior payoff.

Consider the following quadratic program

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{x}'A\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x} \in \Delta \subset \mathbb{R}^n, \end{aligned} \tag{3.2}$$

where A is a $n \times n$ symmetric matrix. We have that $\boldsymbol{x}$ is a Nash equilibrium of a two-player game with payoff matrix A, if and only if it satisfies the first order Karush–Kuhn–Tucker (KKT) conditions [32] for (3.2). Hence, local solution of (3.2) are indeed Nash equilibria, but the converse does not necessarily hold. On the other hand, we have that $\boldsymbol{x}$ is an *ESS* of a two-player game with payoff matrix A, if and only if it is a strict local maxima of (3.2) [53]. Therefore, a link can be established between maximal cliques of a graph G and *ESS*s if we take $A_G + \alpha I$ with $0 < \alpha < 1$ as payoff matrix. We refer to [8] and [10] for a deeper insight of the relation between *ESS* and maximal cliques.

Through this change in perspective, we can move from an optimization to a game–theoretic setting: Instead of finding local solutions of a quadratic program, we look for *ESS*s of two-persons symmetric games. The advantage of this new approach is that we can generalize the Motzkin–Straus result to non-symmetric payoff matrices and hence, to directed graphs.

Let $G = (V, E)$ be a directed graph. A *doubly-linked clique* of G is a set $S \subseteq V$ such that for all $u, v \in S$, we have $(u, v) \in E$ and $(v, u) \in E$. The clique is *saturated* if there is no $t \in V \setminus S$ such that for all $s \in S$, $(t, s) \in E$.

For example, in Fig. 1 we have a directed graph where the subset of vertices $\{A, B, C\}$ forms a doubly-linked clique, but it is not saturated because D is connected to all its vertices. On the other hand, $\{A, D\}$ is a saturated doubly-linked clique.

The following result establishes a one-to-one correspondence between *ESS* and saturated doubly-linked cliques in directed graphs (see also [50]).

Theorem 1. *Let $G = (V, E)$ be a directed graph with adjacency matrix A and $0.5 \leqslant \alpha < 1$. A vector $\boldsymbol{x} \in \Delta$ is an ESS for a two-player game with payoff matrix $B = A + \alpha I$ if and only if it is the characteristic vector of a saturated doubly-linked clique of G.*

Proof. $(\Leftarrow)$ If S is a saturated doubly-linked clique then $\boldsymbol{x}^S$ is a strict Nash equilibrium. In fact
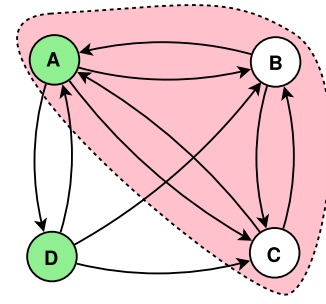

Fig. 1. In this directed graph we have that $\{A, B, C\}$ is doubly-linked clique, but it is not saturated because of D, while $\{A, D\}$ is a saturated doubly-linked clique.

$$(B\boldsymbol{x})_i \begin{cases} = 1 - \frac{(1-\alpha)}{|S|} = \boldsymbol{x}'B\boldsymbol{x}, & \text{if } i \in \sigma(\boldsymbol{x}) \\ \leqslant 1 - \frac{1}{|S|} < \boldsymbol{x}'B\boldsymbol{x} & \text{if } i \notin \sigma(\boldsymbol{x}), \end{cases}$$

where the inequality for vertices outside the support derives from the saturation property. In order for $\boldsymbol{x}$ to be an *ESS* we should have that $\varepsilon'B\varepsilon < 0$ for all $\varepsilon \in M(\boldsymbol{x})$ (see Eq. (3.1)). But this is indeed the case, because $\sigma(\boldsymbol{x})$ is a doubly-linked clique and ε_i is zero outside the support of $\boldsymbol{x}$. Hence,

$$\varepsilon'B\varepsilon = \varepsilon'[(\alpha - 1)I + \boldsymbol{e}\boldsymbol{e}']\varepsilon = (\alpha - 1)\varepsilon'\varepsilon < 0$$

which concludes the first part of the proof.

$(\Rightarrow)$ Assume that $b_{ij} = 0$ for two distinct $i, j \in \sigma(\boldsymbol{x})$. Let $\boldsymbol{y} = \boldsymbol{x} + \varepsilon(\boldsymbol{e}^i - \boldsymbol{e}^j)$, where $0 < \varepsilon \leqslant x_j$ and $\boldsymbol{e}^i$ is the vector with all components equal zero except for the ith-component which is equal to 1. Note that $\boldsymbol{y}$ is a best reply to $\boldsymbol{x}$, in fact

$$\boldsymbol{y}'B\boldsymbol{x} - \boldsymbol{x}'B\boldsymbol{x} = \varepsilon(\boldsymbol{e}^i - \boldsymbol{e}^j)'B\boldsymbol{x} = 0.$$

But then

$$\begin{aligned} \boldsymbol{x}'B\boldsymbol{y} - \boldsymbol{y}'B\boldsymbol{y} &= -\varepsilon(\boldsymbol{e}^i - \boldsymbol{e}^j)'B\boldsymbol{y} \\ &= -\varepsilon(\boldsymbol{e}^i - \boldsymbol{e}^j)'B\boldsymbol{x} + \varepsilon^2(\boldsymbol{e}^i - \boldsymbol{e}^j)'B(\boldsymbol{e}^i - \boldsymbol{e}^j) \\ &= -\varepsilon^2(b_{ii} + b_{jj} - b_{ij} - b_{ji}) = -\varepsilon^2(2\alpha - b_{ji}) \leqslant 0 \end{aligned}$$

which contradicts the evolutionary stability of $\boldsymbol{x}$.

It follows, by the Nash condition, that for all $i \in S$, $(B\boldsymbol{x})_i = 1 - (1 - \alpha)x_i = \boldsymbol{x}'B\boldsymbol{x}$, which implies that x_i is constant for all $i \in S$, i.e., $\boldsymbol{x}$ is the characteristic vector of S.

Finally, the doubly-linked clique S is saturated, because otherwise, by definition, there exists a vertex k such that for all $i \in \sigma(\boldsymbol{x})$ and $i \neq k$, $B_{ki} = 1$. But then for any $i \in \sigma(\boldsymbol{x})$ we have that $(B\boldsymbol{x})_k = 1 > (B\boldsymbol{x})_i = 1 - (1 - \alpha)/|S|$, which contradicts the Nash condition and hence, $\boldsymbol{x}$ cannot be an *ESS*. $\square$

Note that unlike the symmetric case, where α can be chosen between 0 and 1, Theorem 1 does not hold for $0 < \alpha < 0.5$. In fact, take $0 < \alpha < 0.5$ and let G be the cyclic graph of order 3. Then B is given by

$$B = \begin{bmatrix} \alpha & 1 & 0 \\ 0 & \alpha & 1 \\ 1 & 0 & \alpha \end{bmatrix}.$$

Here, $\boldsymbol{x} = \boldsymbol{e}/3$ is a Nash equilibrium for B for any choice of α and $\forall \varepsilon \in M(\boldsymbol{x})$,

$$\varepsilon'B\varepsilon = \frac{1}{2}\varepsilon'(B + B')\varepsilon = \frac{1}{2}\varepsilon'[(2\alpha - 1)I + \boldsymbol{e}\boldsymbol{e}']\varepsilon = (\alpha - 0.5)\varepsilon'\varepsilon < 0.$$

Hence, $\boldsymbol{x}$ is an *ESS*, but $\sigma(\boldsymbol{x})$ is not a doubly-linked clique of G. This shows that the bounds on α are tight and Theorem 1 does not hold for $0 < \alpha < 0.5$.

We have already seen that if we consider an undirected graph G and the payoff matrix $A_G + \alpha I$ with $0 < \alpha < 1$, then the *ESS*s of the

related two-player game are in one-to-one correspondence with maximal cliques of G. Here we can see that if $0.5 \leqslant \alpha < 1$ the concept of saturated doubly-linked clique is a direct generalization to the asymmetric case of the concept of maximal clique, i.e., *ESS*s are in one-to-one correspondence with saturated doubly-linked cliques.

4. Replicator dynamics

In this section, we discuss the replicator dynamics, which is instrumental for finding *ESS*s, and for that it will be used in the subsequent development of our partial clique enumeration algorithm. For a more systematic treatment see [53].

