A Continuous-Based Approach for Partial Clique Enumeration

Samuel Rota Bulò, Andrea Torsello, and Marcello Pelillo

"Ca' Foscari" University of Venice,
{srotabul,torsello,pelillo}@dsi.unive.it

Abstract. In many applications of computer vision and pattern recognition which use graph-based knowledge representation, it is of great interest to be able to extract the K largest cliques in a graph, but most methods are geared either towards extracting the single clique of maximum size, or enumerating all cliques, without following any particular order. In this paper we present a novel approach for partial clique enumeration, that is, the extraction of the K largest cliques of a graph. Our approach is based on a continuous formulation of the clique problem developed by Motzkin and Straus, and is able to avoid extracting the same clique multiple times. This is done by casting the problem into a game-theoretic framework and iteratively rendering unstable the solutions that have already been extracted.

1 Introduction

Many applications of computer vision and pattern recognition which use graph-based knowledge representation have to deal with the problem of finding complete subgraphs (cliques) of their structural descriptions. Examples of problems that have successfully been reduced to a clique-finding problem range from matching [2], to category learning and knowledge discovery [17, 9], to clustering [1, 18], to stereo matching [13], to name just a few. Furthermore, clique finding is also linked with the learning of graphical structure by the Hammersley-Clifford theorem [11].

The maximum clique problem (MCP) deals with the challenge of finding the largest complete subgraph of an undirected and unweighted graph. It falls in the crucial class of NP-Complete problems, whose intractability forces us to fall back on approximation methods. Unfortunately, even approximating the MCP is intractable [12]. Due to this pessimistic state of affairs much attention has gone into developing efficient heuristics for the MCP, for which no formal guarantee of performance may be provided, but are nevertheless useful in practical applications. We refer to Bomze et al. [5] for a survey concerning algorithms, applications, and complexity issues of this important problem.

In a recent series of papers [19, 10, 7] we find approaches that are centered around a classical result from graph theory due to Motzkin and Straus [16], that allows us to formulate the MCP as a continuous quadratic optimization problem with simplex constraints. This program is typically solved by the replicator dynamics, well-known continuous- and discrete-time dynamical systems, developed and studied in the field of evolutionary game theory and for which it can be shown that there exists a one-to-one correspondence between stable points and maximal cliques of the corresponding graph.

In several contexts, it is of great interest to have an approach that can extract several large cliques, in particular, we would like to be able to efficiently extract the K largest cliques in a graph. For example in knowledge discovery, where categories are abstracted in terms of cliques, each element can belong to multiple categories, and hence we are interested in discovering more than one category [17, 9]. In a completely different domain, Horaud and Skordas [13] use the largest cliques to find stereo correspondences in image pairs. While exact search-based enumerative algorithms are guaranteed to generate every maximal clique, in general they cannot guarantee a specific order in which these are found, in particular they give no guarantee about the relative size of the clique obtained at each step.

In this paper we present an approach which uses a continuous formulation to enumerate a user-defined number of large cliques. Ideally, we would like to obtain the K largest maximal cliques after a small number of enumerations. Clearly, the actual size of the extracted cliques depends on the effectiveness of the continuous formulation, but, experimental evidence tells us that the approach performs fairly well [19].

The basis of this approach rests on the fact that under a certain family of quadratic problems, there is a bijection between asymptotically stable points of the replicator dynamics and maximal cliques. Once we have extracted a maximal clique, we would like to avoid that the dynamics converge to the same clique. Intuitively, what our method does is to render unstable the associated rest point. To do this, we deal with directed graphs, and apply a particular asymmetric graph-extension for every maximal clique we want to render unstable. By iterating this extension process, we progressively reduce the set of asymptotically stable points of the replicator dynamics, and, hence, we obtain a continuous-based enumerative algorithm.

2 A Family of Quadratic Programs for Maximum Clique

Let $G = (V, E)$ be an undirected graph without self-loops, where $V = \{1, 2, \ldots, n\}$ is the set of vertices and $E \subseteq V \times V$ the set of edges. Two vertices $u, v \in V$ are *adjacent* if $(u, v) \in E$. A subset C of vertices in G is called a *clique* if all its vertices are mutually adjacent. It is a *maximal clique* if it is not subset of other cliques in G. It is a *maximum clique* if no other cliques of G have a strictly greater cardinality. The cardinality of a maximum clique of G is also called *clique number* and denoted by $\omega(G)$.

