J Optim Theory Appl
DOI 10.1007/s10957-016-1043-6

Solving the Maximum Clique Problem with Symmetric Rank-One Non-negative Matrix Approximation

Melisew Tefera Belachew[1] · **Nicolas Gillis**[2]

Received: 27 May 2015 / Accepted: 8 December 2016

Abstract Finding complete subgraphs in a graph, that is, cliques, is a key problem and has many real-world applications, e.g., finding communities in social networks, clustering gene expression data, modeling ecological niches in food webs, and describing chemicals in a substance. The problem of finding the largest clique in a graph is a well-known difficult combinatorial optimization problem and is called the maximum clique problem. In this paper, we formulate a very convenient continuous characterization of the maximum clique problem based on the symmetric rank-one non-negative approximation of a given matrix and build a one-to-one correspondence between stationary points of our formulation and cliques of a given graph. In particular, we show that the local (resp. global) minima of the continuous problem corresponds to the maximal (resp. maximum) cliques of the given graph. We also propose a new and efficient clique finding algorithm based on our continuous formulation and test it on the DIMACS data sets to show that the new algorithm outperforms other existing algorithms based on the Motzkin–Straus formulation and can compete with a sophisticated combinatorial heuristic.

Communicated by Jim Luedtke.

✉ Nicolas Gillis
nicolas.gillis@umons.ac.be

Melisew Tefera Belachew
melisewt@gmail.com; melisew.belachew@uniba.it

1 Department of Mathematics, Università degli Studi di Bari Aldo Moro, Via E. Orabona 4, 70125 Bari, Italy

2 Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, Rue de Houdain 9, 7000 Mons, Belgium

Published online: 20 December 2016

Keywords Maximum clique problem · Motzkin–Straus formulation · Symmetric rank-one non-negative matrix approximation · Clique finding algorithm

Mathematics Subject Classification 15A23 · 90C26 · 90C27 · 97K30

1 Introduction

A *clique* in an undirected graph is a subset of its vertices such that the corresponding subgraph is complete. A clique is said to be *maximal* if it is not contained in any larger clique. A *maximum* clique is a clique with maximum number of vertices (or, equivalently, edges), and the maximum clique problem (MCP) is the problem of finding such a clique. The number of vertices in a maximum clique in a graph is called the *clique number* of the graph. The MCP is a well-known $\mathcal{NP}$-hard problem, and the associated decision problem, that is, the task of checking whether there is a clique of a given size in a graph, is $\mathcal{NP}$-complete [1].

The MCP arises in many real-life applications. The word "clique" in its graph-theoretic usage was first introduced in [2] where experts used complete graphs to model groups of people who all know each other in social networks. The MCP can also be used to model the problem of clustering gene expression data in bioinformatics [3], to model ecological niches in food webs [4], to analyze telecommunication networks [5], and to describe chemicals in a substance that have a high degree of similarity with a target structure [6].

Since the MCP is an $\mathcal{NP}$-hard problem, it is a challenging task to devise algorithms to identify large cliques in graphs. Experts use various approaches to tackle this problem among which the Motzkin–Straus continuous formulation is a well-known and widely used tool for MCP. There are also a multitude of discrete approaches for the MCP, which is one of the most fundamental problems in graph theory. Performing a literature review of this extremely rich literature is out of the scope of this paper, and we refer the readers to [7–9] for surveys on these methods.

2 Motzkin–Straus Formulation and Outline of the Paper

Let $G = (V, E)$ be an undirected graph where $V = \{1, 2, \ldots, n\}$ is the vertex set, and let $A = \left(a_{ij}\right)_{i,j=1}^{n}$ be the binary adjacency matrix of G, where $a_{ij} = 1$ if and only if $(i, j) \in E$ and $a_{ij} = 0$ otherwise. The *Motzkin–Straus formulation* of the MCP is given by [10]

$$\max_{\mathbf{u}\in\mathbb{R}^n_+} \mathbf{u}^\top A\mathbf{u} \quad \text{such that} \quad \sum_{i=1}^{n} u_i = 1. \tag{MS}$$

Theorem 2.1 (Motzkin and Straus [10]) *The optimal value of the problem* (MS) *is given by* $1 - \frac{1}{\omega(G)}$, *where* $\omega(G)$ *is the clique number of* G.

As mentioned above, many exact and heuristic algorithms have been proposed to solve the MCP; see for example [11–15]. Some of these methods use the above Motzkin–

Straus formulation (MS). Note that using a continuous formulation of a combinatorial problem is a standard approach in global optimization; see, e.g., [16,17] where it is used for the satisfiability problem.

An important result is that if the (zero) diagonal entries of A are replaced by $\frac{1}{2}$, then any local (resp. global) maximum of (MS) corresponds to a maximal (resp. maximum) clique [11]. In this paper, we propose a new continuous formulation for the MCP based on a symmetric rank-one non-negative matrix approximation problem. Below, we give an overview of our continuous formulation and its advantages compared to (MS).

Let us define the associated modified adjacency matrix of G by $B = A + I_n$, where I_n is the identity matrix of dimension n. Moreover, for some parameter $d \geq 0$, we define a (symmetric) matrix $M_d = \left(m_{ij}\right)_{i,j=1}^{n}$ as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } b_{ij} = 1, \\ -d, & \text{if } b_{ij} = 0. \end{cases} \tag{1}$$

We propose in this paper to study the following symmetric rank-one matrix approximation problem

$$\min_{\mathbf{u} \in \mathbb{R}^n_+} \left\| M_d - \mathbf{u}\mathbf{u}^\top \right\|_F^2 . \tag{2}$$

We will show that the optimal solution u^* is the indicator vector of the maximum clique. Unlike other formulations of the MCP, we will draw very precise relationships between stationary points of (2) and cliques of the graph G. Our theoretic result is therefore more complete than for the Motzkin–Straus formulation as we can also associate to any stationary point of (2) a clique of G. Another difference with previous formulations is that the feasible set of (2) is the non-negative orthant onto which it is trivial to project. This is an advantage, for example when designing nonlinear optimization methods, e.g., projected gradient methods; see Sect. 4.

The paper is organized as follows. In Sect. 3.1, we introduce our continuous formulation (2) for the MCP. In Sect. 3.2, we show that

- the local (resp. global) minima of (2) coincide with the maximal (resp. maximum) cliques of a given graph G (Theorems 3.1 and 3.2), and
- every stationary point of the continuous optimization problem (2) coincides with a feasible solution of the MCP (that is, a clique); see Theorem 3.5.

In Sect. 4, we propose a new and efficient clique finding algorithm based on our continuous formulation and show that the limit points of this algorithm coincide with cliques of the graph G. In addition, we present various experimental results that show competitiveness of the new algorithm compared to other clique finding algorithms.

3 Continuous Characterization of the MCP Using Symmetric Non-negative Matrix Approximation

In this section, we derive a new continuous formulation of the MCP using symmetric rank-one non-negative matrix approximation.

3.1 Symmetric Rank-One Non-negative Matrix Approximation and the MCP

The so-called (discrete) *vertex formulation* of the MCP [18] is given by

$$\max_{\mathbf{u}\in\{0,1\}^n} \sum_{i=1}^{n} u_i \quad \text{such that} \quad u_i + u_j \leq 1 + a_{ij} \ \forall i \neq j. \tag{3}$$

The ith vertex belongs to a feasible solution of (3) if and only if $u_i = 1$; otherwise, $u_i = 0$. The constraint $u_i + u_j \leq 1 + a_{ij}$ ensures that if there is no edge between the vertices i and j, that is, if $a_{ij} = 0$, then either $u_i = 0$ or $u_j = 0$. Hence, there is a one-to-one correspondence between the feasible solutions of (3) and the cliques of G.

