Approximating the Maximum Weight Clique Using Replicator Dynamics *

Immanuel M. Bomze [(1)] **Marcello Pelillo** [(2)] **Volker Stix** [(1)]

[(1)] Institut für Statistik und Decision Support Systems

Universität Wien

Universitätsstraße 5, A-1010 Wien, Austria

[(2)] Dipartimento di Informatica

Università Ca' Foscari di Venezia

Via Torino 155, I-30172 Venezia Mestre, Italy

* *IEEE Transactions on Neural Networks*, vol. 11, no. 6, November 2000.

Abstract

Given an undirected graph with weights on the vertices, the maximum weight clique problem (MWCP) is to find a subset of mutually adjacent vertices (i.e., a clique) having largest total weight. This is a generalization of the classical problem of finding the maximum cardinality clique of an unweighted graph, which arises as a special case of the MWCP when all the weights associated to the vertices are equal. The problem is known to be NP-hard for arbitrary graphs and, according to recent theoretical results, so is the problem of approximating it within a constant factor.

Although there has recently been much interest around neural network algorithms for the unweighted maximum clique problem, no effort has been directed so far towards its weighted counterpart. In this paper, we present a parallel, distributed heuristic for approximating the MWCP based on dynamics principles developed and studied in various branches of mathematical biology. The proposed framework centers around a recently introduced continuous characterization of the MWCP which generalizes an earlier remarkable result by Motzkin and Straus. This allows us to formulate the MWCP (a purely combinatorial problem) in terms of a continuous quadratic programming problem. One drawback associated with this formulation, however, is the presence of "spurious" solutions, and we present characterizations of these solutions. To avoid them we introduce a new regularized continuous formulation of the MWCP inspired by previous works on the unweighted problem, and show how this approach completely solves the problem.

The continuous formulation of the MWCP naturally maps onto a parallel, distributed computational network whose dynamical behavior is governed by the so-called *replicator equations.* These are dynamical systems introduced in evolutionary game theory and population genetics to model evolutionary processes on a macroscopic scale. We present theoretical results which guarantee that the solutions provided by our clique finding replicator network are actually the ones being sought. Extensive experiments on both randomly generated and standard benchmark graphs have been conducted, and the results obtained confirm the effectiveness of the proposed approach.

Key words—Maximum weight clique, replicator equations, evolutionary game theory, dynamical systems, quadratic programming.

1 Introduction

The maximum clique problem (MCP) is a well-known example of a combinatorial optimization problem, not only because it was one of the first problems shown to be NP-complete in Karp's classical paper on computational complexity [42], but also for its theoretical as well as practical implications. Given an undirected graph, it consists of finding a subset of pairwise adjacent vertices (i.e., a *clique*) having largest cardinality. The MCP finds applications in a variety of practical problems in such diverse domains as computer vision, experimental design, information retrieval, fault tolerance, etc. (see [19] and references therein). In addition, many important intractable problems turn out to be easily reducible to the MCP, and these include, for example, the Boolean satisfiability problem, the independent set problem, the subgraph isomorphism problem, and the vertex covering problem.

Due to the inherent computational complexity of the MCP, exact algorithms are guaranteed to return a solution only in a time which increases exponentially with the number of vertices in the graph, and this makes them inapplicable even to moderately large problem instances. Moreover, a series of recent theoretical results show that the MCP is in fact difficult to solve even in terms of approximation (see [5] for a recent review concerning approximation issues for NP optimization problems). Strong evidence of this fact came in 1991, when Feige *et al.* [27] (see also [28]) proved that if there is a polynomial-time algorithm that approximates the MCP within a factor of $2^{\log^{1-\varepsilon} n}$, then any NP problem can be solved in "quasi-polynomial" time (i.e., in $2^{\log^{O(1)} n}$ time). The result was further refined by Arora *et al.* [4, 3] one year later. Specifically, they proved that there exists an $\varepsilon > 0$ such that no polynomial-time algorithm can approximate the size of the maximum clique within a factor of n^{ε}, unless $P = NP$. More recent developments along these lines can be found in [15, 16, 34]. In the light of these negative results, much effort has recently been directed towards devising efficient heuristics for the MCP, for which no formal guarantee of performance may be provided, but are anyway of interest in practical applications [19].

An important generalization of the MCP which is receiving increasing attention arises when positive weights are associated to the vertices of the graph. In this case, the problem is known as the maximum weight clique problem (MWCP) and consists of finding a clique in the graph which has largest total weight (note that the maximum weight clique does not necessarily have largest cardinality). It is clear that the classical unweighted version turns out to be a special case when the weights assigned to the vertices are all equal, and for this reason the MWCP has at least the same computational complexity as the unweighted counterpart (but see [7, 10] for classes of graphs for which the problem is solvable in polynomial time).

The MWCP has important applications in such fields as computer vision, pattern recognition and robotics, where graphs are employed as a convenient means of representing high-level pictorial information [11]. Here, the vertices of these graphs typically represent features such as lines, corners, regions, etc., and the edges represent possible relations among them. A fundamental problem in these areas is to match a new image, or image representation, against a model, for recognition purposes. An elegant approach to solve this matching problem is to construct an auxiliary graph structure, called the *association graph*, from the two graphs being matched and then to look for maximum cliques in it [11]. In the association graph, the vertices correspond to putative associations between vertices of the graphs being matched, and the edges represent compatibility relations between associations. Graphs arising in these applications often contains numerical attributes on their vertices and/or edges. In these cases, the matching problem becomes one of finding a maximum weight clique in a vertex-weighted association graph, where the weights on the vertices usually correspond to a measure of the "similarity" between attribute vectors (see, e.g., [37, 60]).

During the past few years, a number of (traditional) algorithms have been devised to solve the MWCP. The first work on this subject appears to have been published in 1975 by Nemhauser and Trotter [51] who formulated the problem as an integer linear program and applied an implicit enumeration algorithm to solve it. Building on previous work on the unweighted version, Loukakis and Tsouros [46] developed a recursive backtracking algorithm for MWCP and presented results over random graphs with up to 250 vertices. In [54], Pardalos and Desai formulated the MWCP as an unconstrained quadratic 0-1 program and developed a branch and bound technique to solve it. They were able to run the algorithm over random graphs with up to 500 vertices. In an unpublished paper [25], Carraghan and Pardalos proposed a simple partial enumerative algorithm which turned out to be more efficient than that presented in [54]. Balas and Xue [8] extended a previous work on the unweighted case and developed an efficient branch and bound procedure for MWCP. Basic to their algorithm is a minimum weighted coloring of a triangulated graph, which they obtain in a time complexity of $O(n^2)$. In [9], Balas and Xue proposed a fast heuristic for the weighted fractional coloring problem and used this heuristic as an upper bounding procedure in a branch and bound algorithm for the MWCP. Compared with the method in [8], computational results show the reduction in search tree size and the improved efficiency of the resulting algorithm. In 1994, Babel [6] presented a fast branch and bound procedure which we briefly describe in the experiment section of this paper. Other algorithms for the MWCP in the literature can be found in [2, 35, 48, 50].

Although in the neural network community there has recently been much interest around the unweighted version of the maximum clique problem [1, 17, 21, 30, 33, 38, 39, 40, 45, 56, 61, 62, 64], no effort has been directed so far to its weighted counterpart. We are only aware of the work of Ballard *et al.* [12], who simply described a method to encode the problem in terms of a connectionist architecture, without presenting any experimental results. In this paper, we expand on previous works done on the unweighted maximum clique problem [21, 56, 58], and present a parallel, distributed heuristic for approximating the MWCP based on dynamics principles developed in various branches of mathematical biology [36]. The approach is centered around a continuous characterization of the problem introduced in a recent paper by Gibbons *et al.* [32], which allows us to formulate the MWCP (a purely combinatorial problem) in terms of a continuous quadratic programming problem. Gibbons *et al.*'s formulation is indeed a generalization of a powerful result proved in the mid-1960's by Motzkin and Straus [49] for the unweighted case, and recently extended in various ways [18, 32, 59]. The Motzkin-Straus formulation has also served as the basis for various clique finding algorithms for unweighted graphs [20, 21, 31, 55, 56]. We will show inhowfar the formulation of the MWCP by Gibbons *et al.* has one major drawback related to the existence of "spurious" solutions, a problem which is also present in the original Motzkin-Straus formulation [59]. These entail a lack of one-to-one correspondence between the solutions to the continuous optimization problem and those to the original, discrete one. We shall present characterizations of these spurious solutions, which sometimes can provide information on the structure of the underlying graph. In principle, however, spurious solutions should be avoided. To this end, we introduce a new regularized continuous formulation of the MWCP inspired by previous works on the unweighted problem, and show how this approach completely solves the problem.

The continuous formulation allows us to map the MWCP onto a parallel, distributed computational network whose dynamical behavior is governed by the so-called *replicator equations* [36, 67]. These are dynamical systems introduced in evolutionary game theory to model the evolution of behavior in animal conflicts, and independently studied, under different names and with different purposes, in such diverse fields as population genetics, macromolecular evolution, mathematical ecology, and computer vision. We present theoretical results which guarantee that the solutions provided by our clique finding replicator networks are actually the ones being sought. Extensive experiments conducted over hundreds graphs also demonstrate the effectiveness of the proposed approach.

The outline of this paper is as follows. In Section 2 we present Gibbons *et al.*'s continuous formulation of the MWCP, discuss its limitations related to the existence of spurious solutions, and develop a straightforward generalization which overcomes these drawbacks. In Section 3 we introduce replicator dynamics and their properties, and show how this class of dynamical systems can be employed to serve our purposes. Section 4 presents experimental results on both randomly generated graphs and standard DIMACS benchmark graphs, and Section 5 concludes the paper.

2 Continuous formulations of the maximum weight clique problem

2.1 Notations and definitions

Let $G = (V, E, \mathbf{w})$ be an arbitrary undirected and weighted graph, where $V = \{1, \ldots, n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set and $\mathbf{w} \in \mathbb{R}^n$ is the *weight* vector, the i-th component of which corresponds to the weight assigned to vertex i. It is assumed that $w_i > 0$ for all $i \in V$. The *order* of G is the number of its vertices (namely, n). The complement graph of G, denoted by $\overline{G}$, is a graph having the same vertex set as G and $\overline{E} = \{(i,j) \in V \times V \ : \ i \neq j \text{ and } (i,j) \notin E\}$ as the edge set. Two distinct vertices $i, j \in V$ are said to be *adjacent* if they are connected by an edge, i.e., $(i,j) \in E$. Given a subset of vertices S, the weight assigned to S will be denoted by

$$W(S) = \sum_{i \in S} w_i \ .$$

As usual, the sum over the empty index set is defined to be zero.