The *adjacency matrix* of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$ where $a_{ij} = \chi_E((i, j))$. Here, $\chi_A(i)$ represents the indicator function that returns 1 if $i \in A$, 0 otherwise.

Consider the following constrained quadratic program.

$$\text{maximize} \quad f_\alpha(\boldsymbol{x}) = \boldsymbol{x}'(A_G + \alpha I)\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x} \in \Delta \subset \mathbb{R}^n, \tag{1}$$

where n is the order of G, I the identity matrix, and α is a real parameter. In 1965 Motzkin-Straus [16] established a connection between the maximum clique problem and the program in (1) with $\alpha = 0$; they related the clique number of G to global solutions $\boldsymbol{x}^*$ of the program through the formula $\omega(G) = (1 - f_0(\boldsymbol{x}^*))^{-1}$, and showed that a subset of vertices C with cardinality $|C|$ is a maximum clique of G if and only if [1] its *characteristic vector* $\boldsymbol{x}^C \in \Delta$, where $x_i^C = \chi_C(i)|C|^{-1}$, is a global maximizer of f_0 on Δ. Gibbons, Hearn, Pardalos and Ramana [10], and Pelillo and Jagota [20], extended the Motzkin-Straus theorem by providing a characterization of maximal cliques in terms of local maximizers of f_0 in Δ.

A drawback of the original Motzkin-Straus formulation is the existence of "spurious" solutions, i.e., maximizers of f_0 that are not in the form of characteristic vectors. Bomze et al.[6] proved that for $0 < \alpha < 1$ all local maximizer of (1) are strict and are in one-to-one relation with the characteristic vectors of the maximal cliques of G, hence, overcoming the problem.

In order to find the maxima of (1) we cast the problem in a game-theoretic setting and use the replicator dynamics, a well-known formalization of the selection process. In the next section we will review some concepts from evolutionary game theory that will be useful throughout the paper and provide the link between game theory and maximal cliques.

3 A game-theoretic perspective

Let $O = \{1, 2, \ldots, n\}$ be the set of *pure strategies* available to the players and $A = (a_{ij})$ the $n \times n$ payoff or utility matrix [23] where a_{ij} is the payoff that a player gains when playing the strategy i against an opponent playing j. In biological contexts, payoff are typically measured in terms of Darwinian fitness or reproductive success whereas in economics applications, they usually represent firms' profits or consumers' utilities.

A *mixed strategy* is a probability distribution $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)'$ over the available strategies in O. Mixed strategies clearly lie in the standard simplex of the n-dimensional Euclidean space $\Delta = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{e}'\boldsymbol{x} = 1, \quad \boldsymbol{x} \geq 0\}$ where $\boldsymbol{e}$ is the vector with all components equal to 1.

The *support* of a mixed strategy $\boldsymbol{x} \in \Delta$, denoted by $\sigma(\boldsymbol{x})$, defines the set of elements with non-zero probability: $\sigma(\boldsymbol{x}) = \{i \in O : x_i > 0\}$

The expected payoff that a player obtains by playing the element i against an opponent playing a mixed strategy $\boldsymbol{x}$ is $u(\boldsymbol{e}^i, \boldsymbol{x}) = (A\boldsymbol{x})_i = \sum_j a_{ij}x_j$, where $\boldsymbol{e}^i$ is the vector with all components equal zero except for the i^{th}-component

[1] In the original paper Motzkin-Straus proved the "only-if" part of this theorem. The converse however is a straightforward consequence of their result (Pelillo & Jagota, 1995) [20].

which is equal to 1. Hence, the expected payoff received by adopting a mixed strategy $\boldsymbol{y}$ is $u(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{y}' A \boldsymbol{x}$.

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

A well-known formalization of the selection process is given by the replicator equations [23]: $\dot{x}_i = x_i(u(\boldsymbol{e}^i, \boldsymbol{x}) - u(\boldsymbol{x}, \boldsymbol{x}))$.

