Theoretical Computer Science 410 (2009) 2024–2032

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

# On parameterized complexity of the Multi-MCS problem

Wenbin Chen [1], Matthew C. Schmidt [1], Nagiza F. Samatova [*]

*Computer Science Department, North Carolina State University, Raleigh, NC 27695, United States*
*Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, United States*

ARTICLE INFO

*Article history:*
Received 6 August 2008
Received in revised form 20 December 2008
Accepted 26 December 2008
Communicated by J. Diaz

*Keywords:*
Algorithms
Maximum common subgraph
Parameterized complexity
Linear FPT reduction

ABSTRACT

We introduce the maximum common subgraph problem for multiple graphs (**Multi-MCS**) inspired by various biological applications such as multiple alignments of gene sequences, protein structures, metabolic pathways, or protein–protein interaction networks. Multi-MCS is a generalization of the two-graph Maximum Common Subgraph problem (MCS). On the basis of the framework of parameterized complexity theory, we derive the parameterized complexity of Multi-MCS for various parameters for different classes of graphs. For example, for directed graphs with labeled vertices, we prove that the parameterized $m$-Multi-MCS problem is $W[2]$-hard, while the parameterized $k$-Multi-MCS problem is $W[t]$-hard ($\forall t \geq 1$), where $m$ and $k$ are the size of the maximum common subgraph and the number of multiple graphs, respectively. We show similar results for other parameterized versions of the Multi-MCS problem for directed graphs with vertex labels and undirected graphs with vertex and edge labels by giving linear FPT reductions of the problems from parameterized versions of the longest common subsequence problem. Likewise, for unlabeled undirected graphs, we show that a parameterized version of the Multi-MCS problem with a fixed number of input graphs is $W[1]$-complete by showing a linear FPT reduction to and from a parameterized version of the maximum clique problem.

Published by Elsevier B.V.

## 1. Introduction

Finding common motifs among multiple objects represented as graphs is a widely known problem in different fields including image processing, pattern recognition, semantic networks, and bioinformatics. Specific applications in chemistry and biology, for example, involve matching of multiple 3D chemical or protein structures, multiple alignment of protein–protein interaction networks, or interpretations of molecular spectra. While mostly restricted to only two objects, algorithms that assist with these tasks often reduce to the problem of finding the maximum common subgraph of two graphs (MCS) [8,12–14]. For achieving higher accuracy, reducing noise and gaining novel scientific insights, the community has recognized the benefit of generalizing these applications to multiple graphs of different properties such as directed and labeled graphs (e.g., matching metabolic pathways) or undirected unlabeled graphs (e.g., multiple alignment of protein structures).

To address these needs, we introduce the problem of finding the maximum common subgraph of a set of $k$ graphs that we refer to as the **Multi-MCS** problem. It is simple to see that the MCS problem is reducible to the Multi-MCS problem by fixing $k = 2$. Since the MCS problem is known to be $NP$-hard [9], then the Multi-MCS problem must be $NP$-hard as well. Therefore, it is unlikely that there exists a general algorithm that gives exact solutions to the Multi-MCS problem in practical time for large graphs.

* Corresponding author at: Computer Science Department, North Carolina State University, Raleigh, NC 27695, United States. Tel.: +1 865 566 5586.
*E-mail address:* samatovan@ornl.gov (N.F. Samatova).

1 The two authors contributed equally.

0304-3975/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.tcs.2008.12.060

Some applications of the Multi-MCS problem, however, do not require an algorithm that can solve *all* instances of the problem. Some applications are concerned only with instances of the Multi-MCS problem in which *certain parameters* of the problem are bounded, or fixed. Our study deals with the complexity of these types of parameterized versions of the Multi-MCS problem. Though there exist some reductions to and from the parameterized MCS problem for two graphs, they are not applicable to reductions to and from the Multi-MCS problem. To be able to prove any sort of parameterized complexity results we must find new reductions to and from the Multi-MCS problem.

We give linear fixed parameter tractable (FPT) reductions of the parameterized versions of the longest common subsequence (LCS) problem presented in [2] to parameterized versions of the Multi-MCS problem for directed graphs with vertex labels. We also give linear FPT reductions of two parameterized versions of the maximum clique problem to parameterized versions of the Multi-MCS problem for unlabeled undirected graphs. These linear FPT reductions give lower bounds on the parameterized complexity of these versions of the Multi-MCS problem.

The remainder of the paper is organized as follows. Section 2 reviews the concept of linear FPT reductions. Section 3 gives an overview of the Multi-MCS problem and the parameterized versions of the problem that we will be focusing on. Section 4 introduces the parameterized problems that we will use in our reductions and discusses their parameterized complexity. Section 5 shows the linear FPT reductions of the parameterized versions of the LCS problem to the parameterized versions of the Multi-MCS problem for directed graphs with vertex labels. Section 6 proves that parameterized versions of the Multi-MCS problem for unlabeled undirected graphs are $W[1]$-hard by showing linear FPT reductions of the problem from the parameterized maximum clique problem. It also shows that a special case of one of the parameterized problems is $W[1]$-complete. Finally Section 7 discusses these results as well as some open problems inspired by these findings.

## 2. Parameterized complexity

In this section we introduce some fundamentals of parameterized complexity theory.