A *clique* is a subset of the vertex set V in which every pair of vertices are adjacent. A clique S is called *maximal* if no strict superset of S is a clique. A maximal weight clique S is a clique which is not contained in any other clique having weight larger than $W(S)$. Since we are assuming that all weights are positive, it is clear that the concepts of maximal and maximal weight clique coincide, and we shall not make any distinction throughout the paper furthermore. A maximum cardinality clique (or, simply, a maximum clique) is a clique whose cardinality is the largest possible, while a maximum weight clique is a clique having largest total weight. The MWCP is the problem of finding a clique S in G having maximum weight $W(C)$ (see [19] for a review). The maximum size of a clique in G is called the *clique number* and is typically denoted by $\omega(G)$; the *weighted clique number* of G, denoted by $\omega(\mathbf{w}, G)$, is the maximum weight of a clique in G.

Note that maximum cliques are obviously also maximal, while the contrary need not be true. Moreover, since we are assuming that all weights are positive, maximum weight cliques are also maximal cliques. Note that the classical (unweighted) version of the maximum clique problem arises as a special case when all vertices are assigned a weight equal to 1, in which case $W(S) = \#S$, where $\#S$ denotes the number of elements in S.

2.2 Basic formulation

In 1965, Motzkin and Straus [49] established a remarkable connection between the unweighted maximum clique problem and a certain standard quadratic programming problem. Let $G = (V, E)$ be an undirected (unweighted) graph, and let Δ denote the standard simplex in the n-dimensional Euclidean space $\mathbb{R}^n$ (see Fig. 1):

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

where a prime denotes transposition, and the letter $\mathbf{e}$ is reserved for a vector of appropriate length, consisting of unit entries (hence $\mathbf{e}'\mathbf{x} = \sum_{i \in V} x_i$). For a subset $S \subseteq V$ of vertices, we shall denote the face of Δ corresponding to S by

$$\Delta_S = \{\mathbf{x} \in \Delta : x_i = 0 \text{ if } i \notin S\}$$

and its relative interior by

$$\Delta_S^\circ = \{\mathbf{x} \in \Delta_S : x_i > 0 \text{ if } i \in S\} \ .$$

****

FIGURE 1 APPROXIMATELY HERE

****

Now, consider the following quadratic function

$$g(\mathbf{x}) = \mathbf{x}' A_G \mathbf{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j \tag{1}$$

where $A_G = (a_{ij})_{i,j \in V}$ is the adjacency matrix of G, i.e., $a_{ij} = 1$ if $(i,j) \in E$, and $a_{ij} = 0$ if $(i,j) \notin E$, and let $\mathbf{x}^*$ be a global maximizer of g on Δ. Motzkin and Straus proved that the clique number of G is related to $g(\mathbf{x}^*)$ according to the following formula

$$\omega(G) = \frac{1}{1 - g(\mathbf{x}^*)} \ .$$

Additionally, Motzkin and Straus proved that a subset of vertices S is a maximum clique of G if and only if its *characteristic* vector $\mathbf{x}^S$, which is the vector in Δ defined by $x_i^S = 1/\#S$ if $i \in S$ and $x_i^S = 0$ otherwise, is a global maximizer of g on Δ.[1] Though probably motivated by genuine mathematical curiosity, the Motzkin-Straus theorem has indeed intriguing computational implications. It suggests a fundamentally new way of solving the maximum clique problem, by allowing us to move from the discrete to the continuous domain. As recently pointed out by Pardalos [53], continuous formulations of discrete optimization problems turn out to be particularly attractive. They not only allow us to exploit the full arsenal of continuous optimization techniques, thereby leading to the development of new efficient algorithms, but may also reveal unexpected theoretical properties. The Motzkin-Straus approach has recently been extended in various ways [18, 32, 59] and has served as the basis of many clique-finding algorithms [21, 31, 39, 55, 56]. It has also been used to determine theoretical bounds on the maximum clique size [24, 55].

In a recent paper, Gibbons *et al.* [32] generalized the Motzkin-Straus theorem to the weighted case. They first reformulated the Motzkin-Straus problem as a minimization problem by considering the function

$$f(\mathbf{x}) = \mathbf{x}'(I + A_{\overline{G}})\mathbf{x} \tag{2}$$

where I is the identity matrix, and $A_{\overline{G}}$ is the adjacency matrix of the complement graph $\overline{G}$. It is straightforward to see that if $\mathbf{x}^*$ is a global minimizer of f in Δ, then we have:

$$\omega(G) = \frac{1}{f(\mathbf{x}^*)} \ .$$

[1]Actually, in their original paper, Motzkin and Straus proved just the "only-if" part of this theorem. The converse direction is however a straightforward consequence of their result [59].

This is simply a different formulation of the Motzkin-Straus theorem. Given a weighted graph $G = (V, E, \mathbf{w})$, Gibbons *et al.* then considered the class of symmetric $n \times n$ matrices defined as:

$$\mathcal{M}(\mathbf{w}, G) = \left\{ B = (b_{ij})_{i,j \in V} : b_{ii} = \frac{1}{w_i}\,,\; b_{ij} = b_{ji} \geq \frac{b_{ii} + b_{jj}}{2} \text{ if } (i,j) \notin E\,, b_{ij} = 0, \text{ otherwise} \right\}$$

and the following quadratic program, which is in general indefinite:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) = \mathbf{x}' B \mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{3}$$

where $B \in \mathcal{M}(\mathbf{w}, G)$. Using a proof technique suggested by Lovász, Gibbons *et al.* proved the following result.

Theorem 1 *Let $G = (V, E, \mathbf{w})$ be an arbitrary weighted graph. Then, for any $B \in \mathcal{M}(\mathbf{w}, G)$ we have:*

$$\omega(\mathbf{w}, G) = \frac{1}{f(\mathbf{x}^*)}$$

where $\mathbf{x}^*$ is a global minimizer of program (3).

Furthermore, denote by $\mathbf{x}^S(\mathbf{w}) \in \Delta_S^\circ$ the *weighted characteristic vector* of S, which is a vector with coordinates

$$x_i^S(\mathbf{w}) = \begin{cases} w_i / W(S) & \text{if } i \in S\,, \\ 0 & \text{otherwise.} \end{cases}$$

From the previous theorem it is possible to derive the following result, which establishes a correspondence between maximum weight cliques and global minima of (3), cf. Proposition 12 in Appendix B.

Theorem 2 *A subset S of vertices of a weighted graph $G = (V, E, \mathbf{w})$ is a maximum weight clique if and only if its weighted characteristic vector $\mathbf{x}^S(\mathbf{w})$ is a global minimizer of (3).*

However, the problem in formulation (3), with a matrix B drawn from the Motzkin-Straus class $\mathcal{M}(\mathbf{w}, G)$, has some *structural* drawbacks. One of them is that maximal cliques

S having less than maximum weight are not necessarily reflected by local solutions to (3) for any $B \in \mathcal{M}(\mathbf{w}, G)$, in that $\mathbf{x}^S(\mathbf{w})$ need not be a local minimizer of $\mathbf{x}'B\mathbf{x}$ on Δ, as the elementary example of two disconnected vertices with different weights shows, where the objective in (3) can be a non-constant affine function although both singletons are maximal cliques. Therefore we propose a regularized variant of the formulation, for which we prove a sharpened version of the preceding results (subsection 2.4). The other advantage of the regularization is that spurious solutions, which can occur with the Motzkin-Straus class and which are discussed in detail in the following subsection, are automatically prohibited with the proposed alternative approach.

Anyhow, as with its unweighted counterpart, the previous result suggests a new way of solving the MWCP which is based on the minimization of the function f in Δ: a continuous optimization problem. Notice that the matrix $I + A_{\overline{G}}$ belongs to $\mathcal{M}(\mathbf{e}, G)$. In other words, the Motzkin-Straus theorem turns out to be a special case of the preceding result.

2.3 Spurious solutions and their characterization

The previous formulation of the MWCP has a major drawback which, as in the unweighted case [59], is related to the presence of "spurious" solutions. A local or global solution of program (3) is said to be *spurious* if it is not in the form of a weighted characteristic vector $\mathbf{x}^S(\mathbf{w})$, for some subset S of vertices. In principle, spurious solutions represent a nuisance for we cannot extract the vertices comprising the clique directly from them; they just provide information about the weighted clique number. In this case, we would be forced to make recourse to iterative or recursive routines to get the clique vertices, as described in [52, 55]. In this subsection we shall give characterizations of spurious solutions which, as a by-product, can sometimes provide information about the structure of the underlying graph. Regularizations which remove these phenomena are discussed in the next subsection.

To illustrate, consider the simple graph having vertex set $V = \{1, 2, 3\}$ and weight vector $\mathbf{w} = (2, 1, 1)'$, with edges only between vertices 1 and 2, and 1 and 3, In this case, the class

$\mathcal{M}(\mathbf{w}, G)$ contains all matrices of the form:

$$B = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1 & b \\ 0 & b & 1 \end{pmatrix}$$

where $b \geq 1$. The subsets $S = \{1, 2\}$ and $T = \{1, 3\}$ are maximum weight cliques, and from Theorem 2 we know that, for any matrix $B \in \mathcal{M}(\mathbf{w}, G)$, their weighted characteristic vectors $\mathbf{x}^S(\mathbf{w}) = (2/3, 1/3, 0)'$ and $\mathbf{x}^T(\mathbf{w}) = (2/3, 0, 1/3)'$ are global minimizers of $\mathbf{x}'B\mathbf{x}$ on Δ. Let us consider a particular member of $\mathcal{M}(\mathbf{w}, G)$, namely the matrix B satisfying $b = 1$, which lies on the boundary of the space $\mathcal{M}(\mathbf{w}, G)$. It is simple to see by direct calculations that for all $\alpha \in [0, 1]$, all vectors of the form $(2/3, \alpha/3, (1-\alpha)/3)'$ also are global minimizers of $\mathbf{x}'B\mathbf{x}$ on Δ, and are therefore spurious solutions. Observe that, in this particular case, spurious solutions contain more information than weighted characteristic vectors in that they allow us to identify *both* cliques of the graph, not just one. Unfortunately, as we shall see below, this is not always the case.

