Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Pattern Recognition Letters 24 (2003) 1099–1106

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Matching graphs by pivoting

Alessio Massaro [a,*], Marcello Pelillo [b,*]

[a] FEI Electron Optics, SEM Software Group, Building HAF13, Postbus 218, 5600MD Eindhoven, Netherlands
[b] Dipartimento di Informatica, Università Ca' Foscari di Venezia, Via Torino 155, 30172 Venezia Mestre, Italy

Abstract

Motivated by a recent (continuous) quadratic formulation, in this paper we present a pivoting-based heuristic for graph matching based on the corresponding linear complementarity problem. Experiments are presented which demonstrate the potential of the proposed method.

Keywords: Graph matching; Linear complementarity problem; Maximum clique; Maximum common subgraph; Quadratic programming; Pivoting

1. Introduction

Graph matching is a fundamental problem in computer vision and pattern recognition, and a great deal of effort has been devoted over the past decades to devise efficient and robust algorithms for it (see Bunke, 2000 for an update on recent developments). Basically, two radically distinct approaches have emerged, a distinction which reflects the well-known dichotomy originated in the artificial intelligence field between "symbolic" and "numeric" methods. The first approach views the matching problem as one of explicit search in state-space (see e.g., Messmer and Bunke, 1998; Shapiro and Haralick, 1981; Tsai and Fu, 1983). The pioneering work of Ambler et al. (1973) falls into this class. Their approach is based on the idea that graph matching is equivalent to the problem of finding maximal cliques in the so-called association graph, an auxiliary graph derived from the structures being matched. This framework is attractive because it casts the matching problem in terms of a pure graph–theoretic problem, for which a solid theory and powerful algorithms have been developed (Bomze et al., 1999).

In the second approach, the relational matching problem is viewed as one of energy minimization. In this case, an energy (or objective) function is sought whose minimizers correspond to the solutions of the original problem, and a dynamical system, usually embedded into a parallel relaxation network, is used to minimize it (Gold and Rangarajan, 1996; Li, 1992; Wilson and Hancock, 1997). Typically, these methods do not solve the problem exactly, but only in approximation terms. Energy minimization algorithms are attractive because they are amenable to parallel hardware implementation and also offer the advantage of biological plausibility.

[*] Corresponding authors.
E-mail address: alessio.massaro@nl.feico.com (A. Massaro).

0167-8655/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(02)00256-8

In a recent paper (Pelillo, 1998), we have developed a new framework for graph matching which does unify the two approaches just described, thereby inheriting the attractive features of both. The approach is centered around a remarkable result in graph theory which allows us to map the maximum clique problem onto the problem of extremizing a quadratic form over a linearly constrained domain (i.e., the standard simplex in Euclidean space). Local gradient-based search methods such as *replicator dynamics* have proven to be remarkably effective when we restrict ourselves to simple versions of the problem, such as tree matching (Pelillo, in press; Pelillo et al., 1999) or graph isomorphism (Pelillo, 1999). However, for more difficult problems the challenge remains to develop powerful heuristics.

It is a well-known fact that stationary points of quadratic programs can be characterized in terms of solutions of a linear complementarity problem (LCP), a class of inequality systems for which a rich theory and a large number of algorithms have been developed (Cottle et al., 1992). Hence, once that the graph matching is formulated in terms of a quadratic programming problem, the use of LCP algorithms naturally suggests itself, and this is precisely the idea proposed in the present paper. Among the many LCP methods presented in the literature, pivoting procedures are widely used and within this class Lemke's method is certainly the best known. Unfortunately, like other pivoting schemes, its finite convergence is guaranteed only for non-degenerate problems, and ours is indeed degenerate. The inherent degeneracy of the problem, however, is beneficial as it leaves freedom in choosing the blocking variable, and we exploited this property to develop a variant of Lemke's algorithm which uses a new and effective "look-ahead" pivot rule. The procedure depends critically on the choice of a vertex in the graph which identifies the driving variable in the pivoting process. Since there is no obvious way to determine such a vertex in an optimal manner, we resorted to iterate this procedure over most, if not all, vertices in the graph. The resulting *pivoting-based heuristic* has been tested on various instances of randomly generated graphs and the preliminary results obtained confirm the effectiveness of the proposed approach.

