



## On the Approximability of the Minimum Weight $t$ -partite Clique Problem

Geoffrey Solano<sup>1,3</sup> Guillaume Blin<sup>2</sup> Mathieu Raffinot<sup>2</sup>  
Jhoirene Clemente<sup>3</sup> Jaime Caro<sup>3</sup>

<sup>1</sup>Mathematics and Computing Sciences Unit, Department Physical Sciences and Mathematics, College of Arts and Sciences, University of the Philippines, Manila

<sup>2</sup>Universite de Bordeaux, CNRS, Bordeaux-INP, LaBRI, UMR 5800, 33405 Talence, France

<sup>3</sup>Algorithms and Complexity Laboratory, Department of Computer Science, College of Engineering, University of the Philippines, Diliman

### Abstract

The MINIMUM WEIGHT  $t$ -PARTITE CLIQUE PROBLEM  $MWtCP$  is the problem of finding a  $t$ -clique with minimum weight in a complete edge-weighted  $t$ -partite graph. The motivation for studying this problem is its potential in modelling the problem of identifying sets of commonly existing putative co-regulated, co-expressed genes, called gene clusters. In this paper, we show that  $MWtCP$  is NP-hard, APX-hard in the general case. We also present a 2-approximation algorithm that runs in  $O(n^2)$  for the metric case and has  $1 + \frac{1}{t}$ -approximation performance guarantee for the ultrametric subclass of instances. We further show how relaxing or tightening the application of the metricity property affects the approximation ratio. Finally insights on the application  $MWtCP$  to gene cluster discovery are presented.

Submitted: August 2018	Reviewed: January 2019	Revised: March 2019	Reviewed: July 2019
Revised: September 2019	Accepted: March 2020	Final: March 2020	Published: April 2020
Article type: Regular Paper		Communicated by: F. Vandin	

## 1 Introduction

Combinatorial optimization problems involve the search for an optimal solution in a finite or countably infinite set of potential solutions. Many of these problems involve searching for a discrete structure of extremal weight in a given graph. Paths, cycles, spanning trees, matchings, and cliques are just some examples of such structures. Some of these problems are polynomially solvable such as the minimum spanning tree problem [15, 26]. Unfortunately, however, many of these combinatorial problems are intractable. Among such problems is the WEIGHTED CLIQUE PROBLEM (WCP), which is the problem of finding a clique of a given size (or number of vertices) that is of extremal weight, measured as the sum of weights of the edges included in the said clique, in a complete weighted graph. Nonetheless, these problems are still extensively studied because of their important applications in different domains. [14, 19, 25].

Cliques have been thoroughly studied in the area of graph theory (*e.g.*, [18, 7, 4, 9, 34, 16, 17, 15, 20]). Up to the most recent years, much is still being explored on the WCP and its variants [2, 22, 27, 32]. It was shown in [19] that the optimization variant of the maximum clique problem is hardly approximable, i.e. it is hard to approximate the problem in polynomial time within a factor of  $n^{1-\epsilon}$ , for any  $\epsilon > 0$  unless  $NP = coR$ . In [12] it was shown that both minimum and maximum WCPs are not approximable in the general case, i.e. if there are no restrictions on the weights, then it would not be possible to come up with a polynomial-time approximation algorithm and a small value  $r$  such that on every input, the algorithm finds a solution whose cost is at most  $r$  times the optimum, for minimization problems (at least  $1/r$  for maximization problems)[33]. A fast 2-approximation algorithm was presented, however, in [12] for two important cases of the problem, in which vertex weights are nonnegative and edge weights either satisfy the triangle inequality (the metric WCP) or are squared pairwise distances for a point configuration in Euclidean space (the quadratic Euclidean WCP).

Here we consider a slightly different version of the problem motivated by the problem of identifying sets of commonly existing putative co-regulated, co-expressed genes, called gene clusters, for which extensive studies are being pursued [28, 6, 21, 10, 11, 13, 29]. In molecular biology, genes belonging to a cluster may share functional dependencies and may be involved in the expression of a specific trait. Identifying these clusters is also essential in establishing relationships between organisms as well as in the discovery of drugs and disease treatments [1, 23]. This biological problem of identifying associations among genes across genomes can be formalized as searching for cliques on undirected weighted graphs [3, 5, 8, 24, 31]. For instance, one-to-one correspondence can be made between genomes and partitions, i.e.  $t$  partitions can represent  $t$  genomes. Gene contents of different intervals of genes in a genome can correspond to vertices in a partition. Co-expression or similarity among the gene content of two interval groups can be represented by the weight placed on the edge joining the pair of vertices corresponding to them. Depending on the weight formulation this task of discovering gene groups that highly resemble each other across the

$t$  genomes, also known as approximate gene clusters, can be reduced intuitively to the problem of finding either the maximum or minimum weighted clique.

In this paper, we present the MINIMUM WEIGHT  $t$ -PARTITE CLIQUE PROBLEM (MW $t$ CP), which is the problem of finding a  $t$ -clique with minimum weight in a complete edge-weighted  $t$ -partite graph. We first show how it is related to WCP. We also show that it is NP-hard, APX-hard in the general case. We present a 2-approximation algorithm that runs in  $O(n^2)$  for the metric case and has a  $1 + \frac{1}{t}$  approximation performance guarantee for the ultrametric subclass of instances. We further show how relaxing or tightening the application of the metricity property affects the approximation ratio. Finally we present some insights when MW $t$ CP is applied to gene cluster discovery.

## 2 Preliminaries

### 2.1 The Weighted Clique Problem (WCP)

Eremin et. al., in [12], presented a general combinatorial problem called the Weighted Clique Problem (WCP). Given a complete simple weighted undirected graph  $G = (V, E, a, c)$  and weight functions  $a : V \rightarrow \mathbb{Q}$  and  $c : E \rightarrow \mathbb{Q}$ , which define the vertex weights and the edge weights respectively. The sum  $\sum_{v \in V} a_v + \sum_{e \in E} c_e$  is called the weight of the graph  $G$ .

#### Weighted Clique Problem (WCP)

**Given:** a complete weighted undirected graph  $G = (V, E, a, c)$ , where  $a : V \rightarrow \mathbb{Q}$  and  $c : E \rightarrow \mathbb{Q}$ , and a positive integer  $k$ .

**Find:** a complete subgraph (clique) of the graph  $G$  of order  $k$ , i.e. having  $k$  vertices, with the smallest (largest) weight.

Without loss of generality, we identify the sets  $V$  and  $N_n = \{1, \dots, n\}$ , where  $n = |V|$ , and use the short notation  $c_{ij}$  for the image  $c_e$  of an edge  $e = ij, j \in E$  under the weight mapping  $c$ .

The problem MIN-EWCP, a variant of WCP, was further defined in [12] as the problem of finding the clique of order  $m$  in the weighted complete graph  $G = (V, E, a, c)$  with the smallest weight, where  $a_i = 0$  and  $c_{ij} \geq 0$ . It was shown that this problem is not in APX.

In the following section we present a related problem to MIN-EWCP which is the MINIMUM WEIGHTED  $t$ -PARTITE CLIQUE PROBLEM (MW $t$ CP).