The previous example is just an instance of a more general property which we present here. To this end, we need to introduce additional notations and definitions. Given two disjoint subsets of vertices S and T, we shall denote by $H_{S,T}$ the set

$$H_{S,T} = (S \setminus T \times T \setminus S) \setminus E\ . \tag{4}$$

Note that, trivially, $(i, j) \in H_{S,T}$ if and only if $(j, i) \in H_{T,S}$. Moreover, let

$$H^*_{S,T} = H_{S,T} \cup H_{T,S}$$

denote the set of edges in the complement graph $\overline{G}$ which connect vertices of $S \setminus T$ with those in $T \setminus S$. Given two disjoint subsets of vertices S and T, we say that a set $E' \subseteq E$ of edges *crosses* S and T if every edge in E' has one endpoint in S and the other in T, i.e., $E' \subseteq (S \times T) \cup (T \times S)$. A set E' crossing S and T is called a *matching* if no two edges in E' have the same endpoint.

Now, we exhibit necessary and sufficient conditions for occurrence of spurious solutions on the segment connecting the weighted characteristic vectors $\mathbf{x}^S(\mathbf{w})$ and $\mathbf{x}^T(\mathbf{w})$, where

S and T are two maximal cliques (which necessarily then must have equal weight). In fact, we show more: namely, that the objective function f is always concave along such a segment, and characterize the case when f is an affine function (which must be constant if $W(S) = W(T)$).

Theorem 3 *Let $G = (V, E, \mathbf{w})$ be a weighted graph, and let S and T be two distinct maximal cliques. Then*

$$f(\alpha \mathbf{x}^S(\mathbf{w}) + (1-\alpha)\mathbf{x}^T(\mathbf{w})) \geq \alpha f(\mathbf{x}^S(\mathbf{w})) + (1-\alpha) f(\mathbf{x}^T(\mathbf{w}))$$

for all $\alpha \in [0,1]$ with equality for some (and hence for all) α with $0 < \alpha < 1$ if and only if

(a) the matrix B satisfies the condition

$$b_{ij} + b_{ji} = b_{ii} + b_{jj} \quad \text{for all } (i,j) \in H_{S,T} \tag{5}$$

and

(b) $H^*_{S,T}$ is a matching in $\overline{G}$ between $S \setminus T$ and $T \setminus S$, in which case S and T have equal cardinality.

Proof: See Appendix A. □

Note that condition (5) is always satisfied for the "distinguished" matrix $B(\mathbf{w}) = (b_{ij}(\mathbf{w}))_{i,j \in V} \in \mathcal{M}(\mathbf{w}, G)$ with entries

$$b_{ij}(\mathbf{w}) = \begin{cases} \frac{1}{2w_i} + \frac{1}{2w_j} & \text{if } (i,j) \notin E\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{6}$$

Observe that in the unweighted case $\mathbf{w} = \mathbf{e}$, we have $B(\mathbf{e}) = \mathbf{e}\mathbf{e}' - A_G$ where A_G is the adjacency matrix of G. Hence, as already noted in [32], problem (3) with $B = B(\mathbf{e})$ is completely equivalent to the original Motzkin-Straus program, namely to maximize $\mathbf{x}' A_G \mathbf{x}$ over Δ.

Theorem 3 suggests us that one way to avoid spurious solutions (or segments joining $\mathbf{x}^S(\mathbf{w})$ and $\mathbf{x}^T(\mathbf{w})$ along which the objective function is affine), is to stipulate that $2b_{ij} >$

$\frac{1}{W(S)} + \frac{1}{W(T)}$ for one pair $(i,j) \in H_{S,T}$, thus violating condition (5). In the next section we shall provide a more systematic regularization. On the other hand, the choice $B = B(\mathbf{w})$ does not rule out a priori the detection of these phenomena. For the readers' convenience, we recapitulate our findings in the next corollary.

Corollary 4 *Let S and T be two distinct maximal cliques of a weighted graph $G = (V, E, \mathbf{w})$ with equal weights $W(S) = W(T)$, such that $\mathbf{x}^S(\mathbf{w})$ and $\mathbf{x}^T(\mathbf{w})$ are local minimizers of $f(\mathbf{x}) = \mathbf{x}'B(\mathbf{w})\mathbf{x}$ on Δ. Then all points lying on the segment connecting $\mathbf{x}^S(\mathbf{w})$ and $\mathbf{x}^T(\mathbf{w})$ are also local minimizers of f if and only if $H^*_{S,T}$ is a matching in $\overline{G}$ between $S \setminus T$ and $T \setminus S$.*

Proof: follows immediately from Theorem 3 and the definition of $B(\mathbf{w})$ in (6). □

The following is a generalization of Theorem 3.

Theorem 5 *Let $S_1, S_2, \ldots, S_q$ be q distinct maximal cliques of a weighted graph $G = (V, E, \mathbf{w})$. Then*

$$f\left(\sum_{h=1}^{q} \alpha_h \mathbf{x}_h^S(\mathbf{w})\right) \geq \sum_{h=1}^{q} \alpha_h f(\mathbf{x}_h^S(\mathbf{w}))$$

for all positive $\alpha_1, \ldots, \alpha_q$ such that $\sum_{h=1}^{q} \alpha_h = 1$ with equality if and only if

(a) the matrix B satisfies condition (5) with $S = S_h$ and $T = S_l$ for all $h, l = 1 \ldots q$; and

(b) $H^*_{S_h,S_l}$ is a matching in $\overline{G}$ between $S_h \setminus S_l$ and $S_l \setminus S_h$ for all $h, l = 1 \ldots q$, in which case $S_1, S_2, \ldots, S_q$ have all equal cardinality.

Proof: See Appendix A. □

As a consequence we have:

Corollary 6 *Let $S_1, S_2, \ldots, S_q$ be q distinct maximal cliques of a weighted graph G, with equal weights $W(S_1) = \cdots = W(S_q)$, such that $\mathbf{x}_1^S(\mathbf{w}), \ldots, \mathbf{x}_q^S(\mathbf{w})$ are local minimizers of $f(\mathbf{x}) = \mathbf{x}'B(\mathbf{w})\mathbf{x}$ on Δ. Then all points lying on the convex hull of $\mathbf{x}_1^S(\mathbf{w}), \ldots, \mathbf{x}_q^S(\mathbf{w})$ are also local minimizers of f if and only if $H^*_{S_h,S_l}$ is a matching in $\overline{G}$ between $S_h \setminus S_l$ and $S_l \setminus S_h$ for all $h, l = 1 \ldots q$.*

It is clear that when more than two cliques are involved it is difficult to establish general criteria which allow us to extract their vertices, as is the case when there are two cliques only.

2.4 A regularized formulation

Even though in certain specific circumstances spurious solutions may provide useful information concerning the structure of the underlying graph, it is clear that in general they represent a problem as they do not allow us to extract the vertices comprising a clique directly from the solution found. The major drawback is in fact the lack of one-to-one correspondence between local solutions to (3) and cliques of maximal weight. This is also reflected by the occurrence of jamming during local optimization procedures based on the so-called replicator dynamics [23]. To avoid jamming, instead of the Motzkin-Straus class $\mathcal{M}(\mathbf{w}, G)$ for the unweighted case, here a different class $\mathcal{C}(\mathbf{w}, G)$ of matrices is proposed to be used as input data for problem (3):

$$\mathcal{C}(\mathbf{w}, G) = \left\{ C = (c_{ij})_{i,j \in V} : c_{ii} = \frac{1}{2w_i}\,,\ c_{ij} = c_{ji} \geq c_{ii} + c_{jj} \text{ if } (i,j) \notin E\,, c_{ij} = 0, \text{ otherwise} \right\}$$

For the unweighted case where $\mathbf{w} = \mathbf{e}$, this regularization goes back to a different characterization of maximal cliques thoroughly studied in [18]. Numerical experiences with this approach on a larger scale are reported in [21, 23]. The class $\mathcal{C}(\mathbf{w}, G)$ represents a natural extension to the weighted case.

By contrast to the Motzkin-Straus class $\mathcal{M}(\mathbf{w}, G)$, we can prove for the class $\mathcal{C}(\mathbf{w}, G)$ again the one-to-one correspondence of strict[2] local minimizers of (3) and cliques S, if $B \in \mathcal{M}(\mathbf{w}, G)$ is replaced with a matrix $C \in \mathcal{C}(\mathbf{w}, G)$. This is done as in the unweighted case, by just adapting the arguments in [32] to this more appropriate setting. As a consequence, we also can rule out the occurrence of spurious solutions, since *all* local solutions to (3) are necessarily strict, provided the matrix is taken from the class $\mathcal{C}(\mathbf{w}, G)$:

[2]A local solution $\mathbf{x}$ to (3) is said to be *strict* if there exists a neighborhood U of $\mathbf{x}$ such that $f(\mathbf{y}) > f(\mathbf{x})$ if $\mathbf{y} \in \Delta \cap U \setminus \{\mathbf{x}\}$.

Theorem 7 *Let $G = (V, E, \mathbf{w})$ be an arbitrary weighted graph and consider a matrix $C \in \mathcal{C}(\mathbf{w}, G)$ in place of B for problem (3). Then, the following assertions hold:*

(a) A vector $\mathbf{x} \in \Delta$ is a local solution to problem (3) if and only if $\mathbf{x} = \mathbf{x}^S(\mathbf{w})$, where S is a maximal clique.

(b) A vector $\mathbf{x} \in \Delta$ is a global solution to problem (3) if and only if $\mathbf{x} = \mathbf{x}^S(\mathbf{w})$, where S is a maximum weight clique.

Moreover, all local (and hence global) solutions to (3) are strict.

A "distinguished" member of the class $\mathcal{C}(\mathbf{w}, G)$ corresponding to $B(\mathbf{w})$ as defined in (6) is the matrix $C(\mathbf{w}) = B(\mathbf{w}) - [2\,\mathrm{diag}(\mathbf{w})]^{-1} \in \mathcal{C}(\mathbf{w}, G)$ with entries

$$c_{ij}(\mathbf{w}) = \begin{cases} \frac{1}{2w_i} & \text{if } i = j\,, \\ \frac{1}{2w_i} + \frac{1}{2w_j} & \text{if } i \neq j \text{ and } (i,j) \notin E\,, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Observe that in the unweighted case, $C(\mathbf{e})$ equals $\mathbf{e}\mathbf{e}' - \widehat{A_G} = \widehat{A_{\overline{G}}}$, the regularized adjacency matrix[3] of the complement graph $\overline{G}$. This reflects the elementary property that an independent set of G is a clique of $\overline{G}$. So, while the local maximizers of $\mathbf{x}'\widehat{A_G}\mathbf{x}$ over Δ exactly are the characteristic vectors $\mathbf{x}^S$ of maximal cliques S of G, the local minimizers of $\mathbf{x}'\widehat{A_G}\mathbf{x}$ over Δ exactly are the characteristic vectors $\mathbf{x}^T$ of maximal independent sets T of G. Note that within the Motzkin-Straus class $\mathcal{M}(\mathbf{e}, G)$, there is no matrix with this straightforward interpretation. See [23] for a global optimization algorithm which needs both large cliques and large independent sets of G in parallel. There it is also shown that all principal minors of $C(\mathbf{e})$ are not vanishing, and that this is a condition sufficient for prevention of jamming in replicator dynamics. Following paths under this dynamics is a frequently used procedure in the local optimization part for finding the minimizer of the general quadratic form $\mathbf{x}'M\mathbf{x}$ over Δ, see Section 3.