The Replicator Dynamics (RD) is a well-known formalization of the selection process that takes place within an evolutionary game, where 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. If successive generations blend into each other, the evolution of behavioral phenotypes can be described by the following set of differential equations [48]:

$$\dot{x}_i = x_i \left[(A\boldsymbol{x})_i - \boldsymbol{x}'A\boldsymbol{x} \right] \quad \text{with} \quad i = 1\ldots n, \tag{4.1}$$

where a dot signifies derivative with respect to time. The basic idea behind this model is that the strategies earning higher payoffs than the population mean payoff will grow faster while those with lower payoff will be driven to extinction.

A discrete-time version of (4.1) is given by

$$x_i^{(t+1)} = x_i^{(t)} \frac{(A\boldsymbol{x}^{(t)})_i}{\boldsymbol{x}^{(t)'}A\boldsymbol{x}^{(t)}} \quad \text{with} \quad i = 1\ldots n, \tag{4.2}$$

which can also be regarded as a very special case of a general class of dynamical systems introduced by Baum and Eagon [4] and studied by Baum and Sell [5] in the context of Markov chain theory. Note that for both the continuous and the discrete dynamics, the simplex Δ is invariant, i.e., if started in Δ, the dynamics will remain there.

A point $\boldsymbol{x}$ is said to be *stationary* (or equilibrium) point for our dynamical systems if $\dot{\boldsymbol{x}} = \boldsymbol{0}$ in the continuous-time case and $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}$ in the discrete-time one. Moreover a stationary point is *asymptotically stable* if any trajectory starting in its vicinity will converge to it as $t \to \infty$. It turns out that both the continuous-time and discrete-time RD have the same set of stationary points in Δ satisfying the condition:

$$x_i((A\boldsymbol{x})_i - \boldsymbol{x}'A\boldsymbol{x}) = 0 \quad \text{with} \quad i = 1\ldots n.$$

We are now interested in studying the dynamical properties of the RD that will allow us to employ them for the partial clique enumeration problem, in particular we will see its relation with *ESS*s.

First, we have that the set of stationary points of the RD is a superset of the set of Nash equilibria related to the payoff function $\boldsymbol{x}'A\boldsymbol{y}$, and if we start the dynamics from the interior of the simplex and it converges to a stationary point $\boldsymbol{x}$, then $\boldsymbol{x}$ is a Nash equilibrium. We also know that by definition the *ESS*s are Nash equilibria, therefore the dynamics can potentially converge towards evolutionary stable states.

Further, from evolutionary game theory we know that asymptotically stable points of the RD are Nash equilibria, and that all the evolutionary stable ones are also asymptotically stable under the continuous-time RD, however, in general, there could still exist asymptotically stable points that are not evolutionary stable.

In the particular case of symmetric payoff matrices we get stronger results. In fact, if we consider a symmetric matrix A, then the quadratic polynomial $\boldsymbol{x}'A\boldsymbol{x}$ is strictly increasing along any non-constant trajectory of both continuous-time and discrete-time replicator equations. Furthermore, any such trajectory converges to a

(unique) stationary point. Finally, a vector $\boldsymbol{x} \in \Delta$ is asymptotically stable under the RD if and only if $\boldsymbol{x}$ is a strict local maximizer of F on Δ and therefore if and only if $\boldsymbol{x}$ is an *ESS* for the payoff function u. In other words, this proposition states that $\boldsymbol{x}'A\boldsymbol{x}$ is a Lyapunov function for the RD. This result is also known in mathematical biology as the Fundamental Theorem of Natural Selection [24,53]. For this property, in [39] the RD were employed as a maximizer for Eq. (2.1).

5. Enumerating maximal cliques

In this section, we present our game–theoretic enumeration approach and study its properties. In order to render a given ESS $\boldsymbol{x}$ unstable, it is sufficient to drop the Nash condition for $\boldsymbol{x}$. A simple way to do it without affecting other equilibria, is to add a new strategy $\boldsymbol{z}$ that is a best reply to $\boldsymbol{x}$, but to no other ESS. This way, $\boldsymbol{x}$ will be no longer asymptotically stable.

Let $G = (V, E)$ be an undirected graph and $G' = (V, E')$ be its directed version, where for all $(u, v) \in E$, $(u, v), (v, u) \in E'$. Given a set Σ of maximal cliques of G, we define the Σ-extension G^Σ of G by adding new nodes to G' as follows. For each clique $S \in \Sigma$, we create a new vertex v, called Σ-vertex, and put edges from v to each vertex in S and from each vertex in $V \setminus S$ to v. After this, each Σ-vertex v dominates a particular clique S of Σ. Further, each vertex not in S dominates the Σ-vertex v so that it cannot form a new asymptotically stable strategy.

Fig. 2 illustrates an undirected graph and its Σ-extension, where Σ contains the maximal cliques $\{A, B, C\}$ and $\{A, D, E\}$. Nodes 1 and 2 are the Σ-vertices. Here, node 1 dominates the clique $\{A, B, C\}$, and node 2 dominates the clique $\{A, D, E\}$.

Theorem 2 guarantees that, given a set of maximal cliques Σ of an undirected graph G, there exists a one-to-one correspondence between the set of maximal cliques of G not in Σ and the set of *ESS*s of a two-player symmetric game associated to the extended graph G^Σ.

Theorem 2. *Let $G = (V, E)$ be an undirected graph, Σ be a set of maximal cliques of G and A be the adjacency matrix of the Σ-extension G_Σ of G. Let Φ be a two-person symmetric game with payoff matrix $A + \alpha I$ with $0.5 \leqslant \alpha < 1$. Then $\boldsymbol{x}$ is an ESS equilibrium of Φ if and only if it is the characteristic vector of a maximal clique of G not in Σ.*

Proof. ($\Rightarrow$) By Theorem 1, if $\boldsymbol{x}$ is an *ESS* of Φ then it is the characteristic vector of a saturated doubly-linked clique S of G_Σ. By construction of G_Σ, the only possible doubly-linked cliques are subsets of V, therefore S is a clique of G. It is also maximal and not in Σ because otherwise it would not be saturated.

($\Leftarrow$) Consider $S \notin \Sigma$ a maximal clique of G. Then by construction of G^Σ, it is a saturated doubly-linked clique of G^Σ and hence, by Theorem 1, $\boldsymbol{x}^S$ is an *ESS* equilibrium of $\mathcal{G}$. $\square$
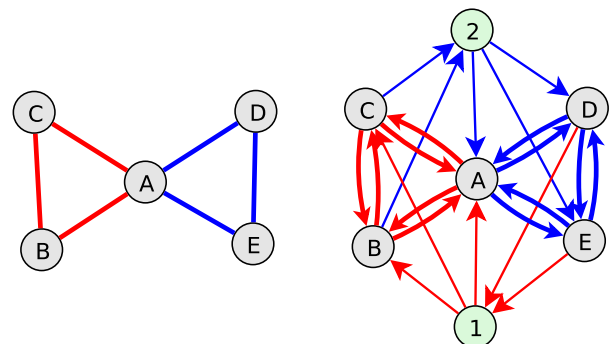

