Fixed Parameter Algorithms and Hardness of Approximation Results for the Structural Target Controllability Problem

Eugen Czeizler[1,4], Alexandru Popa[2,3], and Victor Popescu[1]

[1] Department of Computer Science, Åbo Akademi University, Vesilinnantie 3, 20500, Turku, Finland, `fistname.lastname@abo.fi`
[2] Department of Computer Science, University of Bucharest, Academiei 14, Bucharest, Romania, `fistname.lastname@fmi.unibuc.ro`
[3] National Institute for Research and Development in Informatics, Averescu Bd. 8-10, Bucharest, Romania
[4] National Institute for Research and Development of Biological Sciences, Independentei Bd. 296, Bucharest, Romania

Abstract. Recent research has revealed new applications of network control science within bio-medicine, pharmacology, and medical therapeutics. These new insights and new applications generated in turn a rediscovery of some old, unresolved algorithmic questions, this time with a much stronger motivation for their tackling. One of these questions regards the so-called Structural Target Control optimization problem, known in previous literature also as Structural Output Controllability problem. Given a directed network (graph) and a target subset of nodes, the task is to select a small (or the smallest) set of nodes from which the target can be independently controlled, i.e., it can be driven from any given initial configuration to any desired final one, through a finite sequence of input values. In recent work, this problem has been shown to be NP-hard, and several heuristic algorithms were introduced and analyzed, both on randomly generated networks, and on bio-medical ones. In this paper, we show that the Structural Target Controllability problem is fixed parameter tractable when parameterized by the number of target nodes. We also prove that the problem is hard to approximate at a factor better than $O(\log n)$.

Keywords: Systems biology, Protein interaction networks, Structural network control, Approximation algorithms, Fixed parameter algorithms

1 Introduction

The network control research field has been investigated for more than 50 years, with some of its algorithmic questions only recently being able to be solved. The general topic is concerned with the optimization of output intervention needed in order to drive a linear, time-invariant, dynamical system from an arbitrary initial state, to a precise final configuration, in finite time. Although many real-life dynamical systems tend not to be linear, most of these systems are known to be well approximated by such dynamics, or could behave as such in specific conditions, such as at their steady state. Although inquiries into this field have been initiated in the 60's and 70's, see e.g. the works in [13, 10, 18], only in 2011 Liu et al. [14] succeeded to demonstrate that the algorithmic complexity of the full network control optimization problem is actually of a low polynomial complexity, being reduced to computing the maximum matching in a directed graph. The result was received with a lot of interest, and sparked a renewal of the field. Since then, the network control theory and its newly discovered results have been successively applied to the study of control over power grid networks [9], of bio-medical signaling processes [11, 8, 21], and even the control of social networks [12, 14].

Driven by this new insight into the field as well as by its new applications into the current world of Big (or just Large) Data, researchers have realized that full control can sometime be still too expensive. For example, the network control theory has been recently applied in the case of cancer-related bio-medical networks [11, 8], with the aim of using known drugs in order to drive the system towards a more favorable state. Thus, researchers aimed at using the protein signaling network in order to drive cancerous cells towards apoptosis, i.e., programmed cell death. However, the full controllability of sparse homogeneous networks, such being many bio-medical networks (e.g. gene signaling networks, metabolic networks, gene regulating networks, etc.) requires a lot of effort, sometimes needing a direct outside control over up to 70% of the initial nodes of the network [11, 14]. As in these cases an outside control equivalents to the use of specific drugs, and since these protein networks contain up to 2-3 thousands nodes, a 70% direct outside control would imply an un-viable solution. The key to solving this problem came in the form of a variant of the initial control-theory problem, namely that of target-control. Instead of enforcing the control of the entire network, one would desire to optimize the outside intervention needed to control only a well-specified target, i.e., a subset of the initial network. This proved to be particularly well-fitted with the study of protein signaling networks, as recent research has emphasized the existence of disease-specific essential genes, i.e., disease-specific sets of genes/proteins which, if knocked down, would drive the corresponding cells to apoptosis [1, 22, 23]. As is the case, new formulations lead to new problems. The Structural Target Control (optimization) problem [7, 3] asks to provide an optimum amount of outside intervention in order to drive a linear dynamical system from any initial state to a desired final state of the chosen targets.

