ELSEVIER

Discrete Applied Mathematics 121 (2002) 27–49

DISCRETE APPLIED MATHEMATICS

Annealed replication: a new heuristic for the maximum clique problem

Immanuel M. Bomze[a,*], Marco Budinich[b], Marcello Pelillo[c], Claudio Rossi[c]

[a] *Institut für Statistik & Decision Support Systems, Universität Wien, Universitätsstraße 5, A-1010 Wien, Austria*

[b] *Dipartimento di Fisica, Università di Trieste, Via Valerio 2, I-34127 Trieste, Italy*

[c] *Dipartimento di Informatica, Università Ca' Foscari di Venezia, Via Torino 155, I-30172 Venezia Mestre, Italy*

Received 10 April 1998; received in revised form 9 April 2001; accepted 8 May 2001

Abstract

We propose a new heuristic for approximating the maximum clique problem based on a detailed analysis of a class of continuous optimization models which provide a complete characterization of solutions to this NP-hard combinatorial problem. We start from a known continuous formulation of the maximum clique, and tackle the search for local solutions with replicator dynamics, a class of dynamical systems developed in various branches of mathematical biology. Hereby, we add to the objective used in previous works a regularization term that controls the global shape of the energy landscape, that is the function actually maximized by the dynamics. The parameter controlling the regularization is changed during the evolution of the dynamical system to render inefficient local solutions (which formerly were stable) unstable, thus conducting the system to escape from sub-optimal points, and so to improve the final results. The role of this parameter is thus superficially similar to that of temperature in simulated annealing in the sense that its variation allows to find better solutions for the problem at hand. We demonstrate several theoretical results on the regularization term and we further support the validity of this approach, reporting on its performances when applied to selected DIMACS benchmark graphs.

Keywords: Quadratic optimization; Dynamical systems; Replicator dynamics; Stable set; Independent set; Combinatorial optimization

1. Introduction

The maximum clique problem (MCP) is a well-known problem in combinatorial optimization which finds important applications in many different domains [11]. Since

* Corresponding author.

E-mail addresses: immanuel.bomze@univie.ac.at (I.M. Bomze), mbh@trieste.infn.it (M. Budinich), pelillo@dsi.unive.it (M. Pelillo), rossi@dsi.unive.it (C. Rossi).

0166-218X/02/$ - see front matter
PII: S0166-218X(01)00233-5

the MCP is known to be NP-hard, exact algorithms are guaranteed to return a solution only in a time which increases exponentially with the number of vertices in the graph. 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. Strong evidence of this fact came in 1991, when Feige et al. [17] (see also [18]) 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. [1,2] 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 [3,4,23]. In 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. We refer to [25] for a collection of promising heuristics for the MCP.

We have recently investigated the effectiveness of an approach for approximating the MCP, centered around a continuous formulation due to Motzkin and Straus [33] and its regularization [24,10], which exploits the dynamical properties of the so-called *replicator equations*, a class of dynamical systems developed and studied in various branches of mathematical biology. One problem associated with these models, however, is their inability to escape inefficient local solutions. In this paper, we introduce a class of parametrized quadratic programs, which includes both the Motzkin–Straus program and its regularization as special cases, and investigate the properties of its solutions as a function of its parameter. A detailed analysis of these properties suggests a new algorithm for approximating the MCP which is based on the idea of properly varying the parameter during the replicator optimization process, in much the same spirit as simulated annealing procedures. A related, but different, idea has recently been proposed by Gee and Prager in the neural network domain [20]. Experimental results conducted on various DIMACS benchmark graphs demonstrate the validity of the proposed approach.

The outline of the paper is as follows. In Section 2, we describe the Motzkin–Straus theorem and its parameterization, and present the replicator dynamical systems. These dynamics are used to obtain locally optimal solutions to the MCP. Section 3 is devoted to a few results that enable us to establish bounds on a regularization parameter α which governs stability under the replicator dynamics. For illustration, we investigate in Section 4 a small, but prototypical example in detail. In a more detailed dynamical analysis exceeding the usual perturbation theory approach, we specify explicit ranges within which qualitative features of the dynamics are invariant, and also obtain quantitative sensitivity results for the related optimization problems. This analysis is deferred to an appendix, to promote the flow of the argument. The previously established theoretical properties will lead us to develop in Section 5 an algorithm for properly updating the parameter α with the objective of avoiding poor local solutions. In Section 6 the results of our experiments are presented, and Section 7 concludes the paper.

2. Evolution towards the maximum clique

2.1. A parametrized continuous formulation of the MCP

Let $G=(V,E)$ be an undirected graph, where $V=\{1,\dots,n\}$ is the set of vertices, and $E\subseteq V\times V$ is the set of edges. A *clique* of G is a subset of V in which every pair of vertices is connected by an edge. A clique is called *maximal* if no strict superset of C is a clique, i.e., no vertex external to C is connected with more than $|C|-1$ vertices of C (here, and in the sequel, $|C|$ denotes the cardinality of a set C). A maximal clique C is called *strictly maximal* if no vertex i external to C has the property that the enlarged set $C\cup\{i\}$ contains a clique of the same size as C. In other words, if

$$d_C(i)=|\{j\in C\colon (i,j)\in E\}|$$

denotes the *degree* of i w.r.t. C, then a maximal clique C is strictly maximal if and only if $d_C(i)<|C|-1$ for all $i\notin C$.

A *maximum* clique is a clique having largest cardinality (note that a maximal clique is not necessarily a maximum one). Hence, the MCP consists of finding a clique of maximum size in a graph G. For a recent survey see [11]. In the following, given a set S of vertices in G, we will denote by x^S its characteristic vector, defined as $x^S_i=1/|S|$ if $i\in S$ and $x^S_i=0$ otherwise.

Given a graph G, consider the following quadratic program introduced in [24,10] (x' always denotes the transpose of a column vector x):

$$\begin{aligned}&\text{maximize } x'(A_G+\tfrac{1}{2}I)x\\&\text{subject to } x\in S^n,\end{aligned}\tag{1}$$

where $A_G=(a_{ij})$ is the adjacency matrix of G (i.e., $a_{ij}=1$ if $(i,j)\in E$, and $a_{ij}=0$ if $(i,j)\notin E$), S^n is the standard simplex of $\mathbb{R}^n$, that is

$$S^n=\left\{x\in\mathbb{R}^n\colon x_i\geqslant 0 \text{ for all } i=1,\dots,n \text{ and } \sum_{i=1}^{n}x_i=1\right\}$$

and I is the $n\times n$ identity matrix. This turns out to be a variant of the so-called Motzkin–Straus program [33], which is obtained from (1) by simply dropping the $\frac{1}{2}I$ term. For completeness, we summarize here the original Motzkin–Straus theorem and some recent related results. [1]

Theorem 1. *Let C be a subset of vertices of a graph G, and let x^C be its characteristic vector. Then $(x^C)'A_G(x^C)=1-1/|C|$ if and only if C is a clique. Moreover*:

(a) *x^C is a strict local maximizer of $x'A_Gx$ over S^n if and only if C is a strictly maximal clique.*

(b) *x^C is a global maximizer of $x'A_Gx$ over S^n if and only if C is a maximum clique.*

[1] The original Motzkin–Straus theorem [33] corresponds to the "if" part of Theorem 1(b), while the "only–if" part has been proven in [39]. Part (a) of the theorem if from [39,22].

An immediate consequence of the previous result is that *any* point in S^n provides us with a bound on the size of the maximum clique in G.[2] In fact, if C is a maximum clique of G, for any $x \in S^n$ we have $x'A_G x \leqslant 1 - 1/|C|$, from which it follows that $|C| \geqslant \lceil 1/(1 - x'A_G x)\rceil$.

The Motzkin–Straus theorem has an intriguing computational significance. It suggests a fundamentally new way of solving the maximum clique problem, by allowing us to shift from the discrete to the continuous domain. A pointed out in [35], the advantages of such a reformulation are manifold. It not only allows 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. Additionally, continuous optimization methods are sometimes described in terms of sets of differential equations, and are therefore potentially implementable in analog circuitry. The Motzkin–Straus and related theorems have served as the basis of many clique-finding procedures [36,37,21,11], and have also been used to determine theoretical bounds on the maximum clique size [15].