2. Graph matching and linear complementarity

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, an *isomorphism* between them is any bijection ϕ: $V_1 \to V_2$ such that $(i,j) \in E_1 \iff (\phi(i), \phi(j)) \in E_2$, for all $i,j \in V_1$. Two graphs are said to be isomorphic if there exists an isomorphism between them. The maximum common subgraph problem 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.

The *association graph* derived from $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the undirected graph $G = (V, E)$ defined as follows:

$$V = V_1 \times V_2$$

and

$$E = \{((i,h),(j,k)) \in V \times V : i \neq j, h \neq k, \\ \text{and } (i,j) \in E_1 \iff (h,k) \in E_2\}.$$

Given an arbitrary undirected graph $G = (V, E)$, a subset of vertices C is called a *clique* if all of its vertices are mutually adjacent, i.e., for all $i,j \in C$, with $i \neq j$, we have $(i,j) \in E$. 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.

The following result establishes an equivalence between the graph matching problem and the maximum clique problem (see e.g., Barrow and Burstall, 1976).

Theorem 2.1. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, and let G be the corresponding association graph. Then, all maximal (maximum) cliques in G are in one-to-one correspondence with maximal (maximum) common subgraph isomorphisms between G_1 and G_2.*

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

$$S_n = \{x \in \mathbb{R}^n : e^{\mathrm{T}}x = 1 \text{ and } x_i \geqslant 0, \quad i = 1 \ldots n\},$$

where e is the vector whose components equal 1, and a "T" denotes transposition. Given a subset of vertices C of G, we will denote by 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.

Consider the following quadratic program

$$\begin{aligned} &\text{minimize} && f_G(x) = x^{\mathrm{T}} A_G x, \\ &\text{subject to} && x \in S_n, \end{aligned} \tag{1}$$

where $A_G = (a_{ij})$ is the $n \times n$ symmetric matrix defined as

$$a_{ij} = \begin{cases} \frac{1}{2} & \text{if } i = j, \\ 0 & \text{if } i \neq j \text{ and } (i,j) \in E, \\ 1 & \text{if } i \neq j \text{ and } (i,j) \notin E. \end{cases} \tag{2}$$

The following theorem, recently proved by Bomze (1997), expands on the Motzkin–Straus theorem (Motzkin and Straus, 1965), a remarkable result which establishes a connection between the maximum clique problem and certain standard quadratic programs. This has an intriguing computational significance in that it allows us to shift from the discrete to the continuous domain in an elegant manner.

Theorem 2.2. *Let C be a subset of vertices of a graph G, and let x^c be its characteristic vector. Then, C is a maximal (maximum) clique of G if and only if x^c is a local (global) solution of program* (1). *Moreover, all local (and hence global) solutions of* (1) *are strict and are characteristic vectors of maximal cliques of G.*

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

The following result, which is a straightforward consequence of Theorems 2.1 and 2.2, establishes an elegant connection between graph matching and quadratic programming.

Theorem 2.3. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, and let G be the corresponding association graph. Then, all local (global) solutions to* (1) *are in one-to-one correspondence with maximal (maximum) common subgraph isomorphisms between* G_1 and G_2.