[3] We abbreviate $\widehat{A_G} = A_G + \frac{1}{2}I$ for any graph G.

Of course, one could try to transfer this to the weighted case, and this approach could be backed by the observation that for small $n \leq 3$, the matrices $C(\mathbf{w})$ are all non-singular for all choices of positive weight vectors w. Also, if $\overline{G}$ is complete, one can show that $C(\mathbf{w})$ is non-singular, regardless of the size of n. Admittedly, all these cases are of restricted interest in real-world applications.

However, for general graphs G and (non-rational) weights this property is not always satisfied, as the tridiagonal 4×4-matrix $C(\mathbf{w})$ with $\mathbf{w} = [1, 1, \frac{1}{4+\sqrt{15}}, \frac{1}{4+\sqrt{15}}]'$ shows. On the other hand, an (intelligent) grid search over integer values of $\frac{1}{2w_i}$ up to 1000 for tridiagonal 4×4-matrices did not yield any singular matrix of that kind, which might suggest that any matrix $C(\mathbf{w})$ with *rational* w_i is non-singular. To the best of our knowledge, we are not aware of any clue to this question. But even if $C(\mathbf{w})$ were singular, this does not alter the desired one-to-one correspondence between local solutions of and suitable maximal cliques as in Theorem 7.

3 Clique finding replicator equations: Dynamics from evolutionary game theory

3.1 The model and its properties

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

$$\dot{x}_i(t) = x_i(t)[(M\mathbf{x}(t))_i - \mathbf{x}(t)'M\mathbf{x}(t)]\,, \quad i = 1, \ldots, n \tag{8}$$

where a dot signifies derivative w.r.t. time t, and its discrete-time counterpart

$$x_i(t+1) = x_i(t)\frac{(M\mathbf{x}(t))_i}{\mathbf{x}(t)'M\mathbf{x}(t)}\,, \quad i = 1, \ldots, n\ . \tag{9}$$

It is readily seen that the simplex Δ is invariant under these dynamics, which means that every trajectory starting in Δ will remain in Δ for all future times. Moreover, it turns out that their *stationary points*, i.e. the points satisfying $\dot{x}_i(t) = 0$ for (8) or $x_i(t+1) = x_i(t)$

for (9), coincide and are the solutions of the equations

$$x_i[(M\mathbf{x})_i - \mathbf{x}'M\mathbf{x}] = 0\,, \quad i = 1, \ldots, n\,. \tag{10}$$

As usual, a stationary point $\mathbf{x}$ is said to be *asymptotically stable* (w.r.t. a dynamical system) if every trajectory under this which starts close enough to $\mathbf{x}$, will converge to $\mathbf{x}$ as $t \to \infty$.

Both (8) and (9) are called *replicator equations* in theoretical biology, since they are used to model evolution over time of relative frequencies $x_i(t)$ of interacting, self-replicating entities. Equation (8) has been introduced in evolutionary game theory by Taylor and Jonker [65] to model evolution of behavior in intra-specific conflicts under random pairwise mating in a large, ideally infinite population. It formalizes the idea that the growth rates $\dot{x}_i/x_i$ of relative frequency x_i of the i-th behavior pattern ($i = 1, \ldots, n$) is equal to the (dis)advantage $(M\mathbf{x})_i - \mathbf{x}'M\mathbf{x} = \sum_j m_{ij}x_j - \sum_{j,k} m_{kj}x_jx_k$, measured by incremental fitness relative to the average performance within the population in state $\mathbf{x} = (x_1, \ldots, x_n)'$. Here m_{ij} denotes incremental individual fitness attributed to an i–individual when encountering a j-individual, and $M = (m_{ij})$ is the resulting fitness matrix. The behavior patterns $i \in \{1, \ldots, n\}$ are often called "pure strategies" and the interaction matrix M is also termed "payoff matrix." As it turns out, all asymptotically stationary points are (symmetric) Nash equilibrium strategies, so that the replicator dynamics could also be interpreted as a gradual and adaptive equilibrium selection process.

Note that in contrast to many other equilibrium qualifications, this selection does not need any high-order rationality but rather relies on the assumption that in a large population of actors, every actor compares her or his present performance with that of the statistical average, and then does *not* immediately choose her or his best reply to this average population behavior, but rather gradually changes her or his behavior according to (8). Nevertheless, theory guarantees that in the long run a situation will emerge where all individuals do not deviate from the best reply to the status quo: Nash equilibrium obtains. Especially in situations where transaction costs of adapting somebody's own behavior too fast are high compared to the cost of obtaining a rough statistical description of the population state, this scenario seems to be highly attractive to modern economic theory, cf., e.g. [67].

Similar arguments provide a rationale for the discrete-time version (9). Surprisingly, these dynamical equations can also be regarded as a very special case of a general class of dynamical systems introduced by Baum and Eagon [13] and studied by Baum and Sell [14] in the context of Markov chain theory. This kind of processes have proven to be useful in the speech recognition [44] and computer vision [57] domains. The dynamics (8) and (9) also arise in population genetics under the name *selection equations* in a model assuming separate (non-overlapping) generations, large population size, random union of gametes, and a selection acting only upon one chromosomal locus through different viabilities (i.e., survival probabilities), given by the the fitness matrix M of the genotypes, i.e., pairs of genes drawn from a set $\{1, \ldots, n\}$ of alleles for a single chromosomal locus. Here x_i is the gene frequency of the i-th allele. Obviously the matrix M is in this context always symmetric, since permuted gene pairs belong to the same genotype. The models (8) and (9) as selection equations go way back to Fisher [29] and Kimura [43].

From an optimization point of view, the difference between symmetric and non-symmetric matrices M is also crucial. Indeed, in the symmetric case the quadratic form $\mathbf{x}(t)'M\mathbf{x}(t)$ is increasing along trajectories of the replicator dynamics; this is the Fundamental Theorem of Natural Selection, see, e.g. [26, 47, 36].

Theorem 8 *If $M = M'$ then the function $\mathbf{x}(t)'M\mathbf{x}(t)$ is strictly increasing with increasing t along any non-stationary trajectory $\mathbf{x}(t)$ under both continuous-time (8) and discrete-time (9) replicator dynamics. Furthermore, any such trajectory converges to a stationary point. Finally, a vector $\mathbf{x} \in \Delta$ is asymptotically stable under (8) and (9) if and only if $\mathbf{x}$ is a strict local maximizer of $\mathbf{x}'M\mathbf{x}$ on Δ.*

Under replicator dynamics in continuous time, the trajectory approaches its limit most efficiently in the sense that (8) is a gradient system if one uses the (non-Euclidean) Shahshahani metric [36], which for any point $\mathbf{p} \in \Delta$ is defined by

$$d_{\mathbf{p}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i:p_i>0} (x_i - y_i)^2/p_i}\,, \quad \mathbf{x}, \mathbf{y} \in \Delta\ .$$

This efficiency result is called Kimura's maximum principle. In Appendix B, we will employ the Shahshahani metric frequently, but there the present population state $\mathbf{p}$ will be replaced with the (truncated) weight vector $\mathbf{w}$ as a reference point.

3.2 Application to MWCP

In the previous section we have seen how the MWCP can be formulated as the problem of extremizing a quadratic form over the simplex. On the other hand, we have also seen that both continuous-time and discrete-time replicator equations are able to accomplish precisely this task. Hence, the use of replicator systems to approximately solving the MWCP naturally suggests itself. Indeed, note that replicator equations are maximization procedures, while the MWCP is formulated as a minimization problem. However, it is a straightforward exercise to see that the problem of minimizing a quadratic form $\mathbf{x}'B\mathbf{x}$ on Δ is equivalent to that of maximizing $\gamma\mathbf{e}\mathbf{e}' - B$, where γ is an arbitrary constant.

Now, let $G = (V, E, \mathbf{w})$ be an n-vertex weighted graph and suppose we are looking for maximum weight cliques in it. Consider a replicator model with as many behavior patterns as there are vertices in G, and define the (symmetric and nonnegative) fitness matrix to be

$$M = \gamma\mathbf{e}\mathbf{e}' - C \tag{11}$$

where $C = (c_{ij})$ is any matrix in the class $\mathcal{C}(\mathbf{w}, G)$ defined in the previous section, and

$$\gamma = \max_{i,j \in V} c_{ij}\ .$$

From Theorem 8 we know that the system, starting from an arbitrary initial point $\mathbf{x} \in \Delta$ will iteratively maximize $\mathbf{x}'M\mathbf{x}$ and will eventually converge towards a strict local maximizer. The resulting solution $\mathbf{x}^*$, i.e. the limit of a trajectory under (8) or (9), can then be employed to determine the clique itself from the nonzero components of $\mathbf{x}^*$. Note that, since the algorithm essentially performs a (Shahshahani) gradient ascent search in Δ, there is no guarantee that $\mathbf{x}^*$ yields the global solution to (3). In the light of the preceding results, therefore, what we can say is that the process can only find maximal cliques of G, hoping they have a large weight. This is a common feature of most heuristic clique finding procedures [19]. Using an escape procedure like in [18, 23] one can then try to improve the cliques obtained.

Observe that, since the process cannot leave the boundary of Δ, it is preferable that the relaxation process begins with an interior point, i.e., $x_i(0) > 0$ for all i. A reasonable choice is to make the process start out in the vicinity of the barycenter of the simplex Δ: this would avoid that the search be biased in favor of any particular vertex in G. However, more sophisticated initialization strategies may be employed which take into account the connectivity structure of the graph at hand [24] (but see [22]).

