Annealed Imitation: Fast Dynamics for Maximum Clique

Marcello Pelillo
Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino 155, 30172 Venezia Mestre, Italy

Abstract— **We propose a new class of heuristics for the maximum clique problem (MCP) whose basic ingredients are: (1) a parameterized continuous formulation of MCP, (2) an instability analysis of equilibria of *imitation dynamics* from evolutionary game theory, and (3) a principled way of varying a regularization parameter during the evolution process so as to avoid inefficient solutions. The resulting "annealed imitation" class is shown to contain algorithms that are dramatically faster than and as accurate as state-of-the-art neural network heuristics for maximum clique.**

I. Introduction

The maximum clique problem (MCP) is a well-known graph-theoretic problem which finds important applications in many different domains. Since it is known to be *NP*-hard, however, 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 problem is in fact difficult to solve even in terms of approximation. In light of these negative results, much effort has recently been directed towards devising efficient clique finding heuristics, for which no formal guarantee of performance may be provided, but are anyway of interest in practical applications. In the neural network community, there has also been much recent interest around this important problem. We refer to [2] for a recent review concerning algorithms, applications, and complexity issues related to the MCP.

In the mid-1960s, Motzkin and Straus [11] established a remarkable connection between the MCP and a quadratic programming problem on the standard simplex. The Motzkin-Straus formulation, and variations thereof, has motivated various neural network heuristics for maximum clique. In particular, *replicator equations* from evolutionary game theory have proven to be quite effective in solving this and related combinatorial optimization problems [1], [4], [12], [13], [15].

In this paper, we first introduce a wide family of *payoff-monotonic* dynamical systems of which replicator equations are just a special instance. The models in this family enjoy precisely the same dynamical properties as replicator equations, and hence they naturally suggest themselves as heuristics for the maximum clique problem. However, as standard replicator equations, they are inherently unable to escape from inefficient local solutions.

We then focus on a well-known subclass of payoff-monotonic dynamics which arises in modeling the evolution of behavior by way of imitation processes. We investigate the properties of a parameterized formulation of the Motzkin-Straus program as a function of its parameter. A detailed analysis of these properties suggests an entire class of heuristics for the MCP which is based on the idea of properly varying the parameter during the im itation optimization process, so as to avoid unwanted inefficient solutions. A similar idea has been proposed by Gee and Prager in a different context [5]. Experiments show that this "annealed imitation" class contains algorithms which are dramatically faster than and as accurate as state-of-the-art neural network heuristics for maximum clique.

II. Payoff-monotonic Dynamics and their Properties

Evolutionary game theory considers an idealized scenario whereby in a large population pairs of individuals are repeatedly drawn at random to play a symmetric two-player game. In contrast to traditional game theoretic models, players are not supposed to behave rationally or to have complete knowledge of the details of the game. They act instead according to a pre-programmed behavior pattern, or pure strategy, and it is supposed that some evolutionary selection process operates over time on the distribution of behaviors. We refer the reader to [8], [16] for excellent introductions to this rapidly expanding field.

Let $J = \{1, \cdots, n\}$ be the set of available pure strategies and, for all $i \in J$, let $x_i(t)$ be the proportion of population members playing strategy i, at time t. The state of the population at a given instant is the vector $\mathbf{x} = (x_1, \cdots, x_n)'$, where a prime denotes transposition. Clearly, population states are constrained to lie in the standard simplex of the n–dimensional Euclidean space $\mathbb{R}^n$ (see Fig. 1):

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

where $\mathbf{e} = (1, \ldots, 1)'$, and hence $\mathbf{e}'\mathbf{x} = \sum_i x_i$.

