



Selecting energy efficient inputs using graph structure

Isaac Klickstein & Francesco Sorrentino

To cite this article: Isaac Klickstein & Francesco Sorrentino (2022): Selecting energy efficient inputs using graph structure, International Journal of Control, DOI: [10.1080/00207179.2021.2022218](https://doi.org/10.1080/00207179.2021.2022218)

To link to this article: <https://doi.org/10.1080/00207179.2021.2022218>



Published online: 10 Jan 2022.



Submit your article to this journal [↗](#)



Article views: 31



View related articles [↗](#)



View Crossmark data [↗](#)



Selecting energy efficient inputs using graph structure

Isaac Klickstein  and Francesco Sorrentino 

Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, USA

ABSTRACT

Selecting appropriate inputs for systems described by complex networks is an important but difficult problem that largely remains open in the field of control of networks. Recent work has proposed two methods for energy efficient input selection; a gradient-based heuristic and a greedy approximation algorithm. We propose here an alternative method for input selection based on the analytic solution of the controllability Gramian of the ‘balloon graph’, a special model graph that captures the role of both *distance* and *redundant paths* between a driver node and a target node. The method presented is especially applicable for large networks where one is interested in controlling only a small number of outputs, or target nodes, for which current methods may not be practical because they require computing a typically very ill-conditioned matrix, called the controllability Gramian. Our method produces comparable results to the previous methods while being more computational efficient.

ARTICLE HISTORY

Received 24 March 2021

Accepted 16 December 2021

KEYWORDS

Networked systems; discrete optimisation; optimal control

1. Introduction

Many of the systems we interact with every day are described by complex networks such as social media (Bovet & Makse, 2019), the power grid (Arianos et al., 2009; Pagani & Aiello, 2013), the world wide web (Barabási et al., 2000) and our own biology (Sporns, 2013). As our ability to describe these complex networked systems improves, attention has increasingly turned to our ability to influence, or control, these systems with external signals. For example, targeted media campaigns (both beneficial and malicious) on social media platforms have proven to be incredibly effective (Grinberg et al., 2019), or as our knowledge of human pharmacology grows, multi-drug multi-target therapies become viable for drug developers (Y. H. Li et al., 2016). While the dynamics of each of these systems are significantly different, the first step toward influencing any of them requires a choice of where one should apply the external control signal to the system. In terms of malicious social media campaigns, this means to choose which members of the social network should be targeted by counter measures to provide correct information. In terms of multi-drug therapies, this means to choose which drug targets to activate or inhibit by the therapy cocktail. For power grid networks, this could mean selecting which lines should receive high voltage direct current links to improve the networks stability (Summers et al., 2016).

Here, we focus on linear systems as, at least over short time scales, continuous nonlinear systems can be approximated as linear (Klickstein et al., 2017b; Liu & Barabási, 2016). Rigorous conditions for the controllability of unweighted graphs have been presented in Qu et al. (2020), Ji et al. (2020) and Guo et al. (2021). The problem of selecting the smallest number of control signals to ensure a complex network is controllable has

been addressed in many different frameworks such as structural controllability (Liu et al., 2011), exact controllability (Yuan et al., 2013) and output controllability (Commault et al., 2017; J. Gao et al., 2014; Iudice et al., 2019; Lo Iudice et al., 2015; Zhang et al., 2017; Klickstein et al., 2017a). While the minimum number of inputs is sufficient to ensure controllability, applying this minimum control may lead to extremely ill-conditioned systems of equations (Klickstein et al., 2017; Sun & Motter, 2013; Yan et al., 2012, 2015). Instead, more recently, efficient control problems have garnered interest which look to minimise a control energy metric while constraining the number of control inputs (Summers et al., 2016). The selection of the number of control inputs and their distribution throughout the complex network are vitally important to the feasibility and efficiency of a control action.

Efficient controllability problems, unlike minimum controllability problems, minimise a metric on the control energy while constraining the number of control inputs (Summers et al., 2016). Efficient control problems have previously been shown to be NP hard (Tzoumas et al., 2015, 2016) by mapping them to the hitting set problem following (Olshevsky, 2014). This result removes the possibility of any polynomial time algorithm to find the optimal solution. Instead, heuristic methods and approximation algorithms must be used to find ‘good’, but sub-optimal, solutions. Two such methods are described briefly here.

The projected gradient method (G. Li, Deng, et al., 2018; G. Li, Ding, et al., 2018; G. Li, Hu, et al., 2016) finds a locally optimal solution to a continuous relaxation of the original discrete input selection problem. A rounding procedure, called key component analysis, is used to create a solution to the original discrete problem. Here, we compare our method to a version of

the above heuristic which uses probabilistic projection (L. Gao et al., 2018) to replace the rounding procedure.

A number of control energy metrics have been shown to be submodular set functions (Summers et al., 2016; Summers & Lygeros, 2014). Greedy algorithms have a well-known approximation guarantee when used to minimise submodular set functions (Fisher et al., 1978). Currently, greedy algorithms have not been explicitly used to solve the input selection problem for target control problems, except for the case that the target set coincides with the entire node set (Summers et al., 2016). Nonetheless, the submodularity property holds for the general case of any target control problem (see corollary 2 in Summers et al., 2016) so we also compare our method to a greedy algorithm.

Both of these existing methods are iterative and require computing controllability Gramian matrices at each iteration which can be extremely ill-conditioned. In this paper, we present a novel method for energy efficient selection of driver nodes in general graphs. As opposed to the methods described above, the method we describe here uses structural properties of the graph explicitly. Recent work has derived analytic expressions for the control energy of lattice networks (Klickstein et al., 2018; Klickstein & Sorrentino, 2018a, 2018b; Zhao & Pasqualetti, 2018) and has shown that the control energy in these structurally simple networks can often well approximate the control energy in complex networks. We first analytically compute the output controllability Gramian of the ‘balloon graph’ which consists of a number of disjoint directed paths from a driver node to a target node. This calculation is aided by the particular symmetric structure of this graph. Second, we solve the *facility location problem* (Mirchandani & Francis, 1990) with a cost matrix derived from the pair-wise costs computed using the model graph to select an energy efficient set of driver nodes.