In contrast to the original Motzkin–Straus formulation, however, its regularization (1) has a further merit: as observed by Pardalos and Phillips [36] and later formalized by Pelillo and Jagota [39], the Motzkin–Straus program, in its original formulation, is plagued by the presence of "spurious" solutions, i.e., solutions which are not in the form of characteristic vectors. Clearly, this represents a problem since it prohibits direct extraction of the vertices comprising the clique, and provides information only on its *size*. Therefore, in order to determine the clique vertices, one has to make recourse to iterative or recursive procedure, as those described in [34,36].

The significance of the following result, a sharpening of Theorem 1 proved in [10], is that a local (and hence also a global) maximum of (1) can only be attained at a characteristic vector $x^* = x^C$ for some subset C of vertices which necessarily then forms a maximal clique. This solves the spurious solution problem in a straightforward and definitive manner since it establishes a one-to-one correspondence between local/global solutions to (1) and maximal/maximum cliques of G, respectively.

Theorem 2. *Let G be a graph and consider problem* (1). *Then the following assertions are equivalent*:

(a) $x = x^C$, *where C is a maximal clique of size $k = |C|$*;

(b) x *is a strict local solution to* (1);

(c) x *is a local solution to* (1).

If one of the above conditions (*and therefore all*) *is met, the objective is* $x'(A_G + \frac{1}{2}I)x = 1 - 1/2k$. *Hence C is a maximum clique of G if and only if x^C is a global solution to* (1).

[2] We thank Arun Jagota for pointing this out.

In this paper, we consider the following program, which represents also a regularization of the Motzkin–Straus program and generalizes (1):

$$\begin{aligned} &\text{maximize } f_\alpha(x) = x'(A_G + \alpha I)x \\ &\text{subject to } x \in S^n. \end{aligned} \tag{2}$$

This includes both the Motzkin–Straus ($\alpha = 0$) program and its regularization ($\alpha = \frac{1}{2}$) as special cases. We investigate the properties of its solutions as a function of the parameter α. Specifically, we show that when $\alpha \in]0,1[$ all the properties of program (1) hold true. For negative α, on the other hand, the landscape of $f_\alpha(x)$ changes and "flat regions" can merge in an extremum while other extrema can disappear, depending on the values of the parameter α. A detailed analysis of these effects will suggest a new algorithm for approximating the MCP which is based on the idea of varying the parameter α during an evolutionary optimization process, in such a way as to avoid obtaining characteristic vectors of small cliques.

We point out that the proposed parameterization of the Motzkin–Straus program is completely different, both in content and motivations, from that recently introduced by Gibbons et al. [21]. Their idea was to substitute the sign constraints $x \geqslant 0$ of the Motzkin–Straus program with one of the form $x'x = 1/s$, s being a parameter in the interval $[1, n]$, in an attempt to avoid spurious solutions. With this program it may happen that the solutions have to be projected onto the positive orthant, in order to maintain feasibility.

2.2. Replicator equations and their application to the MCP

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

$$\dot{x}_i(t) = x_i(t)[(Mx(t))_i - x(t)'Mx(t)], \quad i = 1, \ldots, n, \tag{3}$$

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

$$x_i(t+1) = x_i(t)\frac{(Mx(t))_i}{x(t)'Mx(t)}, \quad i = 1, \ldots, n. \tag{4}$$

It is readily seen that the simplex S^n is invariant under these dynamics, which means that every trajectory starting in S^n will remain in S^n for all future times. Moreover, it turns out that their *stationary points*, i.e. the points satisfying $\dot{x}_i(t) = 0$ for (3) or $x_i(t+1) = x_i(t)$ for (4), coincide and are the solutions of the equations

$$x_i[(Mx)_i - x'Mx] = 0, \quad i = 1, \ldots, n. \tag{5}$$

A stationary point x is said to be *asymptotically stable* if every solution to (3) or (4) which starts close enough to x, will converge to x as $t \to \infty$.

Both (3) and (4) 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. Eq. (3) has been introduced in evolutionary game theory by

Taylor and Jonker [40] to model evolution of behavior in intraspecific 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 ith behavior pattern $(i=1,\ldots,n)$ is equal to the (dis)advantage $(Mx)_i - x'Mx = \sum_j m_{ij}x_j - \sum_{j,k} m_{kj}x_jx_k$, measured by incremental fitness relative to the average performance within the population in state $x=(x_1,\ldots,x_n)'$. 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". Similar arguments provide a rationale for the discrete-time version (4). 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 [5] and studied by Baum and Sell [6] in the context of Markov chain theory. This kind of processes have proven to be useful in the speech recognition [31] and computer vision [38] domains. Dynamics (3) and (4) 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 ith allele. The matrix M is in this context always symmetric, since permuted gene pairs belong to the same genotype. Models (3) and (4) as selection equations go way back to Fisher [19] and Kimura [29].

From an optimization point of view, the difference between symmetric and non-symmetric matrices M is crucial. Indeed, in the symmetric case the quadratic form $x'(t)Mx(t)$ is increasing along trajectories of the replicator dynamics; this is the Fundamental Theorem of Natural Selection, see, e.g. [16,26,24].

Theorem 3. *If $M=M'$ then the function $x(t)'Mx(t)$ is strictly increasing with increasing t along any non-stationary trajectory $x(t)$ under both continuous-time* (3) *and discrete-time* (4) *replicator dynamics. Furthermore, any such trajectory converges to a stationary point.*

Apart from the monotonicity result which provides a Lyapunov function for both dynamics, the previous theorem also rules out complicated attractors like cycles, invariant tori, or even strange attractors.

To formulate the results which relate dynamical properties to optimality, we need some further notions and notations. First, consider the general quadratic optimization problem over S^n,

$$\begin{aligned} &\text{maximize } x'Mx \\ &\text{subject to } x\in S^n \end{aligned} \tag{6}$$

and the *generalized Lagrangian*

$$L(x;\lambda,\mu)=x'Mx+\lambda'x+\mu(e'x-1)$$

of (6), where the multipliers λ_i and μ may have arbitrary sign. Call a critical point x of the generalized Lagrangian a *generalized* Karush–Kuhn–Tucker point if $L(x;\lambda,\mu)=x'Mx$ irrespective of the sign of λ_i.

Finally, we need some notions from game theory (see, e.g., [42]): recall that a point $x\in S^n$ is said to be a (*symmetric*) *Nash* (*equilibrium*) *strategy* if and only if

$$y'Mx \leqslant x'Mx \quad \text{for all } y\in S^n. \tag{7}$$

Furthermore, a Nash strategy x is said to be a *neutrally stable strategy* (NSS) if and only if

$$y'Mx = x'Mx \quad \text{implies} \quad x'My \geqslant y'My \tag{8}$$

and an *evolutionarily stable strategy* (ESS) if and only if the inequality in (8) is strict for $y\neq x$.

Now we repeat the characterization results from [12] which link three different fields: optimization theory, evolutionary game theory, and qualitative theory of dynamical systems.

Theorem 4. *Let $M=M'$ be an arbitrary symmetric $n\times n$ matrix and $x\in S^n$. Consider the following properties*:
(a1) *x is an ESS, i.e., satisfies* (8) *with strict inequality, and* (7);
(a2) *x is a strict local solution to* (6);
(a3) *x is an asymptotically stable stationary point of* (3) *and* (4);
(b1) *x is a NSS, i.e., satisfies* (7) *and* (8);
(b2) *x is a local solution of* (6);
(c1) *x is a Nash strategy, i.e., satisfies* (7);
(c2) *x is a Karush–Kuhn–Tucker point for* (6);
(d1) *x is a stationary point under* (3) or (4), *i.e., satisfies* (5);
(d2) *x is a generalized Karush–Kuhn–Tucker point for* (6).
Then the following implications and equivalences hold true: (a1) $\Leftrightarrow$ (a2) $\Leftrightarrow$ (a3) $\Rightarrow$ (b1) $\Leftrightarrow$ (b2) $\Rightarrow$ (c1) $\Leftrightarrow$ (c2) $\Rightarrow$ (d1) $\Leftrightarrow$ (d2).