**Definition 2.1.** A parameterized problem $L$ is a set of pairs $(x, s) \in \Sigma^* \times N^*$, such that for every $(x, s) \in L$ there is no $(x, s') \in L$ for any $s' \neq s$.

We think of $s$ as the parameter of the problem instance $x$. The parameter $s$ is a representation of the parameter values associated with a problem instance $x$. More than one parameter value can be used to form a single parameter $s$. For instance, $s$ may represent a pair of values $(s', s'')$. A problem may have many different parameterized versions of itself. However, an instance of the parameterized version of the problem $x$ has only one parameter string $s$ associated with it.

In order to show how "hard" a parameterized problem is, we show how difficult the parameterized problem is relative to another parameterized problem. To do this we need to have a method of reducing parameterized problems to one another.

**Definition 2.2.** A parameterized problem $L$ is linear FPT reducible, or $\mathrm{FPT}_l$-reducible, to a parameterized problem $L'$ if there exists a function $f(s)$ and an algorithm $\Phi(x, s)$ such that for every instance $(x, s) \in \Sigma^* \times N^*$, $\Phi$ produces a pair $(x', s')$, where $(x', s') \in L'$ if and only if $(x, s) \in L$, and $\Phi$ runs in time $O(f(s)n^{O(1)})$ [7].

In much the same manner that polynomial reductions are used to form the classical complexity classes like $P$ and $NP$, these linear FPT reductions are used to form parameterized complexity classes. The most interesting parameterized complexity class is the class of fixed parameter tractable (FPT) problems. The parameterized problems in FPT run in time $O(f(s)n^{O(1)})$ and thus can be run for large problem sizes as long as the parameter $s$ is kept small. In addition to the class FPT, parameterized complexity theory defines parameterized complexity classes according to the $W$-hierarchy. The parameterized complexity class $W[t]$ contains all of those problems that can be reduced to the Weighted $t$-Normalized Satisfiability problem [11]. For all $t \geq 1$ the problems in these classes are thought to be intractable.

The classes of Downey and Fellow's $W$-hierarchy are the most important parameterized complexity classes of intractable parameterized problems. In recent years, Flum and Grohe introduce a para-$K$ class for every classical complexity class $K$ [6]. They show that no $W$-hierarchy is of the form para-$K$. Thus, the structure theory for the class para-$K$ cannot be used to understand the classes of $W$-hierarchy. In this paper, we focus on the classes of $W$-hierarchy.

## 3. Multi-MCS problem definitions

The main problem that we will be dealing with in this paper is the Multi-MCS problem. The Multi-MCS problem can be defined formally as:

**Definition 3.1.** For a given set of $k$ graphs $H = \{G_1, G_2, \ldots, G_k\}$ and an integer $m$, is there a graph $G_{\mathrm{Max}}$ of size greater than or equal to $m$ that is isomorphic to a subgraph of every $G_i$ for $1 \leq i \leq k$?

The size of $G_{\mathrm{Max}}$ can be measured as either the number of vertices or the number of edges in $G_{\mathrm{Max}}$. If the size is measured as the number of vertices of $G_{\mathrm{Max}}$ then the problem is referred to as the Multi-MCIS (Maximum Common Induced Subgraph for Multiple Graphs) problem. If the size is measured as the number of edges of $G_{\mathrm{Max}}$ then the problem is referred to as the Multi-MCES (Maximum Common Edge Subgraph for Multiple Graphs) problem. These are analogous to the well-known Maximum Common Induced Subgraph and Maximum Common Edge Subgraph problems for two graphs (MCIS and MCES, repectively). Results from [12] show that the MCIS problem is equivalent to the MCES problem. When we refer to the Multi-MCS problem in this paper we are actually referring to the Multi-MCIS problem.

In this paper we will consider three different types of graphs to compose the set $H$. We consider the case when $H$ is composed entirely of directed graphs with vertex labels, entirely of undirected graphs with vertex labels and edge labels, and entirely of undirected graphs with no vertex labels or edge labels.

The Multi-MCS problem can be parameterized in a number of different ways. In this paper we will focus on the following four parameterizations of the Multi-MCS problem.

**Definition 3.2** (*m-Multi-MCS*). Parameter $m$, the size of the isomorphic subgraph $G_{\mathrm{Max}}$ common to all graphs in $H$.

**Definition 3.3** (*k-Multi-MCS*). Parameter $k$, the number of graphs in the set $H$.

**Definition 3.4** (*km-Multi-MCS*). Parameters $k$, the number of graphs in the set $H$, and $m$, the size of the isomorphic subgraph $G_{\mathrm{Max}}$ common to all graphs in $H$.

**Definition 3.5** (*$k|\Gamma|$-Multi-MCS*). For labeled graphs, parameters $k$, the number of graphs in the set $H$, and $|\Gamma|$, the size of the alphabet that labels the components of the graph.

In addition to these parameterized versions of the Multi-MCS, in Section 6 we will also focus on a class of $m$-Multi-MCS problems that have a fixed number of input graphs $k$. When we fix the value of $k$ for a class of $m$-Multi-MCS problems, we are only concerned with those $m$-Multi-MCS that have $k$ input graphs. The original MCS problem is the $m$-Multi-MCS problem with $k$ fixed as the value $k = 2$.

## 4. Background problem definitions

We will show in Section 5 that parameterized versions of the Longest Common Subsequence (LCS) problem are linearly FPT reducible to the parameterized versions of the Multi-MCS problem for directed graphs with vertex labels. The LCS problem can be defined formally as follows.