In Section 2 we introduce necessary background about graphs, the controllability Gramian, and the facility location problem. In Section 3, we present our first results, deriving the controllability Gramian for a model network. The result is used to construct a cost matrix that describes the ability of each potential input to influence each target node. In Section 4, we review two of the main alternative methods to select driver nodes from the literature. In Section 5, we present a comparison of the three methods where we show no method outperforms any other in terms of the cost of the returned solution, but that our method is more computationally efficient. Finally, conclusion is given in Section 6.

2. Background

2.1 Graph symmetries

Graphs are denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which consist of $|\mathcal{V}| = n$ nodes and edges $(v_j, v_k) \in \mathcal{E}$ which may be read ‘from node v_j to node v_k ’. Unless otherwise stated, all graphs considered here are assumed to be directed. The set of neighbours of a node v_j , denoted $\mathcal{N}_j$, is defined as the set of nodes v_k such that $(v_k, v_j) \in \mathcal{E}$. We do not include any loops, that is an edge (v_j, v_j) , in the set of edges, as loops are treated separately. A graph may be represented as an adjacency matrix, $A \in \mathbb{R}^{n \times n}$, which has elements $A_{j,k} > 0$ if $(v_k, v_j) \in \mathcal{E}$ and $A_{j,k} = 0$ otherwise. The diagonal of

the matrix A , $A_{j,j} < 0$, $j = 1, \dots, n$, represents the loops present at each node. In this paper, we assume *uniform edge weights* and *uniform loop weights*, that is, edge weights are equal, $A_{j,k} = A_{j',k'}$ for all $(v_j, v_k), (v_{j'}, v_{k'}) \in \mathcal{E}$ and $(v_{j'}, v_{k'}) \in \mathcal{E}$ and all loop weights are equal, $A_{j,j} = A_{k,k}$, for all $j, k = 1, \dots, n$. The edge weight is denoted $\gamma > 0$ and the loop weight is denoted $-\nu < 0$.

Definition 2.1 (Graph Symmetries and the Automorphism Group (Lauri & Scapellato, 2016)): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $\pi : \mathcal{V} \mapsto \mathcal{V}$ be a bijection on the set of nodes of a graph. After applying a permutation to the nodes of a graph, define the new set of edges as $\mathcal{E}^\pi$ where if $(v_j, v_k) \in \mathcal{E}$ then $(\pi(v_j), \pi(v_k)) \in \mathcal{E}^\pi$. A permutation π is a symmetry if $\mathcal{E} = \mathcal{E}^\pi$. The set of all such symmetries along with function composition form the automorphism group of a graph, $\text{Aut}(\mathcal{G})$.

Let $\mathcal{D} \subseteq \mathcal{V}$ be a subset of the nodes in the graph. The reduced automorphism group $\text{Aut}^{\mathcal{D}}(\mathcal{G})$ consists of all symmetries in $\text{Aut}(\mathcal{G})$ that do not permute any node in $\mathcal{D}$ (this concept is also known as the automorphism group of a coloured graph (McKay & Piperno, 2014) where each driver node is a unique color and all non-driver nodes are the same colour).

$$\text{Aut}^{\mathcal{D}}(\mathcal{G}) = \{\pi \in \text{Aut}(\mathcal{G}) \mid \pi(v_j) = v_j, \forall v_j \in \mathcal{D}\} \quad (1)$$

The automorphism group (and any reduced automorphism group) induces a partition of the nodes, defined as the orbits of the graph, $\mathcal{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_q\}$, where two nodes $v_j, v_k \in \mathcal{O}_\ell$ if and only if there exists a symmetry π that maps $\pi(v_j) = v_k$. This partition is equitable, that is, every node in orbit $\mathcal{O}_\ell$ has the same number of neighbours in each other orbit. As an example, if node v_j is in orbit $\mathcal{O}_k$ and it has m neighbours in orbit $\mathcal{O}_\ell$, then if node $v_{j'}$ is also in orbit $\mathcal{O}_k$ it must also have m neighbours in orbit $\mathcal{O}_\ell$.

Definition 2.2 (Quotient Graph): Given a graph $\mathcal{G}$ and its orbits $\mathcal{O}$, the graph can be compressed to its *quotient graph*, $\mathcal{Q} = (\mathcal{O}, \mathcal{F})$, where each orbit is a node in the quotient graph, and the edges $(\mathcal{O}_j, \mathcal{O}_k) \in \mathcal{F}$ represent those pairs of orbits for which there exists edges passing from the nodes in $\mathcal{O}_j$ to the nodes in $\mathcal{O}_k$.

A permutation of a set of n elements, π , can also be expressed as a matrix, $P \in \{0, 1\}^{n \times n}$, with elements $P_{j,k} = 1$ if $\pi(v_j) = v_k$ and $P_{j,k} = 0$ otherwise. Applying the permutation to the adjacency matrix yields the permuted adjacency matrix $A^\pi = PAP^T$. If π is a symmetry then $A^\pi = A$ (assuming uniform edge weights and loop weights as specified above).

The orbit indicator matrix, $E \in \{0, 1\}^{n \times q}$, has elements $E_{j,k} = 1$ if node $v_j \in \mathcal{O}_k$ and $E_{j,k} = 0$ otherwise. The adjacency matrix of the quotient graph, $A^Q \in \mathbb{R}^{q \times q}$, can be found by applying the orbit indicator matrix,

$$A^Q = E^\dagger A E \quad (2)$$

where the superscript † denotes the Moore–Penrose pseudoinverse, defined as $E^\dagger = (E^T E)^{-1} E^T$. The elements of the quotient graph adjacency matrix, $A_{j,k}^Q$, are equal to the number of neighbours of a node $v_\ell \in \mathcal{O}_j$ that are in $\mathcal{O}_k$ (multiplied by the uniform edge weight γ).

2.2 Minimum energy control

Each node is assigned a time-varying state, denoted $x_j(t)$, $j = 1, 2, \dots, n$, whose behaviour is governed by its neighbours. We are able to influence the dynamics through a subset of the nodes, $\mathcal{D} \subseteq \mathcal{V}$, defined as the $|\mathcal{D}| = m$ driver nodes. The driver node set can be represented as a matrix, $B \in \{0, 1\}^{n \times m}$, where each column of B has a single nonzero element corresponding to a driver node. An independent, external, control input, denoted $u_\ell(t)$, $\ell = 1, \dots, m$, is assigned to each driver node $v_k \in \mathcal{D}$. The states evolve in time according to a system of linear differential equations where the state matrix A is the adjacency matrix of a graph.