The previous result naturally suggests the use of replicator equations for approximating the MCP. In fact, let A_G be the (symmetric) adjacency matrix of graph G; by putting $M=A_G+\frac{1}{2}I$, the replicator dynamical system will iteratively maximize the objective function of (1) and eventually converge (with probability 1) to a local maximizer, which by virtue of Theorem 2, will then correspond to a characteristic vector of a maximal clique of G. One can also put $M=A_G$, in which case we obtain the Motzkin–Straus program, but due to the presence of spurious maximizers, these solutions can only provide an approximation of the size of the maximum clique. The empirical results obtained in [12] over numerous DIMACS benchmark graphs are encouraging and prove the effectiveness of this algorithm. They also show that the approach based on the original (non-regularized) version of the Motzkin–Straus problem performs slightly better than its regularized counterpart (1), in terms of clique size.

This may be intuitively explained by observing that, since all local maxima are strict, the landscape of the new objective function (1) is certainly less flat than the one associated to the non-regularized version and thus a dynamics that increases the objective function at every step will be more prone to end up in a close local maximum.

Finally, let us note that recent empirical investigations [13] indicate that there is no significant gain in varying the starting point of the replicator dynamics by intricate preprocessing, or using a discretization of (3) different from (4).

3. Bounds for the annealing parameter

In this section, we establish bounds for the annealing parameter α related to the stability of x^S under the replicator dynamics. The first results hold for general symmetric matrices; we then specialize these findings to the case of adjacency matrices.

Proposition 5. *If $x \in S^n$ is a (local) maximizer of $x'(A+\alpha I)x$ over S^n and $S = \{i \in V: x_i > 0\}$, then necessarily*

$$\alpha \geqslant \gamma(x) = \max\left\{\frac{(Ax)_i - x'Ax}{x'x}: i \notin S\right\}. \tag{9}$$

Proof. Since x is a local maximizer of $x'(A+\alpha I)x$ on S^n, then Theorem 4 implies the Nash equilibrium condition (7) which for the case $M = A + \alpha I$ entails

$$(Ax)_i + \alpha x_i = (Mx)_i \leqslant x'Mx = x'Ax + \alpha x'x \quad \text{for all } i \notin S$$

(note that equality has to hold if $i \in S$, for otherwise we would arrive at the contradiction $x'Mx = \sum_{i\in S} x_i(Mx)_i < x'Mx$). But $i \notin S$ means $x_i = 0$ so that $\alpha \geqslant \gamma(x)$ follows readily. □

We move to the following result: a local maximizer x of both $x'Ax$ and $x'(A+\alpha I)x$ over S^n necessarily has to be a characteristic vector.

Proposition 6. *If $x \in S^n$ is a (local) maximizer of both $x'Ax$ and $x'(A+\alpha I)x$ over S^n for some $\alpha \neq 0$, then necessarily $x = x^S$ if $S = \{i \in V: x_i > 0\}$.*

Proof. From Theorem 2 of [12] we know that every local maximizer has to be a stationary point under the respective replicator dynamics. Hence for all $i \in S$ we have, due to (5),

$$[Ax]_i = x'Ax \quad \text{and also } [Ax]_i + \alpha x_i = x'(A+\alpha I)x = x'Ax + \alpha x'x, \tag{10}$$

so that all positive coordinates of x have to be equal. Since they sum up to one, the result follows. □

Now we need some additional notation. First, denote by $e=(1,\dots,1)'\in\mathbb{R}^n$ and denote the $(n-1)$-dimensional hyperplane of all vectors the coordinates of which sum up to zero by

$$e^\perp=\{v\in\mathbb{R}^n\ :\ e'v=0\}.$$

Given an arbitrary $n\times n$ matrix M, the action of its quadratic form on $e^\perp$ can be fully described with the help of the *orthoprojector* $P=I-(1/n)ee'$ onto $e^\perp$: indeed, for $u\in e^\perp$ we have $Pu=u$ whence $u'Mu=u'(PMP)u$ results. Now PMP is symmetric if M is symmetric, and e is an eigenvector to the eigenvalue zero of PMP, due to $Pe=0$. Hence we get

$$u'Mu\geqslant\lambda_{\min}(M|e^\perp)u'u\quad\text{for all } u\in e^\perp, \tag{11}$$

where $\lambda_{\min}(M|e^\perp)$ denotes the smallest eigenvalue of PMP, if the zero eigenvalue is ignored with multiplicity one, i.e.,

$$\lambda_{\min}(M|e^\perp)=\min\{\lambda\in\mathbb{R}\ :\ PMPv=\lambda v \text{ for some } v\in e^\perp\setminus\{0\}\}. \tag{12}$$

We recall Theorem 5 of [10] according to which every local maximizer z of $x'Ax$ is maximizing this function over the whole face $\{y\in S^n\ :\ y_i=0 \text{ if } z_i=0\}$, and this face is contained in the basin of attraction of z under the replicator dynamics. For simplicity of exposition, we assume in the next result that this face is the whole simplex S^n, in other words, that $z_i>0$ for all i. Further, in view of Proposition 6 we may and do assume that $z=b$, where $b=x^V$ is the *barycenter* of S^n, i.e., $z_i=1/n$ for all $i\in V$. This gives us an upper bound for the parameter α as follows:

Proposition 7. *If $b=x^V$ is a global maximizer of $x'Ax$ on S^n, then b is also a (global) maximizer of $x'(A+\alpha I)x$ over S^n provided that*

$$\alpha\leqslant\beta=\lambda_{\min}(-A|e^\perp), \tag{13}$$

where $\lambda_{\min}(-A|e^\perp)$ is the smallest eigenvalue corresponding to the action of $-A$ on $e^\perp$, defined in (12).

Proof. The Nash equilibrium condition (7) for b w.r.t. A implies $x'Ab=b'Ab$ for all $x\in S^n$ (again, otherwise we got at least one strict inequality $(Ab)_i<b'Ab$, from which the absurd $b'Ab=\sum b_i(Ab)_i<b'Ab$ would result). But then $x'Ax-b'Ab=x'Ax-2x'Ab+b'Ab=(b-x)'A(b-x)\leqslant-\lambda_{\min}(-A|e^\perp)(b-x)'(b-x)$ due to (11). On the other hand, we also get $0\leqslant(b-x)'(b-x)=b'b-2x'b+x'x=1/n-2\ 1/n+x'x=x'x-1/n=x'x-b'b$, whence

$$\begin{aligned}x'(A+\alpha I)x-b'(A+\alpha I)b&=x'Ax-b'Ab+\alpha(x'x-b'b)\\&\leqslant\alpha(x'x-b'b)-\lambda_{\min}(-A|e^\perp)(b-x)'(b-x)\\&=(\alpha-\beta)(x'x-b'b)\leqslant 0\end{aligned} \tag{14}$$

results, provided $\alpha\leqslant\beta$ holds, since the last expression in parentheses is always non-negative. Hence the result. □

Summarizing, we obtain an admissible range for our parameter:

Theorem 8. *Let $S \subseteq \{1,\dots,n\}$, with $|S| = m$. Let $A_S = (a_{ij})_{i,j\in S}$ be the $m \times m$ submatrix of A corresponding to S, and $e_S = (1,\dots,1)' \in \mathbb{R}^m$. Denote by $\beta_S = \lambda_{\min}(-A_S | e_S^{\perp})$.*

If $x = x^S \in S^n$ is a (local) maximizer of $x'Ax$ over S^n and $\alpha \in]\gamma(x^S), \beta_S[$, then x^S is also a strict local maximizer of $x'(A + \alpha I)x$ over S^n.

On the other hand, if $\alpha < \gamma(x^S)$, then x^S becomes an unstable stationary point of the replicator dynamics under $A + \alpha I$, and thus, with probability one, cannot be approached by an interior path under these dynamics.