Computing the stationary points of (1) can be done by solving the LCP (q_G, M_G), which is the problem of finding a vector x satisfying the system

$$y = q_G + M_G x \geqslant 0, \quad x \geqslant 0, \quad x^{\mathrm{T}} y = 0, \tag{3}$$

where

$$q_G = [0, \ldots, 0, -1, 1]^{\mathrm{T}} \quad \text{and}$$
$$M_G = \begin{bmatrix} A_G & -e & e \\ e^{\mathrm{T}} & 0 & 0 \\ -e^{\mathrm{T}} & 0 & 0 \end{bmatrix}. \tag{4}$$

With the above definitions, it is well known that if z is a complementary solution of (q_G, M_G) with $z^{\mathrm{T}} = (x^{\mathrm{T}}, y^{\mathrm{T}})$ and $x \in \mathbb{R}^n$, then x is a stationary point of (1). Indeed, the matrix A_G is always strictly copositive, hence so is M_G and that is enough to assure that (q_G, M_G) always has a solution (Cottle et al., 1992). [1] Of course, a stationary point of (1) is not necessarily a local minimum, but in practice this is not a problem since there are several techniques that, starting from a stationary point can reach a nearby local optimum. An example is given by the replicator dynamics, but see Massaro et al., in press for a complete discussion on this topic.

3. A pivoting-based heuristic for graph matching

Technical literature supplies a large number of algorithms to go about solving an LCP (Cottle et al., 1992). The most popular is probably Lemke's method, largely for its ability to provide a solution for a large number of matrix classes. Lemke's "Scheme I" belongs to the family of pivoting algorithms. Given the generic LCP

[1] Recall that, given a cone $\Gamma \subseteq \mathbb{R}^n$, a symmetric matrix Q is said to be *Γ-copositive* if $x^{\mathrm{T}} Q x \geqslant 0$ for all $x \in \Gamma$. If the inequality holds strictly for all $x \in \Gamma \setminus \{0\}$, then Q is said to be *strictly Γ-copositive.*

(q, M), it deals with the augmented problem (q, d, M) defined by

$$y = q + [M, d]\begin{bmatrix} x \\ \theta \end{bmatrix} \geqslant 0, \quad \theta \geqslant 0, \quad x \geqslant 0, \quad x^{\mathrm{T}}y = 0. \tag{5}$$

A solution of (q, d, M) with $\theta = 0$ promptly yields a solution to (q, M), and Lemke's method intends to compute precisely such a solution. We refer to Cottle et al. (1992) for a detailed description of Lemke's algorithm. In our implementation, we chose $d = e$, as our problem does not expose peculiarities that would justify a deviation from this common practice.

As usually done for outlining pivoting algorithms, we will use an exponent for the problem data. In practice q^v and M^v will identify the situation after v pivots and A_G^v will indicate the $n \times n$ leading principal submatrix of M^v. Consistently, y^v and x^v will indicate the vectors of basic and nonbasic variables, respectively, each made up of a combination of the original x_i and y_i variables. The notation $\langle x_i^v, y_j^v \rangle$ will be used to indicate pivoting transformations. The index set of the basic variables that satisfy the min-ratio test at iteration v will be denoted with Ω^v, i.e.

$$\Omega^v = \arg\min_i \left\{ \frac{-q_i^v}{m_{is}^v} : m_{is}^v < 0 \right\},$$

where s is the index of the driving column. Also, in the sequel the auxiliary column that contains the covering vector d in (5) will be referred to as the column $n + 3$ of matrix $M = M_G$.

In general, assuming an LCP non-degenerate is a strategy commonly taken to prove finiteness of pivoting schemes. This assumption amounts to having $|\Omega^v| = 1$ for all v, thereby excluding any cycling behavior. In particular, Lemke's method is guaranteed to process any non-degenerate problem (q, M) where M is strictly $\mathbb{R}_+^n$-copositive, and to do so without terminating on a secondary ray (Cottle et al., 1992). Unfortunately our LCP (q_G, M_G) is degenerate and standard degeneracy resolution strategies have proven to yield unsatisfactory results (Massaro et al., in press).