### 2.2 The Minimum Weighted $t$ -partite Clique Problem

A  $t$ -partite graph is a graph  $G = (V, E)$  whose vertices can be partitioned into  $t$  different independent sets  $\{V_1, V_2, \dots, V_t\}$  (i.e.,  $V = V_1 \cup V_2 \cup \dots \cup V_t$  and any graph induced by a  $V_i, 1 \leq i \leq t$ , has no edge). Equivalently, if  $G$  is a  $t$ -partite graph, then each of its vertices can be colored with any one of its  $t$  colors such that no edge is incident to two vertices having the same color. A complete

$t$ -partite graph is a  $t$ -partite graph in which there is an edge between every pair of vertices from different independent sets. If there are  $m$  vertices in each partition, then  $|V| = mt$ , making  $G$   $m(t-1)$ -regular. Using the Handshaking Lemma, the total number of edges in the graph  $G$ , is  $|E| = m\binom{t}{2}$  [30].

Provided a weight for each edge (*e.g.*, defined as a function  $w : E \rightarrow \mathbb{N}$ ), a main problem in graph theory lies in finding a clique with maximum or minimum weight. We formally define the MINIMUM WEIGHTED  $t$ -PARTITE CLIQUE PROBLEM (MWtCP) as follows:

**Minimum Weighted  $t$ -partite Clique Problem (MWtCP)**

**Given:** a complete edge-weighted  $t$ -partite graph  $G = (V, E, w)$ , where  $|V| = n$ .

**Find:** a complete subgraph (clique) of order  $t$  in  $G$  of minimum weight

Every maximal clique in graph  $G$  is of order  $t$ , each of which has  $\binom{t}{2}$  edges, and the total number of unique  $t$ -cliques in  $G$  is  $m^t$ . Clearly, each vertex in the  $t$ -clique belongs to a unique partition. Each unique edge is a part of  $m^{t-2}$  cliques. Naively, we can check all  $t$ -cliques and determine which among them has the minimum total weight in  $O(m^t)$ .

A case of MWtCP is called *metric* if all edge weights in  $G$  follow the triangle inequality, *i.e.*, for any distinct vertices  $a, b, c \in V$ ,  $w(a, b) \leq w(b, c) + w(a, c)$ , where  $w(u, v)$  is the weight of the edge  $(a, b)$ . Similarly, a case of MWtCP is called *ultrametric* if all edge weights in  $G$  follow the rule: for any distinct vertices  $a, b, c \in V$ ,  $w(a, b) = \max(w(b, c), w(a, c))$ .

Following the method presented in [25, 12] we assign to each MWtCP the polynomial time equivalent problem of finding the extremum of the integer linear function

$$\sum_{1 \leq i, j \leq m, 1 \leq p, q \leq t, p \neq q} w_{ij}^{pq} y_{ij}^{pq} \rightarrow \min_{y_{ij}^{pq} \in \{0,1\}}$$

on the feasible set

$$\begin{aligned} \sum_{1 \leq i \leq m} x_i^1 &= 1, \quad \sum_{1 \leq i \leq m} x_i^2 = 1 \quad \dots \quad \sum_{1 \leq i \leq m} x_i^t = 1 \\ \sum_{1 \leq i, j \leq m, 1 \leq p, q \leq t} y_{ij}^{pq} &= \binom{t}{2} \quad y_{ij}^{pq} = x_i^p \wedge x_j^q \\ w_{ij}^{pq} &\geq 0, \quad x_i^p \in \{0, 1\} \end{aligned}$$

where  $x_i^p$  and  $x_j^q$  represents the  $i$ th node in partition  $p$  and the  $j$ th node in partition  $q$  respectively.  $w_{ij}^{pq}$  represents the weight of the edge incident to nodes  $x_i^p$  and  $x_j^q$ , while  $y_{ij}^{pq}$  whose value is either 0 or 1, indicates its inclusion or non-inclusion in the  $t$ -clique, which has  $\binom{t}{2}$  edges.

### 2.3 Minimum Weighted $t$ -partite Clique Problem is Mapping Reducible to Min-EWCP

Earlier it was already mentioned that  $MWtCP$  is closely related to  $MIN-EWCP$ . Here we show that  $MWtCP$  is at most as hard as  $MIN-EWCP$ .

**Theorem 1**  $MWtCP \leq_M MIN-EWCP$

**Proof:** To prove this, there must be a computable function  $f$  such that  $a$  is an input to  $MWtCP$  if and only if  $f(a)$  is an input to  $MIN-EWCP$

Given any instance  $G = (V, E, w)$  of the  $MWtCP$ , i.e. a complete edge-weighted  $t$ -partite graph, where  $V = \{v_1, v_2, v_3, \dots, v_n\}$ , we define a function  $f$  that builds a corresponding instance  $G' = (V', E', a', c')$  of  $MIN-EWCP$ , i.e.  $G' = f(G)$ .

We let  $V' = V$ , and note that  $a'_r = 0$  for  $1 \leq r \leq n$ . For every edge  $e_{ij} = (v_i, v_j) \in E$  where  $v_i$  and  $v_j \in V$ , then  $e'_{ij} = e_{ij}$  where  $e'_{ij} \in E'$  and  $c'_{ij} = w_{ij} = weight(e_{ij})$ . Furthermore, for every  $v_i$  and  $v_j \in V$ , where  $1 \leq i \neq j \leq n$ , if  $e_{ij} \notin E$ , then  $e_{ij} \in E'$  and  $c'_{ij} = weight(e'_{ij}) = \infty$ , making  $G'$  a complete edge-weighted graph.

Clearly, the problem of finding the clique of order  $t$  with the least total weight in  $G'$ , is the same as obtaining the solution to the  $MWtCP$  for the input graph  $G'$ .

Thus,  $f$ , where  $G' = f(G)$ , is a computable function. Therefore,  $MWtCP \leq_M MIN-EWCP$   $\square$

We note, however, that the converse may not be so since reducing any input for  $MIN-EWCP$  into an input for  $MWtCP$ , i.e. converting a complete graph into a complete  $t$ -partite graph, would entail deletion of edges as well as partitioning of vertices. Moreover,  $m$ , which is the order of the clique in  $MIN-EWCP$ , may assume a range of values, i.e.  $1 < m \leq n$ , whereas the order of clique in  $MWtCP$  is specifically the number of partitions.

Nonetheless, the  $MWtCP$ , though it can never be harder than  $MIN-EWCP$ , remains to be of particular interest because of its potential in modelling the problem of identifying approximate gene clusters. In the next section we show that it is both NP-hard and APX-Hard.

## 3 NP and APX Hardness of the $MWtCP$

In this section, we provide the two main hardness results concerning the general case of  $MWtCP$ ; that is its NP and APX hardness.

To prove that  $MWtCP$  is NP hard, we first show a variant of WCP, which is the  $MAXIMUM\ CLIQUE$  problem:

**Given:** a graph  $G = (V, E)$ , where  $|V| = n$  and an integer  $t$ .

**Does there exist:** a clique of order  $t$  in  $G$ , i.e., a vertex set  $V_c \subseteq V$ , such that

$|V_c| = t$  and  $\forall u, v \in V_c, (u, v) \in E$  ?

We now show that there is a reduction from MAXIMUM CLIQUE problem to MWtCP.

