Approximation Hardness of the Cross-Species Conserved Active Modules Detection Problem

Thomas Hume, Hayssam Soueidan, Macha Nikolski, Guillaume Blin

Algorithmic progress on the cross-species conserved active modules detection problem

Thomas Hume*, Hayssam Soueidan*, Macha Nikolski, and Guillaume Blin

Univ. Bordeaux, CNRS / LaBRI, F-33405 Talence, France
Univ. Bordeaux, CBiB, F-33076 Bordeaux, France
Univ. Bordeaux, INSERM U1035, F-33076 Bordeaux, France
{guillaume.blin,thomas.hume,macha.nikolski}@labri.fr,
hayssam.soueidan@gmail.com

Abstract. Biological network comparison is an essential but algorithmically challenging approach for the analysis of underlying data. A typical example is looking for certain subgraphs in a given network, such as subgraphs that maximize some function of their nodes' weights. However, the corresponding MAXIMUM-WEIGHT CONNECTED SUBGRAPH (MWCS) problem is known to be hard to approximate. In this contribution, we consider the problem of the simultaneous discovery of maximum weight subgraphs in two networks, whose nodes are matched by a mapping: the MAXIMUM-WEIGHT CROSS-CONNECTED SUBGRAPHS (MWCCS) problem. We provide inapproximability results for this problem. These results indicate that the complexity of the problem is conditioned both by the nature of the mapping function and by the topologies of the two networks. In particular, we show that the problem is inapproximable even when the mapping is an injective function and the input graphs are two binary trees. We also prove that it remains hard to approximate when the mapping is a bijective function and the input graphs are a graph and a binary tree. We further analyze a variant of the MWCS problem where the networks' nodes are assigned both a weight and a contribution value, that we call MAXIMUM-WEIGHT RATIO-BOUNDED CONNECTED SUBGRAPH (MWRBCS). We provide an FPT-algorithm for trees and an efficient dynamic programming solution for cycles. These algorithms allow us to derive a polynomial solution for MWCCS applicable when (*i*) MWRBCS is polynomially solvable for one of the graphs and (*ii*) the set of subgraphs of the other graph is polynomially enumerable.

1 Introduction

Networks of interacting units are a core concept in modern biology that enables understanding of biological processes at the systems' level. In their most basic form biological networks are graphs where vertices represent biological entities such as genes or proteins and edges represent interactions between these entities. Increasingly advanced experimental methods are used to provide evidence of

* These authors have contributed equally to the paper.

existing interactions and nowadays comprehensive resources provide access to this knowledge (see for example [6] and [10]).

One of the key concepts to understand biological processes is that of *modules* within biological networks. Modules are considered to be sets of entities (genes, proteins, etc.) that function in a coordinated fashion or physically interact (for a review see [9]). The problem of finding gene modules within a biological network was first solved using simulated annealing by Ideker et al. [7].

A possible formulation for the problem of finding modules within a network is to look for connected sub-networks that maximize weights on the nodes. These weights typically represent some measure of biological activity, for example the expression level of genes. Finding the optimal (with respect to sum of weights) module in a biological network has been formally defined as the MAXIMUM (NODE-)WEIGHT CONNECTED SUBGRAPH problem (MWCS) [4].

The MWCS problem is known to be hard to approximate [1]. Despite this complexity, there exist efficient exact solutions to this problem, using either reductions to the PRIZE-COLLECTING STEINER TREE problem [4], or using branch-and-cut mixed integer programming with node separation [1].

One limitation of the existing formulation is that it only considers one network at a time. Indeed, several studies have demonstrated the added value of identifying biological processes that are conserved across different conditions or even different species [13,8] as modules identified in single condition lack robustness [12]. We previously proposed a formulation for the identification of modules that are conserved across species. In our formulation, the two species are represented by two different networks with weighted nodes and we are provided with a mapping between the nodes of these networks. This mapping represents the similarity between genes or proteins across species, for example derived from orthology.

We formalized the identification of conserved modules as the MAXIMUM-WEIGHT CROSS-CONNECTED SUBGRAPHS (MWCCS) problem [5] which consists in the computation of two modules (connected subgraphs, one in each network), such that (*i*) the cumulative sum of their node weights is maximal and (*ii*) the proportion of *conserved* nodes within the solution is greater than a fixed threshold α. We consider a node in one of the modules to be conserved if it is mapped to a node in the other module. We have proposed an efficient mixed-integer programming solution for this problem and provided a fast implementation[1].