Fig. 2. On the left we have an undirected graph G. On the right we have the Σ-extension G^Σ, where $\Sigma = \{\{A, B, C\}, \{A, D, E\}\}$.
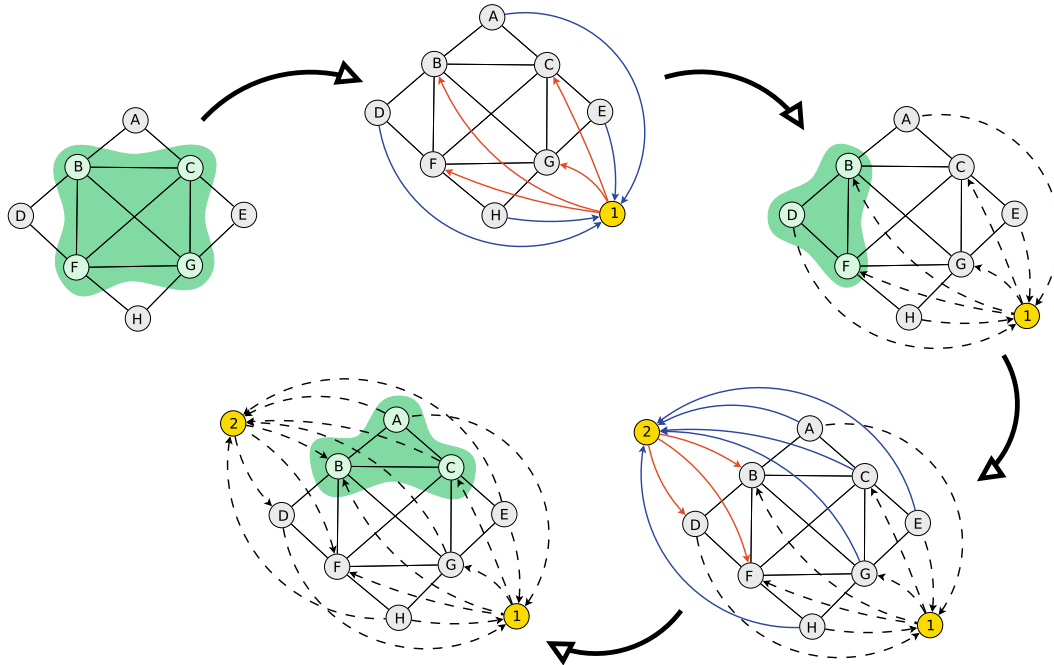
Fig. 3. Example of enumeration with our continuous-based approach.

Our enumerative algorithm uses this result in the following way. We iteratively find an asymptotically stable point through the replicator dynamics. If we have an *ESS*, then we have found a new maximal clique.[2] After that, we extend the graph by adding the newly extracted clique to Σ, thus rendering its associated strategy unstable, and reiterate the procedure until we have enumerated the selected number of maximal cliques. Theorem 2 guarantees that our algorithm is *correct*, i.e., each *ESS* corresponds to a maximal clique that has not already been enumerated, and *complete*, i.e., each remaining maximal clique still corresponds to an *ESS*.

For example, in Fig. 3 we start with an undirected graph and extract the maximal clique $\{B, C, F, G\}$ with plain replicator dynamics. Then we Σ-extend it in order to render the solution found unstable and reapply the dynamics on the extended graph. We converge to a new maximal clique $\{B, D, F\}$. Again we perform a Σ-extension and reapply the dynamics finding $\{A, B, C\}$. Iterating this procedure we can enumerate the desired number of maximal cliques.

The space complexity of this algorithm is $O\{(n + k)^2\}$, where n is the graph order and k is the number of enumerated cliques, while the time complexity is $O\{\gamma k(n + k)^2\}$, where γ is the average number of iterations that the replicator dynamics require to converge (in the experiments we present in the next section we have that $\gamma < 15$).

It is worth pointing out that in order to extract the maximal clique from a characteristic vector, we avoid the standard thresholding technique on the value of each component of the characteristic vector, but rather we use the values of each component as indicators for a New-Best-In heuristic [31]. This is a sequential greedy heuristic that, starting from an empty set of vertices, iteratively constructs a maximal clique by inserting the clique-preserving vertex v that maximizes $w_v + \sum_{j \in S} a_{vj} w_j$ where A is the adjacency matrix of the graph, S is the set of clique-preserving vertices and $\mathbf{w} = (w_1, \ldots, w_n)$ is a weight vector, in our case the mixed strategy obtained through the replicator dynamics. An additional advantage of this approach is that we can stop the dynamics before conver-

gence. This can significantly improve the speed of our approach as a lower number of iterations are needed to extract each clique.

Fig. 4 plots the size of the cliques obtained by iterating our enumerative heuristic 450 times on a random graph of order 100 and density 0.25. For each enumeration the graph plots the average size of the last 40 cliques so as to reduce the variance an making the trend more evident. As it can be seen, the approach enumerates the cliques in approximately decreasing order of size.

6. Experimental results

In order to asses the viability of our game–theoretic Enumeration Heuristic (EH), we conducted two classes of experiments. In the first, we applied the heuristic to the search of a clique of maximum size, using both random graphs with varying order and edge density and the DIMACS benchmark.[3] In the last set of experiments, we applied our approach to the recognition of shapes abstracted in terms of shock trees.

6.1. Experiments on random graphs

In the first set of experiments we applied our approach to random graphs of order 100, 200, 300, 400, 500, 1000, and with edge densities ranging from 0.25 to 0.95. We generated 20 different graphs for each order and density. We run EH enumerating 150 cliques for each graph. The results are compared with the following heuristics based on dynamical systems for maximum clique: Jagota's Continuous Hopfield Dynamics (CHD) and Mean Field Annealing (MFA) [26,27], the Saturated Linear Dynamical Network (SLDN) by Pekergin et al. [38], an approximation approach introduced by Funabiki et al. (FTL) [19], the Iterative Hopfield Nets (IHN) algorithm by Bertoni et al. [6], and the Hopfield Network Learning (HNL) of Wang et al. [52]. Moreover, we also compare with other Motzkin–Straus-based heuristics, namely the Replicator Dynamics (RD) [39] and the Annealed Imitation Heuristic (AIH) [42]. It is

---

[2] We have never experienced an asymptotically stable point that was not an *ESS*, so we conjecture that Theorem (2) can be generalized to asymptotically stable points.

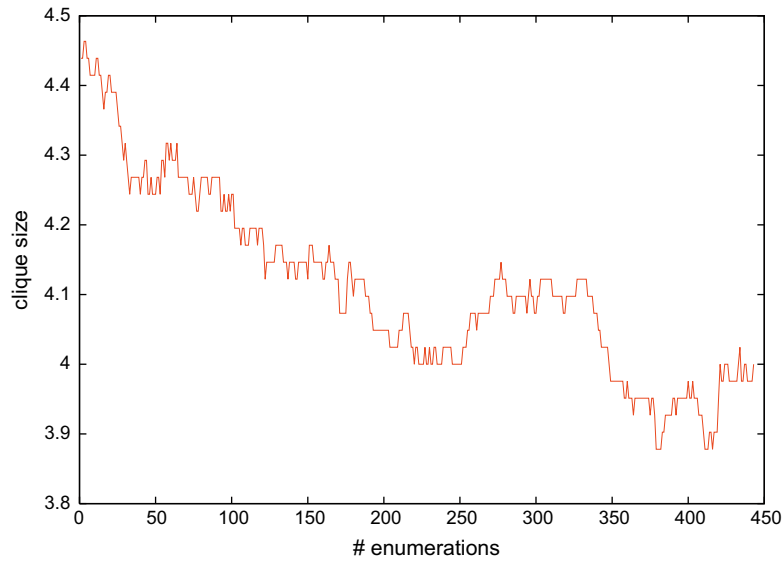[3] Data can be found at http://dimacs.rutgers.edu.

Fig. 4. Average size of the extracted clique over the number of extractions.

worth noting that we did not reimplemented the competing heuristics, but we only compared against the results published in the respective papers.

Table 1 shows the obtained clique sizes and computation times. Here n refers to the graph order and ρ is the edge density. Obviously by construction EH may only improve the results obtained by the plain replicator dynamics, and it is interesting to note that the performance differences are remarkable. The comparisons against the neural network heuristics are not extensive as the re-

sults were reported only on few instances. However, it is clear that EH outperforms CHD, SLDN, FTL, and in most cases also MFA and HNL, while it performs as well as IHN. The comparison against AIH is particularly interesting, as it employs a similar idea of rendering solutions unstable under the replicator dynamics. We can see that EH performs better on all instances with densities smaller that 0.9, while instances having density of 0.9, the two approaches behave comparably well, and on very high densities (0.95) AIH outperforms EH.