**Lemma 2** *An instance of MAXIMUM CLIQUE problem can be transformed to an instance of MWtCP in polynomial time.*

**Proof:** Starting from any instance  $(G = (V, E), t)$  of the MAXIMUM CLIQUE problem where  $V = \{v_1, v_2, v_3, \dots, v_n\}$ , we build a corresponding instance  $G' = (V', E', w)$  of MWtCP problem as follows. The set of vertices is defined as  $V' = V'_1 \cup V'_2 \cup V'_3 \cup \dots \cup V'_t$ , where for  $1 \leq i \leq t$ , and there is a copy of each vertex of  $V$  in every  $V'_i = \{v_i^1, v_i^2, v_i^3, \dots, v_i^t\}$ . The set of edges is defined as  $E' = \{(v_i^x, v_j^y) \mid \forall 1 \leq x, y \leq t; x \neq y; (v_i, v_j) \in E\}$ . Roughly, for each edge  $(v_i, v_j) \in E$ , we build a complete bipartite graph  $K_{t,t}$  using the following two disjoint sets of vertices  $\{v_i^1, v_i^2, \dots, v_i^t\}$  and  $\{v_j^1, v_j^2, \dots, v_j^t\}$  and remove all the following edges from it  $\{(v_i^x, v_j^x) \mid 1 \leq x \leq t\}$ .

Furthermore, for all such edges  $e$  currently in  $E'$ , we set  $w(e) = 0$ . We then add edges to  $E'$  to come up with a complete  $t$ -partite graph, *i.e.*,  $E' = E' \cup E'_r$  where  $E'_r = \{(v_i^x, v_j^y), \mid \forall 1 \leq x \leq y \leq t; i \neq j; (x = y \mid (v_i, v_j) \notin E)\}$ . Each edge  $e \in E'_r$  has a unit weight defined as  $w(e) = 1$ . A complete illustration on a small instance can be seen in Figure 1 (for ease of readability, only the edges not in  $E'_r$  have been drawn). Clearly, this construction can be done in polynomial time.  $\square$

We note some interesting properties of the above construction. As already mentioned, each vertex  $v_i \in V$  of the original graph  $G$  is represented as a clique in  $G'$  composed of the vertices  $\{v_i^1, v_i^2, \dots, v_i^t\}$  with edges weighted to 1 (for ease denoted as  $v_i$ 's). Also, each edge  $(v_i, v_j) \in E$  of the original graph  $G$  is represented as a bipartite graph with null weight between the sets of vertices  $v_i$ 's and  $v_j$ 's representing respectively  $v_i$  and  $v_j$  (for ease denoted as  $B_{t,t}^{(i,j)}$ ).

**Lemma 3** *At most one arc from each  $B_{t,t}^{(i,j)}$  can be part of any solution of null cost to MWtCP.*

**Proof:** To reach a solution of null cost to MWtCP, only edges of null weight can be taken. Therefore, from the previous observation and by construction, if two nodes of  $v_i$ 's were part of the solution, they will not be linked by an edge in the solution. Thus, for any  $i$  and  $j$ , at most one vertex of both  $v_i$ 's and  $v_j$ 's can belong to the solution, inducing that only one arc from each  $B_{t,t}^{(i,j)}$  can be part of the solution.  $\square$

**Lemma 4** *Let  $(G = (V, E), t)$  be an instance of the MAXIMUM CLIQUE problem and the corresponding graph  $G' = (V', E', w)$  of the MWtCP obtained by the above construction.  $G$  contains a  $t$ -clique if and only if  $G'$  contains a  $t$ -clique of null cost.*

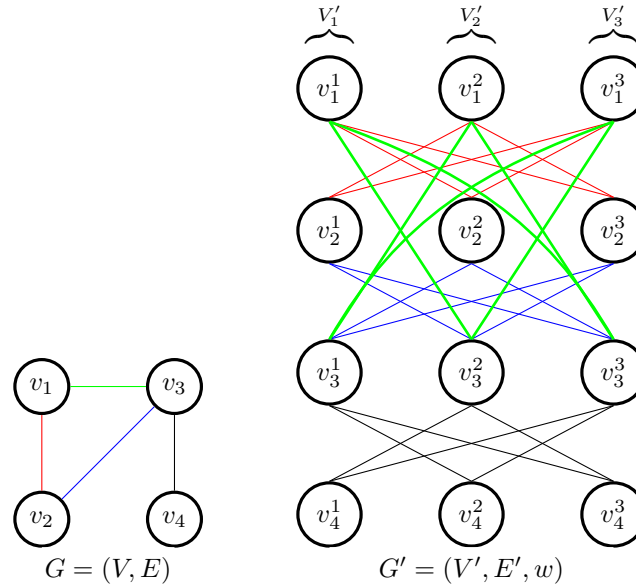


Figure 1: The graph  $G$  of the MAXIMUM CLIQUE problem and the corresponding graph  $G'$  of the MWtCP where only arc of null weight have been drawn; missing arcs (for readability) are of weight 1.

**Proof:** ( $\Rightarrow$ ) Suppose  $G$  contains a  $t$ -clique composed of the vertices in  $V_c \subseteq V$ . We build a solution  $V'_c \subseteq V'$  to MWtCP as follows. For each vertex  $v_i \in V_c$ , add the vertex  $v_i^i$  to  $V'_c$ . Clearly, for each vertex  $v_i$  of the  $t$ -clique  $V_c$ , a copy of  $v_i$  in exactly one of  $V'_j$ 's is selected to be part of  $V'_c$ .

Let us now prove that  $V'_c$  forms a  $t$ -clique in  $G'$ . By definition, since  $V_c$  is a  $t$ -clique then there exists an edge between any two of the  $t$  vertices belonging to  $V_c$ . In  $G'$  there is an edge between any vertex  $v_i^x$  and any vertex of the  $v_j^y$ , where  $i \neq j$  and  $x \neq y$ , if  $v_i$  and  $v_j$  are adjacent in  $G$ . Thus, any two pair of vertices  $v_i^i$  and  $v_j^j$  in  $V'_c$  are adjacent in  $G'$  if the corresponding two vertices  $v_i$  and  $v_j$  are in  $V_c$ . Clearly, it follows that  $V'_c$  also forms a  $t$ -clique.

Let us now prove that the corresponding clique has null cost. By construction of  $V'_c$ , we selected a vertex in each independent set  $V'_j$ , therefore in  $V'_c$ , only edges between vertices belonging to different  $V'_j$ 's exist. Since  $V_c$  is a clique, there is an edge in  $G$  between any pair of vertices of  $V'_c$ . Remind that, by construction, each edge  $(v_i, v_j) \in E$  of the original graph  $G$  is represented as a bipartite graph with null weight between the sets of vertices  $v_i$ 's and  $v_j$ 's representing respectively  $v_i$  and  $v_j$ . Thus, any edge between nodes in  $V'_c$  has a null cost.

( $\Leftarrow$ ) Suppose  $G'$  contains a  $t$ -clique of null cost composed of the vertices in  $V'_c \subseteq V'$ . We build a solution  $V_c \subseteq V$  to the MAXIMAL CLIQUE problem as

follows. For each vertex  $v_i^x \in V'_c$ , add the vertex  $v_i$  to  $V_c$ . Roughly, for any copy of a vertex  $v_i$  belonging to the  $t$ -clique  $V'_c$ , we select the vertex  $v_i$  as part of  $V_c$ .