4 Experimental results

To practically assess the effectiveness of the proposed approach, extensive simulations were conducted over a number of graphs of varying orders and densities. All experiments were done on a Pentium II 300Hz microprocessor using C++ programming language on a Linux operating system. Our implementation was similar to that described in [63], with some adaptations to the weighted case and improvements of some numerical problems. The algorithm starts from the weighted barycenter of Δ (i.e., from the vector $(w_1/W, \cdots, w_n/W)$, where W is the total weight of the graph at hand) and it iterates along a trajectory via the discrete-time version (9) of replicator dynamics (RD) until it reaches a local optimum, i.e. a maximal clique.

Since there is a lack of publicly accessible benchmark graphs, like the DIMACS graphs [41], for the weighted maximum clique case, we compared most of our weighted graph examples with an algorithm developed by L. Babel in [6], which is one of the most efficient algorithms available in the literature for the MWCP, of the standard type. Babel uses a branch and bound approach as follows: upper and lower bounds for the maximum weight clique are found by coloring the weighted graph, where the number of colors represents the total sum of all weights. So if we want to process a graph of order 500 with weights out of $\{1, \ldots, 10\}$ we may have to deal with up to 5000 colors. The branching part of Babel's algorithm divides the bounded search-tree into smaller subproblems, the branching decisions depending on a specific order of all possible remaining nodes. By applying these steps recursively, the maximum weight clique will be found in finite time, and for not too big and too dense graphs in very short time.

The comparison, however, is not thought to be competitive because the approach proposed in this paper is only able to find locally optimal solutions (i.e. maximal weighted cliques) whereas Babel's algorithm computes thousands of subproblems to find the global solution. In some cases (especially in the smallest graphs) Babel's method provided a certificate of global optimality. These cases are labeled with an asterisk in the tables below.

4.1 Experiments on random graphs

To have a flavor of how the algorithm behaves in practice we first ran the algorithm on small randomly generated graphs. In this series of experiments graphs were generated according to the customary model, where each edge out of $\binom{n}{2}$ (n is the number of vertices of the graph) is selected with fixed probability p. Therefore the average density of these graphs is p. We shall call this kind of graphs, *Type I random graphs* to distinguish them from another type of graphs we shall discuss later on in this section.

****

TABLE 1 APPROXIMATELY HERE

****

We constructed 200 such graphs with 20 vertices and varying densities, and weights uniformly distributed in $\{1, \ldots, 10\}$. Table 1 shows the results obtained using our replicator heuristic. The table shows the actual maximum weight clique (column labeled MWC) and the maximal clique found using the replicator dynamics, averaged over all 200 test cases. The results are represented as sum of weights and as size of cliques, respectively. The column "quality" reflects the (rounded) quotient $RD/MWC \cdot 100$ and the column labeled "difference" represents the difference between the average RD clique size and the average MWC-size. The MWC was calculated using a simple enumeration technique, which tests all possible cases. Of course this was possible thanks to the small order of the graphs involved.

Observe that the local solutions found by RD are not so far behind the global ones. Especially in graphs with high density, where more than one global solution typically exists, RD is close to the MWC. Note that RD tends to find larger cliques with respect to cardinality in dense graphs than the size of the actual MWC. This is an interesting result which can be found in all our observations: although RD finds a correct maximal clique, it seems to be more attractive for it to maximize cardinality rather than weight.

Since type I random graphs tend to be very regular (i.e. the degree of all vertices is nearly the same), a feature which is typically not shared by graphs arising in real-world problems, we developed a different construction approach. *Type II random graphs* also have an average density of $p = m \cdot \binom{n}{2}^{-1}$, where m is the number of edges to be set. Knowing this parameter and the order n, these graphs are constructed as follows:

```
let μ = m = p · (n choose 2);
while (μ ≠ 0)
        choose two random numbers v ∈ {1,...,n} and d ∈ {1,...,n−1};
        add d edges to randomly chosen neighbors of node v;
        if this is not possible
                add the maximum of free edges to neighbors of v;
        let μ = μ−number of actually added edges;
endwhile.
```

Obviously this algorithm avoids assigning to every edge the same probability, but rather tries to construct a graph of randomly chosen degrees d distributed over randomly chosen vertices v. Note that adding an edge does not only influence the degree of node v but also that of its neighbors. Therefore for p close to one, these graphs will tend to be regular as well.

****

TABLES 2 AND 3 APPROXIMATELY HERE

****

In the second series of experiments with random graphs we tested our algorithm on larger type I and type II random graphs, and compared our results with those obtained using Babel's procedure. We constructed 100 random graphs with 100 vertices, various density values and equally distributed weights in $\{1, \ldots, 10\}$. Tables 2 and 3 show the results obtained. They contain the same information as Table 1, but Table 2 refers to random graphs of type I and Table 3 refers to random graphs of type II. The results are again represented as sum of weights and as size of cliques, respectively (averaged over all 100 test cases).

For type I graphs the RD algorithm finds in average larger cliques but with respect to the sum of weights it is behind Babel, which found in most cases the actual maximum weight clique. For type II graphs the gap between the two algorithms decreases, but RD is still weaker in finding the MWC, which is not very surprising because RD performs only a local search, as mentioned above. In all cases we can observe that the RD becomes better when graphs are dense. This is remarkable as, typically, dense graphs represent more difficult problem instances. Both algorithms consumed about the same amount of time on these graph instances.

4.2 Experiments on DIMACS graphs

The experiments presented above reveal that RD seems to work better on graphs which are not too regular and which are large and dense. Therefore, our next step was to test the algorithm over a number of DIMACS graphs [41], a class of benchmark graphs explicitly developed to test (unweighted) clique finding algorithms. To be more precise, we considered all graphs of order larger than 500 because on smaller graphs the picture did not change compared to the preceding subsection (i.e. Babel's algorithm found most times the MWC and the RD was behind). On larger graphs, however, we found encouraging results for the RD as well. Since DIMACS graphs are unweighted, we equipped them with random weights equidistributed in $\{1, \ldots, 10\}$. In all test cases the RD computed the first local solution (normally in a few minutes) and we let Babel's algorithm ran until either it found the actual MWC (these cases are marked with an asterisk in Table 4) or the computation time taken by the algorithm exceeded RDs time by a factor of 15.

****

TABLE 4 APPROXIMATELY HERE

****

Table 4 sums up the results obtained. The column "quality" again gives the quotient $RD/Babel \cdot 100$ (rounded). The average of the "quality" column is 98.96%. As far as CPU time is concerned, it is worth mentioning that in all our experiments Babels algorithm was at least 10 times slower than RD. From the table, it can be seen that the denser the graph the harder is it for Babel to prove optimality. On the other hand the denser and larger the graph the better is the local solution of the RD. In some cases (especially on the p_hat graphs and on the hamming10-2) the local solution of RD even exceeds Babel's computation. The Graphs "keller6" and "MANN_a81" of the DIMACS series could not be rendered with Babel because the program aborted due to lack of memory. Graphs of this type still pose no problems to RD, although convergence is gradually slower and takes a lot of time even for a local result only.

4.3 Babel's algorithm vs. replicator dynamics

In applications where weights are real numbers there is clearly no limitation for using RD. In contrast, Babel's algorithm would run into severe difficulties. For example, providing a real number between 1 and 10 with three digits of precision requires 9,000 weights, which means that there are about 1,000,000 different colors necessary in a graph of order 110.

Another advantage of RD over Babel is the symmetry in finding the solution. If one permutes the vertices of a graph in any way, RD will always converge to the same local solution. Contrastingly, Babel sorts the vertices during the branching process, and hence the procedure is sensitive to node permutation, resulting in up to 10% difference in the outcome for the same (p_hat) graph.

It is also worth mentioning that while RD only solves one single local problem, Babel's algorithm solves thousands of local problems. On the other hand, the (straightforward, dense) matrix multiplication used in RD is very time consuming (no attempt has been made to use procedures for sparse matrices; implementation details may be found in [63]), whereas the coloring heuristic by Babel is much faster. But these algorithms should be compared in terms of operational time instead of real time.

Finally, we note that replicator dynamics naturally lends itself to parallel, distributed implementation whereas Babel clearly does not. We refer to [66] for a circuit realization of RD-style dynamical systems.

5 Conclusions

We have presented a new continuous framework for solving the maximum weight clique problem, and a simple parallel, distributed heuristic to approximate it. The work centers around a recently introduced continuous characterization of the MWCP which generalizes an earlier result by Motzkin and Straus. This allows us to formulate the MWCP in terms of a continuous quadratic programming problem. We have seen that one drawback associated with this formulation is the presence of "spurious" solutions which entails a lack of one-to-one correspondence between the solutions of the continuous problem and those in the original, discrete domain. After presenting some characterizations of these spurious solutions, we have introduced a new regularized continuous formulation of the MWCP to avoid them, inspired by previous works on the unweighted problem. The continuous formulation of the MWCP naturally maps onto a parallel, distributed computational network whose dynamical behavior is governed by replicator equations, a class of dynamical systems introduced in evolutionary game theory and population genetics to model evolutionary processes on a macroscopic scale. We have presented theoretical results which guarantee that the solutions provided by our clique finding replicator network are actually the ones being sought. Extensive experiments on both randomly generated and standard benchmark graphs have been conducted, and the results obtained confirm the validity of the proposed approach, despite the inherent inability of replicator dynamical systems to escape from local solutions.

Before concluding, we mention that in a recent work [60] the approach described in this paper has been applied with success to the problem of matching attributed trees arising from visual object recognition tasks. The algorithm worked remarkably well: it was always able to find the globally optimal, or nearly optimal solutions. For problems more difficult than tree matching, however, the challenge remains to devise procedures for escaping from inefficient local solutions. Our future work will be focused on improving the performance of the heuristic along the lines indicated in [18, 20, 23, 39] for the unweighted version of the problem.

Acknowledgment

The authors want to thank Luitpold Babel for kind support, especially for providing us with the original C code of his algorithm.

Appendix A: Proofs of theorems in Section 2.3

We first prove the following useful lemmas:

Lemma 9 *If S is a clique of a weighted graph $G = (V, E, \mathbf{w})$ and $B \in \mathcal{M}(\mathbf{w}, G)$, then*

$$f(\mathbf{x}^S(\mathbf{w})) = [\mathbf{x}^S(\mathbf{w})]' B \mathbf{x}^S(\mathbf{w}) = \frac{1}{W(S)} \tag{12}$$

Proof: Immediate from the definitions of $\mathbf{x}^S(\mathbf{w})$ and $\mathcal{M}(\mathbf{w}, G)$. □

Lemma 10 *Suppose that S and T are maximal cliques of a weighted graph G. The, we have:*

(a) For all $i \in S \setminus T$ there is a $j \in T \setminus S$ such that $(i, j) \in H_{S,T}$;

(b) For all $j \in T \setminus S$ there is a $i \in S \setminus T$ such that $(i, j) \in H_{S,T}$;

(c) The following inequality holds:

$$\sum_{(i,j)\in H_{S,T}} (w_i + w_j) \geq W(S \setminus T) + W(T \setminus S) \tag{13}$$

with equality if and only if $H^*_{S,T}$ is a matching in $\overline{G}$ between $S \setminus T$ and $T \setminus S$, in which case S and T have equal cardinality.