The objective function of (3) can be rewritten as follows. Observe that maximizing $\sum_i u_i$ reduces to requiring as many ones as possible in the vector $\mathbf{u}$, which will lead to having more ones in the matrix $\mathbf{u}\mathbf{u}^\top$, hence maximizing $\sum_{i,j} u_i u_j$. Let $B = A + I_n$ and $\mathbf{u}$ be a feasible solution of (3). Since B and $\mathbf{u}$ are binary and $u_i u_j \leq b_{ij}$ $\forall i, j$, we have that

$$\sum_{i,j=1}^{n} u_i u_j = \sum_{i,j=1}^{n} (u_i u_j)^2 = \sum_{i,j=1}^{n} b_{ij} (u_i u_j)^2 = \|B\|_F^2 - \left\| B - \mathbf{u}\mathbf{u}^\top \right\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm. In fact,

$$\left\| B - \mathbf{u}\mathbf{u}^\top \right\|_F^2 = \|B\|_F^2 - 2\langle B, \mathbf{u}\mathbf{u}^\top \rangle + \langle \mathbf{u}\mathbf{u}^\top, \mathbf{u}\mathbf{u}^\top \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius inner product. Hence, (3) is equivalent to

$$\min_{\mathbf{u}\in\{0,1\}^n} \left\| B - \mathbf{u}\mathbf{u}^\top \right\|_F^2 \quad \text{such that} \quad u_i + u_j \leq 1 + b_{ij} \ \forall i \neq j, \tag{MC}$$

where the objective function is equal to the number of vertices outside the clique, and its minimization is therefore equivalent to maximizing the vertices contained in the clique. Hence, (MC) approximates B via a symmetric rank-one binary approximation.

3.2 New Continuous Formulation of the MCP

We start off by defining the following problem: given a matrix $M \in \mathbb{R}^{n \times n}$, find its best rank-one non-negative matrix approximation, that is, solve

$$\min_{\mathbf{u} \in \mathbb{R}^n_+} \left\| M - \mathbf{u}\mathbf{u}^\top \right\|_F^2. \tag{R1NM}$$

Given a parameter $d \geq 0$, a graph G, and its adjacency matrix $A \in \{0, 1\}^{n \times n}$, we define the associated modified adjacency matrix by $B = A + I_n$ as before and a matrix $M_d = (1 + d)B - d\mathbf{1}_{n \times n}$ as in (1), where $\mathbf{1}_{n \times n}$ is the n-by-n matrix of ones. Then, we define the following instance of (R1NM):

$$\min_{\mathbf{u} \in \mathbb{R}^n_+} F(\mathbf{u}) = \left\| M_d - \mathbf{u}\mathbf{u}^\top \right\|_F^2. \tag{R1N$_d$M}$$

In this paper, we analyze (R1N$_d$M) as a continuous formulation of the MCP.

Remark 3.1 (Similarity with the Motzkin–Straus Formulation) Up to normalization of $\mathbf{u}$, our continuous formulation can be equivalently written as

$$\max_{\mathbf{u} \in \mathbb{R}^n_+} \mathbf{u}^\top M_d \mathbf{u} \quad \text{such that} \quad \|\mathbf{v}\|_2^2 \leq 1, \tag{4}$$

see [20] for the details. The objective is similar to (MS) except that the adjacency matrix A was modified to M_d and that the constraint is now in the ℓ_2 norm instead of the ℓ_1 norm. Note that the optimal objective function value of (4) is $\omega(G)$.

Interpretation of (R1N$_d$M) *and organization of the section.* The parameter d in (R1N$_d$M) can be interpreted as a penalty parameter that is used to satisfy the constraint $u_i u_j \leq b_{ij}$ for all i, j. In fact, the $-d$ entries in the matrix M_d penalize the fact that entries in $\mathbf{u}\mathbf{u}^\top$ are positive when corresponding to the zero entries of B: for each $i \neq j$ such that $b_{ij} = 0$, the term in the objective function is $(-d - u_i u_j)^2 = d^2 + 2du_i u_j + (u_i u_j)^2$. Therefore, as d increases, the nondiagonal entries of $\mathbf{u}\mathbf{u}^\top$ corresponding to the zero entries of B are encouraged to be closer to zero (see Lemma 3.4).

In the next sections, we show that for $d \geq n$, the local (resp. global) minima of the continuous optimization problem (R1N$_d$M) are binary and coincide with the maximal (resp. maximum) cliques of the graph G, respectively (Theorems 3.1 and 3.2), or equivalently with the optimal solutions of the discrete problem (MC). Moreover, we show that the other stationary points of (R1N$_d$M) get arbitrarily close to the cliques of G as d increases (Theorem 3.5).

Remark 3.2 (Link with the Maximum Edge Biclique Problem) Given a bipartite graph, the maximum edge biclique problem (MBP) is the problem of finding a complete subgraph (that is, a biclique) with maximum number of edges. Gillis and Glineur [19] proposed a continuous formulation of the MBP which is a (non-symmetric) rank-one matrix approximation problem. Our proofs will use some arguments from [19], and we will follow a similar organization to prove the one-to-one correspondence between the maximal (resp. maximum) cliques of the graph G and the local (resp. global) minima of (R1N$_d$M).

3.2.1 Definitions and Notations

Let us introduce the definitions and notations that will be used to prove the main results of this paper. Given a positive real number d, we define the following three sets of vectors:

- $\mathsf{S_p}$, the set of nontrivial stationary points of (R1N$_d$M),

$$\mathsf{S_p} := \{\mathbf{u} \in \mathbb{R}^n_+ \mid \mathbf{u} \text{ satisfies (5)}, \mathbf{u} \neq 0\}.$$

- $\mathsf{L_m}$, the set of nontrivial local minima of (R1N$_d$M).
- $\mathsf{G_m}$, the set of nontrivial global minima of (R1N$_d$M).

By definition, $\mathsf{G_m} \subseteq \mathsf{L_m} \subseteq \mathsf{S_p}$.

Let us also define the following three sets of binary vectors:

- $\mathsf{F_s}$, the set of feasible solutions of (MC),

$$\mathsf{F_s} := \{\mathbf{u} \in \mathbb{R}^n_+ \mid \mathbf{u} \text{ is a feasible solution of (MC)}\}.$$

- $\mathsf{C_m}$, the maximal cliques of G: $\mathbf{u} \in \mathsf{C_m}$ if and only if $\mathbf{u} \in \mathsf{F_s}$ and $\mathbf{u}$ corresponds to a maximal clique of G.
- $\mathsf{C_M}$, the maximum cliques of G: $\mathbf{u} \in \mathsf{C_M}$ if and only if $\mathbf{u} \in \mathsf{F_s}$ and $\mathbf{u}$ corresponds to a maximum clique of G.

By definition, $\mathsf{C_M} \subseteq \mathsf{C_m} \subseteq \mathsf{F_s}$.

3.2.2 Key Lemmas

Given an n-by-n symmetric matrix $M \in \mathbb{R}^{n\times n}$, its best symmetric rank-one approximation can be obtained by solving the following unconstrained minimization problem

$$\min_{\mathbf{u}\in\mathbb{R}^n} \left\| M - \mathbf{u}\mathbf{u}^\top \right\|_F^2 . \tag{R1U}$$

In this section, we will prove various results regarding (R1U) that will be crucial for the remainder of the paper. The following lemma is well known although we do not know the reference for the original proof; see, e.g., [21, Theorem 1.14]. The proof can also be found in [20].

