TUCS

Eugen Czeizler | Cristian Gratie | Wu Kai Chiu | Krishna Kanhaiya | Ion Petre

Target Controllability of Linear Networks

Turku Centre *for* Computer Science

TUCS Technical Report
No 1157, June 2016

Target Controllability of Linear Networks

Eugen Czeizler
Turku Centre for Computer Science and
Department of Computer Science, Åbo Akademi University, Turku, Finland
eugen.czeizler@abo.fi

Cristian Gratie
Turku Centre for Computer Science and
Department of Computer Science, Åbo Akademi University, Turku, Finland
cristian.gratie@abo.fi

Wu Kai Chiu
Turku Centre for Computer Science and
Department of Computer Science, Åbo Akademi University, Turku, Finland
kai.wu@abo.fi

Krishna Kanhaiya
Turku Centre for Computer Science and
Department of Computer Science, Åbo Akademi University, Turku, Finland
krishna.kanhaiya@abo.fi

Ion Petre
Turku Centre for Computer Science and
Department of Computer Science, Åbo Akademi University, Turku, Finland
ion.petre@abo.fi

TUCS Technical Report
No 1157, June 2016

Abstract

Computational analysis of the structure of intra-cellular molecular interaction networks can suggest novel therapeutic approaches for systemic diseases like cancer. Recent research in the area of network science has shown that network control theory can be a powerful tool in the understanding and manipulation of such biomedical networks. In 2011, Liu et al. developed a polynomial time optimization algorithm for computing the size of the minimal set of nodes controlling a given linear network. In 2014, Gao et al. generalized the problem for target structural control, where the objective is to optimize the size of the minimal set of nodes controlling a given target within a linear network. The working hypothesis in this case is that partial control might be "cheaper" (in the size of the controlling set) than the full control of a network. The authors developed a Greedy algorithm searching for the minimal solution of the structural target control problem, however, no suggestions were given over the actual complexity of the optimization problem. In here we prove that the structural target controllability problem is NP-hard when looking to minimize the number of driven nodes within the network, i.e., the first set of nodes which need to be directly controlled in order to structurally control the target. We also show that the Greedy algorithm provided by Gao et al. in 2014 might in some special cases fail to provide a valid solution, and a subsequent validation step is required. Also, we improve their search algorithm using several heuristics, obtaining in the end up to a 10-fold decrease in running time and also a significant decrease of the size of the minimal solution found by the algorithms.

Keywords: Network controllability, structural target control, computational complexity, greedy algorithms, driver/driven nodes, output controllability

TUCS Laboratory
Computational Biomdeling Laboratory

1 Introduction

The intrinsic robustness of living systems against perturbations is a key factor that explains why many single-target drugs have been found to provide poor efficacy or lead to significant side effects [5]. The efficacy of multi-target therapies can be understood from a robustness of disease-networks point of view to deal with single node perturbations, due to inherent diversity and redundancy of compensatory signaling pathways that result in highly resilient and resistant network architecture with modular and interconnected topology [5]. Rather than trying to design selective ligands that target individual receptors only, network polypharmacology aims to modify multiple cellular targets to tackle the compensatory mechanisms and robustness of disease-associated cellular systems, as well as to control unwanted off-target side effects that often limit the clinical utility of many conventional drug treatments [1, 3, 5]. However, the exponentially increasing number of potential drug target combinations makes the pure experimental approach quickly unfeasible, and translates into a need for design principles to determine the most promising target combinations to effectively control complex disease systems, without causing drastic toxicity or other side-effects.

Network biology, with the help of mathematical modeling, has revolutionized the human diseasome research and paved the way towards the development of new therapeutic approaches and personalized medicine. Recent work on network controllability has shown that full controllability and reprogramming of intercellular networks can be achieved by a minimum number of control targets [10]. However, the computer-based experimental tests of Liu et al. [10] suggest that the approach is totally unfeasible in practice, as achieving full control over gene regulatory networks requires roughly 80% of the nodes (i.e., on the order of 800 - 1000 nodes) to be directly controlled by an external controller.

Although diseased cells may harbor hundreds of genomic alterations in various biological pathways [8, 17], only a subset of these alterations are driving the disease initiation and progression. These genes form together the sets of (disease specific) essential genes, see [2]. Due to the new CRISPR gene editing technology, researchers can now pinpoint the sets of essential genes, for a very large class of illnesses [11, 16], including many types of cancers [18].

In this research we concentrate over the target structural controllability problem, where the aim is to select a minimal set of driver/driven nodes which can control a given target within a linear network. That is, for every initial configuration of the system and any desired final configuration of the target nodes, there exists a finite sequence of input functions for the driver nodes such that the target nodes can be driven to the desired final configuration, in finite time.

The target controllability problem for linear networks is a particular case of output controllability [14] and a generalization of the full controllability problem, which requires the control over the entire system. In 2011 Liu et al. [10] have provided a polynomial time algorithm (in the size of the network) computing the optimal solution for the full structural controllability problem. Few years latter, Gao et al. [4] developed a Greedy algorithm searching for the minimal solution of the structural target controllability problem. However, in this last research, no suggestions were given over the overall complexity of the target control optimization problem.

In this study we prove that the structural target controllability problem is NP-hard when looking to minimize the number of driven nodes within the network. The driven nodes of a network are those to be directly controlled from an outside agent in order to structurally control the given target. We also show that the Greedy algorithm provided by Gao et al. [4] might sometimes fail to provide a valid solution (i.e., a driver/driven set of nodes actually controlling the target), and thus a subsequent validation step is required. Also, we improve their search algorithm using several heuristics, obtaining up to a 10-fold decrease in the average running time and a significant decrease in the size of the average minimal solution found by the algorithms, especially in the case of proportionally small targets, i.e., less than 15% of the total number of nodes.