Proof: Suppose, on the contrary, that (a) is not true. Then there exists $i \in S \setminus T$ such that $(i, j) \notin H_{S,T}$ for all $j \in T \setminus S$. From the definition of $H_{S,T}$ this implies that $(i, j) \in E$ for all $j \in T \setminus S$, and this means that T is not a maximal clique, contradicting the hypothesis. The proof of (b) proceeds similarly. To show (c), observe that

$$\sum_{(i,j)\in H_{S,T}} (w_i + w_j) = W(S \setminus T) \cdot \#\{j \in T \setminus S : \ (i, j) \notin E\} + W(T \setminus S) \cdot \#\{i \in S \setminus T : \ (i, j) \notin E\}\,.$$

Therefore

$$\begin{aligned}\sum_{(i,j)\in H_{S,T}} (w_i + w_j) - W(S \setminus T) - W(T \setminus S) \quad &= \quad \sum_{i\in S\setminus T} w_i \cdot [\#\{j \in T \setminus S : \ (i, j) \notin E\} - 1] + \\ & \quad\quad \sum_{j\in T\setminus S} w_j \cdot [\#\{i \in S \setminus T : \ (i, j) \notin E\} - 1]\,.\end{aligned}$$

From properties (a) and (b), we see that the expressions in brackets are nonnegative. Hence the above sum vanishes, due to the positivity of the weight vector $\mathbf{w}$, if and only if the brackets are zero, which is equivalent to the assertion. $\square$

Lemma 11 *If S and T are maximal cliques, then*

$$[\mathbf{x}^S(\mathbf{w})]'B\mathbf{x}^T(\mathbf{w}) \geq \frac{1}{2W(S)} + \frac{1}{2W(T)} \tag{14}$$

for all $B \in \mathcal{M}(\mathbf{w}, G)$, with equality if and only if

(a) the matrix B satisfies (5) and

(b) $H^*_{S,T}$ is a matching in $\overline{G}$ between $S \setminus T$ and $T \setminus S$, in which case S and T have equal cardinality.

Proof: By definition the weighted characteristic vectors $\mathbf{x}^S(\mathbf{w}), \mathbf{x}^T(\mathbf{w})$, and from the hypothesis that S and T are cliques, we derive from $B \in \mathcal{M}(\mathbf{w}, G)$

$$\begin{aligned}
[\mathbf{x}^S(\mathbf{w})]'B\mathbf{x}^T(\mathbf{w}) &= \sum_{i,j=1}^{n} b_{ij}x_i^S(\mathbf{w})x_j^T(\mathbf{w}) = \sum_{i,j\in S\cup T} b_{ij}x_i^S(\mathbf{w})x_j^T(\mathbf{w}) \\
&= \sum_{i\in S\setminus T}\sum_{j\in T\setminus S} b_{ij}x_i^S(\mathbf{w})x_j^T(\mathbf{w}) + \sum_{i\in S\cap T} \frac{1}{w_i}x_i^S(\mathbf{w})x_i^T(\mathbf{w}) \\
&= \frac{1}{W(S)W(T)}\left[\sum_{(i,j)\in H_{S,T}} b_{ij}w_iw_j + \sum_{i\in S\cap T} w_i\right] \\
&\geq \frac{1}{W(S)W(T)}\left[\sum_{(i,j)\in H_{S,T}} \frac{(w_i+w_j)}{2w_iw_j}w_iw_j + \sum_{i\in S\cap T} w_i\right] \\
&= \frac{1}{W(S)W(T)}\left[\frac{1}{2}\sum_{(i,j)\in H_{S,T}} (w_i+w_j) + \sum_{i\in S\cap T} w_i\right] \\
&\geq \frac{1}{2W(S)W(T)}\left[W(S\setminus T) + W(T\setminus S) + 2W(S\cap T)\right] \\
&= \frac{1}{2W(S)} + \frac{1}{2W(T)},
\end{aligned}$$

with equality if and only if (a) condition (5) is satisfied; and (b) the claimed matching property holds, due to Lemma 10(c) and strict positivity of the weight vector $\mathbf{w}$. $\square$

Proof of Theorem 3: For any $\alpha \in [0,1]$ we have

$$\begin{aligned}
&f(\alpha\mathbf{x}^S(\mathbf{w}) + (1-\alpha)\mathbf{x}^T(\mathbf{w})) - \left[\alpha f(\mathbf{x}^S(\mathbf{w})) + (1-\alpha) f(\mathbf{x}^T(\mathbf{w}))\right] \\
&\quad = \left[\alpha\mathbf{x}^S(\mathbf{w}) + (1-\alpha)\mathbf{x}^T(\mathbf{w})\right]' B \left[\alpha\mathbf{x}^S(\mathbf{w}) + (1-\alpha)\mathbf{x}^T(\mathbf{w})\right] - \left[\tfrac{\alpha}{W(S)} + \tfrac{(1-\alpha)}{W(T)}\right] \\
&\quad = \alpha^2 f(\mathbf{x}^S(\mathbf{w})) + (1-\alpha)^2 f(\mathbf{x}^T(\mathbf{w})) + 2\alpha(1-\alpha)(\mathbf{x}^S(\mathbf{w}))' B\mathbf{x}^T(\mathbf{w}) - \left[\tfrac{\alpha}{W(S)} + \tfrac{(1-\alpha)}{W(T)}\right] \\
&\quad = \alpha(1-\alpha)\left[2(\mathbf{x}^S(\mathbf{w}))' B\mathbf{x}^T(\mathbf{w}) - \tfrac{1}{W(S)} - \tfrac{1}{W(T)}\right] \\
&\quad \geq 0
\end{aligned}$$

with equality for some α with $0 < \alpha < 1$ if and only if $[\mathbf{x}^S(\mathbf{w})]' B\mathbf{x}^T(\mathbf{w}) = \frac{1}{2W(S)} + \frac{1}{2W(T)}$. Hence the result follows from Lemma 11. $\square$

Proof of Theorem 5: Let $\alpha_1, \ldots, \alpha_q \geq 0$ subject to $\sum_h \alpha_h = 1$, and $B \in \mathcal{M}(\mathbf{w}, G)$. We have:

$$\begin{aligned}
f\left(\sum_{h=1}^q \alpha_h \mathbf{x}_h^S(\mathbf{w})\right) - \sum_{h=1}^q \alpha_h f(\mathbf{x}_h^S(\mathbf{w})) &= \left(\sum_{h=1}^q \alpha_h \mathbf{x}_h^S(\mathbf{w})\right)' B \left(\sum_{h=1}^q \alpha_h \mathbf{x}_h^S(\mathbf{w})\right) - \sum_{h=1}^q \frac{\alpha_h}{W(S_h)} \\
&= \sum_{h,k=1}^q \alpha_h\alpha_k (\mathbf{x}_h^S(\mathbf{w}))' B\mathbf{x}_k^S(\mathbf{w}) - \sum_{h=1}^q \frac{\alpha_h}{W(S_h)} \\
&= \sum_{h\neq k} \alpha_h\alpha_k (\mathbf{x}_h^S(\mathbf{w}))' B\mathbf{x}_k^S(\mathbf{w}) - \sum_{h=1}^q \frac{\alpha_h(1-\alpha_h)}{W(S_h)} \\
&= \sum_{h\neq k} \alpha_h\alpha_k \left[(\mathbf{x}_h^S(\mathbf{w}))' B\mathbf{x}_k^S(\mathbf{w}) - \frac{1}{W(S_h)}\right] \\
&\geq \frac{1}{2}\sum_{h\neq k} \alpha_h\alpha_k \left[\frac{1}{W(S_k)} - \frac{1}{W(S_h)}\right] \\
&= 0
\end{aligned}$$

with equality if and only if $2(\mathbf{x}_h^S(\mathbf{w}))' B\mathbf{x}_k^S(\mathbf{w}) = \frac{1}{W(S_h)} + \frac{1}{W(S_k)}$ for all $h, k = 1 \ldots q$, and the result follows from Lemma 11. $\square$

Appendix B: Proofs of theorems in Section 2.4

In its essence, the following inequality is used already by [32], but for $\mathcal{M}(\mathbf{w}, G)$ rather than for $\mathcal{C}(\mathbf{w}, G)$. Here, a simple proof is provided for both cases.

Proposition 12 *If S is a clique and $\mathbf{y} \in \Delta_S$ then*

$$2\,\mathbf{y}' C\mathbf{y} = \mathbf{y}' B\mathbf{y} \geq 1/W(S)$$

for all $B \in \mathcal{M}(\mathbf{w}, G)$ and all $C \in \mathcal{C}(\mathbf{w}, G)$. Equality is obtained if and only if $\mathbf{y} = \mathbf{x}^S(\mathbf{w})$.