Contrary to the full network control case, the Structural Target Controllability problem was proved to be NP-hard [3]. Several heuristic approaches have been implemented and applied to the study of bio-medical networks [7, 3, 11, 8]. However, no detailed analysis of the hardness of approximation have been developed for this problem.

Assuming the widely believed conjecture, that $P \neq NP$, no polynomial time exact algorithms exist for any NP-hard problems. Thus, there are several alternative methods to tackle the difficulty of these problems, such as *approximation algorithms* and *fixed parameter algorithms*. Approximation algorithms run in polynomial time and provide a suboptimal solution. Nevertheless, unlike heuristic algorithms, approximation algorithms guarantee that on every input instance the solution they return is within a certain factor of the optimal solution. For example, a 2-approximation algorithm for a minimization problem, guarantees that on every input, the solution returned is at most twice the size of the optimal solution on that input. However, some problems, such as the one studied in this paper, might not have approximation algorithms with a constant approximation factor, unless $P = NP$. See [20] for a textbook on approximation algorithms.

In practice instances, many problems have parameters that are typically much smaller than the input size. We can exploit the existence of these parameters in order to design faster algorithms for these problems. *Parameterized complexity* [6, 4] aims to classify problems according to various parameters that are independent of the size of the input. A fixed parameter algorithm runs in time $f(k)O(n^c)$, where, n is the input size, c is a constant, and k is the size of a parameter (independent of the input size). A problem is termed fixed parameter tractable (FPT) if it has an FPT algorithm.

In this paper we show that the Structural Target Controllability problem is fixed parameter tractable when parameterized by the number of target nodes. Also, if a second parameter is allowed, known in practice to have significantly lower values, the resulted fixed parameter algorithm has a considerably improved complexity. Finally, we also formally prove that the Structural Target Controllability problem is hard to approximate at a factor better than O(log n).

2 Notation and Preliminaries

A *linear, time invariant dynamical system* (LTIS) is a system

$$\frac{dx(t)}{dt} = Ax(t) \tag{1}$$

where $x(t) = (x_1(t), ..., x_n(t))^T$ is the n-dimensional vector describing the system's state at time t, and $A \in R^{n\times n}$ is the time-invariant *state transition matrix*; the entry $a_{i,j}$ of matrix A describes the weight of the influence of node j over node i. The elements in x are called the *variables* of the system; we denote with X the set of these variables.

The external control over the system is performed through the action of m external *driver nodes*, $u(t) = (u_1(t), ..., u_m(t))^T$. Their influence over the n variables of the system is described by the time-invariant *input matrix* $B \in R^{n\times m}$; then the LTIS (1), now denoted as (A, B), becomes:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{2}$$

Let $T \subseteq X$, $T = \{t_1, ..., t_k\}$ for some $k \leq n$ be a subset of a particular interest for the variables X, a.k.a., *the target set.* We say that the LTIS (A, B)

is *T-target controllable* if for any initial state of the variables in X and any desired numerical setup of the target variables, there exists a time-dependent input vector $u(t) = (u_1(t), ..., u_m(t))^T$ that can drive the system in finite time from its initial state to a state in which the target variables are in the desired final numerical setup. We associate to the k-target set T the characteristic matrix $C_T \in \{0,1\}^{k\times n}$ where $C_T(i,j) = 1$ iff $i = j$ and $i, j \in T$ (otherwise, $C_T(i,j) = 0$), i.e., C_T is the identity matrix restricted to the subset T. It is known, see e.g. [7], that a system (A, B) is T-target controllable if and only if

$$\operatorname{rank} \mathcal{OC}(A, B, C_T) = |T| \tag{3}$$

where the matrix $\mathcal{OC}(A, B, C_T) := [C_T B \mid C_T AB \mid C_T A^2 B \mid ... \mid C_T A^{n-1} B]$ is called the *controllability matrix*.

In the particular case when the target is the entire n variable set X, the above condition translates to the well known Kalman's condition for full controllability [10], i.e., an LTIS (A, B) is (fully) controllable if and only if $\operatorname{rank}[B \mid AB \mid A^2B \mid ... \mid A^{n-1}B] = n$.