**Definition 4.1.** For a given set of $k$ strings $Y = \{X_1, X_2, \ldots, X_k\}$ where $X_i \in \Gamma^*$, and an integer $m$, is there a sequence $X_{\mathrm{Max}}$ of size greater than $m$ that is a subsequence of every $X_i$ for $1 \leq i \leq k$?

For each parameterized version of the Multi-MCS problem defined in Definitions 3.2–3.5 there is a corresponding parameterized version of the LCS problem defined in [2,5]. These parameterized LCS problems are defined as follows.

**Definition 4.2** (*m-LCS*). Parameter $m$, the length of the sequence $X_{\mathrm{Max}}$ common to each string in $Y$.

**Definition 4.3** (*k-LCS*). Parameter $k$, the number of strings in the set $Y$.

**Definition 4.4** (*km-LCS*). Parameters $k$, the number of strings in the set $Y$, and $m$, the length of the sequence $X_{\mathrm{Max}}$ common to each string in $Y$.

**Definition 4.5** (*$k|\Gamma|$-LCS*). Parameters $k$, the number of strings in the set $Y$, and $|\Gamma|$, the size of the alphabet used to construct the strings in $Y$.

Additional classes of parameterized LCS problems have been researched by others. For instance, the authors of [10] discuss the $k$-LCS problem, where $|\Gamma|$ is fixed as a constant, and give the parameterized complexity of this problem to be $W[1]$-hard. We do not focus on these additional parameterizations of LCS in this paper, but our reduction of the LCS problem to the Multi-MCS problem may have applications to these other parameterized versions of LCS.

In Section 6 we will show linear FPT reductions of parameterized versions of the maximum clique (CLIQUE) problem to some of the parameterized versions of the Multi-MCS problem for unlabeled undirected graphs. The CLIQUE problem can be defined formally as follows.

**Definition 4.6.** For a given graph $G$ and an integer $m$, is there a clique of size $m$ in $G$?

The size of a clique is defined by the number of vertices in it. If we view the size of the maximum clique, $m$, as a parameter, we call the parameterized problem $m$-CLIQUE. The $m$-CLIQUE problem is defined as follows.

**Definition 4.7** (*m-CLIQUE*). Parameter $m$, the size of the clique in $G$.

The reason that we are interested in the parameterized versions of the LCS and CLIQUE problems is because their parameterized complexity is known. The complexity of each parameterized version of the LCS and CLIQUE problem is shown in Table 1. We will obtain parameterized complexity results for the parameterized versions of the Multi-MCS problem when we linearly FPT reduce the parameterized versions of the LCS and CLIQUE problems to them.

**Table 1**
Parameterized complexity results of the parameterized versions of the LCS and CLIQUE problems [1–3].

| Problem | Parameter | Complexity |
|---|---|---|
| LCS | $m$ | $W[2]$-hard |
| | $k$ | $W[t]$-hard, $\forall t \geq 1$ |
| | $km$ | $W[1]$-complete |
| | $k\|\Gamma\|$ | $W[t]$-hard, $\forall t \geq 1$ |
| CLIQUE | $m$ | $W[1]$-complete |

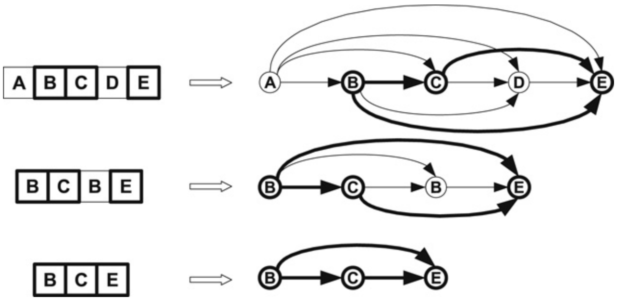


**Fig. 5.1.** An example of the reduction of an instance of the LCS problem to a Multi-MCS problem. The letters at the vertices of the graphs are the labels of the vertices. One can see how the maximum common subgraph of the three graphs maps to the longest common subsequence of the three strings.

## 5. Parameterized complexity of Multi-MCS for labeled directed and undirected graphs via parameterized reduction from LCS

In this section we will show linear FPT reductions of the parameterized versions of the LCS problem in Section 4 to parameterized versions of the Multi-MCS problem for directed graphs with vertex labels. First we will show a reduction of the LCS problem to the Multi-MCS problem for directed graphs with vertex labels. Then we will show how each parameter in the parameterized versions of the Multi-MCS problem can be found as a function of the parameters in the corresponding parameterized versions of the LCS problem. This will suffice to show that the parameterized versions of the LCS problem are linear FPT reducible to the parameterized versions of the Multi-MCS problem (Fig. 5.1).