Let us prove that  $V_c$  is indeed a  $t$ -clique in  $G$ . First note that  $|V_c| = t$ . Indeed, since  $V'_c$  is a  $t$ -clique of null cost, it cannot contain any edge between two vertices  $v_i^x$  and  $v_i^y$  for all  $1 \leq i \leq n$  and  $1 \leq x, y \leq t$  and thus, exactly  $t$  different nodes have been added to  $V_c$ . By Lemma 3, we know that the subgraph induced by  $V'_c$  is a null clique and thus that it only uses edges of null weight; that is edges belonging to  $\{B_{t,t}^{(i,j)} | 1 \leq i < j \leq n\}$ . By Lemma 3, at most one edge of each  $\{B_{t,t}^{(i,j)} | 1 \leq i < j \leq n\}$  is used in  $V'_c$ . This implies by construction, that there exists also an edge between any pair of vertices in  $V_c$ . All together, this leads to prove that  $V_c$  forms a  $t$ -clique inducing the correctness of Lemma 4.  $\square$

**Theorem 5** *MWtCP is NP-hard when the weighting function  $w$  is not a constant function (i.e., at least two different output values are provided).*

**Proof:** First, we note that, in case of a constant function  $w$ , the problem is trivial since any set  $\{v_1, v_2, \dots, v_t | v_i \in V_i\}$  is an optimal solution. In the case when there exists at least two different output values from the weighting function  $w$ , it is clearly shown from the proofs of Lemma 2 and Lemma 4 that MWtCP is NP-hard.  $\square$

Let us now prove moreover that this reduction is an L-reduction provided a slight modification in the construction of  $G'$ .

**Theorem 6** *MWtCP is APX-hard, for any  $t \geq 3$ .*

**Proof:** We proceed with a similar construction as in the previous proof by only adapting the weight of the edges belonging to  $E'_r$  as follows:  $\forall e \in E'_r, w(e) = \frac{2}{t(t-1)}$ .

First, notice that in this new construction, an optimal  $t$ -clique in  $G'$  has still a null cost. Therefore,  $OPT(MWtCP) < OPT(MAXIMUM CLIQUE PROBLEM)$ .

Starting from any given candidate solution to MWtCP, that is a  $t$ -clique of cost  $\zeta$  (which may not be optimal), we can build a solution to the MAXIMUM CLIQUE problem, say, of size 2. This can be done by arbitrarily picking any edge in  $E$ , which indeed forms a clique. Let us now compare the score of the solutions. We obtain that the distance from the optimal solution to MINIMUM WEIGHT  $t$ -PARTITE CLIQUE and the candidate solution is  $|0 - \zeta|$  while the distance between the optimal solution to the MAXIMUM CLIQUE and the built candidate solution is  $|t - 2|$ . In order for the reduction to be an L-reduction, the following inequality should hold  $|t - 2| \leq |0 - \zeta|$ . Since the optimal  $t$ -clique has a null cost, then  $\zeta \geq 0$ . Therefore,  $|t - 2| \leq \zeta$ . One of the possible cases implies that  $t - 2 \leq \zeta$ . Since any  $t$ -clique is composed of  $\frac{t(t-1)}{2}$  edges and, by construction, if any edge has a weight greater than or equal to  $\frac{2}{t(t-1)}$ , then the requested inequality indeed holds for any  $t \geq 3$ . The proposed reduction is therefore an L-reduction. Since the MAXIMUM CLIQUE problem is APX-hard, so is MWtCP.  $\square$



## 4 An Approximation Algorithm for the Metric Case of MWtCP

In the previous section we demonstrated that MWtCP is both NP-hard and APX-hard. By definition, in the general case, it would not be possible to develop an algorithm that yields to a constant approximation ratio.

This opens up the question of whether the presence of restrictions on edge weights, such as the triangle inequality, would provide for a more approximable result. In this section, we present an algorithm for the MWtCP that runs in  $O(n^2)$  that has a relative performance guarantee of 2 for the metric case and a relative performance guarantee of  $1 + \frac{1}{t}$  for the ultrametric subclass of instances. We also demonstrate how the approximation ratio is affected when tightening or relaxing the metric property.

### 4.1 Algorithm Definition

We first define a  $t$ -partite star. An  $n$ -star is the graph  $K_{1,n}$  that has  $n + 1$  vertices. Of these vertices,  $n$  have the degree 1 and one has a degree  $n$ , which, in this study will be referred to as the hub or center of the star. A  $t$ -PARTITE STAR is a  $(t - 1)$ -star in a complete  $t$ -partite graph where all the vertices are part of a distinct partition. The proposed algorithm tries to approximate the solution to the MWtCP by finding the minimum weighted  $t$ -partite star. The approach is based on the fact that a complete graph  $K_n$  can be decomposed as a complete graph  $K_{n-1}$  and a star  $K_{1,n-1}$ . The weight of a candidate star is computed as the sum of the weights of the edges in the  $t$ -partite star. The algorithm searches for the minimum weighted  $t$ -partite star in the complete  $t$ -partite graph, from the  $n$  candidate stars, and uses its vertex set to define the candidate  $t$ -clique.

A formal definition of the algorithm is given in Algorithm 1.

**Input:** A complete weighted  $t$ -partite graph  $G = (V, E, w)$

- 1: for every node  $x$  find the minimum weighted  $t$ -star with  $x$  as the hub or center vertex
- 2: find the minimum weighted  $t$ -star from the  $n$  candidate  $t$ -stars in #1
- 3: return the  $t$ -clique formed from the vertex set of the  $t$ -star in #2

**Output:** a weighted  $t$ -Clique

**Algorithm 1:** Minimum Weighted  $t$ -1 Star Algorithm for MWtCP

### 4.2 A Related Problem and Solution

It has been shown earlier in this study that  $MWtCP \leq_M \text{MIN-EWCP}$ . In the study [12] Eremin et. al. presented the row's subset of symmetric matrix problem (RSSM) as a polynomial time equivalent formulation of MIN-EWCP in the form of property verification problem.

**Definition 4.1 (Row's subset of symmetric matrix problem (RSSM) [12])**

Given a symmetric  $n \times n$  matrix  $W = (w_{ij})$  with nonnegative entries and  $w_{ii} = 0$ , a positive integer  $m$  and a positive number  $D$ , determine whether the set of rows of  $W$  contains a subset  $C$  of cardinality  $m$  such that

$$F(C) = \frac{1}{2} \sum_{i \in C} \sum_{j \in C} w_{ij} \leq D$$

The algorithm presented in the said study, which we will refer to here as Algorithm *A* is as follows:

**Step 1.** For each  $j = 1, \dots, n$ , find a set  $B_j$  that consists of indices of  $m$  smallest entries in the  $j$ th row of  $W$  including the index  $j$  itself. Define  $S(B_j) = \sum_{i \in B_j} w_{ij}$ .

**Step 2.** Denote by  $k^*$  the value  $j$  for which  $S(B_j)$  takes the minimum value  $S^* = S(B^*) = \sum_{i \in B^*} w_{ik^*}$ .

Take  $C = B^*$  as an approximate solution of RSSM.