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \quad (3)$$

An initial condition is assigned to each node at time $t = 0$, $x_j(0) = x_{j,0}$. The set of p target nodes, denoted $\mathcal{T} \subseteq \mathcal{V}$, are those whose states we would like to drive to a particular value at some final time $t = t_f$. The set of target nodes can also be represented as a matrix, $C \in \{0, 1\}^{p \times n}$, where each row has a single nonzero element corresponding to a target node.

$$\mathbf{y}(t) = C\mathbf{x}(t) \quad (4)$$

Definition 2.3 (Controllability Gramian): Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, the time-varying controllability Gramian is the symmetric n -by- n matrix $W(t)$ that satisfies the differential Lyapunov equation (DLE),

$$\dot{W}(t) = AW(t) + W(t)A^T + BB^T, \quad W(0) = O_n, \quad (5)$$

where O_n is the n -by- n matrix of all zeroes. If A is Hurwitz (all of its eigenvalues are in the left-hand side of the complex plane), then there is a unique stable fixed point of the DLE that satisfies the algebraic Lyapunov equation (ALE),

$$AW + WA^T = -BB^T \quad (6)$$

which we call the steady state controllability Gramian.

If the matrix A is Hurwitz and t_f is chosen large enough, then it may be appropriate to use W instead of $W(t_f)$.

Lemma 2.4 (Output Controllability (Kailath, 1980)): Define the matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$ and define $W(t)$ to be the solution to the DLE using A and B . The triplet (A, B, C) is output controllable if, for every vector $\mathbf{x}_0 \in \mathbb{R}^n$, vector $\mathbf{y}_f \in \mathbb{R}^p$ and positive value $t_f > 0$, there exists a time-varying signal $\mathbf{u} : [0, t_f] \mapsto \mathbb{R}^m$ such that,

$$\mathbf{y}_f = C e^{At_f} \mathbf{x}_0 + C \int_0^{t_f} e^{A(t_f-t)} B \mathbf{u}(t) dt. \quad (7)$$

An equivalent statement is that the triplet (A, B, C) is output controllable if the output controllability Gramian,

$$\bar{W}(t) = CW(t)C^T \quad (8)$$

is nonsingular.

The output controllability Gramian appears in the solution to the following optimal control problem.

Lemma 2.5 (Minimum Energy Output Control (Klickstein et al., 2017)): Define the matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$ along with the vectors $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{y}_f \in \mathbb{R}^p$. Then the minimum energy output control problem is

$$\begin{aligned} \min \quad & J = \frac{1}{2} \int_0^{t_f} \|\mathbf{u}(t)\|_2^2 dt \\ \text{s.t.} \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ & \mathbf{x}(0) = \mathbf{x}_0, \quad C\mathbf{x}(t_f) = \mathbf{y}_f \end{aligned} \quad (9)$$

and, if (A, B, C) are output controllable, its unique solution is

$$\begin{aligned} J^* &= \frac{1}{2} (\mathbf{y}_f - C e^{At_f} \mathbf{x}_0)^T (CW(t_f)C^T)^{-1} (\mathbf{y}_f - C e^{At_f} \mathbf{x}_0) \\ &= \frac{1}{2} \boldsymbol{\beta}^T \bar{W}^{-1}(t_f) \boldsymbol{\beta} \end{aligned} \quad (10)$$

where $\boldsymbol{\beta} = (\mathbf{y}_f - C e^{At_f} \mathbf{x}_0)$ is called the control manoeuvre, which is the difference between the desired final output and what the final output would be in the absence of a control input. Note that the control manoeuvre $\boldsymbol{\beta}$ depends on the choice of the target output $\mathbf{y}_f$.

Lemma 2.6 (Symmetries in the Gramian (Klickstein & Sorrentino, 2018b)): Symmetries in the graph from which the adjacency matrix A was constructed appear as repeated values in the controllability Gramian. If two nodes $v_j, v_{j'} \in \mathcal{O}_\ell$ and another two nodes $v_k, v_{k'} \in \mathcal{O}_{\ell'}$, then the elements of the Gramian $W_{j,k}(t) = W_{j',k'}(t)$.

In the following, define the driver node reduced automorphism group, $\text{Aut}^{\mathcal{D}}(\mathcal{G})$ (that is, every driver node is in an orbit of cardinality one). All mentions of orbit indicator matrix or quotient graph refer to those matrices and graphs induced by the driver node reduced automorphism group.

The controllability Gramian for the quotient graph satisfies Equation (5) with A replaced by A^Q (as defined in Equation (2)) and $B^Q = E^\dagger B$.

$$\dot{W}^Q(t) = A^Q W^Q(t) + W^Q(t) A^{Q^T} + B^Q B^{Q^T} \quad (11)$$

If $v_j \in \mathcal{O}_{j'}$ and $v_k \in \mathcal{O}_{k'}$ then $W_{j,k}(t) = W_{j',k'}(t)$ so if $W^Q(t)$ is known, we can ‘expand’ the result to determine $W(t)$.

2.3 Input selection

Given matrices A , B and C , from Lemma 2.5, the minimum energy optimal control problem can be solved for $\mathbf{u} : [0, t_f] \mapsto \mathbb{R}^m$ with associated cost J^* . In this paper, we are interested instead in the scenario where A and C are provided but we may choose the set of driver nodes $\mathcal{D} \subseteq \mathcal{V}$ (equivalently the matrix B with the restrictions described previously) such that we minimise the control energy subject to a cardinality constraint on the set of driver nodes. For a graph with n nodes, there are $\binom{n}{m}$ potential sets of m driver nodes so a brute force search for even a moderate sized network is impossible. In addition, the solution to the minimum energy output control problem depends on the choice of control manoeuvre. To be more general, instead of minimising the optimal cost J^* in Equation (10) which depends on $\boldsymbol{\beta}$ directly, an energy metric that is independent of the particular control manoeuvre, $M(\mathcal{D})$, is defined such that our choice

of driver nodes is good in some average over the possible choices of control manoeuvre.

$$\begin{aligned} \min_{\mathcal{D} \subset \mathcal{V}} \quad & M(\mathcal{D}) \\ \text{s.t.} \quad & |\mathcal{D}| = m \end{aligned} \quad (12)$$

It has previously been shown (Olshevsky, 2014) that minimising energy metrics are at least NP-hard problems, so rather than attempting to derive an algorithm to find the optimal solution to Equation (12), heuristics and approximation algorithms must be used to return ‘good’ solutions (better than could be expected to be found during an extensive random search). The two choices of energy metric, $M(\mathcal{D})$, investigated here are the control volume and the expectation of the control energy.