**Lemma 5.1.** *For every instance of the LCS problem, there is an instance of the Multi-MCS problem for directed graphs with vertex labels that has a maximum common subgraph of size m if and only if the instance of the LCS has a longest common subsequence of length m.*

**Proof.** For a given instance of the LCS problem, $(Y = \{X_1, X_2, \ldots, X_k\})$, we construct an instance of the Multi-MCS problem, $(H = \{G_1, G_2, \ldots, G_k\})$, using Algorithm 5.1. For each string $X_i = x_{i1}x_{i2}\ldots x_{i|X_i|}$ in the instance of the LCS problem, the graph $G_i = (V_i, E_i)$ constructed using Algorithm 5.1 will have the following properties:

1. $|V_i| = |X_i|$.
2. For every $i$ and $j$ such that $1 \leq i \leq k$ and $1 \leq j \leq |V_i|$, the label of a vertex $v_{ij} \in V_i$ is equal to the $j$th character $x_{ij}$ in the string $X_i$.
3. For every pair of vertices $v_{ij}, v_{ij'} \in V_i$, an edge $(v_{ij}, v_{ij'}) \in E_i$ if and only if $j < j'$.

Assume that the string $Z = z_1z_2\ldots z_m$ is the longest string that is a subsequence to each string $X_i$ in the set of strings $Y$. Then on the basis of the properties of the graphs constructed using Algorithm 5.1, each of the graphs, $G_i$, in the set of graphs $H$ will have an induced subgraph isomorphic to the graph $G_{\text{Max}} = (V_{\text{Max}}, E_{\text{Max}})$, where $V_{\text{Max}} = \{v_1, v_2, \ldots, v_m\}$, $E_{\text{Max}} = \{(v_i, v_j) \mid i < j\}$, and the label of $v_i$ is $z_i$. Therefore, the size of the maximum common subgraph of the set $H$ must be at least as large as the size of the longest common subsequence of $Y$.

Now assume that each graph $G_i$ in the set of graphs $H$ has an induced subgraph isomorphic to the graph $G_{\text{Max}} = (V_{\text{Max}}, E_{\text{Max}})$, where $V_{\text{Max}} = \{v_1, v_2, \ldots, v_m\}$, $E_{\text{Max}} = \{(v_i, v_j) \mid i < j\}$, the label of $v_i$ is $z_i$, and there is no graph $G' = (V', E')$, where $|V'| > |V_{\text{Max}}|$, and $G'$ is isomorphic to an induced subgraph of every graph $G_i$. Then on the basis of the properties of the graphs constructed using Algorithm 5.1, the sequence formed by the labels $z_1z_2\ldots z_m$ must be a subsequence of each string, $X_i$ in $Y$. Therefore, the size of the maximum common subgraph of the set $H$ can only be at most the size of the longest common subsequence of $Y$. These two proofs ensure that the set of graphs $H$ has a maximum common subgraph of size $m$ if and only if the set of strings $Y$ has a longest common subsequence of length $m$. □

```
   Input: An LCS instance Y = {X_1, X_2, ..., X_k}
   Output: A Multi-MCS instance H = {G_1, G_2, ..., G_k}
1  forall strings X_i ∈ Y do
2      Add G_i = (V_i, E_i) to H;
3      forall characters x_ij ∈ X_i do
4          Add vertex v_ij to V_i;
5          Label vertex v_ij with character x_ij;
6      end
7      forall pairs of vertices (v_ij', v_ij'') ∈ V_i × V_i do
8          if j' < j'' then
9              Add directed edge (v_ij', v_ij'') to E_i;
10         end
11     end
12 end
```

**Algorithm 5.1**: Reduces an LCS instance to an equivalent Multi-MCS instance.

**Table 2**
Parameterized complexity results of the parameterized versions of the Multi-MCS problem for directed graphs with vertex labels.

| Problem | Parameter | Complexity |
|---|---|---|
| Multi-MCS | $m$ | $W[2]$-hard |
| | $k$ | $W[t]$-hard, $\forall t \geq 1$ |
| | $km$ | $W[1]$-hard |
| | $k\|\Gamma\|$ | $W[t]$-hard, $\forall t \geq 1$ |

In order to ensure that the parameterized reductions of the parameterized versions of the LCS problem to the parameterized versions of the Multi-MCS problem are linear FPT reductions, we must show that Algorithm 5.1 runs in $O(f(s)n^{O(1)})$ time. Since Algorithm 5.1 does not require any parameters as input, it suffices to show that the algorithm runs in $O(n^{O(1)})$ time.

**Lemma 5.2.** *The reduction of an instance of the LCS problem to an instance of the Multi-MCS problem takes $O(n^{O(1)})$ time.*

**Proof.** We use Algorithm 5.1 to reduce an instance of the LCS problem to an instance of the Multi-MCS problem. We define the size of the LCS instance to be $n = |Y| = \sum_{i=1}^{k} |X_i|$. The outermost loop on lines 1–12 of Algorithm 5.1 will run $O(k)$ times. The loop on lines 3–6 will run $O(|X_i|)$ times, and the loop on lines 7–11 will run $O(|X_i|^2)$ times. All of the other lines of Algorithm 5.1 will run in $O(1)$ time. Therefore, the time complexity of Algorithm 5.1 will be in $O(k * (|X_i| + |X_i|^2)) \leq O(|Y| + (|Y|^2/k)) \in O(|Y|^2) = O(n^2)$. Therefore, the reduction of an instance of the LCS problem to an instance of the Multi-MCS problem takes $O(n^{O(1)})$ time. □

Since the reduction of an instance of the LCS problem to an instance of the Multi-MCS problem can be done in $O(n^{O(1)})$ time, we can build a linear FPT reduction of parameterized versions of the LCS problem to parameterized versions of the Multi-MCS problem if we can define the parameters of the Multi-MCS problem as functions of the parameters of the LCS problem.