The notion of target controllability and the focus of imposing a controlling effect only on a subset of the variables of the system, has been introduced and studied only recently, see e.g., [7, 3, 11, 8]. However, this notion can be seen as a special case of output controllability, a topic which received considerate attention in the 80's and 90's, see. e.g. the works of Poljak and Murota [16, 17, 15].

Although the control methodology seem to be very dependent on the numerical setup of the dynamical system of our choice, i.e., the numerical setup of the associated transition matrix A, it turns out that this is not the case. We say that an LTIS (A, B) is *T-structurally target controllable* (with respect to a given size-k target set T) if there exists a time-dependent input vector $u(t) = (u_1(t), ..., u_m(t))^T$ and a numerical setup for the non-zero values within the matrices A and B, that can drive the state of the target nodes to any desired numerical setup in finite time. A deep result of [13, 18] shows that a system is structurally target controllable if and only if it is target controllable for all structurally equivalent matrices A and B , except a so-called thin set of matrices; we say that two matrices are *structurally equivalent* iff they differ only on their non-zero values.[1] Thus, the existence of "a good choice" for the numerical parameters in A and B is (almost) equivalent to picking up any numerical values for these parameters. According to equation (3) above, for a k-sized target T, a system (A, B) is structurally T-target controllable if and only if there exist values for the non-zero entries in A, B such that $\operatorname{rank} \mathcal{OC}(A, B, C_T) = |T| = k$.

It is known, see e.g. [16, 17], that the structural controllability problem has a counterpart formulation in terms of graphs/networks. Given an LTIS (A, B), we associate to it the graph $G_{(A,B)} = (V, E)$ where the n variables of the system $\{x_1, ..., x_n\}$ and the size-m external controller $\{u_1, ..., u_m\}$ are the nodes of the graphs, while directed edges correspond to the non-zero values in the state

[1] It is beyond the goal of this paper to define the topological notion of thin sets; we only give here the intuition that such sets consist of isolated cases that may be easily replaced with nearby favorable cases.

transition matrix and input matrix, respectively. That is, there exists a directed edge from the node corresponding to variable x_i to the node corresponding to x_j if and only if $A(x_j, x_i) \neq 0$.[2] Similarly, there exists a directed edge from u_i to x_j if and only if $B(x_j, u_i) \neq 0$. The nodes $\{u_1, ..., u_m\}$ are called *driver nodes*, while the nodes x_j such that there exists i with $B(x_j, u_i) \neq 0$ are called the *driven nodes* of the network. In the literature, the driver and the driven nodes are sometimes known as *input* and *controlled* nodes [7, 14]. To a rough understanding, the difference between driver and driven nodes is as follows. The set of driver nodes is describing the complexity of an outside controller, assuming this controller can interact/influence independently several well specified nodes of the network. Meanwhile, the set of driven nodes provides exactly the exact collection of network nodes that are used in order to ultimately control the entire set of targets. From an algebraic perspective, the number of driver nodes is given by the number of (nonzero) columns of the control matrix B, while the number of driven nodes is given by the number of nonzero rows of B. It was shown in [3] that from a practical perspective, it is more meaningful to analyze the controllability optimization problem from the point of view of minimizing the number of driven nodes. This is why in this research we focus on this particular formulation of the optimization problem. Thus, we impose that each driver node is connected to exactly one driven node, i.e., the input matrix B contains exactly one non-zero element on each column.

Given an LTIS (A, B) and its associated graph $G_{(A,B)} = (V, E)$, the n variables of the system are (all) structurally controllable from the m-sized input controller u (and control matrix B) if and only if we can select a set of n directed paths from driver nodes as starting points (we denote this set as $\mathcal{U}$) to each of the network nodes, as ending points, such that no two paths would intersect at the same distance d from their end points. The above formulation is closely related to the concepts of *linking* and *dynamic graph* as investigated in [17, 16]. In case of the target controllability problem, for a given target set $T = \{t_1, t_2, \ldots, t_k\} \subseteq X$, the above graph formulation is naturally adjusted as follows. We introduce k new *output nodes* $\mathcal{C}_\mathcal{T} = \{c_1, c_2, \ldots, c_k\}$ (also denoted as $\mathcal{C}$ when clear from the context) and edges (t_i, c_i), for all $1 \leq i \leq k$. Note that the output matrix C_T describes exactly the above wiring. Now, the objective becomes to find a path family containing k directed paths, connecting all the driver nodes (as start-points) to the output nodes (as end-points), such that no two paths would intersect at the same distance d from their end-points. In contrast to the case of full control, the graph condition is only necessary for target control, but not sufficient [16]. However, as investigated in [3], it is only in very restrictive cases where the existence of such a path family would not translate into the algebraic definition of structural control. Thus, from all practical purposes, one can equate the algorithmic process of finding such a family of k directed path to verifying that the system is structural target controllable.