Let $A = (a_{ij})$ be the $n \times n$ payoff matrix. Specifically, for each pair of strategies $i, j \in J$, a_{ij} represents the payoff of an individual playing strategy i against an opponent playing strategy j. In biological contexts payoff is typically measured in terms of Darwinian fitness or reproductive success (i.e., the player's expected number of surviving offspring), whereas in economic applications they usually represent firms' profits or consumers' utility. If the population is in state $\mathbf{x}$, the expected payoff earned by an i-strategist is:

$$\pi_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} x_j = (A\mathbf{x})_i \tag{1}$$

0-7803-7898-9/03/$17.00 ©2003 IEEE

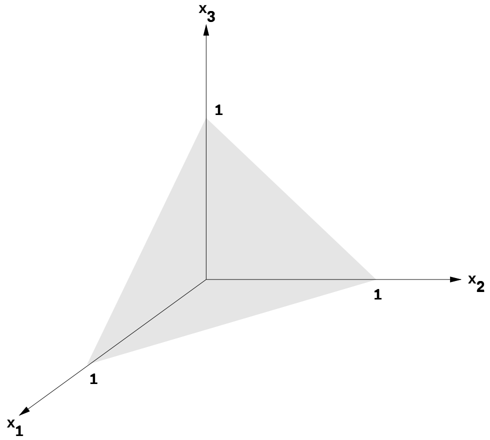

Fig. 1. The simplex Δ in $\mathbb{R}^3$.

while the mean payoff over the entire population is

$$\pi(\mathbf{x}) = \sum_{i=1}^{n} x_i \pi_i(\mathbf{x}) = \mathbf{x}' A \mathbf{x} \ . \tag{2}$$

In evolutionary game theory the assumption is made that the game is played over and over, generation after generation, and that the action of natural selection will result in the evolution of the fittest strategies. If successive generations blend into each other, the evolution of behavioral patterns can be described by a set of ordinary differential equations. A general class of evolution equations is given by:

$$\dot{x}_i = x_i g_i(\mathbf{x}) \tag{3}$$

where a dot signifies derivative with respect to time, and $g = (g_1, \ldots, g_n)$ is a function with open domain containing Δ. Here, the function g_i $(i \in J)$ specifies the rate at which pure strategy i replicates. It is usually required that the growth function g is *regular* [16], which means that it is Lipschitz continuous and that $g(\mathbf{x}) \cdot \mathbf{x} = 0$ for all $\mathbf{x} \in \Delta$. The former condition guarantees us that the system of differential equations (3) has a unique solution through any initial population state. The condition $g(\mathbf{x}) \cdot \mathbf{x} = 0$, instead, ensures that the simplex Δ is invariant under (3), namely any trajectory starting in Δ will remain in Δ.

Payoff-monotonic game dynamics represent a wide class of regular selection dynamics for which useful properties hold. Intuitively, for a payoff-monotonic dynamics the strategies associated to higher payoffs will increase at higher rate. Formally, a regular selection dynamics (3) is said to be payoff-monotonic if:

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \ \Leftrightarrow \ \pi_i(\mathbf{x}) > \pi_j(\mathbf{x}) \tag{4}$$

for all $\mathbf{x} \in \Delta$.

The following result, proved in [7], [13], generalizes the celebrated fundamental theorem of natural selection [8], [16].

Theorem 1: If the payoff matrix A is symmetric, then $\pi(\mathbf{x}) = \mathbf{x}' A \mathbf{x}$ is strictly increasing along any non-constant trajectory of any payoff-monotonic dynamics. In other words, $\dot{\pi}(\mathbf{x}(t)) \geq 0$ for all t, with equality if and only if $\mathbf{x} = \mathbf{x}(t)$ is a stationary point. Furthermore, a vector $\mathbf{x}$ is an asymptotically stable point if and only if it is a strict local maximizer of $\pi(\mathbf{x})$ in Δ.

A well-known subclass of payoff-monotonic game dynamics is given by:

$$\dot{x}_i = x_i \left(\phi(\pi_i(\mathbf{x})) - \sum_{j=1}^{n} x_j \phi(\pi_j(\mathbf{x})) \right) \tag{5}$$

where $\phi(u)$ is an increasing function of u. These models arise in modeling the evolution of behavior by way of imitation processes, where players are occasionally given the opportunity to change their own strategies [7], [16].