The set of all control manoeuvres capable of being performed with E units of energy forms a p -dimensional ellipsoid.

$$\mathcal{S} = \{\boldsymbol{\beta} \in \mathbb{R}^p \mid \boldsymbol{\beta}^T \bar{W}^{-1}(t_f) \boldsymbol{\beta} = E\} \quad (13)$$

The volume of the ellipsoid in Equation (13) is known to be related to the determinant of the matrix $\bar{W}(t_f)$.

$$\log V(\mathcal{S}) = \log \left(\frac{\pi^{p/2}}{\Gamma(p/2 + 1)} \right) + \frac{1}{p} \log \det(\bar{W}(t_f)) \quad (14)$$

The logarithm is taken of the volume as the determinant of the controllability Gramian can fall below the accuracy of double precision floating points values (Sun & Motter, 2013). In Summers et al. (2016), the energy metric in Equation (14) is shown to be a *submodular set function*. The submodular property of Equation (14) has not been directly applied to the target control problem (only the subproblem when $p = n$). Nonetheless, when $p < n$, the submodularity of the log-volume holds (Summers et al., 2016) so we may use a greedy algorithm which retains the same approximation guarantee. For the metric to be minimised, we use the following inverse volume function.

$$\text{Vol}(\mathcal{D}) = -\log \det \bar{W}_{\mathcal{D}}. \quad (15)$$

Decreasing $-\text{Vol}(\mathcal{D})$ means *the set of reachable states, or control manoeuvres, is larger*.

The second energy metric, the expectation of the control energy over initial conditions, assumes that $\mathbf{y}_f = \mathbf{0}_p$ and $\mathbf{x}_0$ is a vector of n independent random variables with mean zero and variance one so that $\mathbb{E}[\mathbf{x}_0 \mathbf{x}_0^T] = \mathbf{I}_n$. The covariance matrix of the control manoeuvre can be written as (G. Li, Hu, et al., 2016)

$$\mathbf{X}_f = \mathbb{E}[\mathbf{e}^{A t_f} \mathbf{x}_0 \mathbf{x}_0^T \mathbf{e}^{A^T t_f}] = \mathbf{e}^{A t_f} \mathbf{e}^{A^T t_f} \quad (16)$$

The expectation of the control energy over the control manoeuvres is the metric to be minimised.

$$\bar{E}(\mathcal{D}) = \text{Tr}(\mathbf{C}^T \bar{W}_{\mathcal{D}}^{-1}(t_f) \mathbf{C} \mathbf{X}_f) \quad (17)$$

The following sections describe heuristics to solve Equation (12) when $M(\mathcal{D}) = \text{Vol}(\mathcal{D})$ and when $M(\mathcal{D}) = \bar{E}(\mathcal{D})$.

2.4 Illustrative example

A directed graph is shown in Figure 1(A) with $n = 20$ nodes and $p = 10$ target nodes highlighted in pink. The goal is to select $m = 5$ driver nodes such that Equation (12) is minimised for either of the two energy metrics considered. This problem is small enough so that a brute force search can be employed. In Figures 1(B) and 1(C), all sets of five nodes are selected as the driver node set successively and both energy metrics are computed. In both cases, the determinant of the output Gramian and the expectation of the control energy span multiple orders of magnitude so choosing an energy efficient set of driver nodes is important. In the following section, we present our heuristic method to find a set of driver nodes that is energy efficient which uses only structural properties of the graph.

3. Graph structure based input selection

It has previously been shown (Klickstein & Sorrentino, 2018a, 2018b) that the optimal cost for the single driver node and single target node problem ($m = 1$ and $p = 1$) is intimately related to the structure of a graph. Two properties were shown to be important, the distance between the driver node and target node, $d_{j,k}$, and the number of nodes that lie along the shortest paths.

Definition 3.1 (Distance): A path of length d is a sequence of d edges, $(v_{\ell_0}, v_{\ell_1}), (v_{\ell_1}, v_{\ell_2}), \dots, (v_{\ell_{d-1}}, v_{\ell_d})$. The distance from node v_j to node v_k is the length of the shortest path such $j = \ell_1$ and $k = \ell_d$.

Definition 3.2 (Redundancy): Let $\mathcal{V}_{j,k}$ be the number of nodes that lie along a shortest path from node v_j to node v_k .

$$\mathcal{V}_{j,k} = \{v_\ell \in \mathcal{V} \mid d_{j,\ell} + d_{\ell,k} = d_{j,k}\}$$

The redundancy between a pair of nodes, $v_j, v_k \in \mathcal{V}$, whose distance apart is $d_{j,k} \geq 2$, is,

$$r_{j,k} = \frac{|\mathcal{V}_{j,k}| - 2}{d_{j,k} - 1} \quad (18)$$

so that if a single path exists between two nodes, then $r_{j,k} = 1$ or if b disjoint paths exist between two nodes then $r_{j,k} = b$.

As mentioned in Introduction, our proposed method to select driver nodes is based on two steps. First, in Section 3.1, we analytically compute the output controllability Gramian for the directed balloon graph, from which we obtain information on the cost to control a target node at distance d from a driver node with b redundant paths. Then, in Section 3.2, we present a method to choose the driver nodes from the solution of the facility location problem with a cost matrix derived from the pair-wise costs found for the directed balloon graph (Figure 2).