Lemma 3.1 *The local minima of* (R1U) *are global minima. All other nontrivial stationary points are either saddle points or local maxima.*

Lemma 3.2 *For the (symmetric) matrix M_d defined in (1) with at least one entry equal to $-d$ with $d \geq n$, any optimal solution* $\mathbf{u}$ *of* (R1U) *with $M = M_d$ is such that* $\mathbf{u}$ *contains at least one non-positive entry.*

Proof Suppose $\mathbf{u}$ is an optimal solution of (R1U) such that $\mathbf{u}\mathbf{u}^\top > 0$. Note that the diagonal entries of M_d are equal to one and that M_d contains at least one $-d$ entry; hence, M_d contains at least two $-d$ entries. Let r denote the number of entries equal to $-d$ in M_d with $r \geq 2$. Therefore, since $\mathbf{u} > 0$, we have $\|M_d - \mathbf{u}\mathbf{u}^\top\|_F^2 > rd^2$. By assumption, there exists (i, j) such that $i \neq j$ and $m_{ij} = -d$, and, by symmetry, $m_{ji} = -d$. Consider the vector $\mathbf{v} \in \mathbb{R}^n$ such that $v_i = \sqrt{\frac{d}{2}}$, $v_j = -\sqrt{\frac{d}{2}}$, and $v_k = 0\ \forall k \neq i, j$. We have that $v_i v_i = v_j v_j = \frac{d}{2}$, $v_j v_i = v_i v_j = -\frac{d}{2}$, and $v_k v_l = 0\ \forall k, l \neq i, j$; hence,

$$\|M_d - \mathbf{v}\mathbf{v}^\top\|_F^2 = 2\left(\frac{d}{2} - 1\right)^2 + 2\left(\frac{d}{2}\right)^2 + (r-2)d^2 + n^2 - (r+2)$$
$$< rd^2 + n^2 - d^2 - (r+2) \overset{r \geq 2}{\leq} rd^2 + n^2 - d^2 \overset{d \geq n}{\leq} rd^2,$$

a contradiction. Therefore, any optimal solution $\mathbf{u}$ of (R1U) must contain at least one non-positive entry. □

3.2.3 Local and Global Minima of (R1N$_d$M)

In this section, we characterize the relationship between the local (resp. global) minima of (R1N$_d$M) and the maximal (resp. maximum) cliques of a given graph G.

Lemma 3.3 *Let $G = (V, E)$ be a graph with at least one edge, with binary adjacency matrix A and modified adjacency matrix $B = A + I_n$, and $d \geq n$. Then, $\mathsf{L_m} \subseteq \mathsf{C_m}$.*

Proof Let $M_d \in \{-d, 1\}^{n \times n}$ be the matrix defined in (1). Let $\mathbf{u} \in \mathsf{L_m}$. To show $\mathsf{L_m} \subseteq \mathsf{C_m}$, we need to show that $\mathbf{u}$ is a feasible solution of (MC) and $\mathbf{u}$ corresponds to a maximal clique of G. The support of a vector $\mathbf{u}$ is defined as the set of indices corresponding to the nonzero entries of $\mathbf{u}$. Let us denote the (non-empty) index set of the support of $\mathbf{u}$ by S and define $\mathbf{u}' = \mathbf{u}(S)$ and $M_d' = M_d(S, S)$ to be the subvector and the submatrix with indices in S and $S \times S$, respectively. Let us also define G' as the graph whose modified adjacency matrix is given by $B' = B(S, S)$. Since $\mathbf{u}$ is a local minimum of (R1N$_d$M) and the objective functions of (R1N$_d$M) and (R1NCG′) differ only by a constant (and also $\mathbf{u}^\top M_d \mathbf{u} = \mathbf{u}'^\top M_d' \mathbf{u}'$), we have that $\mathbf{u}'$ is a local minimum of (R1NCG′)

$$\min_{\mathbf{u}' \in \mathbb{R}_+^{|S|}} \left\| M_d' - \mathbf{u}'\mathbf{u}'^\top \right\|_F^2. \tag{R1NCG′}$$

To show that $\mathbf{u}$ is a feasible solution of (MC), we first suppose there is a $-d$ entry in M_d'. Since $\mathbf{u}'$ is positive, it is located in the interior of the feasible domain of (R1NCG′).

Therefore, it is a local minimum of the unconstrained problem (R1UCM′)

$$\min_{\mathbf{u}'\in\mathbb{R}^{|S|}} \left\| M'_d - \mathbf{u}'\mathbf{u}'^{\top} \right\|_F^2 . \tag{R1UCM′}$$

Thus, Lemma 3.1 implies that $\mathbf{u}'$ is a global minimum of (R1UCM′). Moreover, since M'_d contains at least one $-d$ entry, Lemma 3.2 asserts that $\mathbf{u}'$ contains a non-positive entry, a contradiction. Therefore, M'_d does not contain a $-d$ entry and as a result we have $M'_d = \mathbf{1}_{|S|\times|S|}$. Since $\mathbf{u}'$ is a global minimum of (R1UCM′) and $M'_d = \mathbf{1}_{|S|\times|S|}$, we must have $\mathbf{u}'\mathbf{u}'^{\top} = M'_d = \mathbf{1}_{|S|\times|S|}$, $\mathbf{u}' = \mathbf{1}_{|S|}$ and $\mathbf{u}$ is binary. Therefore, $\mathbf{u}$ is a feasible solution of (MC), that is, $\mathbf{u} \in \mathsf{F_s}$.

Finally, let us show that $\mathbf{u}$ corresponds to a maximal clique of G. Assume that $\mathbf{u}$ corresponds to a clique of G which is not maximal, that is, assume without loss of generality that $\exists i \notin S$ such that $\mathbf{u}+\mathbf{e}_i$ corresponds to a larger clique of G where $\mathbf{e}_i$ is the unit vector whose i-th entry is equal to one. For any $0 < \epsilon \leq 1$, let $\mathbf{v} = \mathbf{u} + \epsilon \mathbf{e}_i$. Then, we have $\left\| M_d - \mathbf{v}\mathbf{v}^{\top} \right\|_F^2 \leq \left\| M_d - \mathbf{u}\mathbf{u}^{\top} \right\|_F^2$, since the entries of M_d corresponding to edges contained only in the larger clique $\{i\} \times S$ are 1's and are approximated by values between 0 and 1 in $\mathbf{v}\mathbf{v}^{\top}$, whereas they are approximated by zeros in $\mathbf{u}$. This contradicts the assumption that $\mathbf{u}$ is a local minimum. Hence, $\mathbf{u}$ must correspond to a maximal clique of G, that is, $\mathbf{u} \in \mathsf{C_m}$. □

The next result shows that all the maximal cliques of a given graph G correspond to the local minima of (R1N$_d$M).

Theorem 3.1 *If G is a graph with at least one edge and $d \geq n$, then $\mathsf{C_m} = \mathsf{L_m}$.*

Proof Since $d \geq n$, by Lemma 3.3 we have $\mathsf{L_m} \subseteq \mathsf{C_m}$. The proof that $\mathsf{C_m} \subseteq \mathsf{Lm}$ is very similar to that of [19, Theorem 1]. It can be found in [20]. □

The next result states the strong relationship between the global solutions of (R1N$_d$M) and the maximum cliques of a graph G.