If the payoff matrix is symmetric then $\boldsymbol{x}' A \boldsymbol{x}$ is strictly increasing along any non-constant trajectory of any payoff-monotonic dynamics [23]. This result allows us to establish a bijective relation between the local solutions of program (1), namely characteristic vectors of maximal cliques of G, and asymptotically stable points of the replicator dynamics with payoff matrix $A_G + \alpha I$ and $0 < \alpha < 1$.

In order to obtain enumeration of maximal cliques through a continuous formulation we move from undirected graphs to directed graphs, or, in other words, from symmetric payoff matrices to asymmetric payoff matrices. If we loosen the symmetry constraint, then all the results that bind local solutions to asymptotically stable points and maximal cliques do not hold any longer, and $\boldsymbol{x}' A \boldsymbol{x}$ is not a Lyapunov function for the dynamics.

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

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

A strategy $\boldsymbol{x}$ is said to be an *evolutionary stable strategy* (ESS) if it is a Nash equilibrium and for all $\boldsymbol{y} \in \Delta$ such that $u(\boldsymbol{y}, \boldsymbol{x}) = u(\boldsymbol{x}, \boldsymbol{x})$ we have that $u(\boldsymbol{x}, \boldsymbol{y}) > u(\boldsymbol{y}, \boldsymbol{y})$. Intuitively, *ESS* are strategies such that any small deviation from them will lead to an inferior payoff.

Consider the following quadratic program

$$\text{maximize} \quad \pi(\boldsymbol{x}) = \boldsymbol{x}' A \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x} \in \Delta \subset \mathbb{R}^n, \tag{2}$$

where A is a symmetric matrix. We have that $\boldsymbol{x}$ is a Nash equilibrium of a two-player game with payoff matrix A, if and only if it satisfies the Karush-Kuhn-Tucker (KKT) conditions for (2). In fact the KKT conditions can be written as

$$u(\boldsymbol{e}^i, \boldsymbol{x}) = (A\boldsymbol{x})_i \begin{cases} = \lambda & \text{if } i \in \sigma(\boldsymbol{x}) \\ \leq \lambda & \text{if } i \notin \sigma(\boldsymbol{x}) \end{cases}$$

for some real λ. However it is clear that $\lambda = \boldsymbol{x}' A \boldsymbol{x} = u(\boldsymbol{x}, \boldsymbol{x})$ and what we obtain is exactly the definition of a Nash equilibrium. Hence local solution of (2) are indeed Nash equilibria, but the converse does not necessarily hold.

A two-player symmetric game where the payoff matrix is also symmetric is called *doubly-symmetric* game. Loser and Akin [15] showed that for all doubly symmetric games the average payoff $u(\boldsymbol{x}, \boldsymbol{x})$ increases along every non-stationary solution path to the replicator dynamics.

If we consider program (1) with $0 < \alpha < 1$, we have that the set of ESS is equivalent to the set of maximal cliques of the related graph. We refer to [8] and [6] for a deeper insight of the relation between ESS and maximal cliques.

Through this change in perspective, we can move from a constrained maximization problem, to a game-theoretic setting. Instead of finding local solutions of a quadratic program, we look for ESS of a doubly symmetric game. The advantage of this new approach is that we can generalize the Motzkin-Straus result to non symmetric payoff matrices and, hence, directed graphs.

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

In [22] we find the following result.

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

We have already seen that if we consider an undirected graph G and the payoff matrix $A_G + \alpha I$ with $0 < \alpha < 1$, then the ESSs of the related two-player game are in one-to-one correspondence with maximal cliques of G. However if we strengthen the constraint on α to lay between 0.5 and 1, then we can see that the concept of saturated doubly-linked clique is a direct generalization to the asymmetric case of the concept of maximal clique, i.e. ESSs are in one-to-one correspondence with saturated doubly-linked cliques.

4 Continuous-based enumeration

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

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

Theorem 2. *Let $G = (V, E)$ be an undirected graph, Σ be a set of maximal cliques of G and A be the adjacency matrix of the Σ-extension G_{Σ} of G. Let Φ be a two person symmetric game with payoff matrix $A + \alpha I$ with $0.5 \leq \alpha < 1$.*

Then $\boldsymbol{x}$ is an ESS equilibrium of $\mathcal{G}$ if and only if it is the characteristic vector of a maximal clique of G not in Σ.