Table 1
Experiments on ρ-random graphs.

n	ρ	EH		RD	AIH	CHD	MFA	SLDN	FTL	IHN	HNL
		Size	Time (s)								
100	0.25	5.60 ± 0.49	0.79	4.90 ± 0.56	5.22 ± 0.46	4.48	–	4.83	4.2	–	–
	0.50	9.20 ± 0.51	0.51	8.07 ± 0.64	8.84 ± 0.63	7.38	8.50	8.07	8.0	9.13	9
	0.75	16.95 ± 0.74	0.40	15.15 ± 1.11	16.43 ± 0.83	13.87	–	15.05	14.1	–	–
	0.90	30.60 ± 1.43	0.47	28.92 ± 1.74	30.20 ± 1.52	27.92	30.02	–	–	–	30
	0.95	43.25 ± 1.97	0.54	42.04 ± 1.78	42.94 ± 1.75	–	–	–	–	–	–
200	0.25	6.40 ± 0.49	1.11	5.35 ± 0.64	5.87 ± 0.51	–	–	–	4.9	–	–
	0.50	10.70 ± 0.56	1.00	9.11 ± 0.77	10.14 ± 0.55	–	–	–	8.5	10.60	11
	0.75	20.50 ± 0.74	0.95	18.05 ± 1.23	19.96 ± 1.00	–	–	–	–	–	–
	0.90	39.70 ± 1.23	1.16	37.41 ± 1.55	39.77 ± 1.50	–	–	–	–	–	39
	0.95	60.45 ± 2.25	1.54	58.24 ± 2.09	60.60 ± 1.64	–	–	–	–	–	–
300	0.25	6.90 ± 0.30	2.06	5.61 ± 0.62	6.30 ± 0.46	–	–	–	5.1	–	–
	0.50	11.60 ± 0.49	1.75	9.57 ± 0.88	10.89 ± 0.60	–	–	–	8.9	11.60	11
	0.75	22.35 ± 0.48	1.75	19.40 ± 1.17	21.78 ± 0.84	–	–	–	–	–	–
	0.90	45.50 ± 1.36	2.27	41.48 ± 1.81	45.33 ± 1.57	–	–	–	–	–	46
	0.95	70.95 ± 1.88	3.18	67.43 ± 2.47	72.09 ± 1.75	–	–	–	–	–	–
400	0.25	7.10 ± 0.30	2.46	5.73 ± 0.65	6.65 ± 0.50	5.53	–	5.70	4.9	–	–
	0.50	12.25 ± 0.43	2.75	10.07 ± 0.77	11.24 ± 0.73	9.24	10.36	9.91	8.9	12.30	–
	0.75	23.95 ± 0.67	2.80	20.42 ± 1.10	22.82 ± 0.89	18.79	–	20.44	17.7	–	–
	0.90	49.10 ± 1.09	3.70	44.44 ± 1.64	48.68 ± 1.51	43.24	49.94	–	–	–	–
	0.95	78.75 ± 2.02	5.43	74.25 ± 2.30	79.19 ± 2.00	–	–	–	–	–	–
500	0.25	7.15 ± 0.36	3.46	5.81 ± 0.69	6.68 ± 0.55	–	–	–	6.2	–	–
	0.50	12.60 ± 0.58	4.10	10.31 ± 0.91	11.73 ± 0.71	–	–	–	9.4	12.80	12
	0.75	24.90 ± 0.62	4.18	21.31 ± 1.27	23.91 ± 0.85	–	–	–	–	–	–
	0.90	52.20 ± 0.75	5.41	46.93 ± 2.29	52.26 ± 1.46	–	–	–	–	–	56
	0.95	85.15 ± 1.31	8.32	79.94 ± 2.68	86.50 ± 2.51	–	–	–	–	–	–
1000	0.25	8.00 ± 0.00	10.44	6.23 ± 0.62	7.17 ± 0.45	6.03	–	6.17	5.8	–	–
	0.50	13.95 ± 0.38	14.34	10.83 ± 0.82	12.73 ± 0.78	10.25	–	10.93	10.4	–	–
	0.75	28.00 ± 0.63	14.54	23.04 ± 1.35	26.63 ± 1.03	21.26	–	23.19	21.4	–	–
	0.90	60.65 ± 1.06	19.72	53.15 ± 2.11	60.46 ± 1.60	–	–	–	–	–	–
	0.95	103.80 ± 1.89	32.00	94.80 ± 2.96	104.93 ± 3.47	–	–	–	–	–	–

Fig. 5 shows for each graph density a plot of the average clique size at varying graph orders. We plotted only the results obtained with EH, RD and AIH, as the other competing methods lack experimental results on most instances. Here, we clearly see that RD performs poorly against AIH and EH. We also note how the comparison between AIH and EH critically depends on the density rather than the order of the graph: EH is superior for densities lower than 0.9, while AIH performs better with higher densities.

6.2. Experiments on the DIMACS benchmark graphs

In the next set of experiments we tested our approach on the DIMACS benchmark graphs for the maximum clique problem.

Table 2 shows the results obtained by our approach, IHN, AIH and other Motzkin–Straus-based heuristics for MCP, namely Continuous Based Heuristic (CBH) [20] and Qualex Motzkin–Straus (QMS) [16]. The table reports for each DIMACS graph, the name

(Name), the number of vertices (#), the graph density (ρ) and the clique number (ω). The results of EH are reported in terms of the minimum (Min), the average (Avg) and the maximum (Max) clique size obtained by running the algorithm 20 times and enumerating 150 cliques for each execution. The average clique size allows us to evaluate EH on enumerations of a limited number of cliques (150), while the maximum clique size focuses on the results obtained restarting the algorithm 20 times. The column labeled (K) provides the number of enumerations expected to achieve the maximum. The running times are referred to a non-optimized Java implementation on a 64-bit PC equipped with a 2-GHz AMD Opteron Processor and 1 GB RAM. We do not report the computation times of the competing methods because they refer to different hardware and software settings, rendering the speed performances not comparable.

The *c-fat* and *johnson* families were the easiest to solve, in fact all algorithms attain the global optima. The *hamming* family is also

(a) 0.25-random graphs.

(b) 0.50-random graphs.

(c) 0.75-random graphs.

(d) 0 90-random graphs .
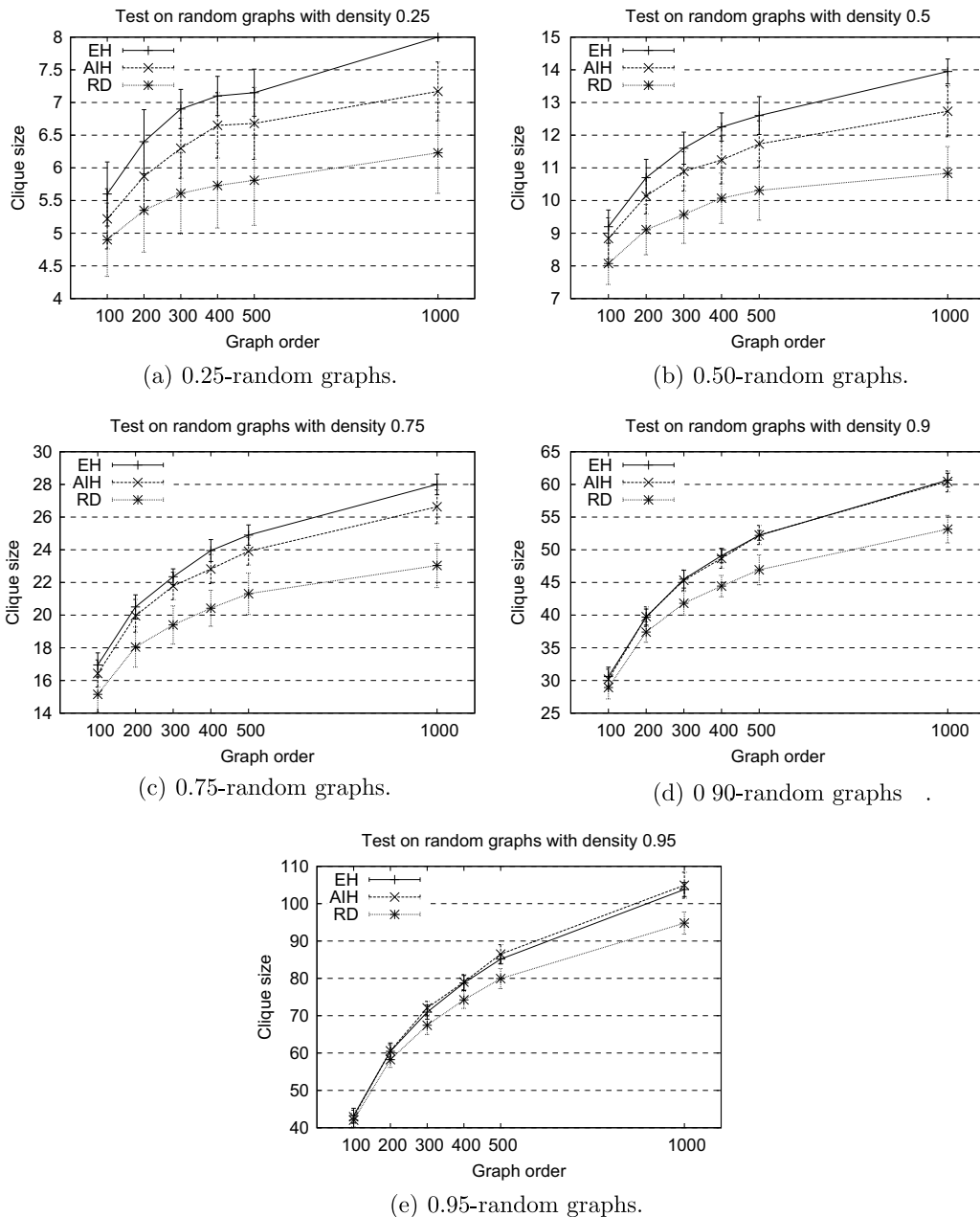

(e) 0.95-random graphs.