Theorem 3.2 *If G is a graph with at least one edge and $d \geq n$, then $\mathsf{G_m} = \mathsf{C_M}$.*

Proof Let $\mathbf{u} \in \mathsf{G_m}$. Then, by definition $\mathbf{u} \in \mathsf{L_m}$ and by Theorem 3.1 $\mathbf{u} \in \mathsf{C_m}$. Hence, $\mathbf{u}$ is binary. Next, observe that the objective functions of (R1N$_d$M) and (MC) differ only by a constant:

$$\left\| M_d - \mathbf{u}\mathbf{u}^{\top} \right\|_F^2 = \left\| B - \mathbf{u}\mathbf{u}^{\top} \right\|_F^2 + (n^2 - \|B\|_F^2)d^2.$$

Therefore, $\mathbf{u} \in \mathsf{G_m}$ if and only if $\mathbf{u} \in \mathsf{C_M}$. □

Corollary 3.1 *The optimal value of (R1N$_d$M) is $\|M_d\|_F^2 - \omega(G)^2$.*

Proof This follows directly from Theorem 3.2: an optimal solution $\mathbf{u}$ of the problem (R1N$_d$M) is the indicator vector corresponding to a maximum clique. □

Corollary 3.2 *(R1N$_d$M) is $\mathcal{NP}$-hard.*

Proof By Theorem 3.2, finding the global optima of (R1N$_d$M) is equivalent to solving (MC) which is equivalent to solving (MS), which is $\mathcal{NP}$-hard [1]. □

3.2.4 Stationary Points and Maximal Cliques

Here, we prove an important result that tells us how the maximal cliques of a given graph G are related to the stationary points of (R1N$_d$M) and to the feasible solutions of (MC).

First, we derive the optimality conditions and stationary points of (R1N$_d$M). The first-order optimality conditions of (R1N$_d$M) are

$$\mathbf{u} \geq 0, \quad \nabla_{\mathbf{u}} F(\mathbf{u}) = \mathbf{u}\mathbf{u}^\top \mathbf{u} - M_d \mathbf{u} \geq 0 \quad \text{and} \quad \mathbf{u} \odot \nabla_{\mathbf{u}} F(\mathbf{u}) = 0, \tag{5}$$

where $\odot$ is the Hadamard product, that is, a vector $\mathbf{u}$ is a stationary point of (R1N$_d$M) if and only if it satisfies the optimality conditions (5). The optimality conditions given in (5) can be written equivalently as

$$\mathbf{u} = 0 \quad \text{or} \quad \mathbf{u} = \max\left(0, \frac{M_d \mathbf{u}}{\|\mathbf{u}\|_2^2}\right) = \frac{[M_d \mathbf{u}]_+}{\|\mathbf{u}\|_2^2}. \tag{6}$$

In fact, if $u_i = 0$, then $(M_d \mathbf{u})_i \leq 0$ (the gradient is non-negative), while if $u_i > 0$, then $u_i = (M_d \mathbf{u})_i$ (the gradient is equal to zero). Hence, the nontrivial stationary points of (R1N$_d$M) satisfy $\mathbf{u} = \frac{[M_d \mathbf{u}]_+}{\|\mathbf{u}\|_2^2}$.

Theorem 3.3 *If G is a graph with at least one edge and $d \geq n$, then $\mathsf{C_m} = \mathsf{F_s} \cap \mathsf{S_p}$.*

Proof We need to show that if $\mathbf{u} \in \mathsf{C_m}$, then $\mathbf{u}$ belongs to both $\mathsf{F_s}$ and $\mathsf{S_p}$. If $\mathbf{u} \in \mathsf{C_m}$, then $\mathbf{u}$ is binary (Theorem 3.1). Let S denote the non-empty support of $\mathbf{u}$. By definition, we have that $\mathbf{u} \in \mathsf{F_s}$ and $\mathbf{u}$ corresponds to a maximal clique of G, that is,

$$\nexists i \text{ such that } u_i = 0 \text{ and } m_{ij} = 1 \ \forall j \in S. \tag{7}$$

It remains to show that $\mathbf{u} \in \mathsf{S_p}$. For all i such that $u_i = 0$, by (7) at least one entry of $M_d(i,:)$ is $-d$. Therefore, we have $M_d(i,:)\mathbf{u} \leq (n-2) - d < 0$ since $d \geq n$. Since $\mathbf{u}$ is binary, we also have

$$u_i = 0 \text{ and } M_d(i,:)\mathbf{u} < 0 \text{ or } u_i = 1 = \frac{\|\mathbf{u}\|_1}{\|\mathbf{u}\|_2^2}. \tag{8}$$

Note that for all $i \in S$ we have $m_{ij} = 1$ if and only if $j \in S$. Thus, $m_{ij} u_j = 1$ if and only if $i, j \in S$. As a result, for all $i \in S$,

$$M_d(i,:)\mathbf{u} = \sum_{j=1}^{n} m_{ij} u_j = \sum_{j \in S} m_{ij} u_j = \sum_{j \in S} u_j = \sum_{j=1}^{n} u_j = \|\mathbf{u}\|_1. \tag{9}$$

By combining (8) and (9), we obtain

$$u_i = 0 \text{ and } M_d(i,:)\mathbf{u} < 0 \quad \text{or} \quad 1 = u_i = \frac{\|\mathbf{u}\|_1}{\|\mathbf{u}\|_2^2} = \frac{M_d(i,:)\mathbf{u}}{\|\mathbf{u}\|_2^2}. \tag{10}$$

We can rewrite (10) as $\mathbf{u} = \max\left(0, \frac{M_d \mathbf{u}}{\|\mathbf{u}\|_2^2}\right)$, which is exactly the stationarity conditions (6) of (R1N$_d$M) for $\mathbf{u} \neq 0$. Therefore, $\mathbf{u} \in \mathsf{S_p}$. Hence, $\mathbf{u} \in \mathsf{C_m}$ if and only if $\mathbf{u} \in \mathsf{F_s} \cap \mathsf{S_p}$. □

Theorem 3.3 implies that if an algorithm converges to a stationary point of (R1N$_d$M) and that this stationary point is binary, then it corresponds to a maximal clique.

3.2.5 Limit Points of (R1N$_d$M) and Feasible Solutions of (MC)

This section shows how close the stationary points of (R1N$_d$M) are to the feasible solutions of (MC). First, we present two lemmas and recall a Theorem from [19] about bipartite graphs. The next lemma shows that entries of $\mathbf{u}\mathbf{u}^\top$ corresponding to the $-d$ entries of M_d are approximated by zeros as d gets larger.

Lemma 3.4 *For any graph G and $\mathbf{u} \in \mathcal{S}_p$, if $m_{ij} = -d$ and $u_i u_j > 0$, we have $0 < u_j < \frac{\|\mathbf{u}\|_1}{d+1}$ and $0 < u_i < \frac{\|\mathbf{u}\|_1}{d+1}$.*

Proof Since u_i and u_j are positive, the optimality condition (6) gives

$$\begin{aligned} 0 < u_i \|\mathbf{u}\|_2^2 = M_d(i,:)\mathbf{u} &= -du_j + \sum_{r \neq j} m_{ir} u_r \\ &\leq -du_j + \sum_{r \neq j} u_r = -du_j + \left(\|\mathbf{u}\|_1 - u_j\right) = \|\mathbf{u}\|_1 - (d+1)u_j. \end{aligned}$$

Therefore, $0 < u_j < \frac{\|\mathbf{u}\|_1}{d+1}$. By symmetry, the same holds for u_i. □

Below, we state and prove a lemma which is useful to draw an important relationship between stationary points of (R1N$_d$M) and feasible solutions of (MC).