The proposed degeneracy resolution technique makes use of the so-called *least-index rule* which amounts to blocking the driving variable with a basic one that has minimum index within a certain subset of Ω^v, i.e. $r = \min \Phi^v$ for some $\Phi^v \subseteq \Omega^v$. The least-index rule per se does not guarantee convergence. In fact we can ensure termination by choosing the blocking variable only among those that make the number of degenerate variables decrease as slowly as possible, i.e. among the index-set

$$\Phi^v = \arg\min_i \left\{ |\Omega^v| - \left|\Omega_i^{v+1}\right| > 0 : i \in \Omega^v \right\} \subseteq \Omega^v,$$

where Ω_i^{v+1} is the index-set of those variables that would satisfy the min-ratio test at iteration $v + 1$ if the driving variable at iteration v was blocked with y_i^v as $i \in \Omega^v$. The previous conditional implies that a pivot step is taken and then reset in a sort of "look-ahead" fashion, hence we will refer to this rule as the *look-ahead* (*pivot*) *rule*.

Before actually proceeding to illustrate a version of Lemke's algorithm applied to our matching problem, let us take a look at the tableaus that it generates. This will help us to identify regularities that are reflected in the behavior of the algorithm itself. The initial tableau follows:

	1	$x_1 \cdots x_n$	x_{n+1}	x_{n+2}	θ
y_1	0	A_G	-1	1	1
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
y_n	0		-1	1	1
y_{n+1}	-1	$1 \cdots 1$	0	0	1
y_{n+2}	1	$-1 \cdots -1$	0	0	1

As q_{n+1} is the only negative entry for the column of q, the first pivot to occur during initialization is $\langle y_{n+1}, \theta \rangle$ thereby producing the following transformation:

	1	$x_1 \cdots x_n$	x_{n+1}	x_{n+2}	y_{n+1}
y_1	1	$A_G - ee^{\mathrm{T}}$	-1	1	1
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
y_n	1		-1	1	1
θ	1	$-1 \cdots -1$	0	0	1
y_{n+2}	2	$-2 \cdots -2$	0	0	1

The driving variable for the second pivot is x_{n+1}. Since $m^1_{i,n+1} = -1$ for all $i = 1, \dots, n$ it is immediate to see that the relative blocking variable can be any one of $y_1, \dots, y_n$. In this case we apply no degeneracy resolution criterion, but rather allow for user intervention by catering for the possibility of deciding the second driving variable a priori. Let thus y_p be the (arbitrary) variable that shall block x_{n+1}. After performing $\langle y_p, x_{n+1} \rangle$, we have the following tableau:

	1	$x_1 \cdots x_n$	y_p	x_{n+2}	y_{n+1}
y_1	0	$A_{G,p}$	1	0	0
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
y_{p-1}	0		1	0	0
x_{n+1}	1		-1	1	1
y_{p+1}	0		1	0	0
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
y_n	0		1	0	0
θ	1	$-1 \cdots -1$	0	0	1
y_{n+2}	2	$-2 \cdots -2$	0	0	1

where $A_{G,p}$ denotes the matrix whose rows are defined as

$$(A_{G,p})_i = \begin{cases} (A_G)_p - e^{\mathrm{T}} & \text{if } i = p, \\ (A_G)_i - (A_G)_p & \text{otherwise.} \end{cases}$$

Algorithm 1 formalizes the previous statements.