Algorithm *A* was shown in the said study as a 2-approximation algorithm for the metric case of RSSM, and consequently, therefore a 2-approximation algorithm for the metric case of MIN-EWCP. It does not automatically follow, however, that Algorithm *A* will provide the same performance guarantee for an input of MIN-EWCP that was converted from an input of MWtCP and that it will always return feasible solutions for MWtCP. Recall that in performing the function  $f$  to produce a graph  $G'$  from an input graph  $G$  of MWtCP, edges are added to  $G'$  that were not in  $G$  and such edges have the weight  $\infty$ . The presence of edges in  $G'$  with such weights violates the metric property. Furthermore, since Algorithm *A* does not take into consideration the presence of partitions, two or more of the  $m$  smallest entries returned for a particular row may actually come from the same partition. However, a solution for MWtCP is a  $t$ -clique, in which, by definition, no two vertices are elements of the same partition. Thus, an approximation solution for MIN-EWCP may not always be applicable for MWtCP.

Some modifications, therefore, must be made on Algorithm *A* to make it applicable for MWtCP. We note that the adjacency matrix of  $G$  has  $n$  columns. Since  $n = m * t$ , the columns of the adjacency matrix can be represented as having  $t$  partitions, each having  $m$  columns. We now consider the following modified approximation algorithm as Algorithm *A'*:

**Step 1.** For each row  $j = 1, \dots, n$ , find a set  $B_j$  that consists of indices of  $t$  smallest entries for each of the  $t$  partitions of  $m$  columns in the  $j$ th row of  $W$  including the index  $j$  itself. Define  $S(B_j) = \sum_{i \in B_j} w_{ij}$ .

**Step 2.** Denote by  $k^*$  the value  $j$  for which  $S(B_j)$  takes the minimum value  $S^* = S(B^*) = \sum_{i \in B^*} w_{ik^*}$ .

Take  $C' = B^*$  as an approximate solution.

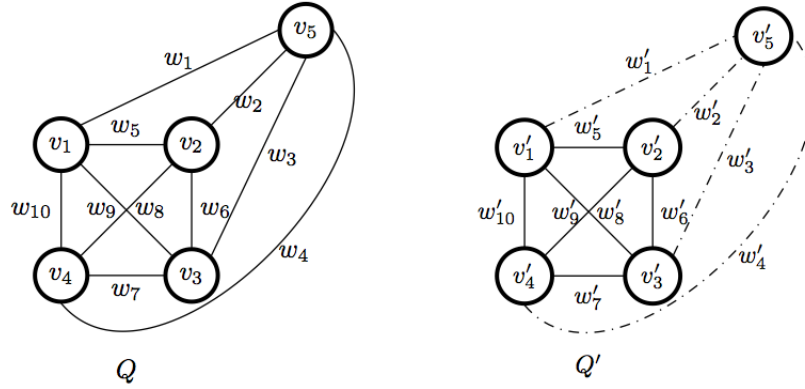


Figure 2: The optimal  $t$ -clique  $Q$  and the solution  $Q'$  in graph  $G$  by Algorithm 1 for  $t = 5$

### 4.3 Analysis of Algorithm 1 for MW $t$ CP

With careful observation, it is easy to see that performing Algorithm 1 to obtain the minimum weight  $(t - 1)$ -star for  $G$  in MW $t$ CP is actually equivalent to performing Algorithm  $A'$  on the adjacency matrix of  $G$  for MW $t$ CP. Equivalently, Algorithm  $A'$  does the following steps in approximating the MW $t$ CP, this time from a graph perspective.

**Step 1.** For each vertex  $u = 1, \dots, n$ , find the set of vertices  $B_u$ , where  $|B_u| = (t - 1)$ , such that if  $u \in \text{partition } U_i$ , then for each vertex  $v \in B_u$ , if  $v \in \text{partition } U_j$ , where  $U_j \neq U_i$ , then  $\text{cost}(u, v)$  is minimum

**Step 2.** For all the  $(t - 1)$ -stars obtained in Step 1, determine the one with minimum weight.

**Theorem 7** *Algorithm 1 is a 2-approximation algorithm for the metric case of MINIMUM WEIGHT  $t$ -PARTITE CLIQUE PROBLEM which runs in  $O(n^2)$  on a complete weighted  $t$ -partite graph of  $n$  vertices.*

**Proof:** Algorithm 1 clearly tries to determine the minimum weighted  $(t - 1)$ -star and then uses this to approximate the minimum weighted  $t$ -clique. We now show through another approach that it is a 2-approximation efficient algorithm, that is, the weight of the clique returned by Algorithm 1 is at the worst case twice that of the optimal solution.

Given a complete graph  $G$  with positive edge weights and  $Q$  as the  $t$ -clique of minimum weight with  $e$  edges, where  $e = \binom{t}{2}$ , having the weights  $w_1, w_2, w_3, \dots, w_e$ , we let  $Q'$  be the  $t$ -clique returned by Algorithm 1 whose edges have the weights  $w'_1, w'_2, w'_3, \dots, w'_e$ . We note that  $Q'$  is derived from the minimal  $(t - 1)$ -star  $Q^*$ . For ease of notation, let  $\{v'_1, v'_2, \dots, v'_t\}$  be the vertices of  $Q^*$  and, without loss of generality,  $v'_t$  be its center (or hub) and  $\sum_{i=1}^{t-1} w'_i$  be the

total weights of its edges ( $w'_i$  being the weight of the edge between  $v'_i$  and  $v_i$ ). An instance where  $t = 5$  is shown in Figure 2, where  $Q^*$  has dashed edges.

By definition,

$$\text{cost}(Q) \leq \text{cost}(Q') = \sum_{i=1}^e w'_i.$$

By correctness of Algorithm 1, there is no  $(t-1)$ -star in  $G$  of cost lower than that of  $Q^*$ , which is  $\sum_{i=1}^{t-1} w'_i$ . Also, of the  $(t-1)$ -stars in  $Q$ , of which there are  $m$ , none would have a cost lower than that of  $Q^*$ . Therefore, at best, there will be  $t$  minimum weighted  $(t-1)$ -stars in  $G$  and all of them are in  $Q$ , that is, each of the  $t$  vertices in  $Q$  is a hub of a minimum weighted  $(t-1)$ -star in  $G$ . We can deduce that

$$\begin{aligned} \frac{1}{t-1} \cdot \binom{t}{2} \cdot \sum_{i=1}^{t-1} w'_i &\leq \text{cost}(Q) \\ \frac{t}{2} \cdot \text{cost}(Q^*) &\leq \text{cost}(Q) \end{aligned}$$

Let us now seek for an upper-bound on  $\text{cost}(Q')$ . We note that the cost of  $Q'$  is actually the sum of the cost of  $Q^*$  and the cost of the remaining edges in  $Q'$  but not in  $Q^*$ :  $\text{cost}(Q') = \sum_{i=1}^e w'_i = \sum_{i=1}^{t-1} w'_i + \sum_{i=t}^e w'_i$ .