Lemma 3.5 *Let M be a symmetric matrix. If $\mathbf{u}$ is a stationary point of $\min_{\mathbf{u} \geq 0} \|M - \mathbf{u}\mathbf{u}^\top\|_F^2$, then $(\mathbf{u}, \mathbf{u})$ is a stationary point of the problem $\min_{(\mathbf{u},\mathbf{v}) \geq 0} \|M - \mathbf{u}\mathbf{v}^\top\|_F^2$.*

Proof If $\mathbf{u} = \mathbf{0}$, the proof is complete since $(\mathbf{0}, \mathbf{0})$ is a stationary point of (MC). Otherwise, since $\mathbf{u}$ is a nontrivial stationary point of $\min_{\mathbf{u} \geq 0} \|M - \mathbf{u}\mathbf{u}^\top\|_F^2$, we have $\mathbf{u} = \max\left(0, \frac{M\mathbf{u}}{\|\mathbf{u}\|_2^2}\right)$; see (6). Moreover, if $(\mathbf{u}, \mathbf{v})$ is a nontrivial stationary point of $\min_{\mathbf{u},\mathbf{v} \geq 0} \|M - \mathbf{u}\mathbf{v}^\top\|_F^2$, by the first-order optimality conditions we have [19, Eq. (6)] $\mathbf{u} = \max\left(0, \frac{M\mathbf{v}}{\|\mathbf{v}\|_2^2}\right)$ and $\mathbf{v} = \max\left(0, \frac{M\mathbf{u}}{\|\mathbf{u}\|_2^2}\right)$. Hence, if $\mathbf{u}$ is a nontrivial stationary point of $\min_{\mathbf{u} \geq 0} \|M - \mathbf{u}\mathbf{u}^\top\|_F^2$, then $(\mathbf{u}, \mathbf{u})$ is a nontrivial stationary point of $\min_{\mathbf{u},\mathbf{v} \geq 0} \|M - \mathbf{u}\mathbf{v}^\top\|_F^2$ (note that the converse is true only when $\mathbf{u} = \mathbf{v}$). □

Theorem 3.4 ([19], Theorem 4 and Corollary 2) *Let $\hat{G}$ be a bipartite graph and $\hat{A} \in \{0,1\}^{m \times n}$ be its binary biadjacency matrix. For some constant $\hat{d} \geq 0$, define*

$\hat{M}_d = (1+\hat{d})\hat{A} - \hat{d}\mathbf{1}_{m\times n}$, where $\mathbf{1}_{m\times n}$ is an m-by-n matrix of ones. Then, when $\hat{d}$ goes to infinity, every stationary point of

$$\min_{\mathbf{u}\in\mathbb{R}^m_+,\mathbf{v}\in\mathbb{R}^n_+} \|\hat{M}_d - \mathbf{u}\mathbf{v}^\top\|_F^2, \tag{11}$$

gets arbitrarily close to some feasible solution of

$$\min_{\mathbf{u}\in\{0,1\}^m,\mathbf{v}\in\{0,1\}^n} \|\hat{A} - \mathbf{u}\mathbf{v}^\top\|_F^2 \quad \text{such that} \quad u_i + v_j \le 1 + \hat{a}_{ij}\ \forall i, j. \tag{12}$$

More precisely, for any $\hat{d} \ge 2\max(m,n)\|\hat{A}\|_F$, we have for any stationary point $(\mathbf{u},\mathbf{v})$ of (11) that

$$\min_{\mathbf{u}_b,\mathbf{v}_b} \|\mathbf{u}\mathbf{v}^\top - \mathbf{u}_b\mathbf{v}_b^\top\|_F < \frac{\max(m,n)\|\hat{A}\|_F}{\hat{d}+1},$$

where $(\mathbf{u}_b,\mathbf{v}_b)$ is a feasible solution of (12).

We can now prove our main result linking the stationary points of (R1N$_d$M) and the feasible solutions of (MC).

Theorem 3.5 *For any graph G, every stationary point of (R1N$_d$M) gets arbitrarily close to some feasible solution of (MC):*

$$\max_{\mathbf{u}\in S_p}\min_{\mathbf{u}_c\in F_s} \|\mathbf{u}-\mathbf{u}_c\|_2 < \frac{n\|B\|_F}{d+1},$$

where B is the modified adjacency matrix of G and $d \ge 2n\|B\|_F$.

Proof If the graph G does not have any edge, $\mathsf{S_p} = \emptyset$ and the proof is complete. Otherwise, let M_d be the symmetric matrix defined in (1), and let $\mathbf{u} \in \mathsf{S_p}$. Then, combining Lemma 3.5 (with $E = M_d$) and Theorem 3.4, we have

$$\min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}\mathbf{u}^\top - \mathbf{u}_c\mathbf{u}_c^\top\|_F^2 < \frac{n^2\|B\|_F^2}{(d+1)^2}.$$

Observe the following:

$$\begin{aligned}\min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}\mathbf{u}^\top - \mathbf{u}_c\mathbf{u}_c^\top\|_F^2 &= \min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}\mathbf{u}^\top - \mathbf{u}_c\mathbf{u}^\top + \mathbf{u}_c\mathbf{u}^\top - \mathbf{u}_c\mathbf{u}_c^\top\|_F^2 \\ &= \min_{\mathbf{u}_c\in\mathsf{F_s}} \|(\mathbf{u}-\mathbf{u}_c)\mathbf{u}^\top + \mathbf{u}_c(\mathbf{u}^\top - \mathbf{u}_c^\top)\|_F^2 \\ &\ge \min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}_c(\mathbf{u}^\top - \mathbf{u}_c^\top)\|_F^2 = \min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}_c\|_2^2\|\mathbf{u}-\mathbf{u}_c\|_2^2.\end{aligned}$$

Therefore, since the vector $\mathbf{u}_c$ is binary and satisfies $\|\mathbf{u}_c\|_2^2 \ge 1$, we have $\min_{\mathbf{u}_c\in\mathsf{F_s}} \|\mathbf{u}-\mathbf{u}_c\|_2 < \frac{n\|B\|_F}{d+1}$. ☐

Corollary 3.3 *Let us define* $\Phi : \mathbb{R}^n_+ \to \{0,1\}^n \;: \mathbf{u} \to \Phi(\mathbf{u})$*, such that* $\Phi(\mathbf{u})_i = 0$ *if* $u_i \leq 0.5$ *and* $\Phi(\mathbf{u})_i = 1$ *if* $u_i > 0.5$ *for all* i*. Then, for any graph* G, $d \geq 2n\|B\|_F$*, and any* $\mathbf{u} \in \mathcal{S}_p$*, we have that* $\Phi(\mathbf{u}) \in \mathcal{F}_s$*.*

Proof This follows directly from Theorem 3.5 since the stationary point $\mathbf{u}$ is at Euclidean distance at most $\frac{n\|B\|_F}{d+1} \leq \frac{n\|B\|_F}{2n\|B\|_F+1} < \frac{1}{2}$ from a binary indicator corresponding to a clique of G. □

4 Proposed Algorithm and Experimental Results

In this section, we propose a new and efficient clique finding algorithm using our continuous formulation (Sect. 4.1). Our algorithm is a projected gradient scheme applied on (R1N$_d$M), using the Armijo procedure for selecting the step sizes. Then, we provide numerical comparisons on the DIMACS data sets in Sect. 4.2.