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

($\Leftarrow$) Consider $S \notin \Sigma$ a maximal clique of G. Then by construction of G^Σ, it is a saturated doubly-linked clique of G^Σ and hence by [22] $\boldsymbol{x}^S$ is an ESS equilibrium of $\mathcal{G}$.

The continuous-based enumerative algorithm uses this result in the following way. We iteratively find an asymptotically stable point through the replicator dynamics. If we have an ESS, then we have found a new maximal clique[2]. After that, we extend the graph by adding the newly extracted clique to Σ, hence rendering its associated strategy unstable, and reiterate the procedure until we have enumerated the selected number of maximal cliques.

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

5 Experimental results

In this section we asses the ability of our continuous-based enumerative heuristic (CEH) to extract large cliques. To this end we apply the enumeration to the extraction of the maximum clique from the DIMACS benchmark graphs. For each graph, we run the method 20 times and took for each run, the maximum between the first 300 enumerated maximal cliques.

In order to extract the maximal clique from a characteristic vector, we avoid the standard thresholding technique on the value of each component of the characteristic vector, but rather we use the values of each component as indicators for a New-Best-In heuristic [14]. This is a sequential greedy heuristic that, starting from an empty set of vertices, iteratively constructs a maximal clique by inserting the clique-preserving vertex v that maximizes $w_v + \sum_{j\in S} \chi_E((v,j))w_j$ where E is the set of edges of the graph, S is the set of clique-preserving vertices and $\boldsymbol{w} = (w_1, \dots, w_n)$ is a weight vector, in our case the mixed strategy obtained through the replicator dynamics. An added advantage of this approach is that we can stop the dynamics before the dominated strategies where driven to a hard zero, and still be able to extract the associated maximal clique. This can significantly improve the speed of the approach as a lower number of iterations are needed to extract each clique. the method.

[2] we have never experienced an AS point that was not an ESS, so we strongly believe that theorem (2) can be generalized to asymptotically stable points.