**Theorem 5.3.** *There exists a linear FPT reduction of the m-LCS problem to the m-Multi-MCS problem; the k-LCS problem to the k-Multi-MCS problem; the km-LCS problem to the km-Multi-MCS problem; and the $k|\Gamma|$-LCS problem to the $k|\Gamma|$-Multi-MCS problem for directed graphs with vertex labels.*

**Proof.** If $\phi$-LCS is one of the parameterized LCS problems given in Definitions 4.2–4.5, where $\phi \in \{k, m, km, k|\Gamma|\}$, then $\phi$-LCS is linearly FPT reducible to $\phi$-Multi-MCS by the following linear FPT reduction. We use the algorithm given in the proof of Lemma 5.1 to construct an equivalent instance of the Multi-MCS problem from the instance of $\phi$-LCS. Then we set the values of the parameters in $\phi$-Multi-MCS to be equal to the values of the parameters in $\phi$-LCS. □

These linear FPT reductions of the parameterized versions of the LCS problem to parameterized versions of the Multi-MCS problem ensure that the parameterized versions of the Multi-MCS problem are at least as hard as the parameterized versions of the LCS problem. Table 2 shows the hardness of each of these parameterized versions of the Multi-MCS problem for directed graphs with vertex labels.

**Theorem 5.4.** *The complexity results in Table 2 also apply to the parameterized versions of the Multi-MCS problem for undirected graphs with both vertex labels and edge labels.*

**Proof.** Since the direction of every edge in a directed graph can be viewed as a label of the edge, the directed graph with vertex labels in the Theorem 5.3 can be viewed as an undirected graph with both vertex labels and edge labels. To do so, modify line 9 of Algorithm 5.1 to add an undirected edge whose label is "$l(v_{ij'}) \rightarrow l(v_{ij''})$" where $l(v_{ij})$ is the vertex label assigned to vertex $v_{ij}$ in line 5 of Algorithm 5.1. The edge label acts as a direction on the edge between $v_{ij'}$ and $v_{ij''}$. Thus, the proof given for Theorem 5.3 holds for a reduction of the LCS problem to the Multi-MCS problem for undirected graphs with both vertex labels and edge labels. □

## 6. Parameterized complexity of Multi-MCS for unlabeled undirected graphs via parameterized reduction with CLIQUE

In this section, we show that the $m$-Multi-MCS and $km$-Multi-MCS problems for unlabeled undirected graphs are $W[1]$-hard by giving a linear FPT reduction of the $m$-CLIQUE problem to them. We also show that when the value $k$ is fixed, the $m$-Multi-MCS problem belongs to the class $W[1]$ by showing a reduction of the problem to the $m$-CLIQUE problem. An algorithm for solving a problem similar to Multi-MCS for degree bounded unlabeled undirected graphs was discussed in [4]. Here we consider the parameterized complexity of algorithms for the Multi-MCS problem on general unlabeled undirected graphs. This work is a generalization of the work in [7] for the MCS problem.

**Lemma 6.1.** *For every instance of the CLIQUE problem, there is an instance of the Multi-MCS problem that has a maximum common subgraph of size m if and only if the instance of the CLIQUE problem has a maximum clique of size m.*

**Proof.** For every instance of the CLIQUE problem, $G$, we create an instance of the Multi-MCS problem by defining the set of graphs $H$ as follows. If the number of graphs, $k$, is not fixed for the Multi-MCS problem, we set $H$ equal to the set $\{G_1, G_2\}$, where $G_1 = G$ and $G_2 = K_{|V(G)|}$. The graph $K_{|V(G)|}$ is a complete graph with $|V(G)|$ vertices. If $k$ is fixed as a constant $c > 2$ for the Multi-MCS problem, then we set $H$ equal to the set $\{G_1, G_2, \ldots, G_k\}$, where $G_1 = G$ and $G_i = K_{|V(G)|}$ for all $i$, $2 \le i \le k$.

Since $K_{|V(G)|}$ is a complete graph, any induced subgraph of $K_{|V(G)|}$ will also be a complete graph. If $H$ has a maximum common induced subgraph of size $m$, then that subgraph must be a complete graph. Thus, $G$ must have a clique of size $m$ and the size of the maximum clique in $G$ must be at least $m$.

Every clique of size less than or equal to $|V(G)|$ is a subgraph of $K_{|V(G)|}$. If $G$ has a maximum clique of size $m$, then the set $H$ must have a common subgraph of size $m$. Thus, the size of the maximum common subgraph of $H$ must be at least $m$. These two proofs ensure that the set of graphs $H$ has a maximum common subgraph of size $m$ if and only if the graph $G$ has maximum clique of size $m$. □

The time needed to construct an instance of the Multi-MCS problem according to the rules described in the proof of Lemma 6.1 is the time needed to construct the copies of $K_{|V(G)|}$. If we consider the size of the instance of the CLIQUE problem to be $n = |G| = |V(G)|$, then the time needed to construct a copy of $K_{|V(G)|}$ is in $O(n^2)$ because there will be $O(|V(G)|^2)$ edges in $K_{|V(G)|}$.

When $k$ is not fixed for the Multi-MCS problem, only one copy of $K_{|V(G)|}$ is needed. Thus, for an instance of the CLIQUE problem, the time complexity of the construction of an equivalent instance of the Multi-MCS problem when $k$ is not fixed is $O(n^2)$. When $k$ is fixed as a constant $c > 2$, $c - 1$ copies of $K_{|V(G)|}$ are needed. However, since $c$ is a constant, the time complexity of the construction of the instance of the Multi-MCS problem remains in $O(n^2)$ time when $k$ is fixed.