Algorithm 1. Lemke's "Scheme I" with the look-ahead rule.

```
Input: A graph $G = (V, E)$ and $p \in V$.
Let $(q_G, e, M_G)$ be the augmented LCP, where $q_G$ and $M_G$ are defined in (4).
$v \leftarrow 0$, $\langle y_{n+1}, \theta \rangle$, $v \leftarrow v + 1$, $\langle x_p, x_{n+1} \rangle$
The driving variable is $x_p$.
Infinite loop
    $v \leftarrow v + 1$
    Let $x^v_s$ denote the driving variable.
    $\Omega^v = \arg\min_i \{\frac{-q^v_i}{m^v_{is}} : m^v_{is} < 0\}$
    If $|\Omega^v| = 1$ then $r = \min \Omega^v$
    else $\Phi^v = \arg\min_i \{|\Omega^v| - |\Omega^{v+1}_i| > 0 : i \in \Omega^v\}$,
    $r = \min \Phi^v$.
    $\langle y^v_{r \in \Omega^v}, x^v_s \rangle$
    If $y^v_r \equiv \theta$ then
        Let $\bar{x}$ denote the complementary solution of $(q_G, M_G)$.
        The result is supp $(\bar{x}) \cap V$
    The new driving variable is the complementary of $y^v_r$.
```

Algorithm 2. The *pivoting-based heuristic* (PBH) for graph matching.

```
Input: Two graphs $G_1$ and $G_2$.
Construct the association graph $G = (V, E)$ of $G_1$ and $G_2$.
Let $G' = (V', E')$ be a permutation of $G$
with $\deg(u') \geqslant \deg(v')$ for all $u', v' \in V'$ with $u' < v'$.
$K^\star \leftarrow \emptyset$
For $v' = 1, \dots, n : \deg(v') > |K^\star|$ do
    Run Algorithm 1 with $G'$ and $v'$ as input.
    Let $K$ be the obtained result.
    If $|K| > |K^\star|$, then $K^\star \leftarrow K$.
The result is the mapping of $K^\star$ in $G$.
```

Empirical evidence indicated p as a key parameter for the quality of the final result of Algorithm 1. We thus had to consider iterating for most, if not all, vertices of V as outlined in Algorithm 2. We also observed that the schema is sensitive to the ordering of nodes and found that the best figures were obtained by reordering G by decreasing node degrees. We will refer to this scheme by the name *pivoting based heuristic* (PBH).

If n denotes the number of vertices in the association graph, computing $\left|\Omega^{v+1}_i\right|$ can be done with $\mathrm{O}(n)$ time complexity as only the driving column is needed for this purpose, and a pivotal transformation takes $\mathrm{O}(n^2)$ computations. Hence, so is the complexity of each step of Algorithm 1. Moreover, PBH iterates over (most of) the nodes of G. This, together with previous observations in (Massaro et al., in press), gives us a strong empirical evidence that the number of steps of PBH is $\mathrm{O}(sn)$, where s is the size of the maximal common subgraph found.

4. Results and conclusion

In this section we present some experimental results of applying PBH to the problem of matching pairs of random graphs. Random structures represent a useful benchmark not only because they are not constrained to any particular application, but also because it is simple to replicate experiments and hence to make comparisons with other algorithms.

We generated random 50-node graphs using edge-probabilities (i.e., densities) ranging from 0.1 to 0.9. For each density value, 20 graphs were constructed so that, overall, 180 graphs were used in the experiments. Each graph had its vertices permuted and was then subject to a corruption process which consisted of randomly deleting a fraction of its nodes. In so doing we obtained a graph isomorphic to a proper subgraph of the original one. Various levels of corruption (i.e., percentage of node deletion) were used, namely 0% (the pure isomorphism case), 10%, 20% and 30%. In other words, the order of the corrupted graphs ranged from 50 to 35.

PBH was applied on each pair of graphs thus constructed and, after convergence, the percentage of matched nodes was recorded. Replicator dynamics, a class of dynamical systems developed in evolutionary game theory and other branches of mathematical biology, have recently proven remarkably powerful on simple versions of the graph matching problem, despite their inherent inability to escape from local optima (Pelillo, 1998; Pelillo, 1999; Pelillo, in press; Pelillo et al., 1999). For the