Proof. From Theorem 4, the claimed assertion follows if we can establish local asymptotic stability of x^S under the replicator dynamics with, say, continuous time, and the matrix $M = A + \alpha I$. Now x^S lies in the relative interior of the face

$$F = \{x \in S^n \,:\, x_i = 0 \text{ for all } i \notin S\}$$

of the simplex, which in turn is also time invariant under dynamics (3) and (4). As a consequence, we can decompose local stability analysis into the question of "internal" stability (concerning convergence of trajectories starting nearby within F) and, separately, "external" stability dealing with trajectories starting in S^n but off F. External stability is governed by the "external" eigenvalues of the linearization of this dynamics around x^S, which are given by the quantities (see [8, Lemma 21])

$$[(A + \alpha I)x^S]_i - (x^S)'(A + \alpha I)(x^S) = [Ax^S]_i - (x^S)'A(x^S) - \alpha(x^S)'(x^S) \quad \text{for } i \notin S.$$

Now if $\alpha > \gamma(x^S)$, then the latter quantity is negative by definition (9). On the other hand, internal stability of x^S follows from $\alpha \leqslant \beta_S$ and the optimality of x^S w.r.t. $x'(A + \alpha I)x$ on the face F as in Proposition 7. Recall that optimality on F is guaranteed by Theorem 5 of [10]. Hence the result for $\alpha > \gamma(x^S)$. The instability result follows by the same argumentation as in [14, Theorem 6]: all starting points of trajectories converging to the non-asymptotically stable point x^S lie on the center-stable manifold [27,28] which always is of codimension at least one. Hence, a trajectory with a randomly chosen starting point will almost surely not converge to x^S. □

A further result holds when A is the adjacency matrix of a graph G:

Theorem 9. *If $A = A_G$ is the adjacency matrix of graph G and S is a strictly maximal clique, then $]-1, 1[\subseteq]\gamma(x^S), \beta_S[$. More precisely, $\gamma(x^S) \leqslant -1$ that becomes $\gamma(x^S) \leqslant 0$ if S is just a maximal clique; on the other side $\beta_S = 1$ for any kind of clique S.*

Proof. If $A = A_G$ and $x = x^S$ is the characteristic vector of a strictly maximal clique of size $k = |S|$, we know that $\gamma(x^S)$ cannot exceed -1, because of

$$k([A_G x^S]_i - (x^S)'A_G(x^S)) = d_S(i) - (k-1) \leqslant -1 \tag{15}$$

for all $i \notin S$ due to the definition of strict maximality of S (similarly, one can show that $\gamma(x^C) \leqslant 0$ for any maximal clique C). Recall that $d_S(i) = \sum_{j\in S} a_{ij}$ denotes the

degree of vertex i w.r.t. S, i.e., the number of vertices in S connected to i. Moreover, in the simplifying hypothesis of Proposition 7, if $x^V = b$ is a global maximizer of $x'Ax$, then G is a complete graph, i.e. $A_G = ee' - I$, so that $-PA_GP = P$ and consequently $\beta_V = 1$. Returning to the general situation where we have to replace V with S, we see that by analogy, $\beta_S = 1$ must hold. □

Theorem 10. *If $0 < \alpha < 1$, then the only strict local maximizers of $x'(A_G + \alpha I)x$ over S^n (i.e. the only attracting stationary points under the replicator dynamics with $A_G + \alpha I$) are characteristic vectors x^S where S is a maximal clique. Conversely, if S is a maximal clique, then x^S represents a strict local maximizer.*

Proof. We show that even every local maximizer $y \in S^n$ (not necessarily strict) is a characteristic vector, by virtually the same proof as of Theorem 9 in [10]: to this end, put $S = \{i : y_i > 0\}$. First we show that the subgraph of G induced by S is complete. Indeed, suppose that for some $i, j \in S$ with $i \neq j$ we had $(i, j) \notin E$, i.e. $a_{ij} = 0$ would hold. Then for small $\delta > 0$, the point $x = y + \delta(e_i - e_j) \in S^n$ where e_i is the ith standard basis vector. Straightforward calculations now yield

$$x'(A + \alpha I)x = y'(A + \alpha I)y + \alpha\delta^2 > y'(A + \alpha I)y,$$

a contradiction to the optimality of y. Hence with A_S and e_S as defined in Theorem 8 we get $A_S = e_S e_S' - I$. Now the Karush–Kuhn–Tucker conditions necessary for local optimality yield, in particular, $A_S y_S + \alpha y_S + \mu e_S = o$ for some $\mu \in \mathbb{R}$, which gives, using $e_S' y_S = 1$,

$$y_S = \frac{1 + \mu}{1 - \alpha} e_S,$$

which, again using $e_S' y_S = 1$, yields $y = x^S$. It remains to show that S is maximal. So suppose that there is a vertex $i \notin S$ such that $d_S(i) = |S|$. But then as in (15),

$$[(A_G + \alpha I)x^S]_i - (x^S)'(A_G + \alpha I)(x^S) = \frac{d_S(i) - |S| + 1 - \alpha}{|S|} > 0,$$

contradicting the Nash equilibrium property (7) of x^S w.r.t. $M = A_G + \alpha I$, which is ensured by local optimality of x^S due to Theorem 4. Hence S is a maximal clique. To show the converse assertion, observe that x^S is a local maximizer of $x'A_Gx$ over S^n due to Theorem 1. Now $\alpha \in [0, 1[\subset]\gamma(x^S), \beta_S[$ (by Theorem 9), and Theorem 8 implies that x^S is also a strict local maximizer of $x'A_G + \alpha Ix$ over S^n. □

For the case $-1 < \alpha < 0$ no general result has been proven, but examples can be provided in which new (spurious) local maxima emerge which are not characteristic vectors of any subset of vertices, and at the same time local solutions in the form of characteristic vectors disappear. In the next section (and in an appendix) we study small examples which illustrate this point.

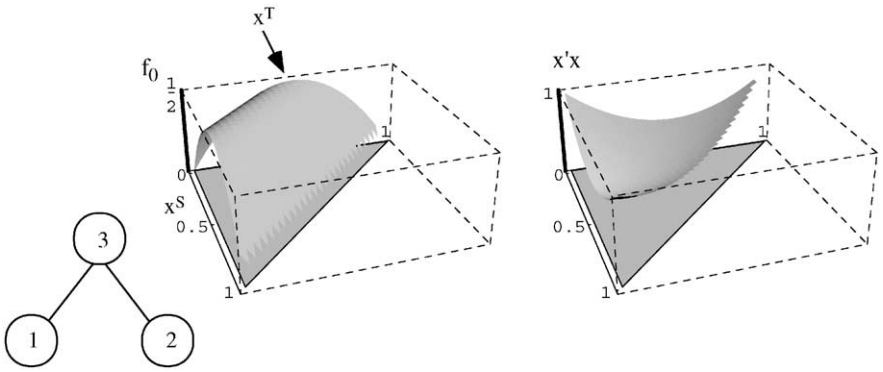

Fig. 1. The Motzkin–Straus program $f_0(x)$ and $x'x$ when x varies over S^3.

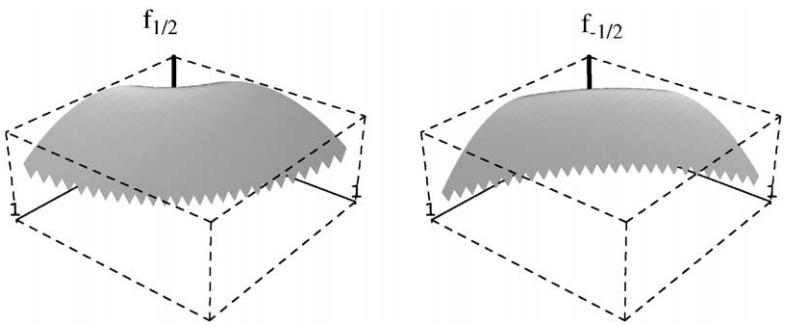

Fig. 2. $f_{1/2}(x)$ and $f_{-1/2}(x)$ on S^3.

4. A prototypical example

In this section we investigate a small, but nevertheless interesting, example, sketched in Fig. 1. It is a graph of size 3 with two maximum cliques of size 2 intersecting in vertex 3. Hence A_G has zero entries with the exception of $a_{13} = a_{31} = a_{23} = a_{32} = 1$. This is a frequently considered counterexample exhibiting spurious solutions to the Motzkin–Straus program [36]. For this simple graph $f_\alpha(x)$ is defined on the two-dimensional simplex S^3 so that we can actually plot it illustrating graphically the findings of the previous paragraphs.

More precisely, S^3 is a triangle spanned by the vertices $[1,0,0]'$, $[0,1,0]'$ and $[0,0,1]'$ contained in the plane described by the equation $x_1 + x_2 + x_3 = 1$. Our plots take this plane as their horizontal plane setting their origin in the vertex $[0,0,1]'$ of S^3. To remind the reader of this situation the plots of Fig. 1 contain the S^3 triangle marked in gray; the third, vertical, axis of the plots report the values of $f_\alpha(x)$.