3.1 Balloon graph

A model that captures these two properties is the *directed balloon graph* that consists of two end nodes, labelled 0 and d , and b disjoint directed paths from node 0 to node d (Klickstein

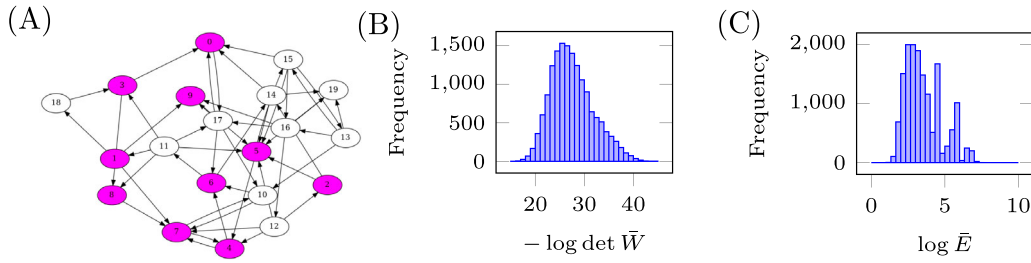


Figure 1. A small example of the input selection problem. (A) A $n = 20$ node directed graph with $p = 10$ target nodes highlighted. (B) The volume cost from Equation (15) binned for all sets of $m = 5$ sets of nodes set as driver nodes. As the determinant of the output controllability Gramian spans over a dozen orders of magnitude, it is clear that choosing a set of driver nodes at random could lead to relatively much worse performance than the optimal solution. (C) The expectation of the control energy in Equation (17) computed for all possible sets of $m = 5$ driver nodes.

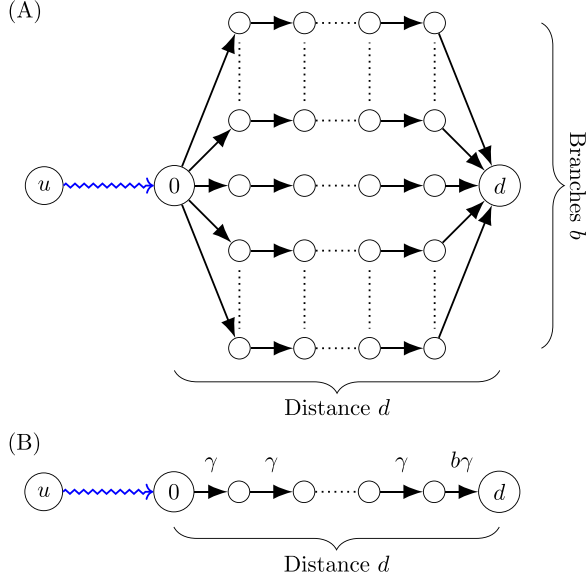


Figure 2. (A) A diagram of the directed balloon graph. There is a single driver node, labelled v_0 , then b parallel, disjoint paths of length d to the terminal node, labelled v_d . All edges have uniform weight γ and loop weight $-\nu$. A directed balloon graph can be completely defined by the two integers, d and b . (B) A diagram of the quotient graph of the balloon graph. This graph is a directed path graph of uniform edge weight γ and uniform loop weight $-\nu$, except for the right-most edge which has weight $b\gamma$.

et al., 2018; Klickstein & Sorrentino, 2018b). A single control input is attached to node 0 while d is the single target node. Each edge is assumed to have uniform weight $\gamma > 0$ and each loop has uniform weight $-\nu < 0$. The driver node 0 and target node d are separated by distance $d_{0,d} = d$ and, from the definition of redundancy in Equation (18), $r_{0,d} = b$. The quotient graph of the balloon graph is a directed path graph with uniform edge weights γ and loop weights $-\nu$ except for the last edge, $(d-1, d)$, which has edge weight $b\gamma$.

The output controllability Gramian when there is a single target node, say $\mathcal{T} = \{d\}$, is only the corresponding diagonal element $W_{d,d}(t)$. To determine the effect that distance and redundancy have on the control energy, this element of the controllability Gramian of the balloon graph's quotient graph is derived analytically.

Theorem 3.3 (Controllability Gramian of the Balloon Graph):
The diagonal element of the controllability Gramian of the balloon

graph corresponding to the node v_d is

$$W_{d,d}(t) = \frac{b^2}{2\nu} \left(\frac{\gamma}{2\nu} \right)^{2d} \binom{2d}{d} \left[1 - e^{-2\nu t} \sum_{k=0}^{2d} \frac{(2\nu t)^k}{k!} \right] \quad (19)$$

For $\nu > 0$, it can be shown that the steady state Gramian's corresponding element is

$$W_{d,d} = \lim_{t \rightarrow \infty} W_{d,d}(t) = \frac{b^2}{2\nu} \left(\frac{\gamma}{2\nu} \right)^{2d} \binom{2d}{d} \quad (20)$$

Proof: Let $W_{j,k}(t)$ be the elements of the controllability Gramian of the quotient graph of the balloon graph (where the superscript Q has been dropped). The elements $W_{j,k}(t)$ for $j, k = 0, 1, \dots, d-1$ satisfy the following system of differential equations:

$$\begin{aligned} \dot{W}_{0,0}(t) &= -2\nu W_{0,0}(t) + 1 \\ \dot{W}_{j,0}(t) &= -2\nu W_{j,0}(t) + \gamma W_{j-1,0}(t), \quad 1 \leq j < d \\ \dot{W}_{0,k}(t) &= -2\nu W_{0,k}(t) + \gamma W_{0,k-1}(t), \quad 1 \leq k < d \\ \dot{W}_{j,k}(t) &= -2\nu W_{j,k}(t) \\ &\quad + \gamma W_{j-1,k}(t) + \gamma W_{j,k-1}(t), \quad 1 \leq j, k < d \end{aligned} \quad (21)$$

From symmetry, $W_{j,0}(t) = W_{0,j}(t)$, so only one set of the boundary elements in Equation (21) must be determined. As every equation in Equation (21) is linear, the Laplace transform is taken of the system where $V(s) = \mathcal{L}\{W(t)\}$.