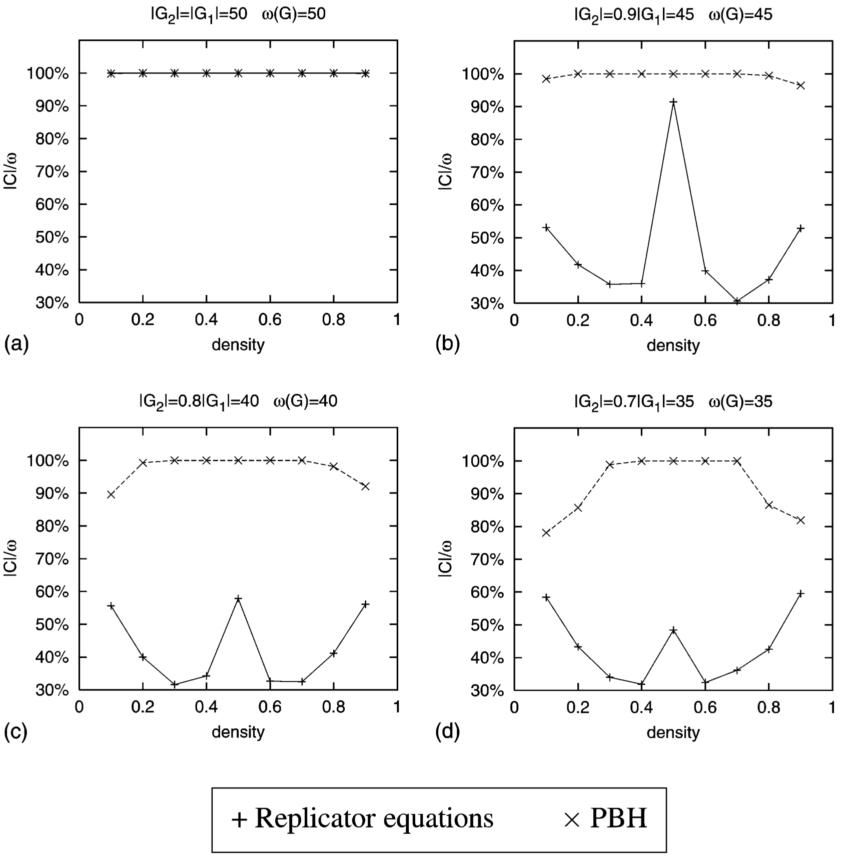

Fig. 1. Results of matching 50-node random graphs, with varying levels of corruption, using PBH and replicator dynamics. The x-axis represents the (approximate) density of the matched graphs, while the y-axis represents the percentage of correct matches. Here ω is the size of the maximum clique of the association graph, i.e., the size of the maximum isomorphism, and $|C|$ is the size of the isomorphism returned by the algorithms, i.e., the size of the maximal clique found. Figures (a)–(d) correspond to different levels of corruption, i.e., 0%, 10%, 20% and 30%, respectively. All curves represent averages over 20 trials.

sake of comparison, we therefore tested their effectiveness on our (more difficult) subgraph isomorphism task.

Fig. 1 plots the results obtained with both PBH and replicator dynamics. As can be seen, whenever no corruption was applied on the original graphs (i.e., in the case of isomorphic graphs), both methods found systematically a maximum isomorphism, i.e. a maximum clique in the association graph (as far as replicator equations are concerned, this is indeed not surprising, as shown in (Pelillo, 1999)). The emerging picture did not change significantly for PBH when we did delete some nodes, whereas the replicator equations underwent a notable deterioration of performance. For the latter case, in fact, the curves (b)–(d) of Fig. 1 have a peculiar "w" shape with a performance peak on 0.5-density graphs, where the corresponding association graphs have minimum density.