				Clique size									
Name	#	ρ	BR	Min	Avg.(S.Dev.)	Max	K	Avg. time	IHN	AIH	CBH	QMS	RLS
brock200_1	200	0.75	21	20	20.050 (0.224)	21	156	7.85s	-	20	20	21	21
brock200_2	200	0.50	12	10	10.400 (0.503)	11	24	7.25s	-	10	12	12	12
brock200_3	200	0.61	15	13	13.750 (0.444)	14	19	7.40s	-	13	14	15	15
brock200_4	200	0.66	17	15	15.850 (0.587)	17	2	7.50s	-	16	16	17	17
brock400_1	400	0.75	27	23	23.800 (0.410)	24	49	19.90s	-	24	23	27	25
brock400_2	400	0.75	29	23	23.450 (0.510)	24	24	20.00s	-	24	24	29	29
brock400_3	400	0.75	31	23	23.700 (0.657)	25	10	19.90s	-	24	23	31	25
brock400_4	400	0.75	33	23	23.900 (0.641)	25	77	19.90s	-	23	24	33	33
brock800_1	800	0.65	23	19	19.600 (0.503)	20	4	51.35s	-	20	20	23	21
brock800_2	800	0.65	24	19	19.900 (0.447)	21	3	51.60s	-	18	19	24	21
brock800_3	800	0.65	25	19	19.750 (0.550)	21	245	51.30s	-	19	20	25	22
brock800_4	800	0.65	26	19	19.550 (0.510)	20	17	51.25s	-	19	19	26	21
c-fat200-1	200	0.08	12	12	12 (0)	12	1	7.40s	12	12	12	12	12
c-fat200-2	200	0.16	24	24	24 (0)	24	1	7.95s	24	24	24	24	24
c-fat200-5	200	0.43	58	58	58 (0)	58	1	18.80s	58	58	58	58	58
c-fat500-1	500	0.04	14	14	14 (0)	14	1	24.70s	14	14	14	14	14
c-fat500-2	500	0.07	26	26	26 (0)	26	1	28.90s	26	26	26	26	26
c-fat500-5	500	0.19	64	64	64 (0)	64	1	41.50s	64	64	64	64	64
c-fat500-10	500	0.37	126	126	126 (0)	126	1	62.75s	-	126	126	126	126
hamming6-2	64	0.90	32	32	32 (0)	32	1	2.79s	32	32	32	32	32
hamming6-4	64	0.35	4	4	4 (0)	4	1	2.23s	4	4	4	4	4
hamming8-2	256	0.97	128	128	128 (0)	128	1	9.90s	128	128	128	128	128
hamming8-4	256	0.64	16	16	16 (0)	16	1	10.15s	16	16	16	16	16
johnson8-2-4	28	0.56	4	4	4 (0)	4	1	1.11s	4	4	4	4	4
johnson8-4-4	70	0.77	14	14	14 (0)	14	1	2.68s	14	14	14	14	14
johnson16-2-4	120	0.76	8	8	8 (0)	8	1	4.28s	8	8	8	8	8
johnson32-2-4	496	0.88	16	16	16 (0)	16	1	25.50s	16	16	16	16	16
keller4	171	0.65	11	11	11 (0)	11	1	2.20s	-	9	10	11	11
keller5	776	0.75	27	25	26.600 (0.681)	27	5	28.55s	-	16	21	26	27
keller6	3361	0.82	$\geq$59	51	52.250 (0.910)	54	45	761.75s	-	31	-	53	59
MANN_a9	45	0.927	16	16	16 (0)	16	1	1.87s	-	16	16	16	16
MANN_a27	378	0.990	126	125	125.100 (0.308)	126	124	36.30s	-	117	121	125	126
MANN_a45	1035	0.996	345	341	342.100 (0.641)	343	85	528.00s	-	-	-	342	345
p_hat300-1	300	0.24	8	8	8 (0)	8	1	11.15s	8	8	8	8	8
p_hat300-2	300	0.49	25	25	25 (0)	25	9	12.80s	25	25	25	25	25
p_hat300-3	300	0.74	36	34	34.550 (0.605)	36	218	13.75s	36	36	36	35	36
p_hat500-1	500	0.25	9	9	9 (0)	9	1	22.65s	9	9	9	9	9
p_hat500-2	500	0.50	36	34	35.300 (0.571)	36	36	31.05s	36	36	35	36	36
p_hat500-3	500	0.75	50	48	48.500 (0.510)	49	7	35.40s	49	49	49	48	50
p_hat700-1	700	0.25	11	9	10.700 (0.571)	11	2	35.45s	11	9	11	11	11
p_hat700-2	700	0.50	44	43	43.400 (0.503)	44	1	56.30s	44	44	44	44	44
p_hat700-3	700	0.75	62	60	60.500 (0.607)	62	1	67.25s	61	60	60	62	62
p_hat1000-1	1000	0.25	10	10	10 (0)	10	1	60.55s	10	-	-	10	10
p_hat1000-2	1000	0.50	46	44	45.250 (0.550)	46	26	104.00s	46	-	-	45	46
p_hat1000-3	1000	0.75	68	63	63.900 (0.718)	65	50	127.80s	68	-	-	65	68
p_hat1500-1	1500	0.25	12	11	11 (0)	11	1	114.95s	-	10	11	12	12
p_hat1500-2	1500	0.50	65	62	63.150 (0.745)	64	51	255.60s	-	64	63	64	65
p_hat1500-3	1500	0.75	94	88	89.750 (1.333)	92	178	326.75s	-	92	94	91	94
san200_0.7_1	200	0.70	30	19	29.050 (2.964)	30	11	7.70s	30	15	15	30	30
san200_0.7_2	200	0.70	18	13	13 (0)	13	1	7.45s	15	12	12	18	18
san200_0.9_1	200	0.90	70	70	70 (0)	70	2	9.60s	70	46	46	70	70
san200_0.9_2	200	0.90	60	57	59.800 (0.696)	60	2	8.95s	41	39	36	60	60
san200_0.9_3	200	0.90	44	36	39.800 (2.375)	44	85	8.70s	-	35	30	40	44
san400_0.5_1	400	0.50	13	7	7.900 (0.308)	8	20	16.55s	-	7	8	13	13
san400_0.7_1	400	0.70	40	40	40 (0)	40	2	20.05s	40	20	20	40	40
san400_0.7_2	400	0.70	30	18	21.250 (4.315)	30	32	18.75s	30	15	15	30	30
san400_0.7_3	400	0.70	22	15	15.800 (0.410)	16	7	18.15s	-	12	14	18	22
san400_0.9_1	400	0.90	100	100	100 (0)	100	1	27.40s	100	51	50	100	100
san1000	1000	0.50	10	8	8.400 (0.503)	9	30	59.90s	10	8	8	15	15
sanr200_0.7	200	0.70	18	17	17.850 (0.366)	18	29	7.60s	17	18	18	18	18
sanr200_0.9	200	0.90	42	39	40.550 (0.686)	42	243	8.80s	41	41	41	41	42
sanr400_0.5	400	0.50	13	12	12.750 (0.444)	13	30	17.25s	12	13	12	13	13
sanr400_0.7	400	0.70	21	19	20.250 (0.550)	21	59	19.10s	21	21	20	20	21