Fig. 5. Plots of the average size of the cliques found by our algorithm (EH), plain Replicator Dynamics (RD) and Annealed Imitation Heuristic (AIH).

Table 2
Comparative results on DIMACS benchmark graphs.

Name	#	ρ	ω	EH			K	Avg. time (s)	IHN	AIH	CBH	QMS
				Min	Avg. (Std. dev.)	Max						
brock200_1	200	0.75	21	20	20.050 (0.218)	21	25	1.04	–	20	20	21
brock200_2	200	0.50	12	10	10.600 (0.490)	11	3	0.99	–	10	12	12
brock200_3	200	0.61	15	13	13.600 (0.583)	15	19	0.95	–	13	14	15
brock200_4	200	0.66	17	15	15.600 (0.583)	17	9	0.97	–	16	16	17
brock400_1	400	0.75	27	23	23.400 (0.490)	24	10	2.87	–	24	23	27
brock400_2	400	0.75	29	23	23.500 (0.500)	24	5	2.83	–	24	24	29
brock400_3	400	0.75	31	23	23.400 (0.583)	25	3	2.89	–	24	23	31
brock400_4	400	0.75	33	23	23.850 (0.477)	25	3	2.83	–	23	24	33
brock800_1	800	0.65	23	19	19.500 (0.592)	21	17	9.93	–	20	20	23
brock800_2	800	0.65	24	19	19.500 (0.592)	21	4	9.95	–	18	19	24
brock800_3	800	0.65	25	19	19.500 (0.671)	21	1	9.95	–	19	20	25
brock800_4	800	0.65	26	19	19.400 (0.490)	20	8	9.93	–	19	19	26
c-fat200-1	200	0.08	12	12	12 (0)	12	1	1.34	12	12	12	12
c-fat200-2	200	0.16	24	24	24 (0)	24	1	0.64s	24	24	24	24
c-fat200-5	200	0.43	58	58	58 (0)	58	1	0.37	58	58	58	58
c-fat500-1	500	0.04	14	14	14 (0)	14	1	12.43	14	14	14	14
c-fat500-2	500	0.07	26	26	26 (0)	26	1	3.94	26	26	26	26
c-fat500-5	500	0.19	64	64	64 (0)	64	1	1.66	64	64	64	64
c-fat500-10	500	0.37	126	126	126 (0)	126	1	1.26	–	126	126	126
hamming6-2	64	0.90	32	32	32.0 (0.00)	32	1	0.51	32	32	32	32
hamming6-4	64	0.35	4	4	4.0 (0.00)	4	1	0.53	4	4	4	4
hamming8-2	256	0.97	128	128	128.0 (0.00)	128	1	1.30	128	128	128	128
hamming8-4	256	0.64	16	16	16.0 (0.00)	16	1	1.51	16	16	16	16
hamming10-2	1024	0.49	512	512	512.0 (0.00)	512	1	22.56	512	512	–	512
hamming10-4	1024	0.41	40	36	38.1 (1.92)	40	11	13.82	36	33	–	36
johnson8-2-4	28	0.56	4	4	4 (0)	4	1	0.87	4	4	4	4
johnson8-4-4	70	0.77	14	14	14 (0)	14	1	0.97	14	14	14	14
johnson16-2-4	120	0.76	8	8	8 (0)	8	1	0.45	8	8	8	8
johnson32-2-4	496	0.88	16	16	16 (0)	16	1	3.07	16	16	16	16
keller4	171	0.65	11	11	11 (0)	11	2	0.64	–	9	10	11
keller5	776	0.75	27	25	26.150 (0.910)	27	9	7.24	–	16	21	26
keller6	3361	0.82	⩾59	50	51.750 (1.374)	56	41	131.59	–	31	–	53
MANN_a9	45	0.927	16	16	16 (0)	16	1	1.87	–	16	16	16
MANN_a27	378	0.990	126	125	125.150 (0.357)	126	1	7.23	–	117	121	125
MANN_a45	1035	0.996	345	341	341.700 (0.557)	343	1	127.99	–	–	–	342
p_hat300-1	300	0.24	8	8	8 (0)	8	1	2.28	8	8	8	8
p_hat300-2	300	0.49	25	24	24.900 (0.300)	25	2	2.02	25	25	25	25
p_hat300-3	300	0.74	36	33	34.150 (0.477)	35	4	2.17	36	36	36	35
p_hat500-1	500	0.25	9	9	9 (0)	9	1	4.34	9	9	9	9
p_hat500-2	500	0.50	36	35	35.150 (0.357)	36	1	4.99	36	36	35	36
p_hat500-3	500	0.75	50	48	48.650 (0.477)	49	1	5.70	49	49	49	48
p_hat700-1	700	0.25	11	10	10.700 (0.458)	11	10	6.30	11	9	11	11
p_hat700-2	700	0.50	44	42	43.300 (0.557)	44	1	9.58	44	44	44	44
p_hat700-3	700	0.75	62	60	60.700 (0.557)	62	1	11.40	61	60	60	62
p_hat1000-1	1000	0.25	10	10	10 (0)	10	2	12.32	10	–	–	10
p_hat1000-2	1000	0.50	46	44	45.150 (0.572)	46	1	18.01	46	–	–	45
p_hat1000-3	1000	0.75	68	63	63.900 (0.768)	65	1	22.65	68	–	–	65
p_hat1500-1	1500	0.25	12	11	11 (0)	11	5	22.95	–	10	11	12
p_hat1500-2	1500	0.50	65	63	63.450 (0.497)	64	1	43.72	–	64	63	64
p_hat1500-3	1500	0.75	94	89	90.200 (1.030)	92	1	54.443	–	92	94	91
san200_0.7_1	200	0.70	30	30	30 (0)	30	2	1.07	30	15	15	30
san200_0.7_2	200	0.70	18	13	13.1 (0.300)	14	1	0.96	15	12	12	18
san200_0.9_1	200	0.90	70	70	70 (0)	70	3	1.56	70	46	46	70
san200_0.9_2	200	0.90	60	56	59.250 (1.401)	60	2	1.25	41	39	36	60
san200_0.9_3	200	0.90	44	35	38.050 (2.617)	44	28	1.17	–	35	30	40
san400_0.5_1	400	0.50	13	7	7.750 (0.433)	8	1	2.70	–	7	8	13
san400_0.7_1	400	0.70	40	40	40 (0)	40	3	3.66	40	20	20	40
san400_0.7_2	400	0.70	30	18	23.300 (4.797)	30	27	3.07	30	15	15	30
san400_0.7_3	400	0.70	22	16	16.450 (0.590)	18	1	2.95	–	12	14	18
san400_0.9_1	400	0.90	100	93	99.650 (1.526)	100	4	4.91	100	51	50	100
san1000	1000	0.50	10	8	8.300 (0.458)	9	1	14.82	10	8	8	15
sanr200_0.7	200	0.70	18	17	17.650 (0.477)	18	12	1.03	17	18	18	18
sanr200_0.9	200	0.90	42	39	40.250 (0.698)	42	27	1.18	41	41	41	41
sanr400_0.5	400	0.50	13	12	12.550 (0.497)	13	1	2.84	12	13	12	13
sanr400_0.7	400	0.70	21	20	20.100 (0.300)	21	3	2.84	21	21	20	20

relatively easy for all approaches, except for the instance *hamming 10-4*, where EH is the only approach that achieves the optimum considering the maximum clique size obtained over the 20 runs, and is even the best one if regard the average clique size.