When ϕ is the identity function, i.e., $\phi(u) = u$, we obtain the standard replicator equations:

$$\dot{x}_i = x_i \left(\pi_i(\mathbf{x}) - \sum_{j=1}^{n} x_j \pi_j(\mathbf{x}) \right) \tag{6}$$

whose basic idea is that the average rate of increase $\dot{x}_i / x_i$ equals the difference between the average payoff of strategy i and the mean payoff over the entire population.

Another popular model arises when $\phi(u) = e^{\kappa u}$ which yields:

$$\dot{x}_i = x_i \left(e^{\kappa \pi_i(\mathbf{x})} - \sum_{j=1}^{n} x_j e^{\kappa \pi_j(\mathbf{x})} \right) \tag{7}$$

where κ is a positive constant. As κ tends to 0, the orbits of this dynamics approach those of the standard, first-order replicator model (6), slowed down by the factor κ; moreover, for large values of κ the model approximates the so-called “best-reply” dynamics [7], [8].

III. Annealed Imitation Dynamics for Maximum Clique

Let $G = (V, E)$ be an undirected graph, where $V = \{1, \cdots, n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges. A subset of vertices C is called a *clique* if all its vertices are mutually adjacent, i.e., for all $i, j \in C$, with $i \neq j$, we have $(i, j) \in E$. A clique is said to be *maximal* if it is not contained in any larger clique, and *maximum* if it is the largest clique in the graph. The maximum clique problem asks for a clique of maximum cardinality. Given a subset of vertices C, we will denote by $\mathbf{x}^C$ its *characteristic vector*, which is the point of Δ defined as

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

where $|C|$ denotes the cardinality of C.

Consider the following family of (standard) quadratic programs:

$$\begin{array}{ll} \text{maximize} & f_\alpha(\mathbf{x}) = \mathbf{x}'(A_G + \alpha I)\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{8}$$

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

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \ , \\ 0, & \text{otherwise} \ , \end{cases}$$

I is the identity matrix, and α is an arbitrary real parameter. The family includes as special cases the original Motzkin-Straus program [11] and its spurious-free regularized version proposed by Bomze [1] (corresponding to the cases $\alpha = 0$ and $\alpha = \frac{1}{2}$, respectively). The following theorem, which generalizes a result proven in [1], establishes a connection between the maximum clique problem and programs (8); see [3], [14] for proof.

Theorem 2: Let C be a subset of vertices of a graph G, and let $\mathbf{x}^C$ be its characteristic vector. Then, for any $0 < \alpha < 1$, C is a maximal (maximum) clique of G if and only if $\mathbf{x}^C$ is a local (global) solution of (8). Moreover, all solutions of (8) are strict and are characteristic vectors of maximal cliques of G.

By virtue of Theorem 1, the previous result implies that (characteristics vectors of) maximal cliques are in one-to-one correspondence to asymptotically stable points of any payoff-monotonic dynamics under payoff $A_G + \alpha I$, provided that $0 < \alpha < 1$. This naturally suggests using payoff-monotonic dynamics as a useful heuristic for the maximum clique problem. Clearly, there is no guarantee that the converged solution will be a *global* maximizer of f_α, and therefore that it will yield a *maximum* clique in G.

In an attempt to avoid local optima, we now follow [3] and investigate the stability properties of equilibria of payoff-monotonic dynamics when the parameter α is allowed to take on negative values. Indeed, we shall restrict our analysis to imitation dynamics (5).

For a given subset of vertices C, let:

$$\gamma(C) = \max_{i \notin C} \deg_C(i) - |C| + 1 \ . \tag{9}$$

where $\deg_C(i) = \sum_{j \in C} a_{ij}$ denotes the degree of vertex i relative to C. Note that if C is a maximal clique then $\gamma(C) \leq 0$. The next theorem shows that $\gamma(C)$ plays a key role in determining the stability of equilibria of imitation dynamics.