$$\begin{aligned} sV_{0,0}(s) &= -2\nu V_{0,0}(s) + \frac{1}{s} \\ sV_{j,0}(s) &= -2\nu V_{j,0}(s) + \gamma V_{j-1,0}(s) \\ sV_{0,k}(s) &= -2\nu V_{0,k}(s) + \gamma V_{0,k-1}(s) \\ sV_{j,k}(s) &= -2\nu V_{j,k}(s) + \gamma V_{j-1,k}(s) + \gamma V_{j,k-1}(s) \end{aligned} \quad (22)$$

The origin element, $V_{0,0}(s)$, is determined by rearranging the first line of Equation (22).

$$V_{0,0}(s) = \frac{1}{s(s + 2\nu)}. \quad (23)$$

The remaining elements are determined using a generating function, denoted,

$$\hat{V}(x, y; s) = \sum_{j,k=0,1,\dots} V_{j,k}(s) x^j y^k. \quad (24)$$

Along the $k = 0$ boundary, the elements are determined by setting $y = 0$.

$$\hat{V}(x, 0; s) = \sum_{j=0,1,\dots} V_{j,0}(s) x^j \quad (25)$$

Multiplying the second line of Equation (22) by x^j and summing over all non-negative j yields,

$$\begin{aligned} (s+2v) \sum_{j=0,1,\dots} V_{j+1,0}(s) x^j &= \gamma \sum_{j=0,1,\dots} V_{j,0}(s) x^j \\ \frac{s+2v}{x} (\hat{V}(x, 0; s) - V_{0,0}(s)) &= \gamma \hat{V}(x, 0; s) \\ \hat{V}(x, 0; s) &= \frac{s+2v}{s+2v-\gamma x} V_{0,0}(s) \end{aligned} \quad (26)$$

Define $\rho(s) = \frac{\gamma}{s+2v}$ so that the boundary elements can more succinctly be written as

$$\begin{aligned} \hat{V}(x, 0; s) &= \frac{1}{1-\rho(s)x} V_{0,0}(s) \\ &= V_{0,0}(s) \sum_{j \geq 0} \rho^j(s) x^j, \end{aligned} \quad (27)$$

which implies the boundary elements are

$$V_{j,0}(s) = \frac{1}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^j. \quad (28)$$

In turn, from symmetry, the other boundary must have elements $V_{0,k}(s) = \frac{1}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^k$. The interior elements are found using the two variable generating function.

$$\begin{aligned} \hat{V}(x, y; s) &= \frac{1}{1-\rho(s)(x+y)} V_{0,0}(s) \\ &= V_{0,0}(s) \sum_{\ell \geq 0} \left(\frac{\gamma}{s+2v} \right)^\ell (x+y)^\ell \\ &= V_{0,0}(s) \sum_{\ell \geq 0} \left(\frac{\gamma}{s+2v} \right)^\ell \sum_{a=0}^{\ell} \binom{\ell}{a} x^{\ell-a} y^a \\ &= V_{0,0}(s) \sum_{j,k \geq 0} \binom{j+k}{k} \left(\frac{\gamma}{s+2v} \right)^{j+k} x^j y^k \end{aligned} \quad (29)$$

The interior elements of the controllability Gramian can be read off as the j, k 'th coefficient, $0 \leq j, k < d$,

$$V_{j,k}(s) = \frac{1}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^{j+k} \binom{j+k}{k}. \quad (30)$$

With all elements now determined for $j, k < d$ we turn to elements when one index is equal to d . First, the element $V_{d,0}(s)$ is determined, then the elements $V_{d,j}(s)$ for $1 \leq j < d$, and then finally $V_{d,d}(s)$.

$$V_{d,0}(s) = \frac{b\gamma}{s+2v} V_{d-1,0}(s)$$

$$= \frac{b}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^d \quad (31)$$

Furthermore, it is straightforward to show that

$$V_{d,j}(s) = \frac{b}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^{d+j} \binom{d+j}{d} \quad (32)$$

Finally, the element of interest in the Laplace domain can be computed,

$$\begin{aligned} V_{d,d}(s) &= \frac{b\gamma}{s+2v} (V_{d-1,d}(s) + V_{d,d-1}(s)) \\ &= \frac{2b\gamma}{s+2v} \left(\frac{b}{s(s+2v)} \right) \left(\frac{\gamma}{s+2v} \right)^{2d-1} \binom{2d-1}{d} \\ &= \frac{b^2}{s(s+2v)} \left(\frac{\gamma}{s+2v} \right)^{2d} \binom{2d}{d}. \end{aligned} \quad (33)$$

The inverse Laplace transform of $V_{d,d}(s)$ is found by using identity 5.2.18 in (Bateman, 1954) which states

$$\mathcal{L}^{-1} \left\{ \frac{1}{s(s+a)^n} \right\} = \frac{1}{a^n} \left[1 - e^{-at} \sum_{k=0}^{n-1} \frac{(at)^k}{k!} \right]. \quad (34)$$

Applying Equations (34) to (33) yields the controllability Gramian element,

$$\begin{aligned} W_{d,d}(t) &= \frac{b^2}{2v} \left(\frac{\gamma}{2v} \right)^{2d} \binom{2d}{d} \left[1 - e^{-2vt} \sum_{k=0}^{2d} \frac{(2vt)^k}{k!} \right] \\ &= \frac{b^2}{2v} \left(\frac{\gamma}{2v} \right)^{2d} \frac{(2d)!}{(d!)^2} [1 - r(t)]. \end{aligned} \quad (35)$$

As there is a single target node, the minimum control energy in Equation (10) can be written as

$$J^* = \frac{\beta^2}{2} \frac{1}{W_{d,d}(t)}. \quad (36)$$

Plugging Equation (35) into Equation (36) completes the proof. $\blacksquare$

From Equation (10), the minimum control energy for the balloon graph is $J^* \propto W_{d,d}^{-1}(t_f)$, or if t_f is large enough, then $J^* \propto W_{d,d}^{-1}$. Using Stirling's approximation for the binomial coefficient, the control energy is approximately

$$W_{d,d}^{-1} \approx \sqrt{\pi d} \frac{2v}{b^2} \left(\frac{v}{\gamma} \right)^d. \quad (37)$$