If we limit EH to enumerations of 150 cliques, QMS turns out to be, in general, superior. Restricting our attention to the other competing approaches, EH is the best performer on the *hamming*, *keller* and *MANN* families, while on the *brock* and *sanr* instances all ap-

proaches perform approximately the same. CBH and IHN perform better against EH and AIH with the *p-hat* family. Finally, on the *san* instances IHN turns out to be the best approach followed by EH, while the remaining approaches perform poorly.

If we allow EH to restart 20 times, which is approximately similar to having visited 3000 cliques, we are able to considerably improve the obtained results. In fact, it turns out that our approach outperforms most of the competing ones on all graph families. EH and IHN are comparable on all families except *sanr*, where EH performes better. QMS seems to be particularly good on the *brock* family, where it outperforms all other approaches. However, it is worth noting that the brock family is specifically designed to foil clique-finding heuristics [14] and does not derive from a real application like the other families in the benchmark. Further, EH outperforms QMS on *MANN*, *keller* and *sanr* families, while the two approaches are comparable on *p_hat* and *san* families, EH performing slightly better on the former and QMS on the latter.

In the following section, we will see how our approach may be useful to solve the computer vision task of recognizing shapes from tree representations.

6.3. Matching shock trees for visual shape recognition

In the final set of experiments our enumeration approach was applied to a shape recognition task based on matching shock trees, a graph abstraction of the morphological skeleton of a shape [30]. This is based on the *shocks* (or singularities) of a curve evolution process, acting on a simple closed curve in the plane.

The morphological skeleton, or medial axis, of a shape is defined as the loci of the centers of the maximal circles inscribed in the shape, i.e., the set of points that are equidistant from opposite boundaries of the object. The skeleton is segmented into branches along which the radius of the maximal circle is either monotonically increasing or decreasing. Each branch segment is then abstracted as a vertex of a graph and vertices are connected by an edge if the associated branches are adjacent. It is straightforward to show that this graph can be turned into a tree [30]. Fig. 6 shows an example of a shape with its skeleton superimposed, and the resulting shock-tree. Numeric labels correspond to the labels on the skeletal branches, while the dummy root is indicated by the symbol #.

The problem of matching rooted shock trees can be cast into a clique problem using a result due to Pelillo et al. [43], who have shown that any maximal/maximum subtree isomorphism is in one-to-one correspondence with a maximal/maximum clique in an auxiliary structure called *tree association graph* (TAG). This allows us to treat the subtree isomorphism between two trees as a clique problem on their TAG. Note that this approach is robust to external structural noise, i.e., noise that does not influence the sub-

tree isomorphism, but it is not as robust with respect to internal noise. However, the experimental evidence shown in [43] suggests that the approach is quite robust when applied to shock-graph matching problems. A common alternative formalization of the tree-matching problem is tree-edit distance, which, by making the deformation explicit, has the potential of being robust against any form of structural deformation. We refer to [49], for a way to cast tree-edit distance into a series of clique problems, which can, hence, be adapted to use our enumeration approach.

Using the TAG we can apply a maximum clique heuristic to extract the best subgraph isomorphism in order to match two shock trees. However, the solution of maximum cardinality is not always the best. Indeed, a similar effect has already been experienced by Horaud and Skordas [25] in the context of stereo correspondences. The results can be improved by enumerating k maximal cliques and then choosing the best one according to a cost function that could not be directly expressed in terms of the maximum weight of a clique. Therefore, we decided to apply our enumerative approach to extract k subtree isomorphisms and then ranked the solutions according to a cost function that allows us to capture global properties of the isomorphism.

The cost function we used is a linear combination of three terms, incorporating the differences in lengths, radii and matching sizes. Each term is normalized to provide a unitless quantity, so that these different geometric properties can be combined. For each shock tree node u we are given a vector of attributes $\alpha(u) = (x_1, y_1, r_1; \ldots; x_m, y_m, r_m)$ where m is the number of shocks in the group, and x_i, y_i, r_i are, respectively, the coordinates and the radius (or time of formation) of each shock i in the sequence, obtained as output of the shock detection process [46]. Let u contain m shocks and v contain n shocks, and without loss of generality, assume that $m \geqslant n$. The Euclidean length of each sequence of shocks is given by:

$$L(u) = \sum_{j=1}^{m-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

Let $\xi(i) : i \to \lceil \frac{ni}{m} \rceil$ be the many-to-one mapping of each index $i \in \{1, \ldots, m\}$ to an index $j \in \{1, \ldots, n\}$ and β_l, β_r and β_s three non-negative constants that sum up to one. Assume we are given two trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ to match and we found a subtree isomorphism, $\varphi : H_1 \to H_2$, where $H_1 \subseteq V_1$ and $H_2 \subseteq V_2$.

The cost function between two nodes $u \in H_1$ and $v \in H_2$ is given by

$$\sigma(u,v) = \beta_l \frac{|L(u) - L(v)|}{\max[L(u), L(v)]} + \beta_r$$

$$\times \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(\frac{r_u(i) - r_v(\xi(i))}{\max[r_u(i), r_v(\xi(i))]} \right)} \quad (6.1)$$

where $r_u(i)$ is the ith radius in $\alpha(u)$. The overall cost of a subtree isomorphism is then given by

$$S(\varphi) = \frac{1}{|H_1|} \sum_{v \in H_1} \sigma(v, \varphi(v)) + \beta_s \left(1 - \frac{2|H_1|}{|V_1| + |V_2|} \right) \quad (6.2)$$

We performed the experiments on the same database used in [43] consisting of 25 silhouettes representing eight different object classes (Table 3 first column). We matched each shape against all the database enumerating 8 cliques and used (6.2) as our cost function. Note that in Eq. (6.1), we took into account the length difference of the two shock groups in the first term and the difference in the radius distribution in the second term. This cost information can easily be recoded as a vertex weight in the association graph and as proposed in [43] the attributed subtree isomorphism can be cast as a maximum weight clique problem. However, one draw-
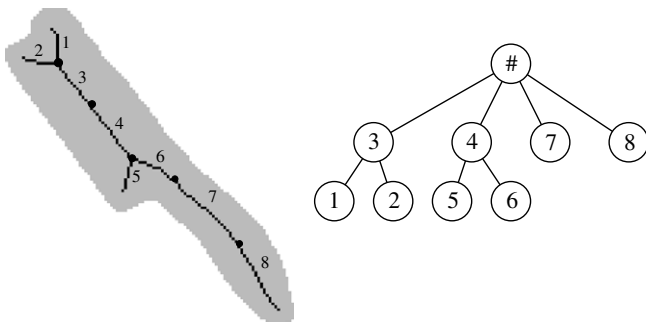

Fig. 6. Example of a silhouette with its skeleton superimposed, and the resulting shock-tree.

Table 3
A tabulation of the top eight topological matches for each query. The scores indicate the cost calculated for each match according to the Function (6.2) and the integer between square brackets indicates the position of extraction of the best match during the clique enumeration.