Theorem 3: Let C be a maximal clique of graph $G = (V, E)$, and let $\mathbf{x}^C$ be its characteristic vector. If $\gamma(C) < \alpha < 1$, then $\mathbf{x}^C$ is an asymptotically stable stationary point under any imitation dynamics (5) with payoff matrix $A = A_G + \alpha I$, and hence a (strict) local maximizer of f_α in Δ. Moreover, assuming $C \neq V$, if $\alpha < \gamma(C)$ then $\mathbf{x}^C$ becomes unstable.

Proof. Assume without loss of generality that $C = \{1, \cdots, m\}$ and suppose that $\gamma(C) < \alpha < 1$. To simplify notations, put $\mathbf{x} = \mathbf{x}^C$. The Jacobian of any regular selection (and hence imitation) dynamics $\dot{x}_i = x_i g_i(\mathbf{x})$ at $\mathbf{x}$ has the following block triangular form:

$$J(\mathbf{x}) = \begin{bmatrix} M(\mathbf{x}) & N(\mathbf{x}) \\ O & D(\mathbf{x}) \end{bmatrix} \tag{10}$$

where the entries of $M(\mathbf{x})$ and $N(\mathbf{x})$ are given by $x_i \frac{\partial g_i(\mathbf{x})}{\partial x_j}$, O is the (possibly empty) matrix containing all zeros, and

$$D(\mathbf{x}) = \text{diag}\{g_{m+1}(\mathbf{x}), \ldots, g_n(\mathbf{x})\} \ .$$

We shall see that the (relevant) eigenvalues of $J(\mathbf{x})$ are real and negative. This implies that $\mathbf{x}$ is a *sink* and hence an asymptotically stable point [6]. The fact that $\mathbf{x}$ is a strict local maximizer of f_α in Δ follows directly from Theorem 1.

The eigenvalues of $J(\mathbf{x})$ are those of $M(\mathbf{x})$ together with those of $D(\mathbf{x})$, and since $D(\mathbf{x})$ is diagonal its eigenvalues coincide with its diagonal entries, i.e., $g_{m+1}(\mathbf{x}), \ldots, g_n(\mathbf{x})$. This set of eigenvalues governs the asymptotic behavior of the external flow under the system obtained by linearization around $\mathbf{x}$, and are usually called *transversal* eigenvalues [8].

For imitation dynamics (5) the growth functions g_i have the following form:

$$g_i(\mathbf{x}) = \phi(\pi_i(\mathbf{x})) - \sum_{k=1}^{n} x_k \phi(\pi_k(\mathbf{x}))$$

but since C is a (maximal) clique, $\pi_k(\mathbf{x}) = \pi(\mathbf{x})$ for all $k \in C$, and therefore

$$g_i(\mathbf{x}) = \phi(\pi_i(\mathbf{x})) - \phi(\pi(\mathbf{x})) \ .$$

Moreover, ϕ is a strictly increasing function, and hence $g_i(\mathbf{x}) < 0$ if and only if $\pi_i(\mathbf{x}) < \pi(\mathbf{x})$. Now, since C is a maximal clique, $\pi_i(\mathbf{x}) = (A_G \mathbf{x})_i = \deg_C(i)/m$ for all $i > m$, and $\pi(\mathbf{x}) = (m - 1 + \alpha)/m$. But, for all $i > m$ we have $\deg_C(i) - m + 1 \leq \gamma(C) < \alpha$, and this yields $\pi_i(\mathbf{x}) < \pi(\mathbf{x})$. Hence all transversal eigenvalues are negative.