The plots of Fig. 1 contain the basic Motzkin–Straus program and the regularizer term $x'x$, respectively; Fig. 2 contain plots of $f_{1/2}(x)$ and of $f_{-1/2}(x)$ seen from a slightly different viewpoint. One can intuitively grasp what is happening by realizing

that these two plots are obtained by the first one of Fig. 1 when the regularizing term is, respectively, added or subtracted.

Let us now examine the figures in detail starting from the basic Motzkin–Straus program in Fig. 1. In this example there are two maximum cliques $S=\{1,3\}$ and $T=\{2,3\}$, and their characteristic vectors $x^S=[\frac{1}{2},0,\frac{1}{2}]'$ and $x^T=[0,\frac{1}{2},\frac{1}{2}]'$ give $f_0(x^S)=f_0(x^T)=\frac{1}{2}$. But more generally, $f_0(x)=\frac{1}{2}$ for all $x=[\frac{1}{2}-s,s,\frac{1}{2}]'$ when $0\leqslant s\leqslant\frac{1}{2}$. This example shows explicitly that in the Motzkin–Straus program the global maximizer of $f_0(x)$ are not necessarily characteristic vectors of maximum cliques, as stated by the following general property (see [39] for a proof and related results):

Theorem 11. *Let G be a graph containing maximum cliques $C_1,\ldots,C_q$ (among possible others). Then every vector belonging to the convex hull of their characteristic vectors is a global maximizer of $x'A_Gx$ over S^n if and only if, for all $i,j=1,\ldots,q$, the number of edges having one endpoint in $C_i\setminus C_j$ and the other in $C_j\setminus C_i$ equals $m_{ij}(m_{ij}-1)$, where $m_{ij}=|C_i\setminus C_j|=|C_j\setminus C_i|$.*

Fig. 2 contains $f_{1/2}(x)$ and shows that in this case the only maxima correspond to the characteristic vectors of the maximum cliques while the second plot, containing $f_{-1/2}(x)$, shows that the isolated maximizer is an interior point not corresponding to any clique vector. For $\alpha>1$, e.g. $f_{3/2}(x)$, the situation is essentially (apart from the vertical scale) that of the regularizer term in Fig. 1. These plots illustrate also the role of the bounds of α that, for this example, are $\gamma=0$ and $\beta=1$ for both x^S and x^T as predicted by Theorem 9. The shapes of $f_{-1/2}(x)$, $f_{1/2}(x)$ and $f_{3/2}(x)$, representing the three possible cases of α with respect to its bounds, confirm the results of Theorems 8 and 10.

A more detailed analysis together with that of another simple example is provided in the appendix.

5. The annealed replication heuristic

As discussed previously, the major drawback of replicator equations is their inherent inability to escape from local maximizers of the objective function. Theorem 8 provides us with an immediate strategy to avoid unwanted local solutions, i.e., maximal cliques which are not maximum. Suppose that S is a maximal clique in G that we want to avoid. By letting $\alpha<\gamma(x^S)$, its characteristic vector x^S becomes an unstable stationary point of the replicator dynamics under f_α, and thus will not be approached by any interior trajectory. Of course, the problem is to obtain a reasonable estimate for $\gamma(x^S)$ without knowing S in advance. Furthermore, if $\alpha\leqslant 0$, it may well happen that the process converges to a vector which does not represent a clique (see below).

Since we are concerned with the maximum clique problem,

$$\gamma(x^S)=\gamma_S=\max_{i\notin S} d_S(i)-|S|+1. \tag{16}$$

As already noted in (15), $\gamma_S \leqslant -1$ if S is strictly maximal while $\gamma_S = 0$ if S is not strictly maximal. In both cases, $\gamma_S \geqslant 1-|S|$ with equality attained if S is isolated in G. So if one wants to avoid cliques with size $|S| \leqslant m$, one could simply run the algorithm with $\alpha < 1 - m \leqslant 1 - |S| \leqslant \gamma_S \leqslant 0$, and if there is a clique T such that still $\gamma_T < \alpha$ holds, there is a (more or less justified) hope to obtain in the limit x^T, which yields *automatically* a larger maximal clique T.

Unfortunately, two other cases could occur:

(a) no other clique T satisfies $\gamma_T < \alpha$, i.e., α has a too large value;
(b) even if there is such a clique, other attractors could emerge which are not characteristic vectors of a clique (note that this is excluded if $\alpha > 0$ by Theorem 10).

The proper choice of the parameter α is therefore a trade-off between the desire to remove unwanted maximal cliques and the emergence of spurious solutions. We present now the strategy we adopted in this choice stressing that, given the lack of precise indications, our prescriptions are supported mainly by numerical results obtained in extensive tests and by the intuitions obtained examining these tests and simple examples like those of Section 4 and of the appendix.

Instead of keeping the value of α fixed, our approach is to start with a sufficiently large negative α and adaptively increase it during the optimization process, in much the same spirit as the simulated annealing procedure [30]. Of course, in this case the annealing parameter has no interpretation in terms of a hypothetical temperature, and the resulting algorithm is completely deterministic. The rationale behind this idea is that for values of α that are sufficiently negative only the characteristic vectors of large maximal cliques will be stable attractive points for the replicator dynamics, together with a set of spurious solutions. As the value of α increases, spurious solutions disappear and at the same time (characteristic vectors of) smaller maximal cliques become stable. We expect that at the beginning of the annealing process the dynamics is attracted toward "promising" regions, and the search is further refined as the annealing parameter increases. In summary, the proposed algorithm is as follows:

1. Start with a sufficiently large negative α.
2. Let b be the barycenter of S^n and set $x = b$.
3. Run the replicator dynamics starting from x, under $A + \alpha I$ until convergence and let x be the converged point.
4. Unless a stopping condition is met, increase α and goto 3.
5. Select $\bar{\alpha}$ with $0 < \bar{\alpha} < 1$ (e.g. $\bar{\alpha} = \frac{1}{2}$), run the replicator dynamics starting from current x under $A + \bar{\alpha} I$ until convergence, and extract a maximal clique from the converged solution.

The last step in the algorithm is necessary if we want to extract also the vertices comprising the clique found, as shown in Theorem 10.

Note that when $\alpha < 0$ we are no longer guaranteed that the trajectories of the replicator dynamics in step (3) will remain in the simplex S^n, and hence $x'(A + \alpha I)x$ will not necessarily increase at every step. Admittedly, in the numerical simulations we

carried out and which are reported in the following section, this phenomenon almost never happened. In a few cases the first iteration yielded negative entries in the iterated vector, but at the following steps the vector was readily projected onto the simplex. In any case, a matrix with negative elements is no problem. It is simple to see that, by adding a sufficiently large constant to the matrix to make it non-negative, the theory and the optimization process are unaffected.

It is clear that for the algorithm to work, we need to select an appropriate "annealing" strategy. To this end, one could employ the following heuristics: suppose for the moment that the underlying graph is a random one in the sense that edges are generated independently of each other with a certain equal probability q (in applications, q could be replaced with $|E|/\binom{n}{2}$, the actual density). Suppose S is an unwanted clique of size m. Take $\delta > 0$ small, say 0.01, and consider a lower bound which is exceeded with probability $1-\delta$:

Theorem 12. *Under the random graph model consider a clique S of size $|S| = m$, put $v = 1/2(n-m)$ and denote by $\bar{\gamma}_m$ the following lower bound for $\gamma(x^S)$:*

$$\bar{\gamma}_m = 1 - (1-q)m - \sqrt{mq(1-q)}\delta^v. \tag{17}$$

Then

$$\mathbb{P}(\gamma(x^S) \leqslant \bar{\gamma}_m) \leqslant \delta.$$

Moreover, $\bar{\gamma}_m$ exceeds $1-m$ for all $m \geqslant m_{q,\delta}$ where

$$m_{q,\delta} = \frac{1-q}{q}\sqrt[n]{\delta}. \tag{18}$$

Proof. Since $\gamma(x^S) = \max_{i \notin S} d_S(i) - m + 1$, and since for different $i \neq j$, the variates $d_S(i)$ and $d_S(j)$ are stochastically independent in the random graph model, we first get the identity

$$\mathbb{P}(\gamma(x^S) \leqslant \bar{\gamma}_m) = [\mathbb{P}(d_S(i) \leqslant \bar{\gamma}_m + m - 1)]^{n-m}.$$

Next, observe that the expected value and variance of $d_S(i)$ is, according to the Binomial Law,

$$\mathbb{E}d_S(i) = mq \quad \text{and} \quad \operatorname{Var} d_S(i) = mq(1-q).$$