In this paper, we investigate the algorithmic complexity of the MWCCS problems. In the case of $\alpha = 0$, the MWCCS problem is as hard as the MWCS problem since it amounts to solving two independent MWCS instances. Here, we (*i*) establish the hardness of the problem when $\alpha = 1$, corresponding to a complete conservation requirement where all nodes in a module must admit a mapped counterpart in the other module; and (*ii*) provide polynomial exact algorithms for certain sub-cases and unfixed α. This paper is organized as follows. We recall basic definitions and problem formulation in Section 2 In Section 3, we provide inapproximability results for this problem when $\alpha = 1$. These results indicate

[1] `http://software.cwi.nl/xheinz`

that the complexity of the problem is conditioned both by the nature of the mapping function and by the topologies of the two networks.

In particular, we show that the problem is inapproximable even when the mapping is an injective function and the input graphs are two binary trees. We also prove that it remains hard to approximate when the mapping is a bijective function and the input graphs are a graph and a binary tree. In Section 4, we study a variant of the MWCS problem where the networks' nodes are assigned both a weight and a contribution value, that we call MAXIMUM-WEIGHT RATIO-BOUNDED CONNECTED SUBGRAPH (MWRBCS). We provide an FPT-algorithm for trees and an efficient dynamic programming solution for cycles. These algorithms allow us to derive a polynomial solution for MWCCS applicable when (i) MWRBCS is polynomially solvable for one of the graphs and (ii) the set of subgraphs of the other graph is polynomially enumerable.

2 Preliminaries

Let us first recall the basic needed material related to graphs. A graph $G = (V, E)$ consists of a set of vertices V and a set of edges (unordered pairs of vertices) E. We say that G is node-weighted if a function $w\colon V \to \mathbb{R}$ is provided. Given a graph $G = (V, E)$, its subgraph $G' = (V', E')$ is said to be *induced* if G' has exactly the edges that appear in G over the vertex set $V' \in V$, that is $E' = \{(x, y) \in E \mid x, y \in V'\}$. We denote the graph *induced* by the node set V' in G by $G[V']$.

The MWCS and MWCCS problems are formally defined as follows.

MAXIMUM (NODE-)WEIGHT CONNECTED SUBGRAPH problem (MWCS): Given a node-weighted graph $G = (V, E)$, w its node-weighting function, find a subset $V^* \subseteq V$, such that the induced graph $G[V^*]$ is connected, and $\sum_{v \in V^*} w(v)$ is maximum. Roughly, MWCS consists in the discovery of the connected subgraph of maximal weight, in a node weighted (possibly negatively) graph.

MAXIMUM-WEIGHT CROSS-CONNECTED SUBGRAPHS (MWCCS): Given two node-weighted graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, w_1 and w_2 their respective node-weighting functions, a symmetric relation $M(V_1, V_2)$, and an interconnection ratio $\alpha \in [0, 1]$, MWCCS asks to find two subsets of nodes ${V_1}^* \subseteq V_1$ and ${V_2}^* \subseteq V_2$ such that:

1. the induced graphs $G_1[{V_1}^*]$ and $G_2[{V_2}^*]$ are connected, and
2. an α-fraction of the solution is M-related:
   $|U^*| \geq \alpha \times |{V_1}^* \cup {V_2}^*|$ where $U^* = \{u \in {V_1}^*, v \in {V_2}^* \mid M(u, v)\}$, and
3. $\sum_{u \in {V_1}^*} w_1(u) + \sum_{v \in {V_2}^*} w_2(v)$ is maximal.

3 Inapproximability of MWCCS

We prove the inapproximability of two specific cases of the MWCCS problem. First, we prove that if the mapping between G_1 and G_2 is an injective function and G_1 is a comb tree while G_2 is a binary tree, MWCCS is APX-hard and can not be approximated within factor 1.0014. Then, we prove that if the mapping is a bijective function, the problem is as hard to approximate as when considering a tree and a graph. These results shade light on the role of the mapping with respect to the difficulty of the problem.

Both proofs consist in an L-reduction from the APX-hard MAX-3SAT(B) problem [11]: Given a collection $C_q = \{c_1, \ldots c_q\}$ of q clauses where each clause consists of a set of three literals over a finite set of n boolean variables $V_n = \{x_1, \ldots x_n\}$ and every literal occurs in at most B clauses, is there a truth assignment of V_n satisfying the largest number of clauses of C_q?

Proposition 1. *The* MWCCS *problem for a comb tree and a binary tree is APX-hard and not approximable within factor* 1.0014 *even when the mapping M is an injective function.*

We first describe how we build an instance of MWCCS corresponding to an instance of MAX-3SAT(B). Given any instance (C_q, V_n) of MAX-3SAT(B), we build a comb tree $G_1 = (V_1, E_1)$ with weight function w_1, a binary tree $G_2 = (V_2, E_2)$ with weight function w_2 and a mapping M as follows.