4.1 Projected Gradient Descent with Armijo Procedure for the Continuous Formulation (R1N$_d$M)

Consider a non-empty closed convex set $\Omega \subset \mathbb{R}^n$ and a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ on Ω. The projected gradient method for solving the minimization problem $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$ is the following [22, Sec. 2.3]: choose some initial $\mathbf{x}^{(1)} \in \Omega$ and, for $k = 1, 2, \ldots$, compute

$$\mathbf{x}^{(k+1)} = \mathcal{P}_\Omega \left[\mathbf{x}^{(k)} - s^{(k)} \nabla f(\mathbf{x}^{(k)})\right],$$

where $\mathbf{x}^{(k)}$ is the kth iterate, $\mathcal{P}_\Omega$ is the projection into Ω, and $s^{(k)}$ is the step size taken at the kth step. The Armijo condition requires the step $s^{(k)}$ to satisfy the condition

$$f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \;\leq\; \sigma \, \nabla f(\mathbf{x}^{(k)})^\top \left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right), \tag{13}$$

for some parameter $0 < \sigma < 1$. To guarantee a sufficient decrease in the objective function at each iteration, the Armijo procedure takes $s^{(k)} = \beta^{m^{(k)}} \bar{s}$, where $\bar{s} > 0$ is a constant, $0 < \beta < 1$ is a parameter, and $m^{(k)}$ is the *smallest* non-negative integer satisfying the above condition. The limit points of a projected gradient method that uses this procedure are stationary points of $\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$ [22, Proposition 2.3.3]; see also [23]. Searching for $s^{(k)}$ can be time-consuming. Since $s^{(k-1)}$ and $s^{(k)}$ usually take values of the same order of magnitude, using $s^{(k-1)}$ as an initial guess for $s^{(k)}$ is usually rather efficient in practice; see, e.g., [24]. Algorithm 1 implements this idea on our continuous formulation (R1N$_d$M) of the clique problem. The parameter d in (R1N$_d$M) is initialized to some value (see Sect. 4.2.1 for a discussion) and is increased progressively (by a factor $\gamma > 1$ at each iteration) until it reaches the upper bound $D = 2n\|A + I_n\|_F \geq n$ that guarantees (i) the one-to-one correspondence between local and global minima of (R1N$_d$M) with the maximal and maximum cliques of G (Theorems 3.1 and 3.2), and that (ii) rounding stationary points of (R1N$_d$M) gives

Algorithm 1 Clique finding algorithm

1: **Require:** Adjacency matrix $A \in \{0, 1\}^{n \times n}$, $\gamma > 1, 0 < \sigma < 1, 0 < \beta < 1$;
2: **Initialize:** $\mathbf{u}, d, \alpha$;
3: **Set:** $D = 2n\|A + I_n\|_F$;
4: **while** stopping criterion is not satisfied **do**
5: $\quad \nabla F(\mathbf{u}) = 2\left[\mathbf{u}\left(\|\mathbf{u}\|_2^2 - 1\right) - (1+d)A\mathbf{u} + d\mathbf{1}_n\|\mathbf{u}\|_1\right]$;
6: $\quad \mathrm{Fo} = -\mathbf{u}^\top M_d \mathbf{u} + \frac{1}{2}\|\mathbf{u}\|_2^4 = -(1+d)\mathbf{u}^\top A\mathbf{u} - (1+d)\|\mathbf{u}\|_2^2 + d\|\mathbf{u}\|_1^2 + \frac{1}{2}\|\mathbf{u}\|_2^4$;
7: $\quad$ **while** Armijo condition is not satisfied **do**
8: $\quad\quad \mathbf{u}_\mathrm{n} \leftarrow \max\left(0, \mathbf{u} - \alpha\nabla F(\mathbf{u})\right)$;
9: $\quad\quad \mathrm{Fn} = -\mathbf{u}_\mathrm{n}^\top M_d \mathbf{u}_\mathrm{n} + \frac{1}{2}\|\mathbf{u}_\mathrm{n}\|_2^4 = -(1+d)\mathbf{u}_\mathrm{n}^\top A\mathbf{u}_\mathrm{n} - (1+d)\|\mathbf{u}_\mathrm{n}\|_2^2 + d\|\mathbf{u}_\mathrm{n}\|_1^2 + \frac{1}{2}\|\mathbf{u}_\mathrm{n}\|_2^4$;
10: $\quad\quad$ **if** $\mathrm{Fn} - \mathrm{Fo} > \sigma\nabla F(\mathbf{u})^\top(\mathbf{u}_\mathrm{n} - \mathbf{u})$ **then**
11: $\quad\quad\quad \alpha \leftarrow \beta\alpha$;
12: $\quad\quad$ **else**
13: $\quad\quad\quad \alpha \leftarrow \frac{\alpha}{\sqrt{\beta}}$;
14: $\quad\quad$ **end if**
15: $\quad$ **end while**
16: $\quad \mathbf{u} \leftarrow \mathbf{u}_\mathrm{n}$;
17: $\quad d \leftarrow \min(\gamma d, D)$;
18: **end while**

cliques of the graph G (Corollary 3.3). The motivation to increase d progressively is the fact that (R1N$_d$M) is an easy problem for small d. In fact, for $d = 0$, $M_d = B$ is non-negative which implies that (R1N$_d$M) is equivalent to computing the eigenvector of M_d associated with the largest eigenvalue of M_d which can be solved efficiently (combining Perron–Frobenius and Eckart–Young theorems; see, e.g., [25]). In fact, we have observed that trying to solve (R1N$_d$M) directly for a large value of d leads in general to worse solutions. Note that the expressions in lines 5, 6, and 9 of Algorithm 1 use (i) half the objective function of (1) minus the constant $\|M_d\|_F^2$ which gives $\|M_d - \mathbf{u}\mathbf{u}^\top\|_F^2 - \|M_d\|_F^2 = -2\mathbf{u}^\top M_d\mathbf{u} + \|\mathbf{u}\|_2^4$, and (ii) the fact that the matrix M_d is equal to $(1+d)(A + I_n) - d\mathbf{1}_{n\times n}$ so that

$$\begin{aligned}\mathbf{u}^\top M_d\mathbf{u} &= \mathbf{u}^\top\left((1+d)(A+I_n) - d\mathbf{1}_{n\times n}\right)\mathbf{u}\\ &= (1+d)\mathbf{u}^\top A\mathbf{u} + (1+d)\|\mathbf{u}\|_2^2 - d\|\mathbf{u}\|_1^2.\end{aligned}$$

This avoids the explicit construction of M_d, which is not practical if A is sparse (since M_d is dense). Since every limit point of Algorithm 1 is a stationary point of (R1N$_d$M), their Φ roundings, as defined in Corollary 3.3, are cliques of the graph G. On all the numerical experiments performed in Sect. 4.2, Algorithm 1 always converged to a maximal clique.

Theorem 4.1 *Every limit point of Algorithm 1 is a stationary point of the problem (R1N$_d$M) and the Φ rounding of this stationary point is a clique of the given graph G.*

Proof A projected gradient algorithm that uses the Armijo procedure converges to a stationary point [23]. Since d will attain the value D in a finite number of steps, the second part of the theorem follows from Corollary 3.3. ◻

4.2 Experimental Setup and Numerical Results

In this section, we assess the performances of the new algorithm (Algorithm 1) compared to two clique finding algorithms based on the Motzkin–Straus formulation from [12,15] (note that these papers do not study the relationship between cliques and the stationary points of the continuous formulation they consider). We will also compare our method with the combinatorial algorithms of Grosso et al. [9] which are effective in solving the MCP. Grosso et al. proposed two iterated local search algorithms based on fast neighborhood search that use multiple restarts and several thousands of node selections per second. Section 4.2.1 describes the parameters, initial values, and stopping criteria used for our algorithm (we used the default values for the algorithms in [12,15]. Section 4.2.2 presents the numerical results on DIMACS data sets, showing that Algorithm 1 performs well on these problems. For more numerical experiments, we refer the reader to [20].