2 Background and Definitions

A *linear, time invariant* (in short lti) *dynamical system* is a system of the form

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

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, describing how each of these states are influencing the dynamics of the system. The elements in x are called the variables/nodes/species of the system; we abuse notation and denote with X the set of these variables. If the system is influenced by a size-m external input controller $u(t) = (u_1(t), ..., u_m(t))^T$, then system (1) becomes:

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

where $B \in R^{n\times m}$ is the time-invariant input matrix describing how each of the n variables are affected by the m inputs. In the additional case when at each time step t the system is also exporting a set of k output values, $y(t) = (y_1(t), ..., y_k(t))^T$ depending on the current state $x(t)$, the system (1) becomes:

$$\begin{aligned} \frac{dx(t)}{dt} = & \ Ax(t) \\ y(t) = & \ Cx(t) \end{aligned} \tag{3}$$

where $C \in R^{k\times n}$ is the output matrix describing how each of the k outputs are influenced by the n variables of the system at time t. For example, in the particular case when the desired output is represented just by the numerical values of a k subset $T \subseteq X$ of the total n variables, such as a target set, the output matrix C_T is a 0–1 matrix, with $C_T(i,j) = 1$ iff $i = j$ and $i,j \in T$, i.e., C_T is the the identity matrix restricted to the subset T. For ease of notation, such m-input, k-output linear, time invariant, dynamical systems are denoted as (A, B, C) with $A \in R^{n\times n}$, $B \in R^{n\times m}$, and $C \in R^{k\times n}$

Given a target set T, a linear time-invariant dynamical system (A, B, C_T) is said to be *target controllable* if there exists a time-dependent input vector $u(t) = (u_1(t), ..., u_m(t))^T$ that can drive the state of the target variables to any desired numerical setup in finite time. It is known, see e.g. [4], that a system (A, B, C_T) is target controllable if and only if

$$\text{rank}[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{n-1} B] = |T| \tag{4}$$

In the particular case when the target is the entire n variable set X, we can see that the above condition is reduced to the well known Kalman condition for full controllability [6], i.e., a linear dynamical system (A, B) is controllable if and only if

$$\text{rank}[B, AB, A^2 B, ..., A^{n-1} B] = n \tag{5}$$

A big step forward in the search for algorithms looking for efficient solutions for the (target) controllability problem has been achieved by translating the problem to graphs. A first step in this direction is implemented by detaching from particular numerical setups of a linear system, and focussing on the intrinsic wiring of the system's variables. We say that a linear time-invariant dynamical system (A, B, C_T) is *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. According to equation (4) above, a system (A, B, C_T) is structurally target controllable if and only if there exist values for the non-zero entries in A and B such that

$$\text{rank}[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{n-1} B] = k \tag{6}$$

The case of full structural controllability is obtained from the above when $T = X$ and $C_T = I_n$. It is known, see e.g. [9, 15], that if a system is structurally (target) controllable, then it is (target) controllable in almost all numerical setups of the non-zero values within the state transition matrix A.

Linear systems can be represented in terms of directed weighted graphs. The n variables of the systems are the nodes of the graphs, while directed edges correspond to non-zero values in the state transition matrix. That is, there exists a directed edge between variables x_i an x_j with weight v if and only if $A(x_j, x_i) = v \neq 0$. Similarly, the size-m controller vector u corresponds to m input nodes, $u_1, ...u_m$, called driver nodes, while the input matrix B determines the edges between the driver nodes and the network. That is, there exists a directed edge between u_i an x_j with weight w if and only if $B(x_j, u_i) = w \neq 0$. 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; these are the first nodes in the network which are directly manipulated in order to drive the entire system to the desired state.

Instead of (target) structural controllability we can now talk of the equivalent network controllability problem, where the variables and the targets are now nodes in the directed network graph. It is known, see e.g. [9], that the structural controllability problem has a counterpart formulation in terms of network graphs. The n variable system (A) is structurally controllable from the m-input/driver controller u and control matrix B if and only if we can select a set of n directed paths from the input/driver nodes (i.e., as starting points) to each of the network nodes (i.e., as ending points), such that no two paths would intersect at the same distance d from their end points. In case of the target controllability problem for a given target set T, with $|T| = m$, the above condition must hold for a path family containing m paths, connecting all the targets to the driver nodes.

From the point of view of bio-medical disease network analysis and control, it is sometimes more advantageous to consider the set of driven nodes instead of that of the driver nodes. To a rough understanding, the set of driver nodes is describing the complexity of an outside controller, assuming this controller can interact/influence with equal impact several of the network nodes; such an interaction could be seen for example as the influence of a drug over the expression of some particular genes. Meanwhile, the set of driven nodes provide the exact collection of network nodes, i.e., genes, that will be used in order to ultimately control the entire set of target nodes. In particular, if we require that each driver node is interacting with at most one network node, i.e., the control matrix B has at most one non-zero entry for every column, then there is a one-to-one correspondence between driver and driven nodes. From now on, within this study we will concentrate over minimizing the set of driven nodes for the control of a given target within a directed network.

Definition We say that for a time invariant dynamical system (A, B, C_T) the input controller is *1-bounded* if and only if matrix B contains only one non-zero value on every column, i.e., each of the m inputs $u_j(t)$ control exactly one of the variables $x_i(t)$, and that variable is independent of the choice for the time point t.

3 Driven Target Control is NP-hard

In this section we are going to prove that the problem of minimizing the number of driven nodes for a given time invariant linear dynamical system (A, B, C_T) and a target set T is NP-hard. If moreover the system (A, B, C_T) is 1-bounded, the problem is equivalent to minimizing its number of driver nodes. We are providing this result by proving that the corresponding decision problem, i.e., whether there exists a size-k 1-bounded controller B which can structurally control the target T, is itself NP-hard. This will be done via a reduction to 3SAT.

We recall that in a directed graph, we say that a node X_i is an *ancestor* of a node X_j if there exists a directed path (possibly empty) from X_i to X_j.