It remains to show that the eigenvalues of $M(\mathbf{x})$ are negative too. When $A = A_G + \alpha I$ we have:

$$M(\mathbf{x}) = \frac{\phi'(\pi(\mathbf{x}))}{m} \left[\left(1 - \frac{\phi(\pi(\mathbf{x}))}{\phi'(\pi(\mathbf{x}))} - \pi(\mathbf{x}) \right) \mathbf{e}\mathbf{e}' + (\alpha - 1) I \right]$$

where $\mathbf{e}\mathbf{e}'$ is the $m \times m$ matrix containings all ones, and the eigenvalues of $M(\mathbf{x})$ are

$$\lambda_1 = \frac{\phi'(\pi(\mathbf{x}))}{m} (\alpha - 1)$$

with multiplicity $m - 1$, and

$$\begin{aligned} \lambda_2 &= \frac{\phi'(\pi(\mathbf{x}))}{m} \left[\left(1 - \frac{\phi(\pi(\mathbf{x}))}{\phi'(\pi(\mathbf{x}))} - \pi(\mathbf{x}) \right) m + \alpha - 1 \right] \\ &= -\phi(\pi(\mathbf{x})) \end{aligned}$$

with multiplicity 1. Since $\alpha < 1$ and ϕ is strictly increasing, we have $\lambda_1 < 0$.

Since we analyze the behavior of imitation dynamics restricted to the simplex Δ, we are interested only in the eigenvalues of $J(\mathbf{x})$ associated with eigenvectors belonging to the tangent space $\mathbf{e}^\perp = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{e}'\mathbf{y} = 0\}$. It is simple to show [14] that the eigenvector associated with $\lambda_2 = -\phi(\pi(\mathbf{x}))$ is $\mathbf{x}$, and if $\mathbf{y}$ is an eigenvector of $J(\mathbf{x})$ associated with an eigenvalue $\lambda \neq -\phi(\pi(\mathbf{x}))$, then $\mathbf{e}'\mathbf{y} = \sum_i y_i = 0$. Hence, the eigenvalue λ_2 can be neglected in our analysis, and the remaining ones, including the transversal eigenvalues, are indeed all relevant. We have shown that these eigenvalues are negative and this concludes the first part of the proof.

Finally, to conclude the proof, suppose that $C \neq V$ (i.e., $m < n$) and $\alpha < \gamma(C) = \max_{i>m} \deg_C(i) - m + 1$. Then, there exists $i > m$ such that $m - 1 + \alpha < \deg_C(i)$ and hence, dividing by m, we get $\pi_i(\mathbf{x}) - \pi(\mathbf{x}) > 0$ and then $g_i(\mathbf{x}) = \phi(\pi_i(\mathbf{x})) - \phi(\pi(\mathbf{x})) > 0$, which implies that a transversal eigenvalue of $J(\mathbf{x})$ is positive, i.e., $\mathbf{x}$ is unstable. □

Theorem 3 provides us with an immediate strategy to avoid unwanted local solutions, i.e., maximal cliques that are not maximum. Suppose that C is a maximal clique in G that we want to avoid. By letting $\alpha < \gamma(C)$, its characteristic vector $\mathbf{x}^C$ becomes an unstable stationary point of any imitation dynamics under f_α, and thus will not be approached by any interior trajectory. Hence, if there is a clique D such that still $\gamma(D) < \alpha$ holds, there is a (more or less justified) hope to obtain in the limit $\mathbf{x}^D$, which yields automatically a larger maximal clique D. Unfortunately, two other cases could occur: (a) no other clique T satisfies $\gamma(T) < \alpha$, i.e., α has a too large absolute 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 2). 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.

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 simulated or mean-field annealing procedures. Of course, in our case the annealing parameter has no interpretation in terms of a hypothetical temperature. 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 imitation 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, a high-level description of the proposed algorithm is as follows:

1. Start with a sufficiently large negative α;
2. Let $\mathbf{b}$ be the barycenter of Δ and set $\mathbf{x} = \mathbf{b}$;
3. Run any imitation dynamics starting from $\mathbf{x}$, under $A_G + \alpha I$, until convergence and let $\mathbf{x}$ be the converged point;
4. Unless a stopping condition is met, increase α and goto 3;
5. Select $\hat{\alpha}$ with $0 < \hat{\alpha} < 1$ (e.g. $\hat{\alpha} = \frac{1}{2}$), run any imitation dynamics starting from current $\mathbf{x}$ under $A_G + \hat{\alpha} I$ 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 2.