Hence Čebyšev's inequality gives

$$\mathbb{P}(|d_S(i) - mq| \geqslant \varepsilon) \leqslant mq(1-q)/\varepsilon^2,$$

which entails, putting $\varepsilon = 1 - (1-q)m - \bar{\gamma}_m > 0$,

$$\begin{aligned}\mathbb{P}(d_S(i) \leqslant \bar{\gamma}_m + m - 1) &= \mathbb{P}(d_S(i) - mq \leqslant \bar{\gamma}_m + (1-q)m - 1) \\ &= \mathbb{P}(d_S(i) - mq \leqslant -\varepsilon) \\ &\leqslant \mathbb{P}(|d_S(i) - mq| \leqslant \varepsilon) \leqslant mq(1-q)\varepsilon^{-2}.\end{aligned} \tag{19}$$

This gives an lower bound $\bar{\gamma}_m$ for γ_S which is exceeded with a probability of least $1-\delta$ as follows:

$$\begin{aligned}\mathbb{P}(\gamma(x^S) \leqslant \bar{\gamma}_m) &= [\mathbb{P}(d_S(i) < \bar{\gamma}_m + m - 1)]^{n-m} \\ &\leqslant [mq(1-q)\varepsilon^{-2}]^{n-m} = \delta \end{aligned} \tag{20}$$

provided $1-(1-q)m-\bar{\gamma}_m = \varepsilon = \sqrt{mq(1-q)}\sqrt[2(n-m)]{\delta}$, which yields (17). Since $v \geqslant 1/2n$ by definition, obviously $\delta^{2v} \leqslant \delta^{1/n}$. On the other hand, $\bar{\gamma}_m > 1-m$ if and only if $qm - \sqrt{mq(1-q)}\delta^v > 0$, which is equivalent to $m > [(1-q)/q]\delta^{2v}$. Hence $m > m_{q,\delta} \geqslant [(1-q)/q]\delta^{2v}$ yields $\bar{\gamma}_m > 1-m$. □

Since $m_{q,\delta} \leqslant 10$ if $q \geqslant 0.1$ for all $\delta > 0$, the previously obtained hard lower bound $1-m$ is relaxed by $\bar{\gamma}_m$ in almost all important applications. Moreover the bound $\bar{\gamma}_m$ decreases with increasing m provided that

$$\frac{1}{\sqrt{m}}\left[1 + \frac{m}{(n-m)^2}\log\delta\right] > -\frac{2}{\delta^v}\sqrt{\frac{1-q}{q}}$$

holds, and this is true for many important cases in practice: indeed, observe that the latter inequality necessarily holds if the expression in brackets is positive, which is true, e.g. for $\delta = 0.01$, whenever $6m \leqslant (n-m)^2$. Thus it makes sense to use $\bar{\gamma}_m$ as a heuristic proxy for the lower bound of $\gamma(x^S)$, to avoid being attracted by a clique of size m.

Furthermore, a well-known result due to Matula [32] accurately predicts the size of the maximum clique in random graphs with sufficiently many vertices. Let

$$M(n,q) = 2\log_{1/q} n - 2\log_{1/q}\log_{1/q} n + 2\log_{1/q}\frac{e}{2} + 1. \tag{21}$$

Matula proved that, as $n \to \infty$, the size of the maximum clique in an n-vertex q-density random graph is either $\lfloor M(n,q)\rfloor$ or $\lceil M(n,q)\rceil$ with probability tending to 1.

The previous results suggest us a sort of "two-level" annealing strategy: the level of clique size, which in turn induces that of the "actual" annealing parameter. More precisely, if we do not have any a priori information about the expected size of the maximum clique, we can use Matula's formula $M(n,q)$ to have an initial (more or less accurate) estimate of it. Let $m = \lceil M(n,q)\rceil$; by setting the initial value for α (step 1 of our algorithm) at some intermediate value between $\bar{\gamma}_m$ and $\bar{\gamma}_{m-1}$, e.g. $\alpha = (\bar{\gamma}_m + \bar{\gamma}_{m-1})/2$, we expect that only the characteristic vectors of maximal cliques having size m will survive in f_α, together with many spurious solutions. After the initial cycle, we decrease m, recalculate $\bar{\gamma}_m$ and $\bar{\gamma}_{m-1}$ and update $\alpha = (\bar{\gamma}_m + \bar{\gamma}_{m-1})/2$ in step 4 as in the previous step. The whole process is iterated until either m reaches 1 or α becomes greater than zero.

6. Experimental results

To assess the effectiveness of the proposed heuristic, extensive simulations were carried out over a selection of DIMACS graphs [25], which represent a standard benchmark for clique finding algorithms.[3] The experiments were conducted using the discrete-time version (4) of the replicator equations. The code was written in the C programming language and run on a Digital AlphaStation Series 200 (no attempt was made to optimize the code). For each graph considered, the proposed algorithm was run by using the two-level annealing schedule described at the end of the previous section. For each internal cycle (step 3), the replicator algorithm was iterated until the (squared) distance between two successive states became smaller than 10^{-10}. At the final cycle (i.e., step 5), the parameter α was set to $\frac{1}{2}$, and the replicator dynamics was stopped when either a maximal clique (i.e., a local maximizer of $f_{1/2}$ on S^n) was found or the distance between two successive points was smaller than a fixed threshold, which was set to $n10^{-15}$ (n being the number of vertices of the graph at hand). In the latter case the converged vector was randomly perturbed, and the algorithm restarted from the perturbed point. Because of the one-to-one correspondence between local maximizers and maximal cliques (see Theorem 10) this situation corresponds to convergence to a saddle point.

In order to assess the relative merits of the proposed heuristic we compared our algorithm with plain replicator dynamics with fixed α, i.e., with no annealing strategy. Specifically, two cases were considered: $\alpha = \frac{1}{2}$, which corresponds to the original spurious-free quadratic program proposed by [24] and recently studied by Bomze et al. [10,12], and $\alpha = 0$ which is the original Motzkin–Straus formulation [33] as studied by Pelillo [37]. In both cases, the replicator process was started from the barycenter of the simplex, and iterated until the squared distance between two successive states became smaller than 10^{-20}. In addition, our results were compared with those reported by Gibbons et al. [21] who proposed a continuous-based heuristic (CBH) also based on a parameterization (completely different from ours) of the Motzkin–Straus program.

The results of our experiments are summarized in Tables 1 and 2, which contain a row for each DIMACS graphs considered. The columns labeled graph, vertices, and dens. represent the name of the corresponding graph, the number of its vertices and its density, respectively. The column Max Clique, contains the size of the maximum clique when known, or a lower bound for it (this information is already available in the file containing the graph). The columns ARH, PRD($\frac{1}{2}$), PRD(0) and CBH contain the size of the clique found using the proposed annealed replication heuristic (ARH), the plain replicator dynamics (PRD) applied to (2) with $\alpha = \frac{1}{2}$, the plain replicator dynamics (PRD) applied to (2) with $\alpha = 0$—these results are taken from [11]—and

[3] We did not consider graphs where the plain algorithm applied to (1) already yields the maximum clique, e.g., the "c-fat" family [11]. Also, a few very large and dense graphs were excluded because of the excessively high computational cost required.

Table 1
Results on DIMACS benchmark graphs (part I)

Graph	Vertices	Dens.	Max clique	Clique size obtained ARH	PRD($\frac{1}{2}$)	PRD(0)	CBH	Time (s)
brock200_1	200	0.745	21	19	17	18	20	167.79
brock200_2	200	0.496	12	10	8	8	12	97.33
brock200_3	200	0.605	15	13	9	10	14	124.44
brock200_4	200	0.658	17	14	12	13	16	150.74
brock400_1	400	0.748	27	20	21	21	23	906.26
brock400_2	400	0.749	29	23	20	22	24	752.69
brock400_3	400	0.748	31	23	18	20	23	554.45
brock400_4	400	0.749	33	23	19	21	24	937.84
brock800_1	800	0.649	23	18	16	17	20	3323.31
brock800_2	800	0.651	24	18	15	17	19	3175.44
brock800_3	800	0.649	25	19	16	18	20	2697.56
brock800_4	800	0.650	26	19	15	17	19	3181.74
san1000	1000	0.501	15	8	8	8	8	1824.60
san200_0.7_1	200	0.700	30	15	15	15	15	39.66
san200_0.7_2	200	0.700	18	12	12	12	12	40.11
san200_0.9_1	200	0.900	70	45	45	45	46	106.29
san200_0.9_2	200	0.900	60	39	36	35	36	56.03
san200_0.9_3	200	0.900	44	31	32	33	30	98.76
san400_0.5_1	400	0.500	13	7	7	7	8	156.77
san400_0.7_1	400	0.700	40	20	20	20	20	232.39
san400_0.7_2	400	0.700	30	15	15	15	15	230.72
san400_0.7_3	400	0.700	22	12	12	12	14	194.12
san400_0.9_1	400	0.900	100	50	40	55	50	425.88
sanr200_0.7	200	0.700	18	16	14	16	18	131.12
sanr200_0.9	200	0.900	$\geqslant 42$	41	37	40	41	158.41
sanr400_0.5	400	0.900	13	13	11	11	12	269.64
sanr400_0.7	400	0.700	$\geqslant 21$	21	18	18	20	838.30

the Gibbons et al. CBH algorithm [21], respectively. Finally, the column labeled time contain the CPU time required by the process to provide the final solution.