Theorem 1 *The 1-Bounded Target Control Optimization Problem is NP-hard, as the following associated decision problem is itself NP-hard. Given a network graph $G = (V, E)$, a target subset $T \subseteq V$, and a number $n \leq |V|$, is there a size-n 1-bounded control scheme for the target T, i.e., a set of n driver nodes, each interacting with exactly one node from the graph, such that we obtain the full control of the target nodes T? In matrix representation, is there a matrix B of size $|V| \times n$, with exactly one non-zero entry per each column, such that rank$[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{n-1} B] = |T|$?*

Proof: We are proving the NP-hardness result via a reduction from the 3SAT problem. Let P be an arbitrary 3SAT boolean formula instance, containing n boolean variables $x_1, ..., x_n$ and m clauses $Cl_1, ..., Cl_m$. We are going to construct a graph $G = (V, E)$ with $|V| = 3m(m+1)/2 + 3n + m$ nodes and select a target subset C with $|C| = m + n$ such that the formula P is satisfiable if and only if the cardinality of a minimal control set for C is n.

The graph G, presented also in Figure 1, can be described as follows. It consists of five types of nodes: *valuation-nodes, clause-nodes, tautology-nodes, and path-nodes*. The valuation set of nodes contains $2n$ nodes, X_j^T, X_j^F, $1 \leq j \leq n$, one for each possible truth assignment of a variable x_j. The clause set of nodes contains m nodes, $CL_1, ..., CL_m$, one for each of the formula's clauses. The tautology set of nodes contains n nodes, $TA_1, ..., TA_n$, each corresponding to a variable-specific tautological clause $(x_j \vee \neg x_j)$. Finally, the path-level set of nodes contains $3m(m+1)/2$ nodes, that is, for each of the clauses Cl_i, with $1 \leq i \leq m$, we have $3i$ nodes $Pa_j^{(1;i)}, Pa_j^{(2;i)}$, and $Pa_j^{(3;i)}$, with $1 \leq j \leq i$.

The directed edges of G can be easily described as follows. In the formula P, every clause Cl_i has exactly three valuations of the variables x_1, ...,x_n which may validate Cl_i; let these be $x_{i_1}^{v_1}$, $x_{i_2}^{v_2}$, and $x_{i_3}^{v_3}$. Using the 3 disjoint sets of vertices $Pa_j^{(1;i)}, Pa_j^{(2;i)}$, and $Pa_j^{(3;i)}$, with $1 \leq j \leq i$, we connect the nodes $X_{i_1}^{v_1}$, $X_{i_2}^{v_2}$, and $X_{i_3}^{v_3}$ to CL_i, using 3 disjoint directed paths (each) of length i. The direction of these edges are from the variable-type nodes towards the clause-type nodes.

In addition to the above edges, for any of the tautology-level nodes TA_j, $1 \leq j \leq n$, we have two directed edges (X_j^T, TA_j) and (X_j^F, TA_j), representing the two valuations which would validate the corresponding clause.

We fix the set $T = \{TA_j \mid 1 \leq j \leq n\} \cup \{CL_i \mid 1 \leq i \leq m\}$ containing n tautology nodes and m clause nodes as our target set.

We prove that the formula P is satisfiable if and only if the target T can be controlled from exactly n control nodes. Moreover, at most one of the valuation nodes associated to a boolean variable can be connected to a driver node.

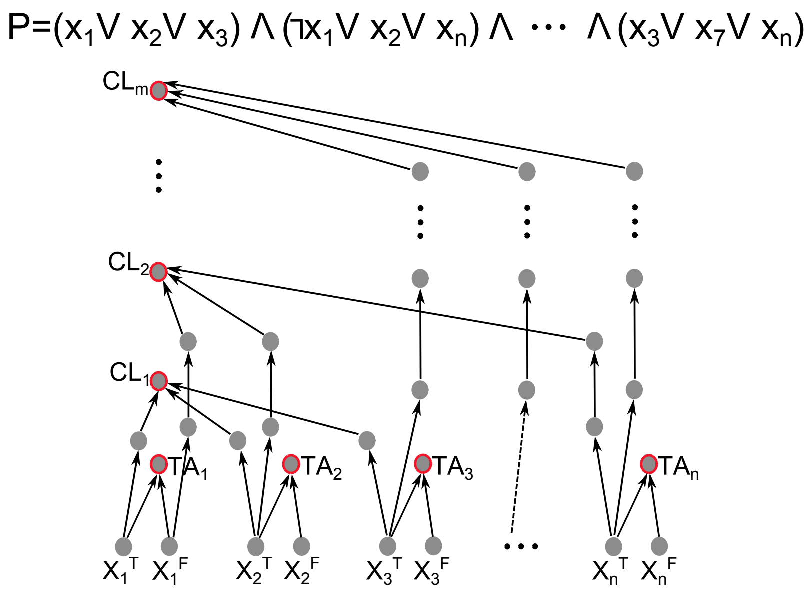


Figure 1: The graph associated to a boolean formula P. For reducing the complexity of the notations, the path nodes $Pa_j^{k;i}$ are not labeled on the figure.

Indeed, the second requirement above can be easily concluded since we need al least n distinct (1-bounded) driver nodes to control the n tautology nodes TA_j, as for any two such nodes there is no node in G having a (directed) path to both of them. Moreover, since these nodes are path-connected only with valuation nodes (X_j^T and X_j^F), controlling any TA_j requires either a direct link to a driver node, or a control via such a valuation node. Thus, if the target T is controlled using exactly n driver nodes, then at most one of the valuation nodes associated to a boolean variable can be connected to each of these driver node. From the above reasoning we can also conclude that controlling all the tautology nodes, and thus also the entire target T, requires at least n distinct driver nodes.