Clearly, the cost of the remaining  $\binom{t-1}{2}$  edges that are part of  $Q'$  but not part of  $Q^*$  is  $\sum_{i=t}^e w'_i$ . We note that these  $\binom{t-1}{2}$  edges form a  $(t-1)$ -clique that is composed of vertices  $\{v'_1, v'_2, v'_3, \dots, v'_{t-1}\}$  via the Handshaking Theorem. Let us study the constraints due to the metric property in triangles using  $v'_i$  and any two vertices  $v'_p$  and  $v'_q$  of  $Q'$ ,  $1 \leq p < q < t$ . Due to the triangular inequality, we can upper-bound the cost of the edge incident to  $v'_p$  and  $v'_q$  by

$$w((v'_p, v'_q)) \leq w((v'_p, v'_i)) + w((v'_q, v'_i)) = w'_p + w'_q$$

We moreover deduce that

$$\begin{aligned} \sum_{i=t}^e w'_i &= \sum_{1 \leq p < q \leq t-1} w((v'_p, v'_q)) \leq \sum_{1 \leq p < q \leq t-1} (w'_p + w'_q) \\ &\leq (t-2) \cdot \sum_{i=1}^{t-1} w'_i \end{aligned}$$

Thus,

$$\begin{aligned} \text{cost}(Q') &= \sum_{i=1}^{t-1} w'_i + \sum_{i=t}^e w'_i \leq \sum_{i=1}^{t-1} w'_i + (t-2) \cdot \sum_{i=1}^{t-1} w'_i \\ &\leq (t-1) \cdot \sum_{i=1}^{t-1} w'_i \\ &\leq (t-1) \cdot \text{cost}(Q^*) \end{aligned}$$

$$\begin{aligned}
 \text{Since } \text{cost}(Q^*) &\leq \frac{2}{t} \cdot \text{cost}(Q), \\
 \text{cost}(Q') &\leq (t-1) \cdot \frac{2}{t} \cdot \text{cost}(Q) \\
 &\leq \left(2 - \frac{2}{t}\right) \cdot \text{cost}(Q) \\
 &\leq 2 \cdot \text{cost}(Q)
 \end{aligned}$$

□

We just showed that the weight of the clique returned by Algorithm 1 is at the worst case twice that of the optimal solution. Intuitively, the performance of the algorithm improves with smaller values of  $t$ .

## 5 Approximation Ratios for Ultrametric Case

We now show performance guarantee of Algorithm 1 in another case of inputs, specifically, the ultrametric case.

### Approximation Ratio for Ultrametric Case for Algorithm 1

**Theorem 8** *Algorithm 1 is a  $1 + \frac{1}{t}$ -approximation algorithm for ultrametric case of MWtCP*

**Proof:** When considering ultrametric weight function  $w$  (i.e., for any distinct vertices  $a, b, c, \in V', w(a, b) \leq \max(w(b, c), w(c, a))$ ), we note the following:

In obtaining the performance guarantee of Algorithm 1 for the ultrametric case of inputs, we wish to obtain  $\alpha$  for which  $\text{cost}(Q') \leq \alpha \cdot \text{cost}(Q)$ . Intuitively, necessary for this is determining and describing the instance when difference between the costs of  $Q'$  and  $Q$  is the greatest.

To describe such instance, we note that  $\text{cost}(Q') \leq \text{cost}(Q^*) + \sum_{i=t}^e w'_i$ , and  $\text{cost}(Q) \leq \text{cost}(Q^A) + \sum_{i=t}^e w_i$ , where, by definition,  $Q^A$  is a  $(t-1)$ -star, such that  $\text{cost}(Q^*) < \text{cost}(Q^A)$ , even if  $\text{cost}(Q) \leq \text{cost}(Q')$ . For  $Q^*$  to be chosen as the  $(t-1)$ -star in  $G$  of minimum weight, it is because though it has  $(t-2)$  edges that have relatively larger weight, say  $b$ , it has an edge that has a very small weight, say  $a$ , making the  $\text{cost}(Q^*)$  still the least among the  $(t-1)$ -stars. The same, however, cannot be said about  $Q'$ . We note that  $\sum_{i=t}^e w'_i$  makes use of edges that are part of  $Q^*$ , as is shown in Figure 2, but because of the ultrametric property, each of the edges in  $Q'$  but not in  $Q^*$ , i.e.  $E(Q' \setminus Q^*) = w'_5, w'_6, \dots, w'_9, w'_{10}$ , will assume the weight value of the higher-weighted edge between the two edges it forms a triangle with. Since each of the edges in  $E(Q' \setminus Q^*)$ , i.e. the non-dashed edges, would have to be from between the dashed edges (i.e. edges in  $Q^*$ ), then  $\sum_{i=t}^e w'_i = \binom{t-1}{2} \cdot b$ . Therefore,  $\text{cost}(Q') = a + (t-2) \cdot b + \binom{t-1}{2} \cdot b$

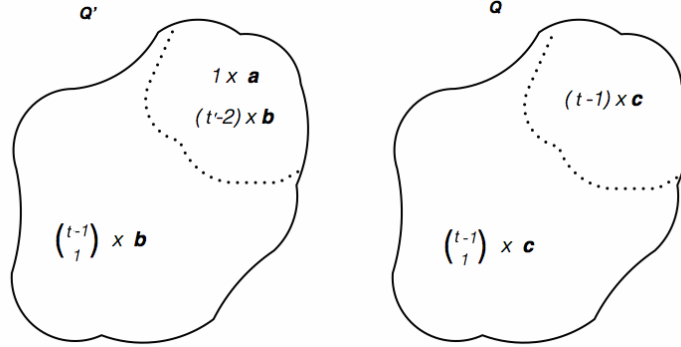


Figure 3: The instance where  $\alpha$  is greatest where  $\text{cost}(Q') \leq \alpha \cdot \text{cost}(Q)$

On the other hand, in describing such instance for  $Q$ , since  $\text{cost}(Q) \leq \text{cost}(Q^A) + \sum_{i=t}^e w_i$ , where  $\text{cost}(Q^*) < \text{cost}(Q^A)$ , then given that  $a$  is a very small positive value, we can let  $\text{cost}(Q^*) = \text{cost}(Q^A) + a$ . For the edges that are not in  $Q^A$ , without loss of generality, if we let  $w_1 \leq w_2 \leq w_3 \leq \dots \leq w_{t-1}$ , where  $w_i \in E(Q^A)$ , then because of the ultrametric property,  $\sum_{i=t}^e w_i = w_2(1) + w_3(2) + w_4(3) + \dots + \leq w_{t-1}(t-2)$ .

Since larger weighted edges would be chosen more often to be in  $E(Q \setminus Q^A)$ , i.e.  $w_i$  for  $t \leq i \leq e$ , consequently making  $\text{cost}(Q)$  larger, then clearly the instance when  $\text{cost}(Q)$  is minimum is when all the edges in  $Q$  are all of equal weight, say  $c$ . Thus,  $\text{cost}(Q) = \binom{t}{2} \cdot c$  as shown in Figure 3, where  $c = \frac{\text{cost}(Q^A) + a}{t-1} = \frac{2a + (t-2) \cdot b}{t-1}$ .

Therefore,

$$\begin{aligned} \text{cost}(Q') &= \frac{a + (t-2) \cdot b + \binom{t-1}{2} \cdot b}{\frac{2a + (t-2) \cdot b}{t-1} \cdot \binom{t}{2}} \cdot \text{cost}(Q) \\ &= (1 + \epsilon) \cdot \text{cost}(Q) \end{aligned}$$

where  $\epsilon = \frac{1}{t} - \frac{2a}{2a + (t-2)b}$ .

We further note that  $\epsilon \approx \frac{1}{t}$  as  $a \rightarrow 0$ .