Our structure based metric that approximates the control energy uses this pair-wise energy cost. Given a set of m driver nodes $\mathcal{D}$ and a set of p target nodes $\mathcal{T}$, and pairwise distances $d_{j,k}$ and redundancies $r_{j,k}$ from each node to each target node, we can construct the pair-wise cost matrix $F \in \mathbb{R}^{n \times p}$ which has elements

$$F_{j,k} = \log \sqrt{\pi d_{j,k}} \frac{2v}{r_{j,k}^2} \left(\frac{v}{\gamma} \right)^{d_{j,k}}. \quad (38)$$

Each term, $F_{j,k}$ in Equation (38) can be thought of as the cost of controlling the k 'th target node with node v_j . The structure

based metric then assigns target nodes to driver nodes by selecting which node $v_j \in \mathcal{D}$ can control the k 'th target node the most cheaply (that is, $F_{j,k}$ is minimised over all other possible choices of driver node).

$$\text{FLP}(\mathcal{D}) = \sum_{k=1}^p \min_{v_j \in \mathcal{D}} F_{j,k}. \quad (39)$$

We propose that this structure based metric can be a surrogate function to replace Equation (15) or Equation (17) when trying to determine a set of driver nodes that minimises one of the energy metrics. To test this proposal, we pick random sets of nodes from graphs using a hill climbing procedure to sample the full range of $\text{FLP}(\mathcal{D})$ and compute both $\text{FLP}(\mathcal{D})$ and $\text{Vol}(\mathcal{D})$ (or $\bar{E}(\mathcal{D})$). The relationship between the ellipsoid volume cost in Equation (15) and the FLP cost in Equation (39) is shown for four example graphs in Figure 3. Each graph has $n = 300$ nodes, $p = 100$ target nodes selected randomly, and a driver node set of $m = 33$ nodes to be determined. The four graphs' method of construction is described in the captions of Figure 3. The log volume cost in Equation (14) appears on the vertical axis of each plot while the FLP cost appears on the horizontal axis. From the trends in Figure 3, it is clear that if we were to find a driver node set $\mathcal{D}$ that minimised $\text{FLP}(\mathcal{D})$, that set of driver nodes would also be a competitive solution for the original optimisation problem minimising $\text{Vol}(\mathcal{D})$.

The pair-wise cost in Equation (39) is also shown to correlate with the expectation cost used by LPGM in Equation (17). A demonstration of this relation is shown in Figure 4 for four types of graphs described in the caption. The two costs, $\bar{E}(\mathcal{D})$ and $\text{FLP}(\mathcal{D})$, are positively correlated as shown by the linear fitted line in red. Again, any driver node set $\mathcal{D}$ that minimises $\text{FLP}(\mathcal{D})$ would be a competitive solution for the $\bar{E}(\mathcal{D})$ minimisation problem as well.

Next, we present a method based on the facility location problem (Mirchandani & Francis, 1990) to minimise $\text{FLP}(\mathcal{D})$ so that we may compare the obtained solutions with those generated by published heuristics to optimise Equation (12) with either $M(\mathcal{D}) = \text{Vol}(\mathcal{D})$ or $M(\mathcal{D}) = \bar{E}(\mathcal{D})$.

3.2 Facility location problem

Facility location problems (FLP) originally arose to address the problem of choosing distribution centres to accommodate demands while minimising transportation costs (Mirchandani & Francis, 1990). Let there be p locations that must be supplied from m distribution centres selected from $n \geq m$ possible choices. The cost of supplying the j 'th location from the k 'th distribution centre is denoted $c_{j,k}$. Each location is assumed to be supplied from a single distribution centre. Let the binary variables $Y_j \in \{0, 1\}$, $j = 1, \dots, n$, be the possible distribution centres where $Y_j = 1$ if it is chosen to be a distribution centre and $Y_j = 0$ otherwise. Let the binary variables $Z_{j,k} \in \{0, 1\}$,