Query Shape	Top 8 Topological Matches							
	1	2	3	4	5	6	7	8
	.000 [1]	.071 [7]	.190 [7]	.203 [5]	.218 [5]	.232 [6]	.234 [1]	.251 [5]
	.000 [1]	.071 [1]	.198 [1]	.204 [5]	.204 [1]	.240 [1]	.251 [1]	.264 [1]
	.000 [5]	.190 [6]	.198 [1]	.211 [8]	.223 [4]	.248 [1]	.259 [6]	.260 [5]
	.000 [7]	.066 [8]	.137 [1]	.190 [2]	.252 [4]	.306 [8]	.316 [6]	.335 [8]
	.000 [1]	.108 [3]	.135 [1]	.159 [1]	.182 [7]	.255 [1]	.264 [1]	.283 [1]
	.000 [6]	.058 [7]	.108 [1]	.156 [6]	.217 [7]	.294 [3]	.313 [1]	.322 [6]
	.000 [1]	.013 [1]	.072 [2]	.075 [3]	.082 [7]	.126 [8]	.146 [1]	.253 [2]
	.000 [1]	.098 [4]	.108 [1]	.126 [4]	.127 [7]	.167 [4]	.265 [5]	.290 [6]
	.000 [6]	.062 [2]	.075 [8]	.091 [5]	.098 [4]	.116 [3]	.223 [2]	.258 [1]
	.000 [1]	.100 [1]	.116 [6]	.130 [6]	.146 [8]	.167 [7]	.197 [6]	.204 [5]
	.000 [7]	.062 [3]	.076 [3]	.082 [3]	.100 [2]	.106 [3]	.247 [2]	.248 [1]
	.000 [1]	.073 [1]	.076 [8]	.091 [8]	.126 [4]	.126 [5]	.259 [1]	.264 [1]
	.000 [8]	.044 [1]	.232 [5]	.242 [5]	.249 [1]	.324 [5]	.328 [1]	.339 [3]
	.000 [1]	.044 [1]	.233 [8]	.248 [5]	.252 [1]	.329 [7]	.338 [7]	.341 [2]
	.000 [6]	.248 [1]	.249 [1]	.272 [5]	.274 [5]	.283 [6]	.291 [1]	.293 [4]
	.000 [7]	.122 [3]	.169 [1]	.217 [4]	.245 [4]	.264 [1]	.321 [4]	.342 [1]
	.000 [1]	.033 [1]	.043 [1]	.259 [8]	.281 [1]	.300 [7]	.303 [5]	.309 [7]
	.000 [8]	.035 [7]	.066 [8]	.272 [1]	.292 [8]	.304 [6]	.318 [1]	.327 [4]
	.000 [8]	.042 [5]	.070 [1]	.286 [1]	.295 [1]	.317 [1]	.328 [1]	.333 [8]
	.000 [6]	.186 [6]	.197 [1]	.203 [1]	.218 [6]	.259 [8]	.272 [7]	.286 [7]
	.000 [7]	.035 [4]	.186 [8]	.272 [8]	.333 [4]	.338 [8]	.344 [5]	.346 [5]
	.000 [7]	.035 [1]	.197 [4]	.273 [1]	.337 [4]	.340 [1]	.344 [1]	.353 [1]
	.000 [1]	.334 [7]	.405 [6]	.409 [2]	.410 [5]	.420 [7]	.432 [1]	.443 [3]
	.000 [1]	.327 [5]	.380 [5]	.391 [5]	.395 [1]	.403 [5]	.416 [1]	.419 [6]

back of this cost function is that it does not penalize matches, where some nodes are missing. Therefore, in Eq. (6.2), we introduce knowledge about the size of the match found. This information clearly cannot be encoded into a vertex weight, but must be considered in a post-processing step on the matches found. This fact justifies our approach of enumerating several cliques and then evaluating the cost function to select the best solution. More in general, with a weighted matching approach such as the one presented in [43] we are limited to quality measures that are additive with respect to the matched nodes, while our enumerative approach allows us to use a wider range of quality measures.

Table 3 shows for each silhouette the top eight matches ranked according to (6.2). Each entry in the table comes also with the extraction order of the best match, indicated in square brackets. Note that all the shapes falling in the same class as the query shape are typically ranked in the first positions. Moreover, we can also notice that most of the time the best match does not derive from the first extracted maximal clique, confirming our expectations. The results compare favorably with those obtained in [43] using a maximum weight clique formulation on the TAG and in [46] using a sequential (level by level) matching approach with backtracking. A clear improvement of our approach can be seen in the query of the first head shape. Here, we correctly extract the two other head shapes first, while in [43] the two next shapes are hands. Fig. 7 reports the tree representations of the query head shape (left), a hand shape (center) and another head shape (right). The green nodes represent vertices belonging to the best structural match, while the red ones are those left outside the match. The dashed vertices are those that are missing preventing to have a perfect submatch with the query shape. As we can see, the hand shape presents a perfect partial match with the query shape. However, in this example there are much more unmatched nodes than what can be found on the head shape on the right. Without comparing the size of the match with the sizes of the original trees, we are unable to rank the head shape higher than the hand shape. Indeed, this is what happens with the formulation based on the maximum weight clique, which can not use this information.

Note that, both our approach and the one proposed in [43] are based on the same formulation, the increase in performance is due to the use of a more discriminative objective function. Hence, in this context, the real advantage of our enumerative method is that it allows one to use a wider range of quality measures. Of course, this feature characterizes any enumerative method, but our approach is, to our knowledge, the only one that applies a heuristic for the maximum clique problem at each iteration. Hence, unlike standard enumerative algorithms, which extract cliques in lexicographical order, ours is explicitly designed to output large cliques first, thereby rendering it especially attractive in conjunction with quality measures that are correlated with the cardinality of the matches.

7. Conclusions and future work

In this paper, we developed a game–theoretic approach to enumerating maximal cliques, which is based on the Motzkin–Straus formulation. In order to perform the enumeration, we deal with a directed form of the clique problem and apply an asymmetric extension to it. This way we lose the original connection with the quadratic problem, but, by casting the problem into a game–theoretic framework, we are able to prove a relationship between the evolutionary stable strategies and maximal cliques that have not yet been enumerated.

In order to asses the effectiveness of the approach we conducted three sets of experiments. The first two were focused on the maximum clique problem, where we compared our approach with several state-of-the-art algorithms on the DIMACS benchmark graphs and on random graphs of varying order and edge density. In the third experiment we used our enumerative heuristic for shape recognition by matching shock-trees. The tests conducted confirm the validity of our approach.

A key advantage of our enumerative approach is that we are able to apply for each clique extraction a continuous-based heuristic for the maximum clique problem, namely the replicator dynamics, and this allows us to extract large cliques first, which is a feature that is not shared by other enumerative approaches. Clearly, we have no guarantee to extract cliques in order of decreasing size, as this strongly depends on the effectiveness of the replicator dynamics as a maximum clique heuristic. However, works due to Pelillo and Bomze [8,11,39,40] indicate that this is indeed a good heuristic.

From a computational point of view, the complexity of the approach increases linearly after each clique extraction and this slows down the algorithm in the long run. Fortunately, the technique of early extraction described at the end of Section 5, helps to speed up the approach considerably. It is worth noting that the main results obtained in this paper are game–theoretic and do not rely on the tool that we use for finding the equilibria. Therefore, a direction to point to in the future may be to try other dynamics for extracting ESSs. In particular, there exists a large class of evolutionary dynamics called payoff-monotonic [53], to which the replicator dynamics belong, that may be employed for our purposes. In [42], for example, it is shown that the exponential replicator dynamics outperform the standard one as maximum clique heuristic from both a computational and a qualitative point of view.
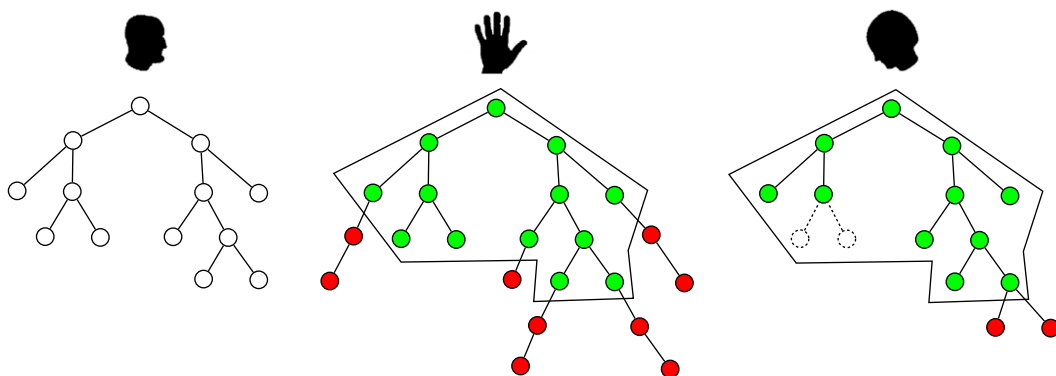

Fig. 7. A comparison of the shock trees of three silhouettes. Here, the head on the left is the query tree. The green vertices belong to the best structural match, while the red ones are those left outside the match. The dashed vertices are those that are missing.

Finally, the game–theoretic approach to enumeration may be generalized to both vertex-weighted and edge-weighted graphs. In the first case, a payoff matrix is derived from the continuous formulation of the maximum weighted clique of Bomze, Pelillo and Stix [11], which is a regularized version of the formulation of Gibbons et al. [20]. By opportunely generalizing Theorems 1 and 2 we may end up with an approach for the enumeration of maximal weighted cliques. In the second case, our approach may be generalized to the enumeration of maximal cliques in edge-weighted graphs, which are also known as dominant sets, a graph-theoretic notion of cluster recently introduced by Pavan and Pelillo [37].