4.2.1 Parameters, Initial Values, and Stopping Criteria

We use random initialization and stop the algorithm when the condition $0 \leq u_i \leq 0.001$ or $0.999 \leq u_i \leq 1.001$ for all i is satisfied. In addition, we used the values $\beta = 0.5, \sigma = 0.01$ and initialized α with $\alpha_0 = 0.1 \frac{\|\mathbf{u}^0\|_2}{\|\nabla F(\mathbf{u}^0)\|_2}$ for all experiments. To make the search for the step sizes more practical, we only try to update α for a maximum of five steps per iteration. An initial value of the parameter d (in Algorithm 1) that is close to the value that balances the positive and negative entries in M_d (that is, choosing d such that $\|\max(M_d, 0)\|_F \approx \|\max(-M_d, 0)\|_F$)) works well in practice [19]. For this reason, we use the initial value $d_0 = \frac{\|A+I_n\|_F^2}{n^2 - \|A+I_n\|_F^2}$ for d and then increase it by a factor $\gamma = 1.1$ at each iteration until it reaches the value D. We have also experimented with SVD initialization, that is, we initialized the algorithm with the best rank-one approximation of the non-negative modified biadjacency matrix $B = A + I_n$ (which is the optimal solution for $d = 0$) and found out that the results are similar to those obtained with random initialization.

Note that the computational cost of our algorithm is $\mathcal{O}(|E|)$ per iteration, where $|E|$ is the number of edges in the graph G (assuming $|E| > |V|$). Most of the time is spent for computing the matrix-vector product $A\mathbf{u}$, and it can be checked that all other operations run in at most $\mathcal{O}(|V|)$ operations where $|V|$ is the number of vertices in G.

4.2.2 Numerical Results

We now present the numerical results. All tests are performed using MATLAB (R2012a) on a laptop Intel(R) Core(TM) i7-6500U CPU @2.50 GHz 2.59 GHz 8 GB RAM. The MATLAB code is available at https://sites.google.com/site/nicolasgillis/.

Table 1 reports the computational costs and clique sizes for the DIMACS data sets from the Web site http://iridia.ulb.ac.be/~fmascia/maximum_clique (maintained by Franco Mascia). In this experiment, we include a combinatorial approach designed by Grosso et al. [9] in which we directly copy the results from their paper (namely, the computational times reported in their paper are scaled to a Pentium IV 2.4 GHz

Table 1 Clique size and computational time in seconds (in brackets) for the different algorithms on the DIMACS instances

Data set (n, $\|E\|$, $\omega(G)$)	[15]	[12]	Algorithm 1	[9]
brock200_1 (200, 14834, 21)	18 (0.14)	18 (0.14)	<u>19</u> (0.05)	**21** (0.02)
brock200_2 (200, 9876, 12)	8 (0.03)	9 (0.52	<u>10</u> (0.05)	**12** (0.02)
brock200_3 (200, 12048, 15)	10 (0.03)	11 (0.09)	<u>13</u> (0.06)	**15** (0.01)
brock200_4 (200, 13089, 17)	13 (0.03)	<u>15</u> (0.09)	<u>15</u> (0.04)	**17** (0.13)
brock400_1 (400, 59723, 27)	21 (0.06)	21 (0.25)	<u>24</u> (0.08)	**27** (9.26)
brock400_2 (400, 59786, 29)	21 (0.09)	18 (0.41)	<u>24</u> (0.08)	**29** (1.20)
brock400_3 (400, 59681, 31)	20 (0.13)	18 (0.28)	<u>23</u> (0.09)	**31** (0.23)
brock400_4 (400, 59765, 33)	20 (0.08)	<u>24</u> (1.03)	<u>24</u> (0.07)	**33** (0.09)
brock800_1 (800, 207505, 23)	16 (0.19)	<u>19</u> (1.58)	18 (0.36)	**22.64** (247)
brock800_2 (800, 208166, 24)	16 (0.14)	16 (1.00)	<u>19</u> (0.34)	**24** (59.24)
brock800_3 (800, 207333, 25)	18 (0.25)	18 (1.77)	<u>19</u> (0.35)	**25** (64.04)
brock800_4 (800, 207643, 26)	17 (0.19)	16 (1.73)	<u>19</u> (0.34)	**26** (27.10)
C500-9 (500, 112332, ≥57)	46 (0.13)	12 (0.03)	<u>50</u> (0.12)	**57** (1.41)
C1000-9 (1000, 450079, ≥68)	51 (0.75)	5 (0.13)	<u>63</u> (0.47)	**67.91** (100)
C2000-5 (2000, 999836, ≥16)	13 (0.91)	<u>14</u> (8.44)	<u>14</u> (2.09)	**16** (1.6)
C2000-9 (2000, 1799532, ≥80)	62 (3.86)	7 (0.31)	<u>73</u> (1.98)	**76.57** (563)
C4000-5 (4000, 4000268, ≥18)	13 (4.47)	3 (1.80)	<u>16</u> (7.97)	**18** (304.18)
ham.10_2 (1024, 518656, 512)	1 (0.001)	1 (0.03)	**512** (0.13)	<u>510.64</u> (2.14)
keller6 (3361, 4619898, ≥59)	31 (10.17)	15 (1.78)	<u>33</u> (8.83)	**59** (118.61)
MANN_a27 (378, 70551, 126)	1 (0.02)	1 (0.03)	<u>123</u> (0.10)	**126** (0.005)
–_a45 (1035, 533115, 345)	1 (0.13)	1 (0.01)	<u>333</u> (0.91)	**344.02** (373)
–_a81 (3321, 5506380, ≥1100)	1 (0.95)	1 (0.27)	<u>1061</u> (9.24)	**1098** (987)
$\hat{p}$1000-1 (1000, 122253, ≥10)	8 (0.14)	9 (1.13)	**10** (0.53)	**10** (0.06)
$\hat{p}$1000-2 (1000, 244799, ≥46)	44 (1.91)	8 (0.25)	**46** (0.72)	**46** (0.01)
$\hat{p}$1000-3 (1000, 371746, ≥68)	<u>63</u> (0.75)	3 (0.13)	62 (0.63)	**68** (0.07)
$\hat{p}$1500-1 (1500, 284923, 12)	9 (0.34)	**10** (1.84)	**10** (1.35)	**10** (5.86)
$\hat{p}$1500-2 (1500, 568960, ≥65)	<u>61</u> (2.09)	22 (0.33)	<u>61</u> (1.65)	**65** (0.07)
$\hat{p}$1500-3 (1500, 847244, ≥94)	88 (2.39)	24 (0.30)	<u>92</u> (1.57)	**94** (0.09)
san400_0.5_1 (400, 39900, 13)	2 (0.03)	2 (0.03)	<u>7</u> (0.04)	**13** (0.03)
san400_0.7_1 (400, 55860, 40)	15 (0.05)	6 (0.05)	<u>22</u> (0.07)	**40** (0.04)
san400_0.7_2 (400, 55860, 30)	5 (0.02)	2 (0.03)	<u>15</u> (0.04)	**30** (0.03)
san400_0.7_3 (400, 55860, 22)	1 (0.05)	1 (0.03)	<u>13</u> (0.04)	**22** (0.05)
san400_0.9_1 (400, 71820, 100)	<u>55</u> (0.13)	12 (0.06)	53 (0.04)	**100** (0.002)
san1000 (1000, 250500, 15)	7 (0.03)	4 (0.06)	<u>8</u> (0.23)	**15** (2.57)
sanr400_0.5 (400, 39984, 13)	11 (0.03)	12 (0.16)	**13** (0.10)	**13** (0.14)
sanr400_0.7 (400, 55869, 21)	18 (0.03)	19 (0.25)	**21** (0.09)	**21** (0.02)