It is clear that for the algorithm to work, we need to select an appropriate annealing schedule. To this end, we employ the following heuristic suggested in [3]. Suppose that the underlying graph is a random one in the sense that edges are generated independently of each other with a certain probability q (in applications, q will be replaced by the actual graph density), and suppose that C is an unwanted clique of size m. Take $\delta > 0$ small, say 0.01, and consider the quantity:

$$\overline{\gamma}_m = 1 - (1-q)m - \sqrt{mq(1-q)}\, \delta^{\nu} \tag{11}$$

where $\nu = 1/2(n-m)$. In [3] it is proven that $\gamma(C)$ exceeds $\overline{\gamma}_m$ with probability $1 - \delta$. Thus it makes sense to use $\overline{\gamma}_m$ as a heuristic proxy for the lower bound of $\gamma(C)$, to avoid being attracted by a clique of size m.

Furthermore, a well-known result due to Matula (see, e.g., [9]) accurately predicts the size of the maximum clique in random graphs with sufficiently many vertices. Let

$$M(n,q) = 2\log_{1/q} n - 2\log_{1/q}\log_{1/q} n + 2\log_{1/q}\frac{e}{2} + 1\ . \tag{12}$$

Matula proved that, as $n \to \infty$, the size of the maximum clique in an n-vertex q-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 $\overline{\gamma}_m$ and $\overline{\gamma}_{m-1}$, e.g, $\alpha = (\overline{\gamma}_m + \overline{\gamma}_{m-1})/2$, 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 $\overline{\gamma}_m$ and $\overline{\gamma}_{m-1}$ and update $\alpha = (\overline{\gamma}_m + \overline{\gamma}_{m-1})/2$ in step 4 as in the previous step. The whole process is iterated until either m reaches 1 or α becomes greater than zero.

IV. Experimental Results

In this section we present experiments of applying our annealed imitation heuristics to a selection of DIMACS benchmark graphs.[1] In our simulations, we used the following discrete-time models in steps 3 and 5 of the algorithm:

$$x_i(t+1) = \frac{x_i(t)\pi_i(t)}{\sum_{j=1}^{n} x_j(t)\pi_j(t)} \tag{13}$$

and

$$x_i(t+1) = \frac{x_i(t)e^{\kappa\pi_i(t)}}{\sum_{j=1}^{n} x_j(t)e^{\kappa\pi_j(t)}} \tag{14}$$

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

TABLE I

PERFORMANCE OF THE ANNEALED IMITATION HEURISTICS (AIH) AND MFA-CM OVER A SELECTION OF DIMACS GRAPHS (SEE TEXT FOR EXPLANATION). THE TIMES REPORTED IN THE MFA-CM COLUMN REFER TO AN IMPLEMENTATION ON A 32-PROCESSOR CONNECTION MACHINE CM-5, WHILE THE TIMES IN THE OTHER COLUMNS WERE OBTAINED ON A 200 MHZ PENTIUM II-BASED MACHINE.