Proof: Denote by $\langle ., . \rangle_{\mathbf{w},S}$ the Shahshahani inner product, cf. [36]:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{w},S} = \sum_{i \in S} \frac{x_i\, y_i}{w_i}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,$$

and by $\|\mathbf{x}\|_{\mathbf{w},S} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{w},S}}$ the corresponding pseudo-norm. Then $\langle \mathbf{x}^S(\mathbf{w}), \mathbf{y} \rangle_{\mathbf{w},S} = 1/W(S)$ for any $\mathbf{y} \in \Delta_S$ and hence also $\|\mathbf{x}^S(\mathbf{w})\|^2_{\mathbf{w},S} = 1/W(S)$. By the inequality of Cauchy, Schwarz, and Bunyakovsky,

$$\sum_{i \in S} \frac{y_i^2}{w_i} = \|\mathbf{y}\|^2_{\mathbf{w},S} \;\geq\; \langle \mathbf{x}^S(\mathbf{w}), \mathbf{y} \rangle^2_{\mathbf{w},S}\, \|\mathbf{x}^S(\mathbf{w})\|^{-2}_{\mathbf{w},S} = 1/W(S)$$

with equality if and only if $\mathbf{y} = \mathbf{x}^S(\mathbf{w})$ (recall that $\mathbf{y} \in \Delta_S$ always). On the other hand, if S is a clique and $\mathbf{y} \in \Delta_S$, then

$$2\,\mathbf{y}'C\mathbf{y} = \mathbf{y}'B\mathbf{y} = \sum_{i \in S} \frac{y_i^2}{2w_i}$$

for any $B \in \mathcal{M}(\mathbf{w}, G)$ and any $C \in \mathcal{C}(\mathbf{w}, G)$, which shows the assertion. $\square$

Proof of Theorem 7: Suppose that $\mathbf{x} = \mathbf{x}^S(\mathbf{w})$ where S is a maximal clique. Then for all $r \in V \setminus S$ we have $(i, r) \notin E$ for at least one $i \in S$ and hence $(C\mathbf{x})_r \geq (1 + w_i/w_r)/[2W(S)] > 1/[2W(S)]$, entailing for $A = \mathbf{e}\mathbf{e}' - C$

$$(A\mathbf{x})_r = 1 - (C\mathbf{x})_r < 1 - 1/[2W(S)] = 1 - (C\mathbf{x})_j = (A\mathbf{x})_j = \mathbf{x}'A\mathbf{x}$$

for all $j \in S$ (the last equality follows by $\mathbf{x} \in \Delta_S$). Now suppose that $\mathbf{y} \in \Delta$ satisfies $\mathbf{y}'A\mathbf{x} = \mathbf{x}'A\mathbf{x}$. Then the strict inequalities above entail $y_r = 0$ for all $r \notin S$, i.e. $\mathbf{y} \in \Delta_S$. Hence by Proposition 12 and the above arguments,

$$\mathbf{x}'A\mathbf{y} = \mathbf{y}'A\mathbf{x} = \sum_{j \in S} y_j(\mathbf{x}'A\mathbf{x}) = \mathbf{x}'A\mathbf{x} = 1 - \frac{1}{2W(S)} \leq 1 - \mathbf{y}'C\mathbf{y} = \mathbf{y}'(\mathbf{e}\mathbf{e}' - C)\mathbf{y} = \mathbf{y}'A\mathbf{y}$$

with equality only if $\mathbf{y} = \mathbf{x}^S(\mathbf{w}) = \mathbf{x}$, which establishes the so-called ESS property of $\mathbf{x}$ for A. Theorem 2 in [21] now yields strict local optimality of $\mathbf{x}$. Next suppose that $\mathbf{x}$ is a local solution to (3). Let $S = \{i \in V : x_i > 0\}$ be the support of $\mathbf{x}$ and assume that S is no clique, which means that there are $i, j \in S$ such that $i \neq j$ and $(i, j) \notin E$, which by definition of $\mathcal{C}(\mathbf{w}, G)$ entails the strict inequality

$$c_{ij} + c_{ji} \geq 2[c_{ii} + c_{jj}] > c_{ii} + c_{jj} \, .$$

Now designate by $\mathbf{v}$ the vector with coordinates

$$v_k = \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } k = j \\ 0 & \text{otherwise} \end{cases}$$

put $\mathbf{x}(\alpha) = \mathbf{x} + \alpha \mathbf{v}$ for $\alpha \in \mathbb{R}$, and denote by s the straight line segment $s = \{\mathbf{x}(\alpha) : \alpha \in \mathbb{R}\}$. Then the objective function $\mathbf{x}'C\mathbf{x}$ is strictly concave along s, since

$$[\mathbf{x}(\alpha)]'C[\mathbf{x}(\alpha)] = \mathbf{x}'C\mathbf{x} + 2\alpha \, \mathbf{x}'C\mathbf{v} + \alpha^2 [c_{ii} + c_{jj} - c_{ij} - c_{ji}] \, .$$

On the other hand, the segment s passes the relative interior Δ_S° in $\mathbf{x} = \mathbf{x}(\alpha = 0)$ and therefore its intersection with Δ is a segment containing $\mathbf{x}$ in its relative interior, which is absurd in view of local optimality of $\mathbf{x}$ for (3) and strict concavity. Hence S has to be a clique. Now join $\mathbf{x}$ with $\mathbf{x}^S(\mathbf{w})$ by a straight line and repeat the preceding arguments. Then Proposition 12 yields $\mathbf{x} = \mathbf{x}^S(\mathbf{w})$. Finally, S has to be a maximal clique. Indeed, suppose that T is a larger maximal clique containing S. By positivity of weights, we have $W(T) > W(S)$. Proposition 12 shows

$$[\mathbf{x}^S(\mathbf{w})]'C\mathbf{x}^S(\mathbf{w}) = 1/[2W(S)] > 1/[2W(T)] = [\mathbf{x}^T(\mathbf{w})]'C\mathbf{x}^T(\mathbf{w}) \, .$$

Now join $\mathbf{x}^S(\mathbf{w})$ and $\mathbf{x}^T(\mathbf{w})$ with a straight line, which intersects Δ_T also in points of the form $(1-\alpha)\mathbf{x}^T(\mathbf{w}) + \alpha\mathbf{x}^S(\mathbf{w})$ with $\alpha < 0$ close to zero. Quadraticity of the objective along this line; local optimality of $\mathbf{x} = \mathbf{x}^S(\mathbf{w})$; and optimality of $\mathbf{x}^T(\mathbf{w})$ on the entire face Δ_T yields a contradiction. This establishes property (a), and the remaining assertions follow as easy consequences. $\square$

References

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, J. Wiley & Sons, Chichester, UK, 1989.

[2] F. Alizadeh, "A sublinear-time randomized parallel algorithm for the maximum clique problem in perfect graphs," in *ACM-SIAM Symposium on Discrete Algorithms*, vol. 2, pp. 188–194, 1991.

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

[4] S. Arora and S. Safra, "Probabilistic checking of proofs: A new characterization of NP," in *Proc. 33rd Ann. Symp. Found. Comput. Sci.*, Pittsburgh, PA, 1992, pp. 2–13.

[5] G. Ausiello, P. Crescenzi, and M. Protasi, "Approximate solutions of NP optimization problems," *Theoret. Comput. Sci.*, vol. 150, pp. 1–55, 1995.

[6] L. Babel, "A fast algorithm for the maximum weight clique problem," *Computing*, vol. 52, no. 1, pp. 31–38, 1994.

[7] E. Balas, V. Chvátal, and J. Nesetril, "On the maximum weight clique problem," *Math. Oper. Res.*, vol. 12, no. 3, pp. 522–535, 1987.

[8] E. Balas and J. Xue, "Minimum weighted coloring of triangulated graphs, with application to maximum weight vertex packing and clique finding in arbitrary graphs," *SIAM J. Comput.*, vol. 20, no. 2, pp. 209–221, 1991. Addendum, vol. 21, no. 5, p. 1000, 1992.

[9] E. Balas and J. Xue, "Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring," *Algorithmica*, vol. 15, no. 5, pp. 397–412, 1996.

[10] E. Balas and C. S. Yu, "On graphs with polynomially solvable maximum-weight clique problem," *Networks*, vol. 19, pp. 247–253, 1989.

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

[12] D. H. Ballard, P. C. Gardner, and M. A. Srinivas, "Graph problems and connectionist architectures," *Technical Report TR 167*, Dept. Computer Science, University of Rochester, 1987.

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

[14] L. E. Baum and G. R. Sell, "Growth transformations for functions on manifolds," *Pacif. J. Math.*, vol. 27, no. 2, pp. 211-227, 1968.

[15] M. Bellare, S. Goldwasser, C. Lund, and A. Russell, "Efficient probabilistically checkable proofs and application to approximation," in *Proc. 25th Ann. ACM Symp. Theory Comput.*, San Diego, CA, 1993, pp. 294–304.

[16] M. Bellare, S. Goldwasser, and M. Sudan, "Free bits, PCPs and non-approximability—Towards tight results," in *Proc. 36th Ann. Symp. Found. Comput. Sci.*, Milwaukee, WI, 1995, pp. 422–431.

[17] A. Bertoni, P. Campadelli and G. Grossi, "A discrete neural algorithm for the maximum clique problem: Analysis and circuit implementation," presented at *WAE'97: Int. Workshop on Algorithm Engineering*, Venice, Italy, 1997.

[18] I. M. Bomze, "Evolution towards the maximum clique," *J. Global Optim.*, vol. 10, pp. 143–164, 1997.

[19] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization (Suppl. Vol. A)*, D.-Z. Du and P. M. Pardalos (Eds.), Kluwer Academic Publishers, Boston, MA, 1999, pp. 1–74.

[20] I. M. Bomze, M. Budinich, M. Pelillo, and C. Rossi, "A new 'annealed' heuristic for the maximum clique problem," in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, P. M. Pardalos (Ed.), Kluwer Academic Publishers, Boston, MA, 2000.

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

[22] I. M. Bomze and F. Rendl, "Replicator dynamics for the evolution towards the maximum clique: Variants and experiments", in *High Performance Algorithms and Software in Nonlinear Optimization*, R. De Leone, A. Murlí, P.M. Pardalos and G. Toraldo (eds.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 53–67.

[23] I. M. Bomze and V. Stix, "Genetic engineering via negative fitness: Evolutionary dynamics for global optimization," *Ann. Oper. Res.*, vol. 89, pp. 297–318, 1999.

[24] M. Budinich, "Properties of the adjacency matrix and the maximum clique problem," presented at *MAX-CLIQUE'97—A Workshop on the Maximum Clique Problem and Its Applications*, Trieste, Italy, June 1997.

[25] R. Carraghan and P. M. Pardalos, "A parallel algorithm for the maximum weight clique problem," *Tech. Rep. CS-90-40*, Department of Computer Science, Pennsylvania State University, 1990.

[26] J. F. Crow and M. Kimura, *An Introduction to Population Genetics Theory.* Harper & Row Publishers, New York, 1970.

[27] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Approximating clique is almost NP-complete," in *Proc. 32nd Ann. Symp. Found. Comput. Sci.*, San Juan, Puerto Rico, 1991, pp. 2–12.

[28] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Interactive proofs and the hardness of approximating clique," *J. ACM*, vol. 43, no. 2, pp. 268–292, 1996.

[29] R. A. Fisher, *The Genetical Theory of Natural Selection.* Clarendon Press, Oxford, 1930.

[30] N. Funabiki, Y. Takefuji, and K. C.Lee, "A neural network model for finding a near-maximum clique," *J. Parallel Distrib. Comput.*, vol. 14, pp. 340–344, 1992.