As can be seen, the results are very encouraging. In fact, in almost all cases we obtained larger cliques with ARH than PRD($\frac{1}{2}$) did (the exceptions being `brock400_1`, `san200_0.9_3` and `p_hat700-2`). In many cases, we obtained the same results as CBH and in a few examples we returned better solutions, e.g., `p_hat1500-2`, `san200_0.9_2`, `sanr400_0.5`. ARH also performed better than PRD(0). Only in six out of 46 cases PRD(0) returned a larger clique size, that is: `brock400_1`, `san_200_0.9_3`, `san_400_0.9_1`, `p_hat500-3`, `p_hat700-2`, and `p_hat1000-3`. However, as discussed in previous sections, due to the presence of spurious solutions in the original Motzkin–Straus program, PRD(0) is not able to always return the nodes comprising the clique found: it only provides information about its size. It is worth noting that the Sanchis graphs (the "san" family) turned out to be very hard for Motzkin–Straus-based optimization algorithms since neither of the three heuristics found good results.

Table 2
Results on DIMACS benchmark graphs (part II)

Graph	Vertices	Dens.	Max clique	Clique size obtained ARH	PRD($\frac{1}{2}$)	PRD(0)	CBH	Time (s)
MANN_a9	45	0.927	16	16	12	12	16	0.833
MANN_a27	378	0.990	126	117	117	117	121	6807.81
p_hat300-1	300	0.244	8	8	6	6	8	107.63
p_hat300-2	300	0.489	25	25	22	24	25	301.69
p_hat300-3	300	0.744	36	35	32	33	36	3221.27
p_hat500-1	500	0.253	9	9	8	8	9	335.83
p_hat500-2	500	0.505	36	36	33	35	35	893.21
p_hat500-3	500	0.752	$\geqslant 49$	47	47	48	49	1729.50
p_hat700-1	700	0.249	11	9	7	9	11	739.92
p_hat700-2	700	0.498	44	41	43	43	44	1893.96
p_hat700-3	700	0.748	$\geqslant 62$	59	57	59	60	2582.18
p_hat1000-1	1000	0.245	10	10	8	8	10	1965.47
p_hat1000-2	1000	0.490	$\geqslant 46$	44	42	44	46	3010.43
p_hat1000-3	1000	0.744	$\geqslant 65$	62	61	63	65	7288.22
p_hat1500-1	1500	0.253	12	10	9	9	11	4100.54
p_hat1500-2	1500	0.506	$\geqslant 65$	64	62	62	63	8598.91
p_hat1500-3	1500	0.754	$\geqslant 94$	91	89	90	94	16251.77
keller4	171	0.649	11	8	7	7	10	34.85
keller5	776	0.751	27	16	15	15	21	610.16

As far as the CPU time is concerned, it is clear that our algorithm turns out to be computationally more expensive than plain replicator dynamics on fixed α (see [12] for comparison) because the latter is simply a single step of our heuristic. Moreover, ARH is slower than CBH [21] which in turn may have serious memory allocation problems. However, we note that the continuous-time version (3) of replicator equations can naturally be mapped onto hardware circuitry [41], thereby making the whole algorithm particularly amenable to parallel, distributed implementations.

From the results obtained, it can be concluded that the proposed annealed replication heuristic does a good job at finding large cliques, and clearly beats the plain replicator dynamics approach, where no annealing strategy is used. Moreover, it should be pointed out that the annealing schedule adopted is entirely based on the assumption that the graphs at hand are random; clearly, DIMACS graphs can hardly said to be "random," but nevertheless the heuristic worked remarkably well. Of course, better annealing strategies could be devised if we knew something about the underlying structure of the graphs, but in absence of this kind of information the random graph assumption seems to be sufficiently robust.

7. Conclusions

We have presented a new heuristic for approximating the maximum clique problem. The approach is centered around an attractive characterization of the problem due to

Motzkin and Straus, which allows us to formulate it as a linearly constrained quadratic maximization program. Specifically, we have introduced a control parameter α and studied the properties of the objective function as α varies. We have shown that when α is positive all the properties enjoyed by the standard regularization approach [10] hold true; specifically, in this case a one-to-one correspondence between local/global maximizers in the continuous space and local/global solutions in the discrete space exists. For negative α's an interesting picture emerges: as the absolute value of α grows larger, local maximizers corresponding to maximal cliques disappear. We have derived bounds on the parameter α which affect the stability of these solutions. These results have suggested the annealed replication heuristic, which consists of starting from a large negative α and then properly reducing it during the optimization process. For each value of α standard replicator equations are run in order to obtain local solutions of the corresponding objective function. The rationale behind this idea is that for values of α with a proper large absolute value only local solutions corresponding to large maximal cliques will survive, together with various spurious maximizers. As the value of α is reduced, spurious solutions disappear and smaller maximal cliques become stable. An annealing schedule is proposed which is based on the assumption that the graphs being considered are random. Experiments conducted over several DIMACS benchmark graphs confirm the effectiveness of the proposed approach and the robustness of the annealing strategy. The overall conclusion is that the annealing procedure *does* help to avoid inefficient local solutions, by initially driving the dynamics towards promising regions in state space, and then refining the search as the annealing parameter is increased.

Acknowledgements

The authors would like to thank one of the anonymous referees for providing critical comments and suggestions.

Appendix Full dynamic analysis of two examples

Here we investigate the dynamic behavior of the replicator dynamics of $A_G + \alpha I$ as α varies over the entire real line, for the graph of size 3 considered in Section 4. Since for the replicator dynamics on S^3 a complete classification of the replicator flow is available [7,9], we refrain from repeating the phase portraits in pictures here, but rather refer to the numbers in the system used in the cited articles (see Fig. 6 in [7] and Fig. 1 in [9]). A prefixed minus sign $(-)$ symbolizes time reversal of the respective phase portrait (PP). In some cases, the PPs have to be rotated accordingly.

There are three bifurcations at $\alpha = 0, 1, 2$: for $\alpha < 0$ the situation is essentially that of the second plot of Fig. 2 and there is only the interior attractor

$$y_\alpha = \left[\frac{1-\alpha}{4-3\alpha}, \frac{1-\alpha}{4-3\alpha}, \frac{2-\alpha}{4-3\alpha}\right]'.$$

The flow is depicted as PP 7 in Fig. 6 of [7]. As α increases reaching the first bifurcation at $\alpha = 0$ the situation changes from that of the second plot of Fig. 2 to that of the Motzkin–Straus program of Fig. 1. The flow stops not only at the edge connecting vertices 1 and 2, but also along the trajectories joining y_α with the (former) saddle points x^S and x^T, so that we arrive at PP 1. If $\alpha \in]0,1[$ the situation is essentially that of the first plot of Fig. 2, and we have the picture of PP 8, which renders both x^S and x^T as (local) attractors, and y_α as a saddle point wandering towards $x^{\{3\}}$ as $\alpha \nearrow 1$. This completely breaks down if $\alpha = 1$ where, again, the flow is stopped at two edges: PP 20 emerges. If α is increased further, only the vertices are attracting with the occurrence of y_α as an interior repellor if α exceeds 2: we get PP -35 for $\alpha \in]1,2]$ and PP -7 if $\alpha > 2$. The regularizing term of Fig. 1 substantially depicts the situation after the last bifurcation at $\alpha = 2$.