			Clique size				*Iterations*			*Time (secs)*			
			AIH	*AIH*	*AIH*	*MFA*	*AIH*	*AIH*	*AIH*	*AIH*	*AIH*	*AIH*	*MFA*
Graph	*n*	ω	*1st*	κ*=3*	κ*=7*	*CM*	*1st*	κ*=3*	κ*=7*	*1st*	κ*=3*	κ*=7*	*CM*
c-fat200_1	200	12	12	12	12	6	324	146	85	1.15	0.55	0.37	3.1
c-fat200_2	200	24	24	24	24	24	41	31	17	0.21	0.14	0.12	2.9
c-fat200_5	200	58	58	58	58	58	37	25	18	0.21	0.16	0.15	2.9
brock200_1	200	21	18	18	19	19	1135	349	153	8.12	2.02	1.11	2.8
brock200_2	200	12	8	8	9	9	268	125	87	2.41	0.87	0.66	2.75
brock200_3	200	15	11	11	12	11	542	205	271	3.48	1.09	1.69	2.75
brock200_4	200	17	14	14	14	14	1367	555	255	19.62	4.98	2.30	2.75
brock400_1	400	27	21	22	23	24	1025	308	1054	20.2	20.2	63.88	18.3
brock400_2	400	29	19	19	22	21	1288	361	180	26.75	17.47	6.06	22.0
brock400_3	400	31	18	21	21	22	1278	301	152	23.79	16.11	5.78	20.9
brock400_4	400	33	22	23	22	23	608	382	356	13.35	24.71	15.55	19.0
brock800_1	800	23	16	15	17	16	602	237	135	62.93	44.18	28.47	151.7
brock800_2	800	24	16	18	18	16	529	308	281	57.22	115.07	67.96	134.7
brock800_3	800	25	16	18	19	16	794	322	192	95.95	93.99	38.51	153.7
brock800_4	800	26	17	17	16	15	648	224	118	66.19	62.65	25.4	147.1
san200_0.7_1	200	30	15	15	15	—	1364	679	391	8.36	2.68	1.74	—
san200_0.7_2	200	18	12	12	12	—	1426	925	658	8.47	4.07	2.95	—
san200_0.9_1	200	70	45	45	45	42	3518	1495	852	20.77	6.04	3.7	2.75
san200_0.9_2	200	60	40	40	38	—	3520	1595	781	24.66	7.03	3.71	—
san200_0.9_3	200	44	32	30	31	33	1613	761	777	10.12	3.49	4.09	2.76
sanr200_0.7	200	18	16	17	18	18	687	279	586	5.07	1.67	6.07	2.75
sanr200_0.9	200	42	37	38	41	41	1890	737	442	10.61	3.34	2.61	2.6
sanr400_0.5	400	13	11	11	11	—	321	173	84	6.76	9.24	3.2	—
sanr400_0.7	400	$\geq$21	17	21	21	—	658	1014	615	14.15	102.2	34.05	—

which are well-known discretizations of equations (6) and (7), respectively. As for the exponential dynamics, here we report results obtained using $\kappa = 3$ and $\kappa = 7$.

For each graph considered, the 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 imitation processes were iterated until the (squared) distance between two successive states became smaller than 10^{-10}. At the final cycle (step 5), the parameter $\hat{\alpha}$ was set to $1/2$, and the dynamics were stopped when either a maximal clique (i.e., a local maximizer of $f_{1/2}$ on Δ) 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 in the graph). In the latter case the converged vector was perturbed and the algorithm restarted from the new perturbed point. Because of the one-to-one correspondence between local maximizers and maximal cliques this situation corresponds to convergence to a saddle point.

In [9], Jagota developed several variations of the Hopfield model, both discrete and continuous, to approximate maximum clique. The best results were obtained using a stochastic steepest descent dynamics and a mean-field annealing (MFA) algorithm. His heuristics rank among the most powerful ones in the neural-network literature. These algorithms, however, were excessively slow and this motivated Jagota et al. [10] to improve their running time. Specifically, the MFA heuristic was implemented on a Connection Machine, and a crude two-temperature annealing strategy was used. The resulting MFA-CM algorithm was found to perform nearly as well as the original version, while being considerably faster (on a 400-vertex test graph MFA-CM was about 20 times faster).