The comb graph G_1 is defined as follows. The vertex set is $V_1 = \{r, l_i, c_j, dl_i, dc_j \mid 1 \leq i \leq n, 1 \leq j \leq q\}$. The edge set is given by the following equation.

$$\begin{aligned} E_1 =& \{(c_j, dc_j), (l_i, dl_i) \mid 1 \leq i \leq n, 1 \leq j \leq q\} \cup \\ & \{(dc_q, r), (r, dl_1)\} \cup \\ & \{(dc_j, dc_{j+1}), (dl_i, dl_{i+1}) \mid 1 \leq i < n, 1 \leq j < q\}. \end{aligned}$$

The weight function w_1 is defined as follows: for all $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_1(l_i) = B$, $w_1(c_j) = 1$ and $w_1(r) = w_1(dc_j) = w_1(dl_i) = 0$.

Roughly, in G_1 there is a node for each clause (denoted by c_j) and for each literal (denoted by l_i) that represent the leaves of the comb. The spine of the comb contains dummy nodes for each clause (denoted by dc_j) and for each literal (denoted by dl_i) separated by a central node (denoted by r).

The binary tree $G_2 = (V_2, E_2)$ with weight function w_2 is defined as follows. The vertex set is $V_2 = \{r, x_i, \overline{x_i}, c_j^k, dx_i, d\overline{x_i}, dc_j^i, dc_j^{\overline{i}} \mid 1 \leq i \leq n, 1 \leq j \leq q, 1 \leq k \leq 3\}$. The edge set E_2 is given by the following equation.

$$\begin{aligned} E_2 =& \{(r, dx_n)\} \cup \\ & \{(c_j^{k'}, dc_j^k) \mid x_k, \text{is the } k'\text{-th literal of clause } c_j\} \cup \\ & \{(c_j^{k'}, dc_j^{\overline{k}}) | \overline{x_k}, \text{is the } k'\text{-th literal of clause } c_j\} \cup \\ & \{(dx_i, d\overline{x_{i+1}}) | 1 \leq i < n\} \cup \\ & \{(dx_i, d\overline{x_i}), (dx_i, x_{n-i+1}), (d\overline{x_i}, \overline{x_{n-i+1}}), (x_i, dc_1^i), (\overline{x_i}, dc_1^{\overline{i}}) | 1 \leq i \leq n\} \cup \\ & \{(dc_j^i, dc_{j+1}^i), (dc_j^{\overline{i}}, dc_{j+1}^{\overline{i}}) | 1 \leq i \leq n, 1 \leq j < q\} \end{aligned}$$

The weight function w_2 is defined as follows: for all $1 \leq i \leq n$, $1 \leq j \leq q$ and $1 \leq k \leq 3$, $w_2(x_i) = w_2(\overline{x_i}) = -B$ and $w_2(r) = w_2(c_j^k) = w_2(dx_i) = w_2(d\overline{x_i}) = w_2(dc_j^i) = w_2(dc_j^{\overline{i}}) = 0$

Roughly, in G_2 there is a node for each literal of each clause (denoted by c_j^k) and for each value of each literal (denoted by x_i and $\overline{x_i}$). Dummy nodes for literals have been duplicated (one for each value of the literal - that is dx_i and $d\overline{x_i}$). Dummy nodes for clauses have also been duplicated (one for each value of all literals - dc_j^i and $dc_j^{\bar{i}}$). The structure is not as easy to informally describe as for G_1 but the reader may refer to an illustration provided in Figure 1.

Finally, the mapping M is an injective function from V_1 to V_2 defined as follows.

$$M(r) = r$$
$$M(l_i) = \{x_i, \overline{x_i}\}, \text{for all } 1 \leq i \leq n$$
$$M(c_j) = \{c_j^k | 1 \leq k \leq 3\}, \text{for all } 1 \leq j \leq q$$
$$M(dl_i) = \{dx_i, d\overline{x_i}\}, \text{for all } 1 \leq i \leq n$$
$$M(dc_j) = \{dc_j^i, dc_j^{\bar{i}}\}, \text{for all } 1 \leq i \leq n \text{ and } 1 \leq j \leq q$$