[31] L. E. Gibbons, D. W. Hearn, and P. M. Pardalos, "A continuous based heuristic for the maximum clique problem," in [41], pp. 103–124.

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

[33] T. Grossman, "Applying the INN model to the MaxClique problem," in [41], pp. 125–146.

[34] J. Hastad, "Clique is hard to approximate within $n^{1-\varepsilon}$," in *Proc. 37th Ann. Symp. Found. Comput. Sci.*, Burlington, VE, 1996, pp. 627–636.

[35] M. Hifi, "A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems," *J. Oper. Res. Soc.*, vol. 48, pp. 612–622, 1997.

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

[37] R. Horaud and T. Skordas, "Stereo correspondence through feature grouping and maximal cliques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 11, pp. 1168–1180, 1989.

[38] A. Jagota, "Approximating maximum clique with a Hopfield neural network," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 724–735, 1995.

[39] A. Jagota, M. Pelillo, and A. Rangarajan, "A new deterministic annealing algorithm for maximum clique," in *Proc. IJCNN'2000—Int. J. Conf. Neural Networks*, Como, Italy, July 2000.

[40] A. Jagota, L. Sanchis, and R. Ganesan, "Approximately solving maximum clique using neural networks and related heuristics," in [41], pp. 169–204.

[41] D. S. Johnson and M. A. Trick (Eds.), *Cliques, Coloring, and Satisfiability—Second DIMACS Implementation Challenge*, American Mathematical Society, 1996. (see also `http://dimacs.rutgers.edu/Volumes/Vol26.html`).

[42] R. M. Karp, "Reducibility among combinatorial problems," In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher (Eds.), Plenum Press, New York, 1972, pp. 85–103.

[43] M. Kimura, "On the change of population fitness by natural selection," *Heredity* vol. 12, pp. 145–167, 1958.

[44] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1035–1074, 1983.

[45] F. Lin and K. Lee, "A parallel computation network for the maximum clique problem," in *Proc. 1st Int. Conf. Fuzzy Theory Tech.*, Baton Rouge, Louisiana, 1992.

[46] E. Loukakis and C. Tsouros, “An algorithm for the maximally internally stable set in a weighted graph,” *Int. J. Computer Math.*, vol. 13, pp. 117–129, 1983.

[47] Yu. Lyubich, G.D. Maistrowskii, and Yu.G. Ol'khovskii, “Selection-induced convergence to equilibrium in a single-locus autosomal population,” *Problems of Information Transmission* vol. 16, 66–75, 1980.

[48] C. Mannino and E. Stefanutti, “An augmentation algorithm for the maximum weighted stable set problem,” *Comput. Optim. Appl.*, vol. 14, no. 3, pp. 367–381, 1999.

[49] T. S. Motzkin and E. G. Straus, “Maxima for graphs and a new proof of a theorem of Turán,” *Canad. J. Math.*, vol. 17, no. 4, pp. 533–540, 1965.

[50] G. L. Nemhauser and G. Sigismondi, “A strong cutting plane/branch-and-bound algorithm for node packing,” *J. Opl. Res. Soc.*, vol. 43, no. 5, pp. 443–457, 1992.

[51] G. L. Nemhauser and L. E. Trotter, “Vertex packings: Structural properties and algorithms,” *Math. Programming*, vol. 8, pp. 232–248, 1975.

[52] C. H. Papadimitriou and K. J. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[53] P. M. Pardalos, “Continuous approaches to discrete optimization problems,” In *Nonlinear Optimization and Applications*, G. Di Pillo and F. Giannessi (Eds.), Plenum Press, 1996, pp. 313–328.

[54] P. M. Pardalos and N. Desai, “An algorithm for finding a maximum weighted independent set in an arbitrary graph,” *Int. J. Computer Math.*, vol. 38, pp. 163–175, 1991.

[55] P. M. Pardalos and A. T. Phillips, “A global optimization approach for solving the maximum clique problem,” *Int. J. Computer Math.*, vol. 33, pp. 209–216, 1990.

[56] M. Pelillo, “Relaxation labeling networks for the maximum clique problem,” *J. Artif. Neural Networks*, vol. 2, no. 4, pp. 313–328, 1995.

[57] M. Pelillo, “The dynamics of relaxation labeling processes,” *J. Math. Imaging Vision*, vol. 7, no. 4, pp. 309–323, 1997.

[58] M. Pelillo, "Replicator equations, maximal cliques, and graph isomorphism," *Neural Computation*, vol. 11, no. 8, pp. 1933–1955, 1999.

[59] M. Pelillo and A. Jagota, "Feasible and infeasible maxima in a quadratic program for maximum clique," *J. Artif. Neural Networks*, vol. 2, no. 4, pp. 411–420, 1995.

[60] M. Pelillo, K. Siddiqi, and S. W. Zucker, "Matching hierarchical structures using association graphs," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 11, pp. 1105–1120, 1999.

[61] J. Ramanujam and P. Sadayappan, "Optimization by neural networks," in *Proc. IEEE Int. Conf. Neural Networks*, 1988, pp. 325–332.

[62] Y. Shrivastava, S. Dasgupta, and S. M. Reddy, "Guaranteed convergence in a class of Hopfield networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 951–961, 1992.

[63] V. Stix, *The Maximum Clique Problem—A Fresh Approach: Empirical Evidence and Implementation in C++.* Thesis, University of Vienna, 1997. Also available as I.S.O.C. Technical Report TR-97-05.

[64] Y. Takefuji, L. Chen, K. Lee, and J. Huffman, "Parallel algorithms for finding a near-maximum independent set of a circle graph," *IEEE Trans. Neural Networks*, vol. 1, pp. 263–267, 1990.

[65] P. Taylor and L. Jonker, "Evolutionarily stable strategies and game dynamics," *Math. Biosci.*, vol. 40, pp. 145–156, 1978.

[66] A. Torsello and M. Pelillo, "Continuous-time relaxation labeling processes," *Pattern Recognition*, vol. 33, no. 11, pp. 1897–1908, 2000.

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

Figure captions

Figure 1. The simplex Δ in $\mathbb{R}^3$.

Table captions

Table 1. Performance of RD on 20-vertex type I random graphs.

Table 2. RD and Babel on 100-vertex type I random graphs.

Table 3. RD and Babel on 100-vertex type II random graphs.

Table 4. RD and Babel on weighted DIMACS graphs.

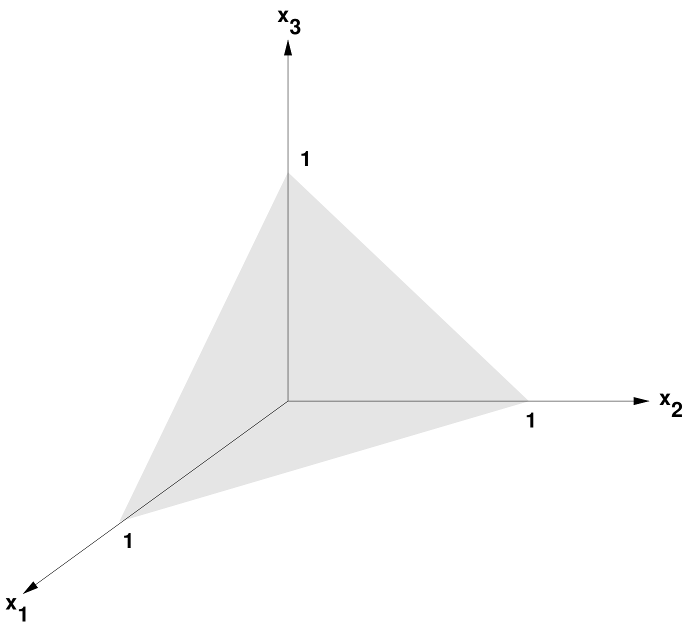

Figure 1: The simplex Δ in $\mathbb{R}^3$.

Table 1: Performance of RD on 20-vertex type I random graphs.

	sum of weights			size of clique		
density	RD	MWC	quality	RD	MWC	difference
0.10	16.745	18.955	88%	2.230	2.330	-0.100
0.50	33.105	35.580	93%	4.755	5.025	-0.270
0.75	51.320	53.240	96%	7.810	7.950	-0.140
0.90	73.695	74.200	99%	11.885	11.870	0.015

Table 2: RD and Babel on 100-vertex type I random graphs.

	sum of weights			size of clique		
density	RD	Babel	quality	RD	Babel	difference
0.10	21.6	29	74%	2.9	2.4	0.5
0.50	52.8	65.6	80%	7.1	7	0.1
0.75	97.3	113.2	86%	13.7	13.2	0.5
0.90	180.9	194	93%	26.8	25.7	1.1

Table 3: RD and Babel on 100-vertex type II random graphs.

	sum of weights			size of clique		
density	RD	Babel	quality	RD	Babel	difference
0.10	45.1	46.3	97%	6.7	5.9	0.8
0.50	158.7	161.6	98%	26.6	25.6	1.0
0.75	268.5	270.7	99%	46.6	45.6	1.0
0.90	378.8	380.2	100%	66.6	65.7	0.9

Table 4: RD and Babel on weighted DIMACS graphs.

graph			sum of weights		
name	order	density	RD	Babel	quality
c-fat500-1	500	0.036	93	96*	97%
c-fat500-10	500	0.374	755	755*	100%
c-fat500-2	500	0.073	181	181*	100%
c-fat500-5	500	0.186	371	395*	94%
hamming10-2	1024	0.990	2674	1923	139%
hamming10-4	1024	0.829	258	277	93%
san1000	1000	0.501	80	94	85%
brock800_1	800	0.649	136	156	87%
brock800_2	800	0.651	142	157	90%
brock800_3	800	0.649	141	148	95%
brock800_4	800	0.650	136	147	93%
p_hat500-1	500	0.253	56	72*	78%
p_hat500-2	500	0.505	213	228*	93%
p_hat500-3	500	0.752	307	294	104%
p_hat700-1	700	0.249	54	76*	71%
p_hat700-2	700	0.498	275	272	101%
p_hat700-3	700	0.748	371	316	117%
p_hat1000-1	1000	0.245	77	80*	96%
p_hat1000-2	1000	0.490	285	284	100%
p_hat1000-3	1000	0.744	435	349	125%
p_hat1500-1	1500	0.253	74	87*	85%
p_hat1500-2	1500	0.506	327	309	106%
p_hat1500-3	1500	0.754	490	390	126%
MANN_a45	1035	0.996	2643	2653*	100%