References

[1] A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall, R.J. Popplestone. A versatile computer-controlled assembly, in: Proc. Third Int. J. Conf. Artif. Intell., Standford, CA, 1982, pp. 298–307.
[2] J.G. Augustson, J. Minker, An analysis of some graph theoretical cluster techniques, J. ACM 17 (4) (1970) 571–588.
[3] H.G. Barrow, R.M. Burstall, Subgraph isomorphism, matching relational structures, and maximal cliques, Inform. Process. Lett. 4 (4) (1976) 83–84.
[4] L.E. Baum, 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 (1967) 360–363.
[5] L.E. Baum, G.R. Sell, Growth transformations for functions on manifolds, Pacific J. Math. 27 (1968) 211–227.
[6] A. Bertoni, P. Campadelli, G. Grossi, A neural algorithm for the maximum clique problem: analysis, experiments and circuit implementation, Algorithmica 331 (2002) 71–88.
[7] R.C. Bolles, R.A. Cain, Recognizing and locating partially visible objects: the local-feature-focus method, Int. J. Robot. Res. 1 (3) (1982) 57–82.
[8] I. Bomze, Evolution towards the maximum clique, J. Global Optimiz. 10 (2) (1997) 143–164.
[9] I. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, Handbook Combinat. Optimiz. A (Suppl.) (1999) 1–74.
[10] I. Bomze, M. Pelillo, R. Giacomini, Evolutionary approach to the maximum clique problem: empirical evidence on a larger scale, Dev. Global Optimiz. (1997) 95–108.
[11] I.M. Bomze, M. Pelillo, V. Stix, Approximating the maximum weight clique using replicator dynamics, IEEE Trans. Neural Networks 11 (6) (2000) 1228–1241.
[12] I. Bomze, M. Budunich, M. Pelillo, C. Rossi, Annealed replication: a new heuristic for the maximum clique problem, Discrete Appl. Math. 121 (2002) 27–49.
[13] R.E. Bonner, On some clustering techniques, IBM J. Res. Dev. 8 (1964) 22–32.
[14] M. Brockington, J.C. Culberson, Camouflaging independent sets in quasi-random graphs, in: D. Johnson, M. Trick (Eds.), Cliques Coloring and Satisfiability: Second DIMACS Implementation Challenge, American Mathematical Society, 1996, pp. 75–88.
[15] C. Bron, J. Kerbosch, Algorithm 457: finding all cliques of an undirected graph, Commun. ACM 16 (1973) 575–577.
[16] S. Busygin, A new trust region technique for the maximum weight clique problem, Discrete Appl. Math. 154 (15) (2006) 2080–2096.
[17] R.T. Chin, C.R. Dyer, Model-based recognition in robot vision, Comput. Surv. 18 (1) (1986) 67–108.
[18] D. Dmitry, R. Ari. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words, in: Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL, Association for Computational Linguistics, 2006, pp. 297–304.
[19] N. Funabiki, Y. Takefuji, K.C. Lee, A neural network model for finding a near-maximum clique, J. Parallel Distrib. Comput. 14 (1992) 340–344.
[20] L.E. Gibbons, D.W. Hearn, P.M. Pardalos, M.V. Ramana, Continuous characterizations of the maximum clique problem, Math. Oper. Res. 22 (1997) 754–768.
[21] J. Hammersley, P. Clifford, Markov fields on finite graphs and lattices, unpublished manuscript, 1971.
[22] F. Harary, I.C. Ross, A procedure for clique detection using the group matrix, Sociometry 20 (3) (1957) 205–215.

[23] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$, in: Proc. 37th Ann. Symp. Found. Comput. Sci., 1996, pp. 627–636.
[24] J. Hofbauer, K. Sigmund, The Theory of Evolution and Dynamical Systems, Cambridge University Press, Cambridge, UK, 1988.
[25] R. Horaud, T. Skordas, Stereo correspondence through feature grouping and maximal cliques, IEEE Trans. Pattern Anal. Mach. Intell. 11 (11) (1989) 1168–1180.
[26] A. Jagota, Approximating maximum clique with a Hopfield neural network, IEEE Trans. Neural Networks 6 (1995) 724–735.
[27] A. Jagota, L. Sanchis, R. Ganesan, Approximately solving maximum clique using neural networks and related heuristics, in: D. Johnson and M. Trick (Eds.), Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge, 1996, pp. 169–204.
[28] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[29] D.S. Johnson, M. Yannakakis, C.H. Papadimitriou, On generating all maximal independent sets, Inform. Proc. Lett. 27 (1988) 119–123.
[30] B.B. Kimia, A. Tannenbaum, S.W. Zucker, Shape, shocks, and deformations: the components of two-dimensional shape and the reaction-diffusion space, Int. J. Comp. Vis. 15 (1995) 189–224.
[31] R. Kopf, G. Ruhe, A computational study of the weighted independent set problem for general graphs, Found. Control Eng. 12 (1987) 167–180.
[32] D.G. Luenberger, Linear and Nonlinear Programming, Addison Wesley, Reading, MA, 1984.
[33] T.S. Motzkin, E.G. Straus, Maxima for graphs and a new proof of a theorem of Turán, Can. J. Math. 17 (1965) 533–540.
[34] M. Nina, R. Dana, S. Ram, A new conceptual clustering framework, Mach. Learn. 56 (2004) 115–151.
[35] H. Ogawa, Labeled point pattern matching by Delaunay triangulation and maximal cliques, Pattern Recogn. 19 (1) (1986) 35–40.
[36] Y. Okubo, M. Haraguchi. Creating abstract concepts for classification by finding top-N maximal weighted cliques, in: Proc. Discovery Science, vol. 2843, 2003, pp. 418–425.
[37] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 167–172.
[38] F. Pekergin, Ö Morgül, C. Güzelis, A saturated linear dynamical network for approximating maximum clique, IEEE Trans. Circuits Syst. 46 (6) (1999) 667–685.
[39] M. Pelillo, Relaxation labeling networks for the maximum clique problem, J. Artif. Neural Networks 2 (4) (1995) 313–328.
[40] M. Pelillo, Replicator equations, maximal cliques, and graph isomorphism, Neural Comput. 11 (8) (1999) 2023–2045.
[41] M. Pelillo, A. Jagota, Feasible and infeasible maxima in a quadratic program for maximum clique, J. Artif. Neural Networks 2 (1995) 411–420.
[42] M. Pelillo, A. Torsello, Payoff-monotonic game dynamics and the maximum clique problem, Neural Comput. 18 (5) (2006) 1215–1258.
[43] M. Pelillo, K. Siddiqi, S.W. Zucker, Matching hierarchical structures using association graphs, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1105–1120.
[44] B. Radig, Image sequence analysis using relational structures, Pattern Recogn. 17 (1) (1984) 161–167.
[45] X. Shi, Luo Liang, Y. Wan, J. Xu, A finding maximal clique algorithm for predicting loop of protein structure, Appl. Math. Comput. 180 (2) (2006) 676–682.
[46] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, Int. J. Comp. Vis. 35 (1999) 13–32.
[47] P. Suetens, P. Fua, A.J. Hanson, Computational strategies for object recognition, Comput. Surv. 24 (1) (1992) 5–61.
[48] P. Taylor, L. Jonker, Evolutionary stable strategies and game dynamics, Math. Biosci. 40 (1978) 145–156.
[49] A. Torsello, E.R. Hancock, Computing approximate tree edit distance using relaxation labeling, Pattern Recogn. Lett. 24 (2003) 1089–1097.
[50] A. Torsello, S. Rota Bulò, M. Pelillo, Grouping with asymmetric affinities: a game–theoretic perspective, in: Proc. of the 2006 IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 2006, pp. 292–299.
[51] E. Tomita, A. Tanaka, H. Takahashi, The Worst-case Time Complexity for Finding all the Cliques. Technical Report UEC-TR-C5, 1988.
[52] R.L. Wang, Z. Tang, Q.P. Cao, An efficient approximation algorithm for finding a maximum clique using Hopfield network learning, Neural Comput. 15 (2003) 1605–1619.
[53] J.W. Weibull, Evolutionary Game Theory, MIT Press, Cambridge, MA, 1995.