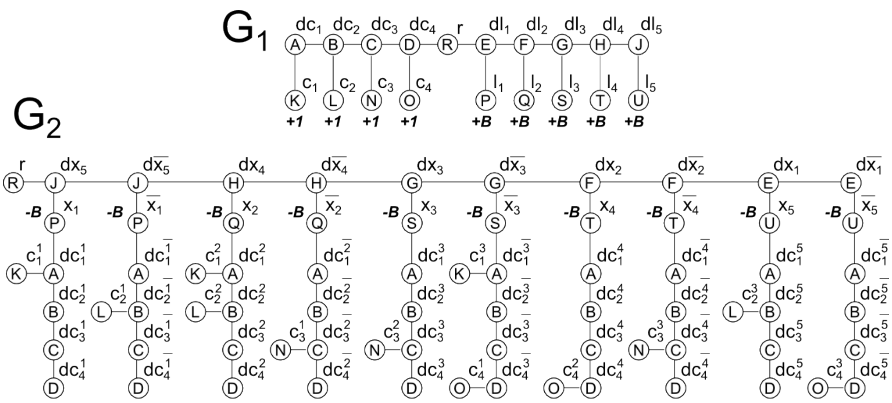

Fig. 1. Illustration of the construction of G_1, G_2, and M, given $C_q = \{(x_1 \vee x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_5), (\neg x_2 \vee x_3 \vee \neg x_4), (\neg x_3 \vee x_4 \vee \neg x_5)\}$. For readability, the mapping M is not drawn but represented as labels located on the nodes: any pair of nodes (one in G_1 and one in G_2) of similar inner label are mapped in M.

Let us prove that this construction is indeed an L-reduction from MAX-3SAT(B). More precisely, we will prove the following property.

Lemma 1. *There exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution to* MWCCS *of weight at least m.*

Proof. $\boxed{\Rightarrow}$ Given an assignment $\mathcal{A}$ of V_n satisfying m clauses of C_q, we construct a solution to MWCCS of weight m as follows.

Let $V_1^* = V_1 \setminus \{c_j \mid c_j$ is not satisfied by the assignment$\}$ and

$$\begin{aligned} V_2^* = &\{r\} \cup \\ &\{c_j^k \mid c_j \text{is satisfied by its } k\text{-th literal}\} \cup \\ &\{x_i, dc_j^i \mid x_i = 1, 1 \leq j \leq q\} \cup \\ &\{\overline{x_i}, dc_j^{\bar{i}} \mid x_i = 0, 1 \leq j \leq q\} \cup \\ &\{dx_i, d\overline{x_i} \mid 1 \leq i \leq n\}. \end{aligned}$$

By construction, $G_1[V_1^*]$ is connected since all the vertices of the spine of the comb have been kept. Moreover, $G_1[V_1^*]$ contributes $B \times n + m$ to the overall weight of the solution, that is B for each of the l_i and $+1$ for each satisfied clause. By construction, all the sub-trees rooted at x_i (resp. $\overline{x_i}$) are kept in $G_2[V_2^*]$ if $x_i = 1$ (resp. $x_i = 0$) in $\mathcal{A}$. Moreover, all the dummy nodes for literals (dx_i and $d\overline{x_i}$) and the root r have been kept. Thus, $G_2[V_2^*]$ is also connected. Furthermore, $G_2[V_2^*]$ contributes to $-B \times n$ to the overall weight of the solution since exactly one of each variable node (x_i and $\overline{x_i}$) has been kept. One can easily check that any node of V_1^* has a mapping counterpart in V_2^*. The overall solution is valid and of total weight m.

$\boxed{\Leftarrow}$ Given any solution $\{V_1^*, V_2^*\}$ to MWCCS of weight m, we construct a solution to the MAX-3SAT(B) problem satisfying at least m clauses as follows.

First, note that we can assume that any such solution to MWCCS is *canonical*, meaning that V_2^* does not contain both vertices x_i and $\overline{x_i}$ for all $1 \leq i \leq n$. Indeed, by contradiction, suppose there exists a solution such that $\{x_i, \overline{x_i}\} \subseteq V_2^*$ for a given $1 \leq i \leq n$. Then, $\{x_i, \overline{x_i}\}$ in G_2 induce a negative weight of $-2B$. This negative contribution can at most be compensated by the weight of the corresponding literal node in G_1 ($w_1(l_i) = B$) and at most B clause nodes in G_1 ($B \geq \sum w_1(c_j)$ where $x_i \in c_j$ *or* $\overline{x_i} \in c_j$) since every literal occurs in at most B clauses in C_q. Therefore, such local configuration does not provide any positive contribution to the solution and can be transformed into a better solution by removing one of the sub-trees rooted in $\{x_i, \overline{x_i}\}$. We will consider hereafter that m is the weight of the resulting canonical solution. We further assume that $m > 1$ since otherwise we can build a trivial assignment $\mathcal{A} = \{c_1^1 = 1\}$ of V_n that is satisfying at least one clause of C_q.

Let $\mathcal{A}$ be an assignment of V_n such that for all $1 \leq i \leq n$ if $x_i \in V_2^*$ then $x_i = 1$ and $x_i = 0$ otherwise. Note that, since our solution is canonical, each literal has been assigned a single boolean value in $\mathcal{A}$. Let us now prove that this assignment satisfies at least m clauses of C_q.