It is also possible to compare our approach with the well-known Graduated Assignment method (GA) of Gold and Rangarajan (1996). Their algorithm is based on the minimization of an objective function which is significantly different from ours. In (Gold and Rangarajan, 1996, Fig. 8) they present results of applying GA on 100-node random graphs with density values ranging from 4% to 28%, and various corruption levels up to 30%. From their results a significant sensitivity of GA to node deletion emerges, similarly to what happens for the replicator dynamics, but in a less pronounced manner. In contrast, the performance of PBH (on smaller graphs) seems to be more insensitive to the corruption level, a feature which is clearly desirable. Notice, however, that the results tend to degrade slowly for the denser association graphs that arise for densities close to 0 and 1. This phenomenon tends to strengthen up slowly as more nodes are deleted, but the average efficiency never goes below 85%. This figure is superior to those obtained with replicator dynamics and GA.

These experimental results are very encouraging and indicate that the proposed framework offers a promising new way to tackle graph matching and related combinatorial problems. Note also that our algorithm is completely devoid of working parameters, a valuable feature which distinguishes it from other heuristics proposed in the literature. Clearly, more experimental work needs to be done in order to fully assess the potential of the method. Also, generalizations of the proposed approach to attributed graphs and error-tolerant (many-to-many) matching problems are possible, along the lines suggested in (Pelillo, 1998; Pelillo et al., 2001). All this will be the subject of future work.

References

Ambler, A.P. et al., 1973. A versatile computer-controlled assembly system. In: Proc. 3rd Int. J. Conf. Artif. Intell., Stanford, CA, pp. 298–307.

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

Bomze, I.M., 1997. Evolution towards the maximum clique. J. Global Optim. 10, 143–164.

Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M., 1999. The maximum clique problem. In: Du, D.-Z., Pardalos, P.M., (Eds.), Handbook of Combinatorial Optimization-Supplement, vol. A. Kluwer Academic Publishers, Dordrecht, pp. 1–74.

Bunke, H., 2000. Recent developments in graph matching. In: Proceedings of 15th International Conference on Pattern Recognition, vol. 2. IEEE Computer Society Press, pp. 117–124.

Cottle, R.W., Pang, J., Stone, R.E., 1992. The Linear Complementarity Problem. Academic Press, Boston, MA.

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

Li, S.Z., 1992. Matching: invariant to translations, rotations, and scale changes. Pattern Recognition 25, 583–594.

Massaro, A., Pelillo, M., Bomze, I.M. A complementary pivoting approach to the maximum weight clique problem. SIAM J. Optim. 12 (4), 928–948.

Messmer, B., Bunke, H., 1998. A new algorithm for error tolerant subgraph isomorphism. IEEE Trans. Pattern Anal. Machince Intell. 20, 493–505.

Motzkin, T.S, Straus, E.G., 1965. Maxima for graphs and a new proof of a theorem of Turán. Canad. J. Math. 17 (4), 533–540.

Pelillo, M., 1998. A unifying framework for relational structure matching. In: Proceedings of 14th International Conference on Pattern Recognition. IEEE Computer Society Press, pp. 1316–1319.

Pelillo, M., 1999. Replicator equations, maximal cliques, and graph isomorphism. Neural Comput. 11, 1933–1955.

Pelillo, M. Matching free trees, maximal cliques, and monotone game dynamics. IEEE Trans. Pattern Anal. Machince Intell., in press.

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

Arcelli, C., Siddiqi, K., Zucker, S.W., 2001. Many-to-many matching of attributed trees using association graphs and game dynamics. In: Cordella, L.P., Sanniti di Baja, G. (Eds.), Visual Form 2001. Springer, Berlin, pp. 583–593.

Shapiro, L.G, Haralick, R.M., 1981. Structural descriptions and inexact matching. IEEE Trans. Pattern Anal. Machine Intell. 3, 504–519.

Tsai, W.-H., Fu, K.-S., 1983. Subgraph error-correcting isomorphisms for syntactic pattern recognition. IEEE Trans. Systems Man Cybernet. 13, 48–62.

Wilson, R.C., Hancock, E.R., 1997. Structural matching by discrete relaxation. IEEE Trans. Pattern Anal. Machince Intell. 19 (6), 634–648.