**Theorem 6.2.** *The m-CLIQUE problem is linear FPT reducible to both the m-Multi-MCS problem and the km-Multi-MCS problem.*

**Proof.** We construct a linear FPT reduction of the $m$-CLIQUE problem to the $m$-Multi-MCS and $km$-Multi-MCS problem as follows. We use the algorithm given in the proof of Lemma 6.1 to construct the new instance of the Multi-MCS problem. For the $m$-Multi-MCS problem we determine the value of the new parameter $m$ to be equal to the value of the parameter $m$ in the CLIQUE instance. For the $km$-Multi-MCS problem we determine the value of the new parameter $k$ to be equal to 2 and the new value of the parameter $m$ to be equal to the value of the parameter $m$ in the CLIQUE instance. □

The existence of a linear FPT reduction of the $m$-CLIQUE problem to both the $m$-Multi-MCS problem and the $km$-Multi-MCS problem for unlabeled undirected graphs means that these problems are $W[1]$-hard.

We would like to show that the parameterized versions of the Multi-MCS problem are linear FPT reducible to the parameterized CLIQUE problem. This would prove that these parameterized problems are in complexity class $W[1]$.

**Lemma 6.3.** *For every instance of the Multi-MCS problem, there is an instance of the CLIQUE problem that has a maximum clique of size m if and only if the instance of the Multi-MCS problem has a maximum common subgraph of size m.*

**Proof.** For every instance of the Multi-MCS problem, $H = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_k = (V_k, E_k)\}$, we can construct an instance of the CLIQUE problem for graph $G$ according to the Algorithm 6.1. The graph $G$ constructed using Algorithm 6.1 has the following properties:

1. $V(G) = V_1 \times V_2 \times \cdots \times V_k$.
2. An edge $(\{u_1, u_2, \ldots, u_k\}, \{v_1, v_2, \ldots, v_k\})$ is a member of $E(G)$ if and only if the following apply:
   (a) For all $i$ such that $1 \le i \le k$, $u_i \ne v_i$.
   (b) One of the following two apply:
      i. For all $i$ such that $1 \le i \le k$, $(u_i, v_i) \in E_i$.
      ii. For all $i$ such that $1 \le i \le k$, $(u_i, v_i) \notin E_i$.

**Input**: A Multi-MCS instance $H = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_k = (V_k, E_k)\}$
**Output**: A CLIQUE instance $G$

Create $G = (V, E)$;
**forall** sets of vertices $\{v_1, v_2, \ldots, v_k\} \in V_1 \times V_2 \times \cdots \times V_k$ **do**
    Add the vertex $v = \{v_1, v_2, \ldots, v_k\}$ to set $V(G)$;
**end**
**forall** pairs of vertices $(u = \{u_1, u_2, \ldots, u_k\}, v = \{v_1, v_2, \ldots, v_k\}) \in V(G) \times V(G)$ **do**
    **forall** integers $i$ such that $1 \le i \le k$ **do**
        **if** $((u_1, v_1) \in E_1$ AND $(u_i, v_i) \in E_i)$ OR $((u_1, v_1) \notin E_1$ AND $(u_i, v_i) \notin E_i)$ **then**
            **if** $u_i \ne v_i$ **then**
                *add_edge* $\leftarrow$ TRUE;
            **else**
                *add_edge* $\leftarrow$ FALSE then **break**;
            **end**
        **else**
            *add_edge* $\leftarrow$ FALSE then **break**;
        **end**
    **end**
    **if** *add_edge* $=$ TRUE **then**
        Add edge $(u, v)$ to $E(G)$;
    **end**
**end**

**Algorithm 6.1**: Reduces a Multi-MCS instance to an equivalent CLIQUE instance.

Assume that a clique $C = \{c_1, c_2, \ldots, c_m\}$ of size $m$ exists in $G$ such that there is no clique $C'$ in $G$ where $|C'| > |C|$. Because $c_j$ where $1 \le j \le m$ is a vertex in the constructed graph $G$, $c_j$ can also be defined as the set $c_j = \{v_{1j}, v_{2j}, \ldots, v_{kj}\}$. The vertex $v_{ij}$ is a vertex in the graph $G_i$ of the set of graphs $H$ in the given instance of a Multi-MCS problem. Construct the set $S_i$ such that $S_i = \{v_{i1}, v_{i2}, \ldots, v_{im}\}$.

On the basis of the properties of the graph $G$ constructed using the Algorithm 6.1 expressed above we know the following. For two integers $j'$ and $j''$ such that $1 \le j' \le m$ and $1 \le j'' \le m$, if an edge $(v_{ij'}, v_{ij''})$ exists in any graph $G_i$ for $i$ such that $1 \le i \le k$, then the edge $(v_{ij'}, v_{ij''})$ must exist in all graphs $G_i$ for $i$ such that $1 \le i \le k$. Conversely, if an edge $(v_{ij'}, v_{ij''})$ does not exist in all graphs $G_i$ for $i$ such that $1 \le i \le k$, then the edge $(v_{ij'}, v_{ij''})$ must not exist in any graph $G_i$ for $i$ such that $1 \le i \le k$. This ensures that the subgraphs induced in the graph $G_i$ by the vertices in the set $S_i$ will be isomorphic to one another. Thus, the size of the maximum common subgraph of the instance of the Multi-MCS problem must be at least as large as the size of the maximum clique in the instance of CLIQUE constructed using Algorithm 6.1.