First, note that since our solution is canonical and we require any node of V_1^* to have a mapping counterpart in V_2^*, this implies that if $l_i \in V_1^*$ then its contribution (that is $w_1(l_i) = B$) is cancelled by the negative contribution of either x_i or $\overline{x_i}$ in V_2^* (that is $w_2(x_i) = w_2(\overline{x_i}) = -B$). Therefore, the weight m of the solution can only be realized by m clause nodes of G_1, say $\mathcal{C}_1 \subseteq V_1^*$ – since $w_1(c_j) = 1$ for all $1 \leq j \leq q$.

As already stated, to be part of the solution any node in V_1^* has a mapping counterpart in V_2^*. Thus, for each node in $\mathcal{C}_1$, there should be a node of $\mathcal{C}_2 \subseteq \{c_j^k \mid 1 \leq j \leq q, 1 \leq k \leq 3\}$ in V_2^*. More precisely, by construction, any node c_j in V_1 has exactly three mapping counterparts in V_2 (that is $\{c_j^k \mid 1 \leq k \leq 3\}$) and for each $c_j \in \mathcal{C}_1$ at least one of these mapping counterparts has to belong to $\mathcal{C}_2$.

Finally, since both $G_1[V_1^*]$ and $G_2[V_2^*]$ have to be connected, each node in $\mathcal{C}_2$, say c_j^k, should be connected by a path to a node x_i or $\overline{x_i}$, say x_i, for some $1 \leq i \leq n$, in $G_2[V_2^*]$. By construction, this is the case if x_i is the k-th literal of the clause c_j for some $1 \leq k \leq 3$. Thus, $\mathcal{A}$ is an assignment that satisfies any clause c_j such that the clause node c_j belongs to V_1^*. As already stated $|\mathcal{C}_1| = m$. □

The above reduction linearly preserves the approximation since the weights of optimal solutions of the problems correspond and there exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution to MWCCS of weight at least m. Hence, given an approximation to MWCCS, one can derive an algorithm for MAX-3SAT(B) with the same approximation ratio. Since MAX-3SAT(B), $B \geq 3$, is APX-hard [11] and MAX-3SAT(B) for $B = 6$ is not approximable within factor 1.0014 [3], so is MWCCS, which proves Proposition 1.

Let us now prove a similar result for MWCCS problem when the mapping is a bijective function.

Proposition 2. *The* MWCCS *problem for a graph and a tree is APX-hard and not approximable within factor* 1.0014 *even when the mapping is a bijective function.*

Proof. Given any instance (C_q, V_n) of MAX-3SAT(B), we build a graph $G_1 = (V_1, E_1)$ with weight function w_1, a tree $G_2 = (V_2, E_2)$ with weight function w_2 and a mapping M as follows. The graph G_1 has the vertex set $V_1 = \{r, l_i, x_i, \overline{x_i}, c_j, c_j^k \mid 1 \leq i \leq n, 1 \leq j \leq q, 1 \leq k \leq 3\}$ and the edge set defined by the following equation.

$$
\begin{aligned}
E_1 = &\{(l_i, x_i), (l_i, \overline{x_i}), (r, x_i), (r, \overline{x_i}) \mid 1 \leq i \leq n\} \cup \\
&\{(c_j, c_j^k), (r, c_j^k) \mid 1 \leq k \leq 3, 1 \leq j \leq q\}.
\end{aligned}
$$

The weight function w_1 is defined as follows: for all $1 \leq k \leq 3$, $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_1(l_i) = B$, $w_1(c_j) = 1$ and $w_1(r) = w_1(c_j^k) = w_1(x_i) = w_1(\overline{x_i}) = 0$.

Roughly, in G_1 there is a node for each clause (denoted by c_j), for each of the three literals of each clause (denoted by c_j^k), for each literal (denoted by l_i) and for each valuation of each literal (denoted by x_i, $\overline{x_i}$). Clause nodes and literal nodes are separated by a central node r.

The tree G_2 is defined as follows. The vertex set is $V_2 = V_1$, the edge set is given by the following equation:

$$
\begin{aligned}
E_2 = &\{(l_i, r), (c_j, r), (x_i, r), (\overline{x_i}, r) \mid 1 \leq i \leq n, 1 \leq j \leq q\} \cup \\
&\{(c_j^k, x_i) \mid x_i \text{ is the } k\text{-th literal of clause } c_j\} \cup \\
&\{(c_j^k, \overline{x_i}) \mid \overline{x_i} \text{ is the } k\text{-th literal of clause } c_j\}.
\end{aligned}
$$

The weight function w_2 is defined as follows: for all $1 \leq k \leq 3$, $1 \leq i \leq n$ and $1 \leq j \leq q$, $w_2(x_i) = w_2(\overline{x_i}) = -B$, $w_2(r) = w_2(c_j^k) = w_2(l_i) = w_2(c_j) = 0$.