We define the notion of optimization for structural target controllability in case of LTIS as follows:

[2] We implicitly interchange the usage of x_i and i for matrix indices.

Definition 1. The Structural Target Control (Optimization) problem (in short STC):

Input: The size-n variable set X, the associate transition matrix A of size $n \times n$, and a size-k target subset $T \subseteq X$, with $k \leq n$.

Output: Matrix B of size $n \times m$ such that

1. *every column of B contains exactly one non-zero value,*
2. *$Srank\,\mathcal{OC}(A, B, C) = k$,*
3. *m (i.e., the number of columns of B) is minimum among all feasible matrices.*

3 Fixed Parameter Algorithm

In this section we prove that the STC problem is fixed parameter tractable, parameterized by some of the secondary variables of our problem. First, we show that one parameter, namely the number of target nodes $|T| = k$, suffice in generating such a fixed parameter algorithm. On the other hand, from the practical instances from where this problem was generated, namely the targeted control of human protein signaling networks in cancer, we identify several other variables of this problem which are known to have significantly lower numerical values, i.e., one or even two orders of magnitude lower than the total number of input nodes. Thus, we will involve these parameters in order to generate some lower complexities for the structural target control optimization problem.

3.1 A one-Parameter STC Algorithm

Informally, our algorithm carries out the following steps. First we compute for each vertex v in the input graph, all the possible subsets of T that v can control. Since $T = k$, there can be at most 2^k such subsets for each node v. Then, we enumerate over all possible subsets of 2^T (notice that there are precisely 2^{2^k} of such subsets). For each such subset of $\mathcal{D} \subseteq 2^T$ we check if there exists a collection of $|D|$ nodes such that each node controls precisely one set in $\mathcal{D}$. If so, we solve exactly the set cover instance $(\mathcal{D}, T)$ and store the solution if it is better than the previously found solutions (i.e., controls the target nodes with less nodes than the previous solutions). Algorithm 1 describes our procedure in detail.

Theorem 2. *Given a graph $G = (V, E)$ and a target set $T \subseteq V$ with $|T| = k$, Algorithm 1 solves the STC problem in time $O(f(k)p(n))$. Thus, the STC problem is fixed parameter tractable.*

Proof. We present in more detail and analyze the running time of each step of Algorithm 1.

Step 1. For each node $v \in V$ we compute and store as follows all the sets of nodes in T that v can simultaneously control. First, we show how to decide if a node $v \in V$ covers a given subset of nodes $T' \subseteq T$ in polynomial time in $|V|$. Given a set of vertices $X \subseteq V$, let $N(X)$ be the neighborhood of X, that is $N(X) = \{v \in V : \exists a \in X \text{ s.t. } (v, a) \in E\}$. Define the following graph $G_{v,T'} = (V', E')$ where:

Algorithm 1 An FPT algorithm for the STC problem

Input: An undirected graph $G = (V, E)$ and a set of nodes $T \subseteq V$, $|T| = k$
Output: A set of nodes $S \subseteq V$ of minimum cardinality that controls T.