Let us assume first that formula P is satisfiable. Thus, there exists a valuation $x_1^{v_1}, ...x_n^{v_n}$ for each of the n variables such that each clause is validated by at least one of these truth assignments. Let B_i, $1 \leq i \leq n$ be a set of n driver nodes, each connected by an edge $(B_i, X_i^{v_i})$ to the associated valuation node from the validating truth assignment. Since the formula P is validated by this truth assignment, it implies that all the clause nodes $CL_1, ..., CL_n$ are connected to at least one of the valuation nodes $X_1^{v_1}, ..., X_n^{v_n}$, and thus to at least one of the driver bodes $B_1 ... B_n$. Moreover, by the way we constructed the graph G, if one of the above valuation nodes is connected to several clause nodes, the length of all these paths, i.e., the number of edges in-between the driver nodes and the clause nodes, are all different, and of length greater or equal than 3.

In the same time, each of the n driver nodes is connected to exactly one tautology node, using a path of size 2. Thus, the length of the above path is different from any of the path which could connect the corresponding valuation node to any of the clause nodes.

To conclude, the n driver nodes are overall connected to all the nodes in the target T. Also, for each of these driver nodes, the length of the paths connecting it to nodes in the target T are all different. Also, for any such path of length k between B_i and a target node $Y \in T$, there is no other path of length k between B_i and any other target $Y' \in T$; thus, the column $C_T A^k B_i$ from the target controllability matrix contains only one non-zero entry. Thus, in the target controllability matrix we can successfully choose $m + n$ linearly independent columns, i.e., its rank is $m + n$.

Assume now that the target T, containing n tautology nodes and m clause nodes, can be controlled from exactly n driver nodes $b_1, ..., b_n$, under the 1-bounded principle (i.e., allowing for exactly 1 driven node per every driver node).

As previously demonstrated, none of the n driver nodes can control two tautology nodes at the same time. Moreover, no two valuation nodes corresponding to the same variable can be both connected to the same driver node.

Since the target T can be controlled from the $b_1...b_n$ nodes, it means there exists a matrix $B \in R^{|V| \times n}$ such that the target controllability matrix satisfies:

$$\text{rank}[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{|V|-1} B] = m + n \tag{7}$$

Moreover, from the reasoning above we have that each column of B contains exactly one non-zero entry, either corresponding to a tautology-node or to a valuation node. From the structure of the underlying graph G we have that the target controllability matrix contains at most one non-zero entry per column, and that such an entry corresponds to exactly one path from a driver node, to a target node.

Consider now the m clause-nodes from the target T. Since the target is entirely controlled from the size n driver set, it means that the size-n set of driven nodes has to be connected to the m clause nodes. In particular, since no edge is outgoing from any of the tautology nodes, the subset of valuation nodes within the set of driven nodes has to be connected to all the m clause nodes. That is, for each of the clause nodes CL_i there must exist at least one valuation node within the set of driven nodes connected to CL_i. According to the graph construction procedure from the formula P, it means that every clause cl_i of P is validate by at least one of the chosen valuations. Moreover, as demonstrated above, for each variable x_j we are picking (in the set of driven nodes) at most one of its valuation nodes. Thus, the chosen valuations given by the set of driven nodes validate the formula P, i.e., P is satisfiable.

The above result can be generalized for the case when each of the driver nodes is connected to the network by at most k edges, for any given constant k. That is, each driver node is controlling at most k driven nodes; we call such a system k-bounded.

Definition We say that for a linear time invariant dynamical system (A, B, C_T) the input controller is *k-bounded* if and only if the matrix B contains at most k

non-zero values on every column, i.e., each of the m inputs $u_j(t)$ control at most k of the variables in X.

Theorem: The k-Bounded Target Control Optimization Problem is NP-hard. Namely, given a linear time invariant dynamical system (A, B, C_T) with a k-bounded input controller and target control set T, finding the minimal set of driven nodes controlling the target T is an NP-hard optimization problem.

We omit the proof here.

4 Approximation algorithms for target control

We have demonstrated in the previous section that trying to provide the optimal solution for the Target Control problem is computationally hard. An alternative choice is to develop approximation algorithms, trying to get close to the optimal solution in a time-efficient manner.

A first Greedy algorithm for the Target Control problem has been described by Gao et al.[4]. The authors approach the problem from a different perspective, that of generating a linking in an associated network, called the dynamic graph; the method has its roots in earlier studies of Poljak and Murota [13, 14].

In the following we present first the approach of Poljak and Murota [14] which connects the target control problem to the linking graph structure. Then, we proceed to presenting the Gao et al. approximation algorithm for target controllability, show its connection to the linking graph approach, and analyze the algorithm's shortcomings. Finally, we introduce three new heuristic improvements of the optimization algorithm, and analyze their performance.

Let (A, B, C_T) be an lti dynamical system over n variables, m inputs, and l targets (i.e., $|T| = l$), and let $G = (V, E)$ be the associated network graph. The dynamical graph $\overline{G}$ is a time-disjoint representation of the network graph, where each state (from $t = 1$ to $t = n$) and each input variable (from $t = 0$ to $t = n - 1$) is viewed as a distinct node at different time points, whereas the target states are associated only with the time-point $t = n + 1$. Formally, it is defined as the graph $\overline{G} = (\overline{V}, \overline{E})$, with the set of nodes $\overline{V} = \overline{V_A} \cup \overline{V_B} \cup \overline{V_C}$, where

- $\overline{V_A} = \{v_{i,t} \mid i = 1..n,\ t = 1..n\}$,
- $\overline{V_B} = \{v_{n+j,t} \mid j = 1..m,\ t = 0..n-1\}$, and
- $\overline{V_C} = \{v_{n+m+k} \mid k = 1..l\}$.

Note that the nodes in $\overline{V_C}$ are in one-to-one correspondence with the nodes V_C, as well as with the target T. The graph $\overline{G}$ has the following set of edges $\overline{E}$:

- $\{(v_{j,t}v_{i,t+1}) \mid \text{for all } i \text{ and } j \text{ such that } A_{i,j} \neq 0,\ t = 1..n\} \cup$
- $\{(v_{n+j,t}v_{i,t+1}) \mid \text{for all } i \text{ and } j \text{ such that } B_{i,j} \neq 0,\ t = 0..n-1\} \cup$

- $\{(v_{j,n}v_{n+m+i}) \mid$ for all i and j such that $C_{i,j} \neq 0\}$.