Roughly, in G_2 all the nodes except the ones in $\{c_j^k \mid 1 \leq j \leq q, 1 \leq k \leq 3\}$ form a star centered in node r. The nodes representing the literal of the clause (that is c_j^k) are connected to their corresponding variable nodes (that is x_i or $\overline{x_i}$).

Finally, the mapping M is a bijective function from V_1 to V_2 defined as the identity (that is each node in V_1 is mapped to the node of similar label in V_2).

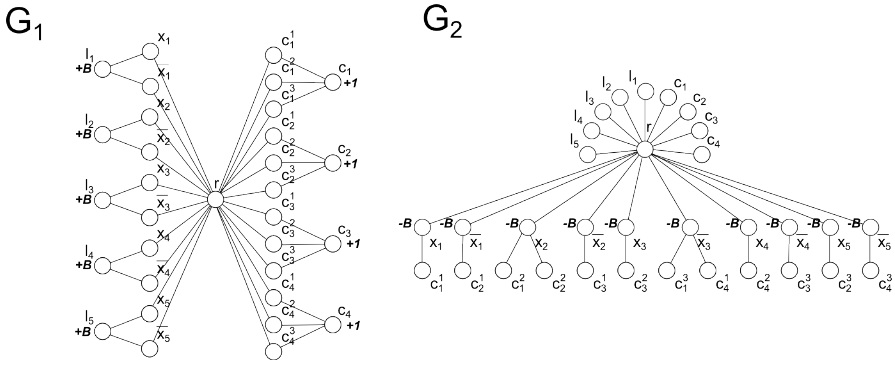

Fig. 2. Illustration of the construction of G_1, G_2, and M, given $C_q = \{(x_1 \vee x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_5), (\neg x_2 \vee x_3 \vee \neg x_4), (\neg x_3 \vee x_4 \vee \neg x_5)\}$. For readability, the mapping M is not drawn but deduced from the labels of the nodes; any pair of nodes (one in G_1 and one in G_2) of similar label are mapped in M.

Let us prove that this construction is indeed an L-reduction from MAX-3SAT(B). More precisely, we will prove the following property.

Lemma 2. *There exists an assignment of V_n satisfying at least m clauses of C_q if and only if there exists a solution (not necessarily optimal) to* MWCCS *of weight at least m.*

Proof. $\boxed{\Rightarrow}$ Given an assignment $\mathcal{A}$ of V_n satisfying m clauses of C_q, we construct a solution to MWCCS of weight m as follows.

Let $V_1^* = V_2^* = \{c_j \mid c_j \text{ is satisfied by } \mathcal{A}\} \cup \{c_j^k \mid c_j^k \text{ is satisfying } c_j \text{ by } \mathcal{A}\} \cup \{x_i \mid x_i = 1\} \cup \{\overline{x_i} \mid x_i = 0\} \cup \{r, l_i \mid 1 \leq i \leq n\}$.

By construction, $G_1[V_1^*]$ and $G_2[V_2^*]$ are connected. Moreover, $G_1[V_1^*]$ contributes $B \times n + m$ to the overall weight of the solution, that is B for each of the l_i and $+1$ for each satisfied clause, while $G_2[V_2^*]$ contributes $-B \times n$ to the overall weight of the solution since exactly one of each variable node (*i.e.*, x_i and $\overline{x_i}$) has been kept. The overall solution is valid and of total weight m.

$\boxed{\Leftarrow}$ Given any solution $V^* \subseteq V_1$ to MWCCS of weight m, we construct a solution to the MAX-3SAT(B) problem satisfying at least m clauses as follows.

First, note that, as in the previous construction, we can assume that any such solution to MWCCS is *canonical* meaning that V^* does not contain both vertices x_i and $\overline{x_i}$ for any $1 \leq i \leq n$.

Let $\mathcal{A}$ be an assignment of V_n such that for all $1 \leq i \leq n$, if $x_i \in V^*$ then $x_i = 1$ and $x_i = 0$ otherwise. Note that, since our solution is canonical, each literal has been assigned a single boolean value in $\mathcal{A}$. Let us now prove that this assignment satisfies at least m clauses of C_q.

First, note that since our solution is canonical, as in the previous construction, the weight m of the solution can only be induced by m clause nodes of G_1, say $\mathcal{C}_1 \subseteq V^*$.

Since both $G_1[V^*]$ and $G_2[V^*]$ have to be connected, any solution with $m > 1$ will include node r in V^*. Thus, for each node $c_j \in \mathcal{C}_1$ there should be a node of $\{c_j^k \mid 1 \leq k \leq 3\}$ in $G_1[V^*]$ to connect c_j to r. In $G_2[V^*]$, in order for nodes r and c_j^k to be connected, the corresponding literal node (that is x_i or $\overline{x_i}$), say x_i – has to be kept in V^*. By construction, this is the case if x_i is the k-th literal of clause c_j. Thus, $\mathcal{A}$ is an assignment that satisfies any clause c_j such that the clause node c_j belongs to V^*. As already stated $|\mathcal{C}_1| = m$. □

The above reduction linearly preserves the approximation and proves Proposition 2.