1. For every node $v \in V$, compute all possible sets of target nodes that v can control in the same time $C_v \subseteq 2^T$.
2. $OPT := \infty$, $S = \emptyset$
3. For every $\mathcal{D} \subseteq 2^T$ do:
   (a) Let $\mathcal{D} = \{D_1, D_2, \ldots, D_\ell\}$. If there exists nodes $v_1, v_2, \ldots, v_\ell$ such that $C_{v_1} = D_1, C_{v_2} = D_2, \ldots, C_{v_\ell} = D_\ell$, then:
   (b) Solve exactly the set cover problem on instance $(\mathcal{D}, T)$. Let $\mathcal{D}' = \{D_{u_1}, D_{u_2}, \ldots, D_{u_x}\}$ be the sets in the optimal set cover. If $x < OPT$, then $OPT := x$ and $S := \{u_1, u_2, \ldots, u_x\}$

return S

1. Let $T_0 = T$ and $T_{i+1} = N(T_i)$, $\forall 0 \leq i < n$. The vertex set V' of the graph $G_{v,T'}$ is the *multiset* consisting of all the sets T_i plus two other vertices $\{s, t\}$. We refer to a vertex $p \in V$ that is in the set T_i as p^i. Notice that a vertex p cannot appear twice in a set T_i.
2. In the edge set E' of the graph $G_{v,T'}$ we add an edge (a^{i+1}, b^i) if $(a, b) \in E$. Moreover, we add an edge between (s, v^i), if $v^i \in T_i$. Finally we add an edge (a, t), $\forall a \in T'$

The vertex v can control simultaneously the nodes in the set V if and only if there exists k-vertex disjoint paths from s to t. Observe that graph $G_{v,T'}$ was constructed such that any two vertex disjoint paths from s to t in $G_{v,T'}$ correspond to paths in G from v to a vertex in T' that do not intersect at the same distance from the vertices in T'. The k vertex disjoint paths problem between two vertices is solvable in time $O(k(n + m))$ on a graph with n vertices and m edges [2]. Thus, since $G_{v,T'}$ has at most $|V|^2$ vertices and $|V|^3$ edges, to find k disjoint paths between s and t, takes time at most $k|V|^3$

Then, to complete step 1, we repeat the procedure described above for every vertex $v \in V$ and any subset $T' \subseteq T$. Since there are 2^k subsets of T, the total running time of step 1 is $O(2^k k|V|^4)$.

Step 2(a)

Since any set C_v has at most 2^k elements and any set $\mathcal{D}$ has at most 2^{2^k} elements, step 2(a) of the algorithm is solved in time $O(n2^{2^k})$: for every node $v \in V$ we simply search each element of C_v in $\mathcal{D}$.

Step 2(b)

Notice that since the number of sets in the set cover instance is bounded by 2^{2^k} and the number of elements is k, then we can solve the set cover in $O(2^{2^{2^k}})$ time by a simple brute force algorithm that chooses all the possible subsets of $\mathcal{D}$ and verifies if such a subset covers T.

Since Steps 2(a) and 2(b) are executed 2^{2^k} times, the total running time of Step 2 is $O(2^{2^{2^k}})$.

Thus, the overall running time of Algorithm 1 is $O(k2^k|V|^4 + 2^{2^{2^k}})$

3.2 Towards Tractable Full Search Algorithms by Multiple-Parameterization

In the following, we present a fixed parameter tractable algorithm for STC whose runtime complexity is exponential in the parameters k and p, corresponding to the size of the target set T and the maximal length of the controlling path from a driver to a target node, respectively, and low polynomial in n, the total number of nodes in the network. The algorithm is a full search expansion of a Greedy approach first reported in [7] and later analyzed and improved in [3, 11, 8].

Algorithm 2 An FPT algorithm for the STC problem parametrized by k, the size of the target set and p, the maximal length of the controlling path

Input: A directed graph $G = (V, E)$, a set of nodes $T \subseteq V$, $|T| = k$, and an integer p.
Output: A set of nodes $U \subseteq V$ of minimum cardinality that controls T.

1. We create a new graph $G' = (V', E')$. For determining V' we add to V a number of k nodes (denoted $u_1, u_2, ..., u_k$) and for E' we add to E a number of k edges, such that the edge $(u_i \to t_i) \in E'$, $\forall i = \overline{1, k}$.
2. We set $S_{best} = T$, $|S_{best}| = k$ and $S = \emptyset$.
3. We apply the iterative algorithm Control (Algorithm 3) for $(G' = (V', E'), i = 1, T_0 = T, p, S)$.

return S_{best}

Theorem 3. *Given a graph $G = (V, E)$ and a target set $T \subseteq V$ with $|T| = k$ and $|V| = n$, Algorithm 2 solves the Target Controllability Problem in time $O(kn \cdot (\frac{e(n+k)}{k})^{kp})$.*[3] *By further assuming a ratio of 1/10 between the size of the target set vs. the total nodes, we obtain an approximate time complexity $O((11e)^{kp} \times n)$.*

Proof. In the following, we present in more details and analyze the running time of each step of the Algorithm 2 and of its Control sub-function, i.e., Algorithm 3
.