A collection $L = (p_1, p_2, ..., p_k)$ of k edge disjoint paths in the dynamical graph $\overline{G}$ is called a linking of size k. If $S, T \subseteq \overline{V}$ are the sets of initial and terminal nodes of the path L, then we say that L is an (S, T)-linking.

It has been shown in [14] that if (A, B) is an lti dynamical system with m driver nodes (i.e., the number of columns of B is m) and T is a size-l target set which is controllable from these driver nodes, then there must exist an $(\overline{V_B}, \overline{V_C})$-linking of size l. It has been a question for many years whether the converse of the above result also holds. Namely, if for an lti system (A, B, C_T) there exists an $(\overline{V_B}, \overline{V_C})$-linking of size l, then does it imply that the size-m driver set associated to B is controlling the target T, i.e., rank$[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{n-1} B] = l$? Although the answer to this question was proved in [14] to be negative, it became clear that any counter-example for this claim must obey some very strict design conditions regarding the controlling path from the driver nodes to the target.[1] Thus, in practice, finding a collection of nodes V_B such that there exists a $(\overline{V_B}, \overline{V_C})$-linking of size l provides a good candidate for the set of driver nodes controlling the target V_C.

The above approach has been employed by Gao et al. [4] which introduced a Greedy algorithm for the target control problem. Namely, given an lti A and a target T, their algorithm searches for a small set V_B for which there exists a $(\overline{V_B}, \overline{V_C})$-linking. In turns, such a set V_B would have a very high probability for defining a set of driver nodes for the target T. However, after applying this algorithm, one has to perform a validation step which verifies whether the selected set of driver/driven nodes selected by the algorithm are indeed controlling the target. This can be done by checking that the rank of the controllability matrix $[C_T B, C_T AB, , C_T A^2 B, ..., C_T A^{n-1} B]$ is indeed equal to $|T|$.

In the following we describe the Gao et al. algorithm [4] and we introduce three new heuristically improved variants of it. The comparative analysis of all these algorithms is performed in Section 5

4.1 The basic Target Control Algorithm (TarCo)

Let A be an lti over n variables and let $G = (V_A, E_A)$ be the directed graph associated to it. Let $T \subseteq V_A$ be a set of target variables/nodes. The following algorithm outputs a set of driven nodes D which has a one-to-one correspondence to the searched set V_B for which there exists a $(\overline{V_B}, \overline{V_C})$-linking.

Step 1: Let $i = 0$, $C^i = T$, and $D = D^i = \emptyset$.

Step 2: Define a bipartite graph G_{bi} with nodes $L \cup R$, where $L = V_A$, $R = C^i$, and any node appearing both in V_A and in C^i is treated distinctly in L and R. For

[1] An intuitive description of those systems for which a linking is not translated to a valid controlling path is when there exist two targets t_1 and t_2 such that for every path from a driver note d to t_1 there exists another path from d to t_2 using the exact same collection of edges (as a multiset).

$l \in L$ and $r \in R$ there exists an edge (l, r) in G_{bi} iff $(l, r) \in E_A$ is an edge in the initial directed graph G.

Step 3: Find a maximum matching (M_L, M_R) in G_{bi}, $M_L \subseteq L$ and $M_R \subseteq R$, and let $C^{i+1} = M_L$ be the set of the left sided matched nodes and let $D_i = R \setminus M_R$ be the set of right sided un-matched nodes. Let $D = D \cup D_i$.

Step 4: We consider C^{i+1} as the new set of target nodes. If $C^{i+1} = \emptyset$ then we complete the algorithm and output D. If not, we proceed to Step 5.

Step 5: If $i < n$ then $i = i + 1$ and proceed to Step 2 with the updated target C^i and driven set D. Else, proceed to Step 6.

Step 6: Output D as the set of driven nodes.

Note: The previous algorithm is focussed on minimizing the set of generic driver nodes, and not the set of 1-bounded driver nodes (i.e., driven nodes) focussed on this research. In particular, the algorithm might not output the complete set of driven nodes, but rather a subset of it which is in one-to-one correspondence with the set of generic driver nodes. Indeed, if the algorithm ends in Step 6, then it implies that the target set C^n is non-empty. Since the total number of nodes in G is n, it implies that all the remaining nodes in C^n can be partitioned into a number of cycles. Since the 1-bounded condition for driver nodes is not imposed, all the nodes in these cycles, including the ones in C^n, can be controlled from any driver nodes.

In order to modify the $TarCo$ algorithm for finding a suitable set of driven nodes, instead of driver nodes, we implemented an update/optimization step.

4.2 The Optimized Target Control Algorithm (OpTarCo)

In the $TarCo$ algorithm, once a node x is selected for being a driven node, i.e., added to D in Step 3, we do not check whether until that stage the node x appeared before in some previous control path. If so, now that we know that node x is selected for being a driven node, we can prune that control path after reaching node x. This leads to the following modified algorithm:

As before, let A be an lti over n variables and let $G = (V_A, E_A)$ be the directed graph associated to it. Let $T \subseteq V_A$ be the set of target variables/nodes.

Step 1 (Similar to $TarCo$): Let $i = 0$, $C^i = T$, and $D = D^i = \emptyset$.

Step 2 (Similar to $TarCo$): Define a bipartite graph G_{bi} with nodes $L \cup R$, where $L = V_A$, $R = C^i$, and any node appearing both in V_A and in C^i is treated distinctly in L and R. For $l \in L$ and $r \in R$ there exists an edge (l, r) in G_{bi} iff $(l, r) \in E_A$ is an edge in the initial directed graph G.

Step 3.1: Find a maximum matching (M_L, M_R) in G_{bi}, $M_L \subseteq L$ and $M_R \subseteq R$, and let $C^{i+1} = M_L$ be the set of the left sided matched nodes and $D_i = R \setminus M_R$ be the set of right sided un-matched nodes.