Now assume that each graph $G_i$ in the set of graphs $H$ has an induced subgraph, $I_i$, isomorphic to the graph $G_{\text{Max}} = (V_{\text{Max}}, E_{\text{Max}})$ such that there is no graph $G' = (V', E')$ where $|V'| > |V_{\text{Max}}|$ and $G'$ is isomorphic to an induced subgraph of every graph $G_i$. Because each induced subgraph $I_i$ is isomorphic to the graph $G_{\text{Max}}$ we know that there is a one-to-one mapping $g_i$ between the vertices in $V_{\text{Max}}$ and $V(I_i)$. This ensures that if the edge $(u, v) \in E_{\text{Max}}$, then the edge $(g_i(u), g_i(v))$ is in the graph $G_i$ for all $i$ such that $1 \le i \le k$. It also ensures that if the edge $(u, v) \notin E_{\text{Max}}$, then the edge $(g_i(u), g_i(v))$ is not in the graph $G_i$ for all $i$ such that $1 \le i \le k$.

On the basis of the properties of the graph $G$ constructed using the Algorithm 6.1 we know that for every pair of vertices $u, v \in E_{\text{Max}}$, the edge $(\{g_1(u), g_2(u), \ldots, g_k(u)\}, \{g_1(u), g_2(u), \ldots, g_k(u)\})$ is in the graph $G$. Thus, the size of the maximum clique of the constructed instance of the CLIQUE problem must be at least as large as the size of the maximum common subgraph of the instance of the Multi-MCS problem. These two proofs ensure that the instance of the Multi-MCS problem has a maximum common subgraph of size $m$ if and only if the constructed instance of CLIQUE has a maximum clique of size $m$. □

Since Algorithm 6.1 does not require any parameters as input, we need the algorithm to run in $O(n^{O(1)})$ time in order for us to be able to use it as part of a linear FPT reduction. We will show that this is only possible if certain parameters of the Multi-MCS problem are fixed as constants.

**Lemma 6.4.** *The reduction of an instance of the CLIQUE problem to an instance of the Multi-MCS problem takes $O(n^k)$ time.*

**Proof.** We use Algorithm 6.1 to reduce an instance of the Multi-MCS problem to an instance of the CLIQUE problem. We define the size of the Multi-MCS instance to be $n = |H| = \sum_{i=1}^{k} |G_i|$. In order to avoid defining the size of $G_i = (V_i, E_i)$ according to the number of edges in $E_i$ we assume that $E_i$ is represented by an adjacency matrix. When we do this, $|G_i| = |V_i| + |V_i|^2 \in O(|V_i|^2)$. Also, we assume that $|V_i|$ is the same for all $i$ where $1 \le i \le k$. This can be accomplished by padding the sets $V_i$ whose sizes are smaller. Because $|V_i|$ is the same for all $i$ where $1 \le i \le k$, we set $|V_i| = |V_1|$. With these assumptions $n \in O(|V_1|^2)$.

**Table 3**
Parameterized complexity of parameterized versions of the Multi-MCS problem for unlabeled undirected graphs obtained via linear FPT reductions to and from the $m$-CLIQUE problem.

| Problem | $k$ | Hardness | Complexity class |
|---|---|---|---|
| $m$-Multi-MCS | Fixed | $W[1]$-hard | $W[1]$ |
| $m$-Multi-MCS | Not fixed | $W[1]$-hard | Unknown |
| $km$-Multi-MCS | Parameter | $W[1]$-hard | Unknown |

**Table 4**
Parameterized complexity results of the parameterized versions of the Multi-MCS problem for unlabeled undirected graphs.

| Multi-MCS parameter | Complexity | | |
|---|---|---|---|
| | Directed graphs | Undirected graphs | |
| | Vertex labeled | Vertex and edge labeled | Unlabeled |
| $m$ | $W[2]$-hard | $W[2]$-hard | $W[1]$-hard, $k$ not fixed<br>$W[1]$-complete, $k$ fixed |
| $k$ | $W[t]$-hard, $\forall t \geq 1$ | $W[t]$-hard, $\forall t \geq 1$ | Unknown |
| $km$ | $W[1]$-hard | $W[1]$-hard | $W[1]$-hard |
| $k\|\Gamma\|$ | $W[t]$-hard, $\forall t \geq 1$ | $W[t]$-hard, $\forall t \geq 1$ | Not applicable |

In Algorithm 6.1 the lines 2–4 run in $O(|V_1|^k)$ time. The `for` loop on lines 5–20 runs $(|V_1|^k)^2 = |V_1|^{2k}$ times. Inside the `for` loop on lines 5–20, the `for` loop on line 6–16 runs $k$ times. All of the other statements in Algorithm 6.1 run in $O(1)$ time. Thus, the total run time of the algorithm is $O(|V_1|^k) + (|V_1|^{2k} * k * O(1)) \in O(|V_1|^{2k})$. Since $n \in O(|V_1|^2)$, the run time of Algorithm 6.1 is $O(n^k)$. □

Since the run time of Algorithm 6.1 is $O(n^k)$ the run time cannot be $O(n^{O(1)})$ unless $k$ is a fixed constant.