4 A general algorithm for some polynomial cases of MWCCS

In this section, we consider the general version of the problem where α is given in the input rather than being fixed, but where the mapping is restricted to a partial function (any element of V_1 has at most one image in V_2) and G_1 to a polynomially enumerable graph. We will consider each subgraph of G_1 as part of a candidate solution and will try to find the best subgraph in G_2, that is a subgraph that maximizes the total weight of the candidate solution and such that at least an α-fraction of the nodes of G_1 and G_2 in the solution are M-related. The optimal solution will be the maximum among the candidate ones.

We suppose that there is a polynomial number of subgraphs of G_1. For every subgraph $G_1' = (V_1', E_1')$ of G_1, we define the corresponding G_2 *contribution function* $c\colon V_2 \to \mathbb{N}$ to be $c(v) = |\{u \mid M(u,v), u \in V_1'\}|$. Informally, the contribution function provides for each node of V_2 the number of inverse images, given that G_1' is supposed to be the candidate solution.

Given $G_2 = (V_2, E_2)$, its weight-function w_2 and its contribution function c, the problem now corresponds to the discovery of the connected subgraph of maximum weight such that the ratio of the sum of contributions over the number of nodes in the solution is greater than or equal to α. We call this problem the MAXIMUM-WEIGHT RATIO-BOUNDED CONNECTED SUBGRAPH (MWRBCS) problem defined formally as follow: Given a node-weighted graph $G = (V, E)$, its node-weighting function $w\colon V \to \mathbb{R}$, its contribution function $c\colon V \to \mathbb{N}$, and a ratio[2] $\alpha \in [0,1]$ find a subset $V^* \subseteq V$ such that:

1. the induced graph $G[V^*]$ is connected, and
2. the ratio of the sum of contributions over the number of nodes in the solution is greater than or equal to α, that is $\sum\limits_{v \in V^*} c(v) \geq \alpha \times |V^*|$, and
3. $\sum\limits_{v \in V^*} w(v)$ is maximum.

Proposition 3. MWRBCS *is as difficult as* MWCS*.*

Proof. Indeed, when $\forall v \in V, c(v) = 1$ the MWCS and MWRBCS problems are equivalent. Thus, MWRBCS is hard to approximate for general graphs. □

Let us show now that it is polynomial for bounded degree trees.

Proposition 4. MWRBCS *is solvable in* $O(n^{2d+1})$ *time for* d*-ary trees.*

Proof. Let us consider the MWRBCS problem for a d-ary tree. We define a dynamic programming strategy with a $O(n^{2d+1})$ time complexity. This leads to a Fixed-Parameter Tractable algorithm for the problem. The basic idea is to define a 3-dimensional table T of size $|V| \times \sum_{v \in V} c(v) \times |V|$ that stores the maximum weight of a subtree rooted in v of size s and of total contribution tc.

Formally, $\forall v \in V$, $0 \leq tc \leq \sum_{v \in V} c(v)$, $0 \leq s \leq |V|$, let us note $v_{\langle i \rangle}$ the i-th child of v, $1 \leq i \leq d$, we have:

$$
\begin{aligned}
&T[v][0][0] = 0 \\
&T[v][tc][s] = \max_{tc_1,\ldots,tc_d,\ s_1,\ldots,s_d} \left(w(v) + \sum_{1 \leq i \leq d} T[v_{\langle i \rangle}][tc_i][s_i] \right) \\
&\quad s.t. \quad tc = c(v) + \sum_{1 \leq i \leq d} tc_i \\
&\qquad\qquad s = 1 + \sum_{1 \leq i \leq d} s_i
\end{aligned}
$$

The optimal subtree can be reconstructed from the table by finding the entry with the maximal weight and where the contribution ratio is not violated, and backtracking from that entry on the selected tc_i's and s_i's from the max function. Each entry of the table can be computed in $O(n^{d-1})$ (that is, an integer partition of $|V|$ into d parts) time and there are $O(n^3)$ of them, which leads to the overall complexity. □

As paths and cycles are trees of degree 1, using the preceding result leads to an $O(n^3)$ algorithm for these cases. However, one can achieve a better complexity.

Proposition 5. MWRBCS *is solvable in* $O(n^2)$ *time for paths and cycles.*

[2] We use α here for ease of notation, but formally, this parameter is derived from the original α and the partial candidate solution V_1'.

Proof. Let us first consider the MWRBCS problem for paths. Leveraging the linearity of the graph structure, we define a dynamic programming strategy with an $O(n^2)$ time complexity.