Therefore, Algorithm 1 is a  $1 + \frac{1}{t}$ -approximation algorithm for the ultrametric case of MWtCP  $\square$

## 6 Varying the strictness of the metricity through a given factor $\sigma$

We introduce a variable  $\sigma$  in analyzing Algorithm 1 as a means of tightening or relaxing the metric property.

**Theorem 9** *If for any distinct vertices  $a, b, c, \in V$ ,  $w(a, b) \leq \sigma(w(b, c) + w(c, a))$ , then Algorithm 1 is a  $(2\sigma - \frac{4\sigma-2}{t})$ -approximation algorithm for MWtCP*

**Proof:** The triangle inequality states that for any distinct vertices  $a, b, c, \in V$ ,  $w(a, b) \leq w(b, c) + w(c, a)$ . We observe the effect if  $w(a, b) \leq \sigma(w(b, c) + w(c, a))$ . It has already been established that

$$\begin{aligned} \text{cost}(Q) &\leq \text{cost}(Q') \\ \sum_{i=1}^e w_i &\leq \sum_{i=1}^e w'_i \\ &\leq \sum_{i=1}^{t-1} w'_i + \sum_{i=t}^e w'_i \\ &\leq \text{cost}(Q^*) + \sum_{i=t}^e w'_i \end{aligned}$$

and that what is being approximated here is  $\sum_{i=t}^e w'_i$ . Applying the proposed bounds on the third side of the triangle, the weights of each of the  $\binom{t-1}{2}$  remaining edges may be computed as follows for  $1 \leq p < q \leq t - 1$

$$\begin{aligned} \text{cost}(v'_p, v'_q) &\leq \sigma(\text{cost}(v'_t, v'_p) + \text{cost}(v'_t, v'_q)) \\ &\leq \sigma(w'_p + w'_q) \end{aligned}$$

As has already been shown, each of the vertices  $v'_1, v'_2, v'_3, \dots, v'_{t-1}$  in the  $(t - 1)$ -clique, by definition, is adjacent to  $t - 2$  vertices in the  $(t - 1)$ -clique as is illustrated in Figure 4 for  $t = 5$ .

Hence,

$$\sum_{i=t}^e w'_i \leq \sigma(t - 2) \cdot \sum_{i=1}^{t-1} w'_i$$

Using this to provide bounds for the cost of  $Q'$ :

$$\begin{aligned} \text{cost}(Q') &= \sum_{i=1}^{m-1} w'_i + \sum_{i=t}^e w'_i \leq \sum_{i=1}^{t-1} w'_i + \sigma(t - 2) \cdot \sum_{i=1}^{t-1} w'_i \\ &= \sum_{i=1}^{t-1} w'_i + \sum_{i=t}^e w'_i \leq (1 + \sigma(t - 2)) \cdot \sum_{i=1}^{t-1} w'_i \\ &= \sum_{i=1}^e w'_i \leq (1 + \sigma(t - 2)) \cdot \sum_{i=1}^{t-1} w'_i \end{aligned}$$

Therefore,

$$\frac{t}{2} \cdot \text{cost}(Q^*) \leq \text{cost}(Q) \leq \text{cost}(Q') \leq (1 + \sigma(t - 2)) \cdot \text{cost}(Q^*)$$

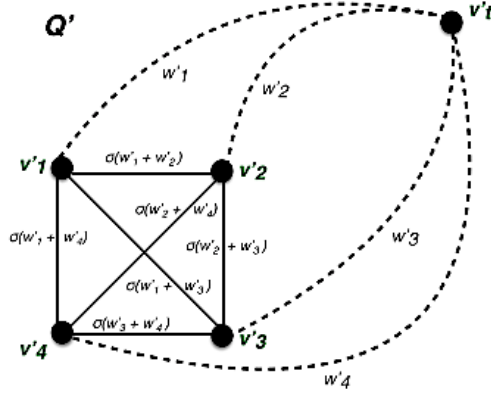


Figure 4: The weights of the edges of  $Q^*$  are used to provide variable upper bounds on the remaining edges of the solution  $Q'$  for  $t = 5$

The upper bound for  $cost(Q')$  will therefore be  $(1 + \sigma(t - 2)) \cdot cost(Q^*)$ . Noting that  $cost(Q^*) \leq \frac{2}{t} \cdot cost(Q)$ , the computation for the approximation ratio is as follows.

$$\begin{aligned} cost(Q') &\leq (1 + \sigma(m - 2)) \cdot \frac{2}{t} \cdot cost(Q) \\ &\leq 2\sigma - \frac{4\sigma - 2}{t} \cdot cost(Q) \end{aligned}$$

Therefore, applying the property for any distinct vertices  $a, b, c, \in V$ ,  $w(a, b) \leq \sigma(w(b, c) + w(c, a))$  will make Algorithm 1 a  $(2\sigma - \frac{4\sigma - 2}{t})$ -approximation algorithm for MWtCP  $\square$

## 7 On Returning a Set of Candidate Gene Clusters

It was mentioned at the beginning that this study on MWtCP is motivated by the problem of identifying approximate gene clusters. Correspondence can be made between genomes and partitions, gene contents in linear interval in a genome can correspond to vertices in a partition, while co-expression value between two gene groups can be represented by weights placed on edges joining a pair of vertices. The output of MWtCP would be a clique representing a set of highly identical gene groups across a set of genomes which will be a candidate gene cluster. In practice, however, experts are more concerned with obtaining not just one candidate gene cluster but a set of candidate gene clusters, which can then be validated by experiments.

For this, a modified version of Algorithm 1 can be made - one that does not return only one solution but returns a list of candidate solutions sorted according to a metric. The algorithm can be modified to allow the selection



of  $r$  candidate gene clusters, where  $1 \leq r \leq n$ . Recall that in the algorithm, the weight of each  $t$ -clique in  $G$  is approximated using the weight of the star. A slight modification can be made on Algorithm 1 that would allow storage of weights for each of the stars and which can then be sorted later. The  $r$  stars with least total weight will be the candidate genes.

With this, the modified algorithm provides a sorted list of approximate gene clusters. Since Algorithm 1 runs in  $O(n^2)$ , then this modified version will run in  $O(n^2) + n \log n = O(n^2)$ .

## 8 Conclusion and Future Work

Searching for cliques in undirected weighted graphs has been used in many studies to formalize the biological problem of identifying associations among genes. In this study we explored MINIMUM WEIGHT  $t$ -PARTITE CLIQUE PROBLEM (MWtCP), which is the problem of finding the  $t$ -clique of minimum weight in a complete edge-weighted  $t$ -partite graph. It was shown that MWtCP is NP-hard and APX-hard in the general case. An algorithm that runs in  $O(n^2)$  which uses the minimum  $(t-1)$ -star to approximate the minimum  $t$ -clique is presented. It was shown that the said algorithm has a relative performance guarantee of 2 for the metric case and a relative performance guarantee of  $1 + \frac{1}{t}$  for the ultrametric subclass of instances was also presented. It was further shown how relaxing or tightening the application of the metricity property of input instances affects the approximation ratio. Some insights on the application of MWtCP to approximate gene cluster discovery were also presented such as providing multiple gene clusters.

Further investigation needs to be done in MWtCP, specifically in applying it to real genomic data. Studies are ongoing and we look forward to present them soon.

## Acknowledgements

G. Solano is supported by the Engineering Research and Development for Technology (ERDT) Scholarship Program of the Department of Science and Technology (DOST) and the Doctoral Sandwich Scholarship Program of the Commission on Higher Education (CHED) Philippines.