Step 3.2: For each $x \in D_i \setminus D$, do:

Step 3.2.1: If node x appears in any previously computed C^j, $j < i$, then remove the entire control path from that occurrence (in C^j) onward, and update all the sets C^k, D^k with $j \leq k \leq i + 1$ accordingly. Then update D as $D = \bigcup_{p=0toi} D^p$.

End For (from Step 3.2)

Step 4: We consider $D = D \cup D^i$ as the new set of driven nodes, and $C^{i+1} \setminus D$ as the new set of targets. If $C^{i+1} = \emptyset$ then we complete the algorithm and output D. If not, we proceed to Step 5.

Step 5 (Similar to $TarCo$): If $i < n$ then $i = i + 1$ and proceed to Step 2 with the updated target C^i and driver set D. Else, proceed to Step 6.

Step 6: For all the remaining nodes in C^n, add them one by one to the driven set D and, at each new addition to D, perform the check from Step 3.2.1, i.e., pruning the existing controlling path for each new addition in D.

Step 7: Output D as the set of driven nodes.

4.3 Heuristically Optimized Target Control Algorithms (HeTarCo1-3)

In Step 3 (resp. 3.1) of the previous two algorithms, at each iteration of the search process we find a maximum matching in between the nodes of G and the current target C^i. However, such maximum matchings might not be unique, in which case some of these maximum matchings might be more suitable to be chosen. Let us assume the algorithm is at some iteration i in its search procedure. Let $C^1, ..., C^i, D^1, ..D^{i-1}$ and D be the already computed sets of targets and driven nodes. Let G_{bi} be the bipartite graph constructed in iteration i, with nodes $L \cup R$, where $L = V_A$, $R = C^i$, and any node appearing both in V_A and in C^i is treated distinctly in L and R. When searching for a maximum matching (M_L, M_R) in G_{bi}, $M_L \subseteq L$ and $M_R \subseteq R$, we are setting the following heuristic criteria for guiding the process towards a minimum number of driven nodes. Note, not all criteria below can be followed in the same time.

- Criteria 1: When computing the maximum matching (M_L, M_R), maximize the use of already driven nodes in M_L.
- Criteria 2: When computing the maximum matching (M_L, M_R) try to avoid the creation of cyclic controlling path. That is, avoid selecting nodes $x \in M_L$ such that there exists $j \leq i$ and a sequence $u_{i+1}, ..., u_j$ such that $u_k \in C^k$ for all $j \leq k \leq i$, $u_{i+1} = u_j = x$, and for all $j \leq k \leq i$, u_j is matched to u_{j+1} in the corresponding bipartite graph.
- Criteria 3: When computing the maximum matching (M_L, M_R), maximize the use of nodes in M_L which have appeared in some previous $C^j, j < i$, on a path that is already controlled (ends with a driven node).
- Criteria 4: When computing the maximum matching (M_L, M_R), maximize the use of nodes in M_L which have appeared in some previous $C^j, j < i$, on a path that is not controlled yet.

- Criteria 5: When computing the maximum matching (M_L, M_R), maximize the use of edges (u, v) (with $u \in M_L$ and $v \in M_R$) which have been used in some previous matching and are part of at least one path that is already controlled.

- Criteria 6: When computing the maximum matching (M_L, M_R), maximize the use of edges (u, v) (with $u \in M_L$ and $v \in M_R$) which have been used in some previous matching, but are not part of any path that is already controlled.

Following subsets of the above selection criteria, as well as the previously introduced optimized control algorithm ($OpTarCo$) as a base algorithm, we define in the following a series of three heuristically optimized target control algorithms, as follows.

Algorithm $HeTarCo1$**:** Within Step 3.1 of the $OpTarCo$ algorithm select a maximum matching (M_L, M_R) following Criteria 2.

Algorithm $HeTarCo2$**:** Within Step 3.1 of the $OpTarCo$ algorithm select a maximum matching (M_L, M_R) following Criteria 1, 2, 3, and 4, in this exact order of importance.

Algorithm $HeTarCo3$**:** Within Step 3.1 of the $OpTarCo$ algorithm select a maximum matching (M_L, M_R) following Criteria 2, 5, 6, 1, 3, and 4, in this exact order of importance.

5 Results: A comparative analysis of the four algorithms

We analyzed the performance of the four approximation algorithms against both randomly generated networks and targets, as well as against a human protein-protein interaction network, using as target a set of Breast Cancer specific essential genes.[2]

We predict the performance of all four algorithms to be highly dependent on the size of the network, i.e., the number of nodes and edges, the average degree, the size and the choice of the target set, as well as the overall control-affinity of the network, i.e., the size of its minimal driven set controlling the entire set of nodes. Thus, in order to perform a fair analysis of the algorithms against one-another we impose several conditions for our test cases.

We generated randomly a set of 100 networks, each over 1000 nodes and having exactly 4000 directed edges, i.e., all networks have equal average node (in/out)

[2]Note that there is a one-to-one correspondence between genes and proteins; thus, having as target a set of essential genes means the equivalent set of essential proteins.

degree 4.[3] For each network, we randomly selected 10 target sets of size 100, 200, ..., up to 1000 (i.e., all the network's nodes), respectively. We performed 10 (independent) runs for each of the target sets (and networks) with each of the four algorithms; all runs were performed on the same Xeon 6/3GHz core computer.

In order to compare the overall performance of all algorithms, as well as the performance of each individual algorithm applied over different test-cases, we normalize the performance of each run of an algorithm against the minimal total driven control set of the network, i.e., against the size of the minimum set of driven nodes controlling the entire network, as obtained over all runs with all algorithm (40 runs for each of the complete 1000 target node set).[4] Thus, a reported value of 1 for an algorithm's run (for a target and a network) signifies that the size of that solution is equal to the minimal total driven control set of that respective network.