The final controlling set, S_{best}, can be updated only after p nested applications of the iterative Control algorithm. In each of these p nested steps, we need to generate a bipartite graph, compute/enumerate all possible maximal matchings, and form the set S, which will then be fed into the next application of the iterative function Control. While the construction of the bipartite graph can be done in $O(kn)$, enumerating all its maximal matchings requires $O(n)$ per maximal matching, see e.g. [19]. In the worst case scenario, when we are dealing with a complete graph G, all of the intermediary bipartite graphs G_i will also

---

[3] We use the following upper bound for the binomial coefficient $\binom{n+k}{k} \leq (\frac{e(n+k)}{k})^k$

Algorithm 3 The iterative function Control called in the main program

Input: A directed graph $G = (V, E)$, an integer i- the current level in the linking graph, two sets of nodes S- the current solution (incomplete if $i < p$) and T_{i-1}- the current target in the i^{th} level of the linking graph, and an integer p- the maximum expansion of the linking graph.

Output: The set T_i which is the target in the $(i+1)^{\text{th}}$ level of the linking graph and an update of S, the current solution for the driven set. If $i = p$, a possible update of the S_{best} solution.

1. We build a bipartite graph G_i with the nodes in V on the left side (denoted T_i), and the nodes in T_{i-1} on the right side. We add to G_i all of the edges in E that have the source node in T_i and the destination node in T_{i-1}.
2. We compute/enumerate all maximal matchings in the graph G_i between the nodes in T_i and the nodes in T_{i-1}.
3. For each maximal matching, do:
   (a) We remove from T_i all of the nodes left unmatched. We add all unmatched nodes from T_{i-1} to S, if they are not already there. If they are, then they are left unchanged.
   (b) (Optionally, to speed up the search, we check if $|S| \geq |S_{best}|$, and if so we backtrack)
   (c) If $i = p$, we add to S all of the nodes in T_i. If $|S| < S_{best}$, then $S_{best} \leftarrow S$.
   (d) If $i \neq p$, we repeat again the iterative algorithm for $(G' = (V', E'), i+1, T_i, p, S)$

be complete. Thus, in each case, the number of edges will be bounded by $k \cdot (n + k)$ (since we have $|V'| = n + k$ nodes on the left side, and $|T_i| \leq k$ nodes on the right side) while the number of maximal matchings will be upper bounded by $\binom{n+k}{k}$. Therefore, the overall time complexity can be upper bounded by $O(kn + \binom{n+k}{k} \cdot (_1 kn + \binom{n+k}{k} \cdot (_2 ...p \text{ times}...)_2)_1)$, i.e., $O(\binom{n+k}{k}^p \cdot kn)$. As $\binom{n+k}{k} \leq (\frac{e(n+k)}{k})^k$, we get that the running time of the algorithm can be upper bounded by $O(kn \cdot (\frac{e(n+k)}{k})^{kp})$.

4 Hardness of Approximation

In this section we show that the Structural Target Controllability (optimization) problem cannot be approximated within a factor of $(1 - \epsilon) \ln k, \forall \epsilon > 0$ where k is the number of nodes in the target set T. We prove this via an approximation preserving reduction from the Set Cover problem, which is known to be hard to approximate by Feige [5].

Definition 4 (Set Cover). *Given an universe of elements $\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$ and a family consisting of n subsets of $\mathcal{U}$, $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, find the smallest sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that the union of all the sets $\mathcal{S}'$ is $\mathcal{U}$.*

Theorem 5. *Unless $NP \subseteq DTIME(n^{\log \log n})$, the STC problem cannot be approximated within a factor of $(1 - \epsilon) \ln n, \forall \epsilon > 0$.*

Proof. Given an instance of the Set Cover problem, i.e., a set $\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$ with k elements and n sets $S_1, S_2, \ldots, S_n \subseteq \mathcal{U}$, we construct the following instance of the STC problem.