## References

- [1] J. A. Aborot, H. Adorna, J. B. Clemente, B. K. de Jesus, and G. Solano. Search for a star: Approximate gene cluster discovery problem (agcdp) as a graph problem. *Philippine Computing Journal*, 7(2), 2012.
- [2] B. Alidaee, F. W. Glover, G. A. Kochenberger, and H. Wang. Solving the maximum edge weight clique problem via unconstrained quadratic programming. *Eur. J. Oper. Res.*, 181:592–597, 2007.
- [3] U. Alon. Biological networks: The tinkerer as an engineer. *Science*, 301(5641):1866–1867, 2003. doi:10.1126/science.1089072.
- [4] J. G. Augustson and J. Minker. An analysis of some graph theoretical cluster techniques. *J. ACM*, 17(4):571–588, Oct. 1970. doi:10.1145/321607.321608.
- [5] A.-L. Barabasi and Z. N. Oltvai. Network biology: understanding the cell’s functional organization. *Nature reviews. Genetics*, 5(2):101–uc0u128 u148 113, February 2004. doi:10.1038/nrg1272.
- [6] A. Bergeron, S. Corteel, and M. Raffinot. The algorithmic of gene teams. In R. Guigo and D. Gusfield, editors, *Algorithms in Bioinformatics*, pages 464–476. Springer Berlin Heidelberg, 2002.
- [7] E. Bierstone. Cliques and generalized cliques in a finite linear graph. unpublished, 1960.
- [8] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, 1999.
- [9] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575’96577, Sept. 1973. doi:10.1145/362342.362367.
- [10] G. Cabunducan, J. B. Clemente, H. N. Adorna, and R. T. Relator. Probing the hardness of the approximate gene cluster discovery problem (AGCDP). In *Theory And Practice Of Computation - Proceedings Of Workshop On Computation: Theory and Practice*, 2014. doi:10.1142/9789814612883\_0003.
- [11] J. L. Dela Rosa, A. E. A. Magpantay, A. C. Gonzaga, and G. A. Solano. Cluster center genes as candidate biomarkers for the classification of leukemia. In N. G. Bourbakis, G. A. Tsihrintzis, and M. Virvou, editors, *5th International Conference on Information, Intelligence, Systems and Applications, IISA 2014*, pages 124–129. IEEE, 2014. doi:10.1109/IISA.2014.6878769.

- [12] I. I. Eremin, E. K. Gimadi, A. V. Kel'manov, A. V. Pyatkin, and M. Y. Khachai. 2-approximation algorithm for finding a clique with minimum weight of vertices and edges. *Proceedings of the Steklov Institute of Mathematics*, 284(SUPPL.1):87–95, 4 2014. doi:10.1134/S0081543814020084.
- [13] P. D. V. Florendo and G. A. Solano. Integene: An integer linear programming tool for discovering approximate gene clusters. In *10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019*, pages 1–8. IEEE, 2019. doi:10.1109/IISA.2019.8900755.
- [14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.
- [15] E. K. Gimadi, A. V. Kel'manov, A. V. Pyatkin, and M. Y. Khachai. Efficient algorithms with performance guarantees for some problems of finding several cliques in a complete undirected weighted graph. *Proceedings of the Steklov Institute of Mathematics*, 289(SUPPL.1):88–101, 7 2015. doi:10.1134/S0081543815050089.
- [16] O. Goldschmidt, D. S. Hochbaum, C. A. J. Hurkens, and G. Yu. Approximation algorithms for the k-clique covering problem. *SIAM J. Discrete Math.*, 9:492–509, 1996. doi:10.1137/S089548019325232X.
- [17] R. Gupta, J. C. Walrand, and O. Goldschmidt. Maximal cliques in unit disk graphs: Polynomial approximation. In *Proceedings INOC*, 2006.
- [18] F. Harary and I. C. Ross. A procedure for clique detection using the group matrix. *Sociometry*, 20(3):205–215, 1957. URL: <http://www.jstor.org/stable/2785673>.
- [19] J. Hastad. Clique is hard to approximate within  $n^{1-\epsilon}$ . In *Acta Mathematica*, pages 627–636, 1996.
- [20] G. He, J. Liu, and C. Zhao. *Approximation Algorithms for Some Graph Partitioning Problems*, pages 21–31. World Scientific, 2004. doi:10.1142/9789812794741\_0002.
- [21] R. Hoberman and D. Durand. The incompatible desiderata of gene cluster properties. In A. McLysaght and D. H. Huson, editors, *Comparative Genomics*, pages 73–87. Springer Berlin Heidelberg, 2005.
- [22] D. Kumlander. A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search. In C. M. S. B.H.V. Topping, editor, *Proceedings of the Fourth International Conference on Engineering Computational Technology*, pages 202–208, 2004. doi:10.4203/ccp.80.60.
- [23] M. A. Langston, C. Cotta, and P. Moscato. Combinatorial and algorithmic issues for microarray analysis. In *Handbook of Approximation Algorithms and Metaheuristics*, 2007.

- [24] Z. N. Oltvai and A.-L. Barabási. Systems biology. life's complexity pyramid. *Science*, 298 5594:763–764, 2002. doi:10.1126/science.1078563.
- [25] K. Park, K. Lee, and S. Park. An extended formulation approach to the edge-weighted maximal clique problem. *European Journal of Operational Research*, 95(3):671 – 682, 1996. doi:10.1016/0377-2217(95)00299-5.
- [26] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957. doi:10.1002/j.1538-7305.1957.tb01515.x.
- [27] W. J. Pullan. Approximating the maximum vertex/edge weighted clique using local search. *Journal of Heuristics*, 14:117–134, 2008. doi:10.1007/s10732-007-9026-2.
- [28] S. Rahmann and G. W. Klau. *Integer Linear Programming Techniques for Discovering Approximate Gene Clusters*, chapter 9, pages 203–221. John Wiley & Sons, Ltd, 2007. doi:10.1002/9780470253441.ch9.
- [29] B. C. Silmaro and G. A. Solano. Clique-finding tool for detecting approximate gene clusters. In *10th International Conference on Information, Intelligence, Systems and Applications, IISA 2019*, pages 1–8. IEEE, 2019. doi:10.1109/IISA.2019.8900766.
- [30] G. Solano, G. Blin, M. Raffinot, and J. Caro. *A Clique Finding Algorithm for the Approximate Gene Cluster Discovery Problem*, pages 72–88. World Scientific, 2018. doi:10.1142/9789813279674\_0006.
- [31] G. A. Solano, B. C. Silmaro, S. A. Ojeda, and J. Caro. Discovering approximate gene clusters as a minimum weighted clique problem. *Philippine Computing Journal*, 13(2), 2017.
- [32] M. M. Sorensen. New facets and a branch-and-cut algorithm for the weighted clique problem. *European Journal of Operational Research*, 154(1):57 – 70, 2004. doi:https://doi.org/10.1016/S0377-2217(02)00852-4.
- [33] L. Trevisan. *Inapproximability of Combinatorial Optimization Problems*, chapter 13, pages 381–434. John Wiley & Sons, Ltd, 2014. doi:10.1002/9781119005353.ch13.
- [34] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977. doi:10.1137/0206036.