Table 1. Comparative results on DIMACS benchmark graphs.

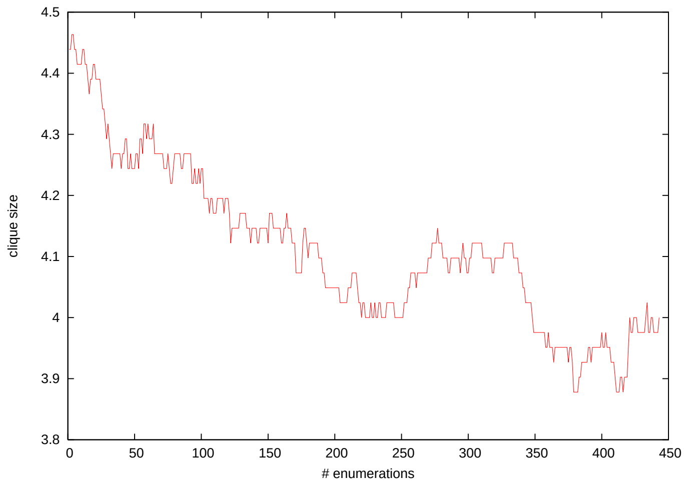

Fig. 1. Average size of the extracted clique over the number of extractions.

In figure (1) we show the results obtained by enumerating about 450 maximal cliques of a random graph of order 100 and density 0.25. For each enumeration the graph plots the average size of the last 40 cliques in order to clarify the descending tendency. As it can be seen, the approach enumerates the clique in approximately decreasing order of size.

Table (1) shows the results obtained with CEH on the DIMACS benchmark. We compared our approach with a neural-network-based heuristic, Inverted Neural Network (IHN) [4] and with other Motzkin-Straus -based heuristics for MCP, i.e. Annealed Imitation Heuristic (AIH) [21], Continuous Based Heuristic (CBH) [10] and Qualex Motzkin-Straus (QMS) [7]. Furthermore, we also compare the approach with Reactive Local Search (RLS) [3], a state-of-the-art heuristic search-based algorithm for MCP.

The table includes the name of the DIMACS graph (Name), the number of vertices (#), the graph density (ρ), the optimum size (BR) . In the second part we find the results obtained with CEH: the minimum (Min), the average size and standard deviation (Avg), and the maximum size (Max) obtained among 20 runs of CEH, each enumerating 300 cliques. The column labeled with K provides the number of enumerations required before the maximum was found. The running times are referred to an unoptimized C implementation on 64-bit PC with a 2 GHz AMD Opteron Processor and 1 Gb RAM. The computation times of the other methods can be found in their respective papers, however they are not comparable because they refer to experiments conducted with different hardware and software settings.

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

Though CEH, AIH and CBH use the same continuous-based technique, CEH outperforms both algorithms on all DIMACS graphs. The comparison with IHN is not so meaningful because it has been tested on few graph instances, but we can notice that for all families except *sanr* the approaches are comparable, while on the *sanr* graphs CEH is the best performer.

QMS seems to be particularly good on the *brock* family, where it outperforms all other approaches. However, CEH outperforms QMS on *MANN*, *keller* and *sanr* families and performs slightly better on the *p_hat* family, while QMS performs slightly better on the *san* family.

We can see that RLS provides the best performance on almost all DIMACS benchmarks, with the exception of the *brock* family, where QMS is indeed the best. It is worth reminding that RLS is a search based-approach while all the other are continuous-based.

The column K of the tables represent the minimum number of enumerations before the best clique size for the algorithm has been reached. It is in some sense a measure of the action of the enumeration in order to achieve the maximum result. We see that the easy instances of the benchmark are solved within the first enumeration, while more difficult ones, for example *brock*, *san*, *sanr*, require a higher number of enumerations.