1. Add a vertex $s_i \in V$ corresponding to each set S_i in the Set Cover instance.
2. Add a vertex $t_i \in V$ corresponding to each element u_i in the set $\mathcal{U}$.
3. For each S_i add $q_i = |S_i|(|S_i| - 1)/2$ *auxiliary* vertices in V. We term these vertices $a_1^i, a_2^i, a_3^i, \ldots, a_{q_i}^i$
4. The target set T consists of all the nodes $t_i \in V$
5. For each set S_i of the set cover instance we construct $|S_i|$ paths of length $2, 3, 4 \ldots |S_i| + 1$ as follows. Let $S_i = \{u_1, u_2, \ldots u_{|S_i|}\}$. Then we construct the paths: $\{s_i, u_1\}, \{s_i, a_1^i, t_1\}, \{s_i, a_2^i, a_3^i, t_2\}, \ldots, \{s_i, a^i_{q_i-|S_i|+1}, a^i_{q_i-1}, \ldots, a^i_{q_i}, t_{|S_i|}\}$

We show now that the Set Cover instance has a solution with x sets if and only if the target set of nodes T can be controlled with x driver nodes. Thus, the existence of an approximation algorithm of $(1 - \epsilon) \ln n$, for some $\epsilon > 0$, implies the existence of an approximation algorithm with the same factor for the Set Cover problem (which leads to a contradiction).

Given a Set Cover with x sets $S_{i_1}, S_{i_2}, \ldots S_{i_x}$, then the driver nodes $s_{i_1}, s_{i_2}, \ldots s_{i_x}$ control all the target nodes since each s_{i_j} controls precisely the target nodes corresponding to the elements in S_{i_j}. This holds since each path from the node s_{i_j} to vertices in T has a different length.

Conversely, given a set of x driver nodes that control all the target nodes we reconstruct a valid Set Cover with x sets, by choosing the sets corresponding to the driver nodes. Thus, the theorem follows.

5 Conclusions and Future Work

Network Science has been proven to be highly relevant within the current developments of medicine and of personalized therapeutics. Within this field, structural network control is a powerful and efficient tool for steering the involved bio-medical systems towards desirable configurations. Thus, the algorithmic optimization problems studied in this manuscript are relevant for the computational biomedicine community, as highly optimized solutions have a significant chance of translating into efficient therapeutics. Although the Structural Target Control (Optimization) problem has been proven to be NP-hard in its general case and can not even be approximated within a constant factor, and although it is a known fact that bio-medical networks are rather large, containing thousands of nodes and (tens of thousands of) interactions, in practice, several of the involved parameters can still be considerably bounded to significantly lower values. In this research we took advantage of these insights in order to provide two optimization algorithms which remain of low polynomial complexity with regards to the size of the network, and are exponential only in those chosen parameters.

However, Structural Controllability is only one of network science methods which can be used in order to influence the dynamics of these systems. Other methods, such as the Minimum Dominating Set (MDS) approach, or the Target Reachability approach come with new challenges, but also with several advantages.

Thus, the optimization and approximation of these algorithms is of a similar practical importance, and worth of detailed investigation and analysis.

Acknowledgments

This work was supported by the Academy of Finland through grant 272451, by the Finnish Funding Agency for Innovation through grant 1758/31/2016, by the Romanian National Authority for Scientic Research and Innovation, through the POC grant P 37 257, and by the (Romanian) Ministry of Research and Innovation through the Institutional Research Programme PN 1819, project PN 1819-01-01.