The idea is to define two 2-dimensional tables T_w and T_{tc} with n^2 entries each and that store respectively, for each pair of indices, the maximum weight and the total contribution, of the corresponding graph. Let us consider a given orientation in the path with the node at the starting end as the reference node, of index 0. Every candidate solution (a subpath) in the path can then be defined as a pair of positions, the first element being the starting position as an index number, the second element being the size of the candidate solution. The main idea being that increasing the indices one by one enables us to update the weights and total contributions incrementally.

Formally, let us denote the k-th node of the graph in the predefined orientation by n_k, we have for all $0 \leq i \leq j \leq n$:

$$T_w[i][i] = 0$$
$$T_w[i][j] = w(n_{i+j-1}) + T_w[i][j-1]$$

$$T_{tc}[i][i] = 0$$
$$T_{tc}[i][j] = c(n_{i+j-1}) + T_{tc}[i][j-1]$$

The optimal subpath is defined by the indices of the entry with the maximal weight and where the contribution ratio is not violated (*i.e.*, for any (i, j) s.t. $T_{tc}[i][j] \geq \alpha \cdot j$). Each $O(n^2)$ entry of the tables can be computed in constant time, leading to the overall complexity. For cycles, the trick consists in taking any linearization of the cycle and merging two copies of the corresponding linearization as the input path. This ensures that we will consider any candidate solution (*i.e.*, simple subpath of the cycle). The time complexity is preserved. □

5 Conclusion

In this contribution we provide the first deep complexity analysis of the MWCCS problem and show several interesting results. There still remain numerous pertinent questions to be answered. First of all, generalizing the problem to more than two graphs is of interest; even if the hardness results will hold, what practical solutions can be derived? We also would like to study the complexity effect of the relaxation of the connectivity constraints. Finally, it would be relevant to further analyse the links that can be set up between MWRBCS and variants of MWCS such as the budget constraint one.

References

1. Eduardo Álvarez-Miranda, Ivana Ljubić, and Petra Mutzel. The maximum weight connected subgraph problem. In *Facets of Combinatorial Optimization*, pages 245–270. Springer, 2013.

2. Eduardo Álvarez-Miranda, Ivana Ljubić, and Petra Mutzel. The rooted maximum node-weight connected subgraph problem. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 300–315. Springer, 2013.
3. Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 200–209. Springer Berlin Heidelberg, 1999.
4. Marcus T Dittrich, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.
5. Mohammed El-Kebir, Hayssam Soueidan, Thomas Hume, Daniela Beisser, Marcus Dittrich, Tobias Müller, Guillaume Blin, Jaap Heringa, Macha Nikolski, Lodewyk F. A. Wessels, and Gunnar W. Klau. Conserved cross-species network modules elucidate th17 t cell differentiation in human and mouse. *Under review*, 2014.
6. Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, and Lars J. Jensen. String v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):D808–D815, 2013.
7. T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–40, 2002.
8. Yong Lu, Roni Rosenfeld, Gerard J Nau, and Ziv Bar-Joseph. Cross species expression analysis of innate immune response. *J Comput Biol*, 17(3):253–68, Mar 2010.
9. Koyel Mitra, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker. Integrative approaches for finding modular structure in biological networks. *Nat Rev Genet*, 14(10):719–732, 10 2013.
10. Sandra Orchard, Samuel Kerrien, Sara Abbani, Bruno Aranda, Jignesh Bhate, Shelby Bidwell, Alan Bridge, Leonardo Briganti, Fiona S L Brinkman, Gianni Cesareni, Andrew Chatr-aryamontri, Emilie Chautard, Carol Chen, Marine Dumousseau, Johannes Goll, Robert E W Hancock, Linda I Hannick, Igor Jurisica, Jyoti Khadake, David J Lynn, Usha Mahadevan, Livia Perfetto, Arathi Raghunath, Sylvie Ricard-Blum, Bernd Roechert, Lukasz Salwinski, Volker Stumpflen, Mike Tyers, Peter Uetz, Ioannis Xenarios, and Henning Hermjakob. Protein interaction data curation: the international molecular exchange (imex) consortium. *Nat Meth*, 9(4):345–350, 04 2012.
11. Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425 – 440, 1991.
12. C. Staiger, S. Cadot, R. Kooter, M. Dittrich, T. Muller, G. W. Klau, and L. F. A. Wessels. A critical evaluation of network and pathway-based classifiers for outcome prediction in breast cancer. *PLoS ONE*, 7(4):e34796, 2012.
13. Peter Waltman, Thadeous Kacmarczyk, Ashley R Bate, Daniel B Kearns, David J Reiss, Patrick Eichenberger, and Richard Bonneau. Multi-species integrative biclustering. *Genome Biol*, 11(9):R96, 2010.