6 Conclusions

In this paper we developed a partial clique enumeration algorithm based on the Motzkin-Straus formulation. In order to perform the enumeration, we deal with a directed form of the clique problem and we deal with an asymmetric extension. This way we lose the original connection with the quadratic problem, but, by casting the problem into a game-theoretic framework, we are able to prove a relationship between the evolutionary stable strategies and maximal cliques that have not yet been enumerated. In order to asses the usefulness of the approach we compared it with several state-of-the-art approaches on the problem of extracting the maximum clique from the DIMACS benchmark graphs. The approach proved to be superior to other continuous-based approaches and competitive with the state of the art search heuristics.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, Seattle, Washington, June 1998
2. H.G. Barrow and R.M. Burstall, Subgraph Isomorphism, Matching Relational Structures, and Maximal Cliques, *Information Processing Letters*, vol. 4, no. 4, pp. 83-84, 1976.
3. R. Battiti and M. Protasi, Reactive local search for the maximum clique problem, *Technical Report TR-95-052*, International Computer Science Institute, Berkeley, CA, 1995.

4. A. Bertoni, P. Campadelli, and G. Grossi, "A neural algorithm for the maximum clique problem: analysis, experiments and circuit implementation," *Algorithmica*, vol. 33, no. 1, pp. 71–88, 2002.
5. I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization, Supplement Volume A*, D. Z. Du and P. M. Pardalos, Eds. Boston, MA: Kluwer Academic Publishers, 1999, pp. 1–74.
6. I. M. Bomze, M. Pelillo, and R. Giacomini, "Evolutionary approach to the maximum clique problem: Empirical evidence on a larger scale," in *Developments in Global Optimization*, I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, Eds. Dordrecht, The Netherlands: Kluwer, 1997, pp. 95–108.
7. S. Busygin, "A new trust region technique for the maximum weight clique problem," 2002.
8. C. Cannings and G. Veckers, "Patterns of ess's." *J. Theor. Biol.*, no. 132.
9. D. Dmitry and R. Ari, "Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words," *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Association for Computational Linguistics*, pp. 297–304, 2006
10. L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana, "Continuous characterizations of the maximum clique problem," *Math. Oper. Res.*, vol. 22, pp. 754–768, 1997.
11. J. Hammersley, and P. Clifford, Markov fields on finite graphs and lattices, unpublished manuscript 1971.
12. J. Hastad, "Clique is hard to approximate within $n^{1-\varepsilon}$," in *Proc. 37th Ann. Symp. Found. Comput. Sci.*, 1996, pp. 627–636.
13. R. Horaud and T. Skordas, "Stereo correspondence through feature grouping and maximal cliques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 11, pp. 1168–1180, 1989.
14. R. Kopf and G. Ruhe, A computational study of the weighted independent set problem for general graphs, *Found. Control Engin.*, Vol. 12: 167–180, 1987.
15. V. Losert and E. Akin, "Dynamics of games and genes: Discrete versus continuous time," *J. Math. Biol.*, vol. 17, pp. 241–251, 1983.
16. 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.
17. M. Nina and R. Dana and S. Ram, A New Conceptual Clustering Framework, *Machine Learning*, Vol. 56: 115–151, 2004
18. M. Pavan and M. Pelillo, Dominant sets and pairwise clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):167-172, 2007.
19. M. Pelillo "Replicator equations, maximal cliques, and graph isomorphism," *Neural Computation*, vol. 11, no. 8, pp. 2023–2045, 1999.
20. M. Pelillo and A. Jagota, "Feasible and infeasible maxima in a quadratic program for maximum clique," *J. Artif. Neural Networks*, vol. 2, pp. 411–420, 1995.
21. M. Pelillo and A. Torsello, Payoff-Monotonic Game Dynamics and the Maximum Clique Problem, *Neural Computation*, 18: 1215–1258, 2006.
22. A. Torsello, S. R. Bulò, and M. Pelillo, "Grouping with asymmetric affinities: A game-theoretic perspective," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 2006, pp. 292–299.
23. J. W. Weibull, *Evolutionary Game Theory.* Cambridge, MA: MIT Press, 1995.