Table 1 shows the performance figures obtained by our annealed imitation heuristics (columns AIH) and by MFA-CM (data for the latter algorithm are from [10]). Columns marked with n and ω contain the number of vertices in the graph and the size of the max imum clique (or a lower bound), respectively. The labels “1st” refers to the first-order replicator dynamics (13) whereas labels “$\kappa = 3$” and “$\kappa = 7$” refer to the exponential dynamics (14). Columns “Clique size” contain the size of the clique found by the competing algorithms, and columns “Iterations” list the total number of iterations performed by AIH (no such data are available for MFA-CM). Computing times (columns “Time”) for our algorithms refer to a (non-optimized) C implementation on a machine equipped with a 200 MHz Pentium II processor. As for the MFA-CM the times listed in the table refer to a 32-processor Connection Machine CM-5 implementation.

Several conclusions can be drawn from these experiments. Firstly, it is clear that the exponential version of the algorithm not only performed typically less iterations than its linear counterpart, but it produced quite often better results (and only once worse). Also, as expected, it is evident that the parameter κ governs the speed of the imitation dynamics: usually, the higher the κ the faster the process. In discrete-time simulations, however, large values of κ can lead to oscillatory behavior [12], [14]. Finally, it is impressive that our heuristics perform on average as well as MFA-CM in terms of quality of solutions, while being dramatically faster. Again, note that our simulations were performed on a sequential machine, while MFA-CM was executed on a 32-processor one.

V. Conclusions

We have introduced a wide class of heuristics for the maximum clique problem (MCP) whose basic ingredients are: (1) a parameterized continuous formulation of MCP, (2) an instability analysis of equilibria of imitation dynamics from evolutionary game theory, and (3) a principled way of varying a regularization parameter during the evolution process so as to avoid inefficient local solutions. Experiments on various benchmark graphs have shown that this "annealed imitation" class contains algorithms which are dramatically faster than and as accurate as state-of-the-art neural network heuristics for maximum clique. More extensive experimental results that confirm these findings can be found in a forthcoming paper [14].

Acknowledgments

The author would like to thank C. Rossi for his help in early stages of this work and I. M. Bomze for many useful discussions.

References

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

[2] 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. Boston, MA: Kluwer, 1999, pp. 1–74.

[3] I. M. Bomze, M. Budinich, M. Pelillo, and C. Rossi, "Annealed replication: A new heuristic for the maximum clique problem," *Discr. Appl. Math.*, vol. 121, pp. 27–49, 2002.

[4] I. M. Bomze, M. Pelillo, and V. Stix, "Approximating the maximum weight clique using replicator dynamics," *IEEE Trans. Neural Networks*, vol. 11, no. 6, pp. 1228–1241, 2000.

[5] A. H. Gee and R. W. Prager, "Polyhedral combinatorics and neural networks," *Neural Computation*, vol. 6, pp. 161–180, 1994.

[6] M. H. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. New York: Academic Press, 1974.

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

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

[9] A. Jagota, "Approximating maximum clique with a Hopfield neural network," *IEEE Trans. Neural Networks*, vol 6, pp. 724–735, 1995.

[10] A. Jagota, L. Sanchis, and R. Ganesan, "Approximately solving maximum clique using neural networks and related heuristics," In *Cliques, Coloring, and Satisfiability*, D. Johnson and M. Trick, Eds. Providence, RI: American Mathematical Society, 1996, pp. 169–204.

[11] T. S. Motzkin and E. G. Straus, "Maxima for graphs and a new proof of a theorem of Turán," *Canad. J. Math.*, vol. 17, pp. 533–540, 1965.

[12] M. Pelillo, "Replicator equations, maximal cliques, and graph isomorphism," *Neural Computation*, vol. 11, no. 8, pp. 2023–2045, 1999.

[13] M. Pelillo, "Matching free trees, maximal cliques, and monotone game dynamics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 11, pp. 1535–1541, 2002.

[14] M. Pelillo, "Payoff-monotonic game dynamics and the maximum clique problem," forthcoming.

[15] 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.

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