**Theorem 6.5.** *The m-Multi-MCS problem with a fixed value of k is linear FPT reducible to the m-CLIQUE.*

**Proof.** We construct a linear FPT reduction of the $m$-Multi-MCS problem with a fixed value of $k$ to the $m$-CLIQUE as follows. We use the algorithm given in the proof of Lemma 6.3 to construct the new instance of the CLIQUE problem. For the $m$-CLIQUE problem we determine the value of the new parameter $m$ to be equal to the value of the parameter $m$ in the Multi-MCS instance. This reduction is linear FPT because it runs in $O(n^k)$ time. When the value of $k$ is fixed as a constant $c > 2$, $O(n^k) \in O(n^{O(1)})$. If $k$ is either a parameter or not fixed, $O(n^k) \notin O(n^{O(1)})$. Thus, the reduction of Lemma 6.1 is not FPT for either the $km$-Multi-MCS problem or the $m$-Multi-MCS problem, where $k$ is variable. □

The existence of a linear FPT reduction for the $m$-Multi-MCS problem where the number of input graphs $k$ is fixed for unlabeled undirected graphs to the $W[1]$-complete problem $m$-CLIQUE means that the $m$-Multi-MCS problem is in the complexity class $W[1]$. Since the $m$-Multi-MCS problem for undirected unlabeled graphs with a fixed $k$ is known to be $W[1]$-hard by Theorem 6.2, the problem is $W[1]$-complete. The complexity results for all of the parameterized versions of Multi-MCS discussed in this section are presented in Table 3.

## 7. Conclusions

We have shown that there exist linear FPT reductions of parameterized versions of the LCS problem to parameterized versions of the Multi-MCS problem for directed graphs with vertex labels and shown its corollary for undirected graphs with vertex and edge labels. We have also shown a linear FPT reduction of the parameterized CLIQUE problem to and from parameterized versions of the Multi-MCS problem for unlabeled undirected graphs. These reductions prove the complexity results summarized in Table 4.

There are still open problems concerning the parameterized complexity of the Multi-MCS. We would like to find linear FPT reductions of the parameterized versions of the Multi-MCS problem to other parameterized problems to prove membership in a $W[t]$ complexity class. We would like to extend the parameterized complexity results for labeled graphs to unlabeled graphs. There are also questions about whether these complexity results hold when the class of graphs is restricted to trees. The theoretical results proven in this paper will also be useful as we try to develop algorithms for solving the Multi-MCS problem and to apply these algorithms to biological problems.

## Acknowledgments

This research was supported by the "Exploratory Data Intensive Computing for Complex Biological Systems" project from US Department of Energy (Office of Advanced Scientific Computing Research, Office of Science). The work of NFS was also sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory. Oak Ridge National Laboratory is managed by UT-Battelle for the LLC US D.O.E. under contract no. DEAC05-00OR22725.

## References


[1] H.L. Bodlaender, R.G. Downey, M.R. Fellows, Harold T. Wareham, The parameterized complexity of sequence alignment and consensus, in: CPM, 1994, pp. 15–30.
[2] H.L. Bodlaender, R.G. Downey, M.R. Fellows, H.T. Wareham, The parameterized complexity of sequence alignment and consensus, Theoret. Comput. Sci. 147 (1–2) (1995) 31–54.
[3] H.L. Bodlaender, R.G. Downey, M.R. Fellows, H.T. Wareham, Parameterized complexity analysis in computational biology, Comput. Appl. Biosci. 11 (1) (1995) 49–57.
[4] D.M. Bayada, R.W. Simpson, A.P. Johnson, An algorithm for the multiple common subgraph problem, J. Chem. Inf. Comput. Sci. 32 (1992) 680–685.
[5] R.G Downey, M.R. Fellows, Parameterized Complexity, Springer, 1999.
[6] J. Flum, M. Grohe, Describing parameterized complexity classes, Inf. Comput. 187 (2) (2003) 291–319.
[7] X. Huang, J. Lai, S.F. Jennings, Maximum common subgraph: Some upper bound and lower bound results, BMC Bioinformatics 7 (Suppl. 4) (2006) 1–9.
[8] R. Horaud, T. Skordas, Stereo correspondence through feature grouping and maximal cliques, IEEE Trans. Pattern Anal. Mach. Intell. 11 (11) (1989) 1168–1180.
[9] V. Kann, On the approximability of the maximum common subgraph problem, in: Proc. 9th Annual Symposium on Theoretical Aspects of Computer Science, in: Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 377–388.
[10] K. Pietrzak, On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems, J. Comput. Syst. Sci. 67 (4) (2003) 757–771.
[11] R. Niedermeier, Invitation to Fixed-parameter Algorithms, Oxford University Press Inc., New York, 2006.
[12] J.W. Raymond, P. Willett, Maximum common subgraph isomorphism algorithms for the matching of chemical structures, J. Comput.-Aided Mol. Des. 16 (2002) 521–533.
[13] K. Shearer, H. Bunke, S. Venkatesh, Video indexing and similarity retrieval by largest common subgraph detection using decision trees, No. IDIAP-RR 00-15, Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, Valais, Switzerland, 2000.
[14] P. Willett, IMA Vol. Math. Appl. 108 (1999) 11.