The best and second best clique sizes are highlighted in bold and underlined, respectively with 512 MB RAM running Linux, and the value of the clique corresponds to running their algorithms 100 times and taking the average of the maximum and the minimum clique as the average clique size). Two heuristics were proposed by Grosso et al., but we consider the one with (a slightly) better performance in terms of clique size; see the results of Algorithm 2 in Table 4 of their paper. For 13 out of 36 cases, Grosso et al. finds larger cliques faster than the other methods, whereas on other 9 cases this method is found to be very expensive. In most cases, the two algorithms from [12,15] are faster than the other methods, but the solutions are poor. When comparing the clique sizes of the continuous approaches, Algorithm 1 outperforms the other two clique finding algorithms from [12,15]. When comparing all approaches, the discrete method due to Grosso et al. provides cliques having larger sizes in many cases except in one case (hamming10_2) where it gives the second best value (with Algorithm 1 scoring the best value) and in other five cases where there is a tie with Algorithm 1. The success of Grosso et al. can be attributed to the fact that it uses a fast neighborhood search in combination with multiple restarts which is based on selecting several thousands of nodes per second (on average 250,210 nodes per second for the tested DIMACS instances). On the contrary, our algorithm is a simple single-start method which could be improved using standard techniques such as genetic algorithms or simulated annealing, for example similarly as it was done very recently for the (closely related) non-negative matrix factorization problem [26]. Designing such heuristics based on our approach is a direction for further research and out of the scope of this paper.

4.3 Generalizations of Algorithm 1

In this paper, we focused on finding cliques in unweighted graphs. However, Algorithm 1 can be straightforwardly used in the following two more general scenarios, which are implemented in the code available online:

- *Weighted graphs*. If the graph is weighted (that is, a weight is assigned to each edge of the graph indicating the importance of the links between vertices), Algorithm 1 can be used and will try to identify a clique whose corresponding submatrix has the largest possible first singular value.
- *Finding dense subgraphs*. In case one is looking for dense subgraphs instead of fully connected ones, the parameter D can be kept smaller. In fact, when d is small, zero entries of the matrix B can be approximated by positive ones. At the limit, for $d = 0$, Algorithm 1 computes the first singular vector $\mathbf{u}$ of M_d which is positive (given that M_d is a primitive matrix, that is, M_d^p is positive for some p [27]). The density of the graph found by Algorithm 1 will depend on the value of D; see also [19] where the idea is experimented in the case of bicliques.

5 Conclusions

In this paper, we introduced a new continuous formulation of the maximum clique problem (MCP) using symmetric rank-one non-negative matrix approximation;

see (R1N$_d$M). We showed a one-to-one correspondence between the local (resp. global) optimal solutions of our continuous formulation and the maximal (resp. maximum) cliques of a given graph (Theorems 3.1 and 3.2). In addition, we showed that the other stationary points can be made arbitrarily close to the cliques of the graph (Theorem 3.5). We then proposed a new clique finding algorithm (Algorithm 1), applying a standard projected gradient method on our continuous formulation, and showed that the limit points of this algorithm coincide with the cliques of a given graph (Theorem 4.1). Finally, we tested our algorithm on 36 benchmark instances from the DIMACS data sets. The experimental results were compared with two other continuous clique finding algorithms based on the Motzkin–Straus formulation and one discrete approach based on a fast neighborhood search that uses multiple restarts. The results show that Algorithm 1 outperforms the two continuous methods and gives reasonable results compared to the discrete approach given that it is a single-start local search heuristic.

Acknowledgements This research was conducted when the first author was visiting the Department of Mathematics and Operational Research at the University of Mons. We would like to thank the reviewers and Prof. Kunal Narayan Chaudhury for their insightful feedback that helped us improve the paper significantly. NG acknowledges the support of the F.R.S–FNRS (incentive Grant for Scientific Research No. F.4501.16) and of the ERC (Starting Grant No. 679515).

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
2. Luce, R.D., Perry, A.D.: A method for matrix analysis of group structure. Psychometrika **14**, 94–114 (1949)
3. Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. J. Comput. Biol. **6**, 281–297 (1999)
4. Sugihara, G.: Graph theory, homology and food webs. Proc. Symp. Appl. Math. **30**, 83–101 (1984)
5. Prihar, Z.: Topological properties of telecommunication networks. Proc. IRE **44**, 927–933 (1956)
6. Rhodes, N., Willett, P., Calvet, A., Dunbar, J.B., Humblet, C.: CLIP: similarity searching of 3D databases using clique detection. J. Chem. Inf. Comput. Sci. **43**(2), 443–448 (2003)
7. Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: The maximum clique problem. In: Du, D.-Z., Pardalos P.M. (eds.) Handbook of Combinatorial Optimization, pp. 1–74. Springer, New York (1999). doi:10.1007/978-1-4757-3023-4
8. Wu, Q., Hao, J.-K.: A review on algorithms for maximum clique problems. Eur. J. Oper. Res. **242**(3), 693–709 (2015)
9. Grosso, A., Locatelli, M., Pullan, W.: Simple ingredients leading to very efficient heuristics for the maximum clique problem. J. Heuristics **14**, 587–612 (2008)
10. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turan. Can. J. Math. **17**, 533–540 (1965)
11. Bomze, I.M.: Evolution towards the maximum clique. J. Glob. Optim. **10**, 143–164 (1997)
12. Ding, C., Li, T., Jordan, M.I.: Non-negative matrix factorization for combinatorial optimization: spectral clustering, graph matching and clique finding. IEEE Int. Conf. Data Mining **44**, 183–192 (2008)
13. Gibbons, L.E., Hearn, D.W., Pardalos, P.M., Ramana, M.V.: Continuous characterization of the maximum clique problem. Math. Oper. Res. **22**(3), 754–768 (1997)
14. Pardalos, P.M., Phillips, A.T.: A global optimization approach for solving the maximum clique problem. Int. J. Comput. Math. **17**, 533–540 (1990)
15. Pelillo, M.: Relaxation labeling networks for the maximum clique problem. J. Artif. Neural Netw. **2**(4), 313–328 (1995)
16. Du, G., Gu, J., Pardalos, P.M.: Satisfiability Problem: Theory and Applications, DIMACS Series, vol. 35. American Mathematical Society, Providence (1997)

17. Horst, R., Pardalos, P.M., Van Thoai, N.: Introduction to Global Optimization. Springer, Berlin (2000)
18. Pardalos, P.M., Xue, J.: The maximum clique problem. J. Glob. Optim. **4**, 301–328 (1992)
19. Gillis, N., Glineur, F.: Continuous characterization of the maximum-edge biclique problem. J. Glob. Optim. **58**, 439–464 (2014)
20. Belachew, M.T., Gillis N.: Solving the Maximum Clique Problem with Symmetric Rank-One Non-negative Matrix Approximation. arXiv:1505.07077 (2015)
21. Ho, N.D.: Non-negative Matrix Factorization Algorithms and Applications. Doctoral dissertation, Université catholique de Louvain (2008)
22. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
23. Calamai, P.H., Moré, J.J.: Projected gradient methods for linearly constrained problems. Math. Prog. **39**, 93–116 (1987)
24. Lin, C.-J., Moré, J.J.: Newton's method for large bound-constrained optimization problems. SIAM J. Optim. **9**(4), 1100–1127 (1999)
25. Golub, G.H., Loan, C.F.V.: Matrix Computations, 3rd edn. The John Hopkins University Press, Baltimore (1996)
26. Vandaele, A., Gillis, N., Glineur, F., Tuyttens, D.: Heuristics for exact non-negative matrix factorization. J. Glob. Optim. **65**(2), 369–400 (2016)
27. Berman, A., Plemmons, R.J.: Non-negative Matrices in the Mathematical Sciences. SIAM, Philadelphia (1994)