In another example, G consists of three vertices with only two of them connected, so A_G has only zero entries with the exception of $a_{12} = a_{21} = 1$. Hence $S = \{1,2\}$ is the unique maximum clique. In this case, replicator dynamics undergoes a simple exchange-of-stability bifurcation as α passes through $\alpha = -1$, and a more dramatic, but similar phenomenon occurs as $\alpha = 0$ where the flow on two edges is reversed simultaneously: indeed, for $\alpha < -1$ we obtain, again, PP No. 7 with interior global attractor of the form

$$x_\alpha = \left[\frac{\alpha}{3\alpha+1}, \frac{\alpha}{3\alpha+1}, \frac{\alpha+1}{3\alpha+1}\right]'$$

(observe that x_α approaches $[\frac{1}{3},\frac{1}{3},\frac{1}{3}]'$ as $\alpha \searrow -\infty$ while $x_\alpha \to x^S = [\frac{1}{2},\frac{1}{2},0]'$ as $\alpha \nearrow -1$). If α now increases, this stable stationary point remains at x^S. Indeed, the PP is for $\alpha \in [-1,0[$ qualitatively the same as PP 35 while for $\alpha = 0$ we obtain the PP -20. Still x^S is the global attractor. If α is further increased to positive numbers (e.g. $\alpha = \frac{1}{2}$), then local stability of x^S is retained, but there emerges a second local attractor $x^{\{3\}}$ (corresponding to the maximal clique $\{3\}$) together with an interior saddle point, again at x_α as above, and the PP -8 results. If $\alpha = 1$, stability of $x^{\{3\}}$ is retained, but the flow at the edge containing x^S stops: PP -1 depicts the situation, which in some sense corresponds to the occurrence of spurious solutions in the previous example (emerging at $\alpha = 0$ there). Finally, if $\alpha > 1$, we arrive again at PP -7 where x_α is now a repellor and all vertices become attractors. This is in accordance with the theory as in Theorem 8.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and the hardness of approximation problems, Proceedings of the 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 14–23.

[2] S. Arora, S. Safra, Probabilistic checking of proofs: a new characterization of NP, Proceedings of the 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 2–13.

[3] M. Bellare, S. Goldwasser, C. Lund, A. Russell, Efficient probabilistically checkable proofs and application to approximation, Proceedings of the 25th Annual ACM Symposium on Theory Comput., San Diego, CA, 1993, pp. 294–304.

[4] M. Bellare, S. Goldwasser, M. Sudan, Free bits, PCPs and non-approximability—towards tight results, Proceedings of the 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, 1995, pp. 422–431.
[5] L.E. Baum, J.A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bull. Amer. Math. Soc. 73 (1967) 360–363.
[6] L.E. Baum, G.R. Sell, Growth transformations for functions on manifolds, Pacific J. Math. 27 (2) (1968) 211–227.
[7] I.M. Bomze, Lotka–Volterra equation and replicator dynamics: a two-dimensional classification, Biol. Cybernet. 48 (1983) 201–211.
[8] I.M. Bomze, Non-cooperative two-person games in biology: a classification, Internat. J. Game Theory 15 (1986) 31–57.
[9] I.M. Bomze, Lotka–Volterra equation and replicator dynamics: new issues in classification, Biol. Cybernet. 72 (1995) 447–453.
[10] I.M. Bomze, Evolution towards the maximum clique, J. Global Optim. 10 (1997) 143–164.
[11] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: D.Z. Du, P.M. Pardalos (Eds.), Handbook of Combinatorial Optimization (suppl. Vol. A), Kluwer, Dordrecht, 1999, pp. 1–74.
[12] I.M. Bomze, M. Pelillo, R. Giacomini, Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale, in: I.M. Bomze, T. Csendes, R. Horst, P.M. Pardalos (Eds.), Developments in Global Optimization, Kluwer, Dordrecht, 1997, pp. 95–108.
[13] I.M. Bomze, F. Rendl, Replicator dynamics for evolution towards the maximum clique: variations and experiments, in: R. De Leone, A. Murli, P.M. Pardalos, G. Toraldo (Eds.), High Performance Algorithms and Software in Nonlinear Optimization, Kluwer, Dordrecht, 1998, pp. 53–68.
[14] I.M. Bomze, V. Stix, Genetic engineering via negative fitness: evolutionary dynamics for global optimization, Ann. Oper. Res. 89 (1999) 297–318.
[15] M. Budinich, Bounds on the maximum clique of a graph, Disc. Appl. Math., to appear.
[16] J.F. Crow, M. Kimura, An Introduction to Population Genetics Theory, Harper & Row, New York, 1970.
[17] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, Approximating clique is almost NP-complete, Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1991, pp. 2–12.
[18] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, Interactive proofs and the hardness of approximating clique, J. ACM 43 (2) (1996) 268–292.
[19] R.A. Fisher, The Genetical Theory of Natural Selection, Clarendon Press, Oxford, 1930.
[20] A.H. Gee, R.W. Prager, Polyhedral combinatorics and neural networks, Neural Comput. 6 (1994) 161–180.
[21] L.E. Gibbons, D.W. Hearn, P.M. Pardalos, A continuous based heuristic for the maximum clique problem, in: D.S. Johnson, M. Trick (Eds.), Cliques, Coloring, and Satisfiability—Second DIMACS Implementation Challenge, American Mathematical Society, Providence, RI, 1996, pp. 103–124.
[22] L.E. Gibbons, D.W. Hearn, P.M. Pardalos, M.V. Ramana, Continuous characterizations of the maximum clique problem, Math. Oper. Res. 22 (3) (1997) 754–768.
[23] J. Hastad, Clique is hard to approximate within $n^{1-\epsilon}$, Proceedings of the 37th Annual Symposium on Foundations of Computer Science, 1996, pp. 627–636.
[24] J. Hofbauer, K. Sigmund, The Theory of Evolution and Dynamical Systems, Cambridge University Press, Cambridge, UK, 1988.
[25] D.S. Johnson, M.A. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26, American Mathematical Society, Providence, RI, 1996 (see also `http://dimacs.rutgers.edu/Volumes/Vol26.html`).
[26] S. Karlin, Mathematical models, problems and controversies of evolutionary theory, Bull. Amer. Math. Soc. 10 (1984) 221–273.
[27] A. Kelley, The stable, center-stable, center, center-unstable, unstable manifolds, J. Differential Equations 3 (1967) 546–570.
[28] A. Kelley, Stability of the center-stable manifold, J. Math. Anal. Appl. 18 (1967) 336–344.
[29] M. Kimura, On the change of population fitness by natural selection, Heredity 12 (1958) 145–167.

[30] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–679.
[31] S.E. Levinson, L.R. Rabiner, M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, Bell System Tech. J. 62 (1983) 1035–1074.
[32] D.W. Matula, The largest clique size in a random graph, Technical Report CS 7608, Department of Computer Science, Southern Methodist University, 1976.
[33] T.S. Motzkin, E.G. Straus, Maxima for graphs and a new proof of a theorem of Turán, Canad. J. Math. 17 (1965) 533–540.
[34] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[35] P.M. Pardalos, Continuous approaches to discrete optimization problems, in: G. Di Pillo, F. Giannessi (Eds.), Nonlinear Optimization and Applications, Plenum Press, New York, 1996, pp. 313–328.
[36] P.M. Pardalos, A.T. Phillips, A global optimization approach for solving the maximum clique problem, Internat. J. Computer Math. 33 (1990) 209–216.
[37] M. Pelillo, Relaxation labeling networks for the maximum clique problem, J. Artificial Neural Networks 2 (1995) 313–328.
[38] M. Pelillo, The dynamics of nonlinear relaxation labeling processes, J. Math. Imaging Vision 7 (4) (1997) 309–323.
[39] M. Pelillo, A. Jagota, Feasible and infeasible maxima in a quadratic program for maximum clique, J. Artificial Neural Networks 2 (1995) 411–419.
[40] P. Taylor, L. Jonker, Evolutionarily stable strategies and game dynamics, Math. Biosci. 40 (1978) 145–156.
[41] A. Torsello, M. Pelillo, Continuous-time relaxation labeling processes, Pattern Recognition 33 (2000) 1897–1908.
[42] J.W. Weibull, Evolutionary Game Theory, MIT Press, Cambridge, MA, 1995.