References

1. Blomen, V.A., et al.: Gene essentiality and synthetic lethality in haploid human cells. Science **350**(6264), 1092–1096 (2015). https://doi.org/10.1126/science.aac7557, `http://science.sciencemag.org/content/350/6264/1092`
2. Bondy, A., Murty, U.: Graph Theory. Springer (2011), `https://books.google.ro/books?id=HuDFMwZOwcsC`
3. Czeizler, E., et al.: Structural target controllability of linear networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics **PP**(99), 1–1 (2018). https://doi.org/10.1109/TCBB.2018.2797271
4. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (2012)
5. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (Jul 1998). https://doi.org/10.1145/285055.285059, `http://doi.acm.org/10.1145/285055.285059`
6. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, Springer (2006). https://doi.org/10.1007/3-540-29953-X, `https://doi.org/10.1007/3-540-29953-X`
7. Gao, J., Liu, Y.Y., D'Souza, R.M., Barabási, A.L.: Target control of complex networks. Nature Communications **5**, 5415 (nov 2014). https://doi.org/10.1038/ncomms6415
8. Guo, W.F., et al.: A novel algorithm for finding optimal driver nodes to target control complex networks and its applications for drug targets identification. BMC Genomics **19**(1), 924 (Jan 2018). https://doi.org/10.1186/s12864-017-4332-z, `https://doi.org/10.1186/s12864-017-4332-z`
9. Hines, P., Blumsack, S., Sanchez, E.C., Barrows, C.: The topological and electrical structure of power grids. In: 2010 43rd Hawaii International Conference on System Sciences. pp. 1–10 (Jan 2010). https://doi.org/10.1109/HICSS.2010.398
10. Kalman, R.E.: Mathematical description of linear dynamical systems. Journal of the Society for Industrial and Applied Mathematics Series A Control **1**(2), 152–192 (jan 1963). https://doi.org/10.1137/0301010
11. Kanhaiya, K., Czeizler, E., Gratie, C., Petre, I.: Controlling directed protein interaction networks in cancer. Scientific Reports **7**(1), 10327 (Sep 2017), `https://doi.org/10.1038/s41598-017-10491-y`
12. Li, A., et al.: The fundamental advantages of temporal networks. Science **358**(6366), 1042–1046 (2017). https://doi.org/10.1126/science.aai7488, `http://science.sciencemag.org/content/358/6366/1042`

13. Lin, C.: Structural controllability. IEEE Trans. Automat. Contr **19**, 201–208 (1974)
14. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. Nature **473**(7346), 167–173 (may 2011). https://doi.org/10.1038/nature10011
15. Murota, K.: Systems analysis by graphs and matroids: structural solvability and controllability. Algorithms and combinatorics, Springer-Verlag (1987), https://books.google.fi/books?id=DkHvAAAAMAAJ
16. Murota, K., Poljak, S.: Note on a Graph-theoretic Criterion for Structural Output Controllability. KAM series, discrete mathematics and combinatorics, operations research, mathematical linguistics, Department of Applied Mathematics, Charles University (1989), https://books.google.fi/books?id=5RrPHAAACAAJ
17. Poljak, S.: On the generic dimension of controllable subspaces. IEEE Transactions on Automatic Control **35**(3), 367–369 (Mar 1990). https://doi.org/10.1109/9.50361
18. Shields, R., Pearson, J.: Structural controllability of multiinput linear systems. IEEE Transactions on Automatic Control **21**(2), 203–212 (apr 1976). https://doi.org/10.1109/TAC.1976.1101198
19. Uno, T.: Algorithms for enumerating all perfect, maximum and maximal matchings in bipartite graphs. In: Leong, H.W., Imai, H., Jain, S. (eds.) Algorithms and Computation. pp. 92–101. Springer Berlin Heidelberg, Berlin, Heidelberg (1997)
20. Vazirani, V.V.: Approximation Algorithms. Springer-Verlag New York, Inc. (2001)
21. Vinayagam, A., et al.: Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets. Proceedings of the National Academy of Sciences **113**(18), 4976–4981 (apr 2016). https://doi.org/10.1073/pnas.1603992113
22. Wang, T., et al.: Identification and characterization of essential genes in the human genome. Science **350**(6264), 1096–1101 (2015). https://doi.org/10.1126/science.aac7041, http://science.sciencemag.org/content/350/6264/1096
23. Zhan, T., Boutros, M.: Towards a compendium of essential genes from model organisms to synthetic lethality in cancer cells. Critical Reviews in Biochemistry and Molecular Biology **51**(2), 74–85 (2016). https://doi.org/10.3109/10409238.2015.1117053, https://doi.org/10.3109/10409238.2015.1117053, pMID: 26627871