In Fig. 2 we report the comparative analysis of the four algorithms with regards to the average (normalized) size of solutions after each run of the algorithm, Fig. 2a), by taking to minimum of 10 runs for the same algorithm over the same target, Fig. 2b), as well as the average time complexity of each individual run, Fig. 2c). Our analysis shows that in terms of minimality of the driven set solution, all three heuristic algorithms, $HeTarCo1-3$ perform slightly better than $OpTarCo$, when the target set is proportionally small compared to the total number of nodes, i.e., less than 15%. When the target size increases on the other hand, two out of three heuristically improved algorithms perform considerably worse, whereas the third one has a very similar performance as $OpTarCo$. However, in terms of average time taken by each of the algorithms to perform a run, there are up to 10-fold decreases for all heuristically improved algorithms.

In order to better analyze the importance of multiple runs over the solution size decrease we considered testing the four algorithms by doing multiple runs over a fixed time period, namely 12 hours. For that we have selected a human protein-protein interaction network consisting of approx. 3000 nodes (i.e., proteins) and 1000 directed edges (i.e., protein interactions); the network was obtained from the SIGNOR (SIGnaling Network Open Resource) database [12]. For targets, we have intersected the set of nodes from the previous network with the list of Breast Cancer essential genes taken from the COLT-Cancer database [7]. In particular, we considered the MDA-MBD-231 cell line and followed the GARP (Gene Activity Rank Profile) and GARP-P values of corresponding proteins mentioned in the database, selecting those entries with negative GARP score and GARP-P

[3]Similar analyses were performed for networks of average degree from 2 to 6, but due to space limitations we concentrate here over average degree 4 networks; similar results were obtained in all cases, with more pronounced differences (for the normalized values) in the case of higher degree networks.

[4]Note that computing the driven target control is different than computing the driver control, as for the latter one there is a known polynomial time algorithm computing the size of the minimal (total) driver control set, see [10]. In practice however, we observed that both values, i.e., for driver and driven control, are very close to one-another in the case of randomly generated networks and real-life bio-medical networks.

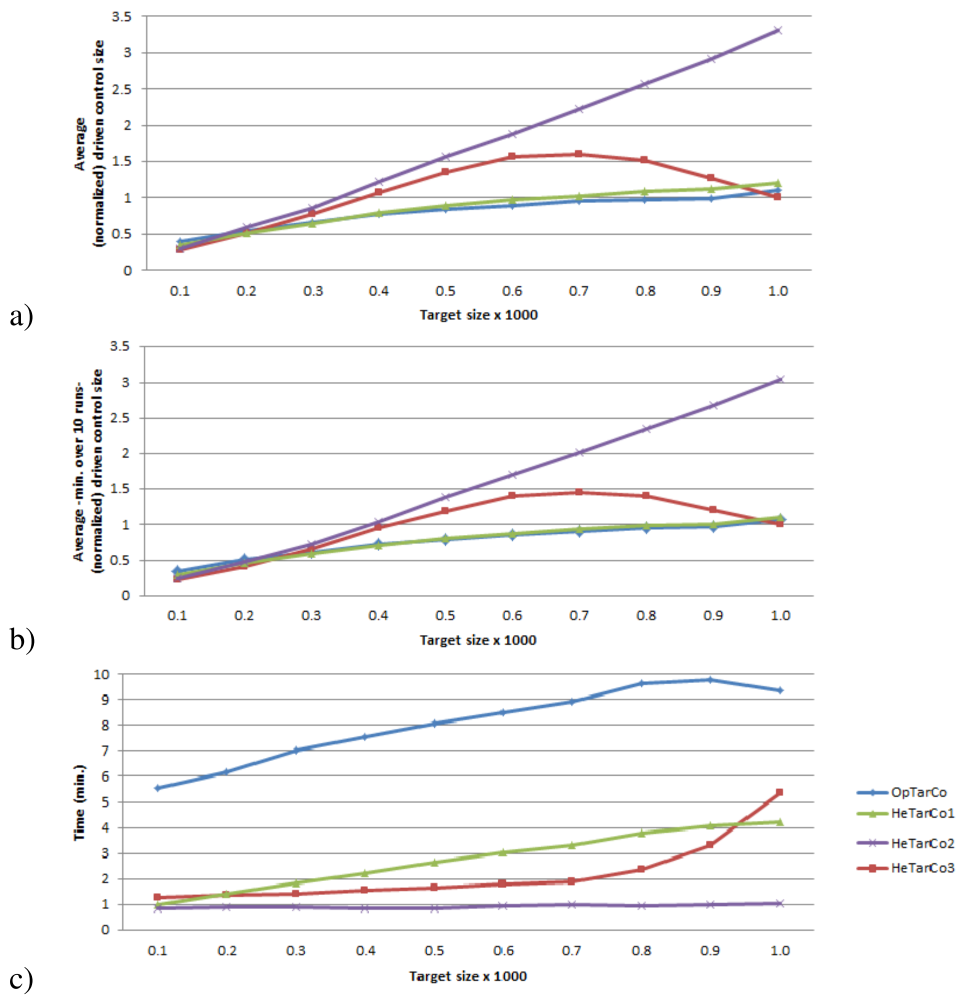
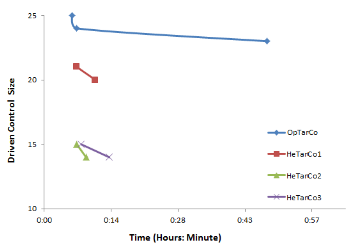


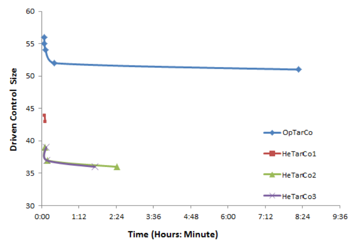
Figure 2: Comparative analysis of the four algorithms over randomly generated networks. a) The average (normalized) size of the driven set per algorithm and per target size; b) The average (normalized) size of the minimal size (over 10 runs) of the driven set per algorithm and per target size; c) the average time required for a single run, per algorithm and per target size.

value less than 0.05. The rationale behind this particular test-case can be found in network pharmacology: Identifying a relative small set of proteins which could control larger proportions of target essential genes would be advantageous for the development of efficient drugs. By cascading effects, these drugs could target several Breast Cancer essential genes in the same time, with minimal effect over the healthy cells (by definition, disease-specific essential genes are not taken among those genes which are essential for normal, i.e. healthy, cell survival).

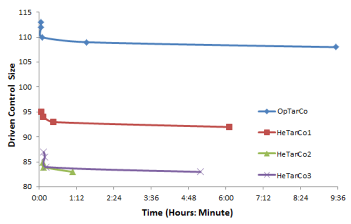
The above procedure provided us with a set of 145 essential genes which could be used as target pool. We selected 3 target sets containing 30, 72, and 145 nodes, respectively; the choice of nodes for the smaller/incomplete targets was done according to an increasing ordering of their corresponding GARP values. The results of the 12 hour runs of the algorithms are presented in Fig. 3. As it can be seen from this analysis, the heuristically improved algorithms, especially $HeTarCo2$ and $HeTarCo3$, performed much better and faster.



a) Target size 30;



b) Target size 72;



c) Target size 145;

Figure 3: Comparative analysis of the four algorithms for multiple runs over a fixed 12 hour time period. The reported performance is cut short if no smaller sized solutions appear after a certain time point.

6 Conclusions

The network controllability approach provides an interesting insight into a system modeled as a directed graph: given a set of target nodes, we can identify a set of driven nodes that allow an external user to gain control over the target nodes through an external intervention on the driven nodes, taking advantage of the internal 'wiring' of the network. We established in this paper that calculating a minimal set of drive nodes is an NP-hard problem – this makes it hopeless to apply the approach to real-life networks, such as signaling networks, that may have thousands of nodes and edges. Even more, we established this hardness result for a more practical version of the problem, where the external intervention mimics that obtained through drug delivery. The drug delivery constraint is modeled in our approach through a driver node that interacts with exactly one node in the graph (seen as the main target of that drug) or, in a different formulation, with at most k nodes (seen as the main and the secondary targets of that drug). At the same time, we introduced several different heuristics for approximated target control; these algorithms find a set of driven nodes (perhaps not the smallest one) that control a given set of target nodes. Our algorithms improve significantly the currently known algorithm for the problem and we demonstrated in this paper that they are efficiently applicable even to real-life-size networks.

There are several highly interesting research avenues that may be explored in this area. On the theoretical side, an open problem is to establish the approximation threshold of our heuristics. Another one, on which almost nothing is known, is on the general, rather than on the structural controllability of networks; in other words, this is the problem in which we also ask about the timing and the level of the external intervention, in addition to identifying the driver nodes where it should be applied. On the applied side, an interesting problem is to connect the network controllability approach with data on FDA-approved drug targets, and with data on gene-essentiality for different types of diseases; this has the potential of helping in the design of more diverse therapeutic strategies using currently known drugs.

References

[1] Ashworth A, Lord CJ, Reis-Filho JS. Genetic interactions in cancer progression and treatment. Cell 2011; 145(1):30-8.

[2] Blomen, Vincent A. et. al. Gene essentiality and synthetic lethality in haploid human cells. Science. 2015; 350(6264): 1092-1096.

[3] Brough et al. Searching for synthetic lethality in cancer. Curr Opinion in Genetics & Development 2011; 21: 34-41.

[4] Gao J, Liu Y, D'Souza MR, Barabsi LA. Target control of complex networks. Nat Comm 2014; 5415.

[5] Hopkins AL. Network pharmacology: the next paradigm in drug discovery. Nat Chem Biol 2008; 4: 682-90.

[6] Kalman, R. E. Mathematical description of linear dynamical systems. J. Soc. Indus. Appl. Math. Ser. 1963; A 1: 152192.

[7] Koh YLJ,Brown RK,Sayad A,Kasimer D,Ketela T,Moffat1 J. COLT-Cancer: functional genetic screening resource for essential genes in human cancer cell lines. Nucl. Acids Res. 2012 Jan; 40(Database issue): D957D963. doi:10.1093/nar/gkr959

[8] Kolch W et al. The dynamic control of signal transduction networks in cancer cells. Nat Rev 2015; 15: 515-525.

[9] Lin,C.-T.Structuralcontrollability. IEEE Trans. Automat. Contr. 1974; 19: 201208.

[10] Liu Y, Slotine j, Barabsi LA. Controllability of complex networks. Nature 2011; 473: 167-73.

[11] Marcotte R, Brown RK, Suarez F et al. Essential Gene Profiles in Breast, Pancreatic, and Ovarian Cancer Cells. Cancer Discovery. 2012 Feb; 2(2):172-89. doi:10.1 158/2159-8290

[12] Perfetto L, Briganti L, Calderone A. SIGNOR: a database of causal relationships between biological entities. Nucl. Acids Res. 2016 Jan 4; 44(D1):D548-54. doi:10.1093/nar/gkv1048

[13] S. Poljak. On the generic dimension of controllable subspaces. IEEE Transactions on Automatic Control. 1990; 35(3): 367-369. doi: 10.1109/9.50361

[14] S. Poljak and K. Murota. Note on a graph-theoretic criterion for structural output controllability. IEEE Transactions on Automatic Control. 1990; 35: 939-942.

[15] Shields, R. W. and Pearson, J. B. Structural controllability of multi-input linear systems. IEEE Trans. Automat. Contr. 1976; 21: 203212.

[16] Wang T et al. Identification and characterization of essential genes in the human genome. Science 2015; 350(6264).

[17] Zañudo JGT, Albert R. Cell Fate Reprogramming by Control of Intracellular Network Dynamics. PLoS Comput Biol 2015; 11(4).

[18] Zhan T, Boutros M. Towards a compendium of essential genes - From model organisms to synthetic lethality in cancer cells. Crit Rev Biochem Mol Biol 2016; 51(2): 74-85

ISBN 978-952-12-3402-6
ISSN 1239-1891