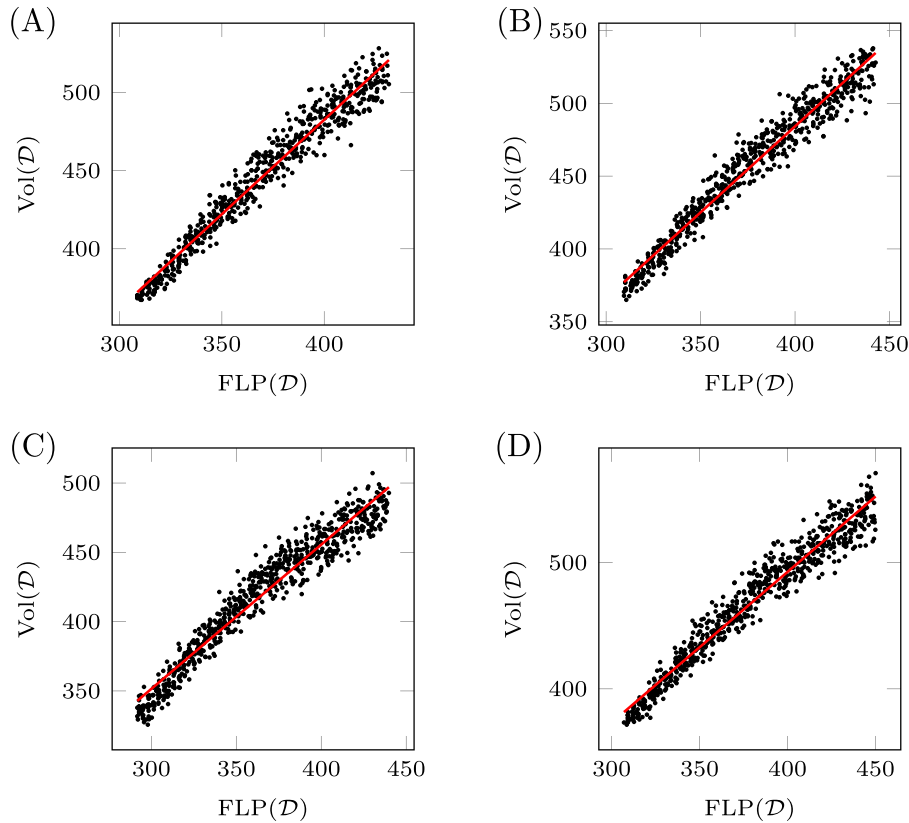


Figure 3. Comparison of the volume based cost, $\text{Vol}(\mathcal{D})$ in Equation (15), and the structure based metric $\text{FLP}(\mathcal{D})$. The graphs used for the analysis are (A) a k -regular graph with $\kappa = 10$, (B) an Erdős–Rényi graph with $\kappa_{av} = 10$, (C) a Watts–Strogatz graph with average degree $\kappa_{av} = 8$ and (D) a graph with a power law degree distribution with exponent $\gamma = 3$ and average degree $\kappa_{av} = 10$ created using the configuration model. All graphs are directed with $n = 300$ nodes. The set of $p = 100$ targets are chosen uniformly at random. The set of $m = 33$ driver nodes are determined using the hill climbing process to achieve a desired value of $\text{FLP}(\mathcal{D})$. The four plots shown here are typical of all graphs examined, directed and undirected.

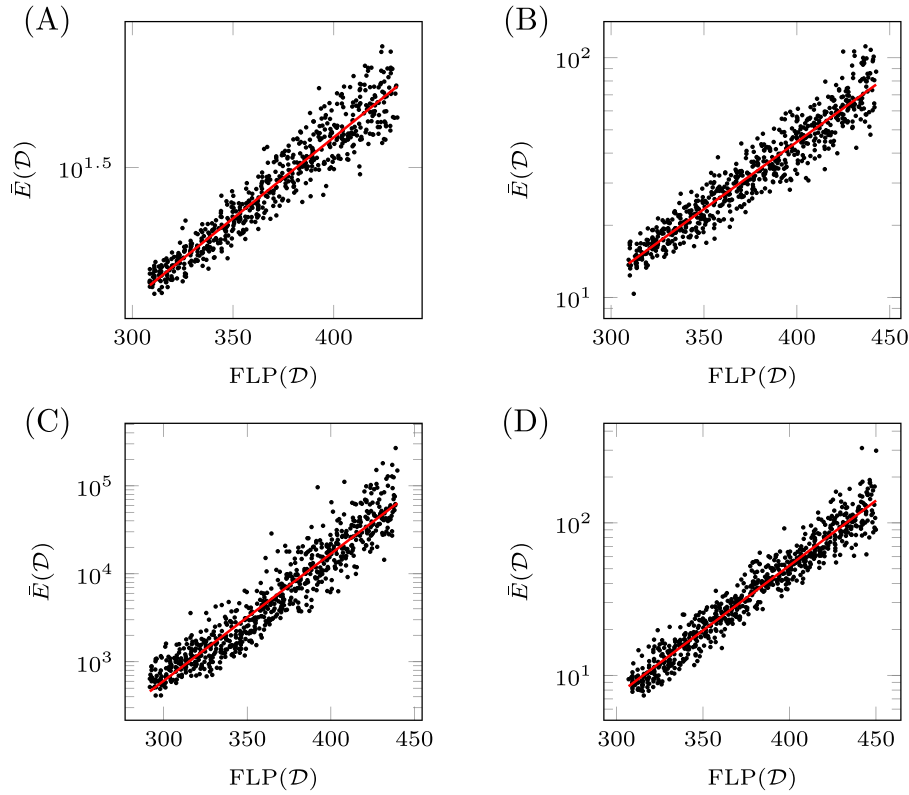


Figure 4. Comparison of the expectation energy cost in Equation (17) and the structure based metric in Equation (39). The graphs used for the analysis are (A) a k -regular graph with $\kappa = 10$, (B) an Erdős–Rényi graph with $\kappa_{av} = 10$, (C) a Watts–Strogatz graph with average degree $\kappa_{av} = 8$ and (D) a graph with a power law degree distribution with exponent $\gamma = 3$ and average degree $\kappa_{av} = 10$ created using the configuration model. Each graph has $n = 300$ nodes, $p = 100$ targets chosen uniformly at random, and $m = 33$ driver nodes chosen the same way as in Figure 3.

$j = 1, \dots, n, k = 1, \dots, p$, denote assignments so that if distribution centre j supplies location k then $Z_{j,k} = 1$ and $Z_{j,k} = 0$ otherwise.

The FLP can be posed as an integer linear programming (ILP) with binary variables.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^n \sum_{k=1}^p Y_j Z_{j,k} c_{j,k} \\
 \text{s.t.} \quad & \sum_{j=1}^n Y_j = m \\
 & \sum_{j=1}^n Z_{j,k} = 1, \quad k = 1, \dots, p \\
 & Z_{j,k} \leq Y_j, \quad j = 1, \dots, n, k = 1, \dots, p
 \end{aligned} \tag{40}$$

The first constraint ensures that precisely m locations are chosen to be distribution centres. The second constraint ensures that each location to be supplied is assigned to a single distribution centre. The third constraint ensures locations to be supplied are only assigned to distribution centres that are opened.

Even large instances ($n \approx 1000$) can be solved efficiently with ILP solvers such as the GNU Linear Programming Kit (Makhorin, 2018). For larger instances of Equation (40), one can use recently developed specialised algorithms to approximately solve the FLP with an approximation guarantee (Jain et al., 2002) efficiently.

4. Alternative methods

4.1 Greedy algorithm

A greedy algorithm that starts with an empty set and at each iteration adds the single node to the driver node set that improves the cost function the most has an approximation guarantee of 63% when the cost function is submodular (Fisher et al., 1978).

By the definition of the matrix B we impose, the matrix product can be decomposed into the individual contributions of each driver node $BB^T = \sum_{k \in \mathcal{D}} \mathbf{e}_k \mathbf{e}_k^T$ where $\mathbf{e}_k$ is the unit vector with the single non-zero element corresponding to each driver node. This decomposition can be used to split the differential Lyapunov equation in Equation (5) into the contribution of each driver node as well.

$$\begin{aligned}
 \dot{W}_k(t) &= A W_k(t) + W_k A^T + \mathbf{e}_k \mathbf{e}_k^T, \quad W_k(0) = O_n \\
 W(t) &= \sum_{k \in \mathcal{D}} W_k(t)
 \end{aligned} \tag{41}$$

A greedy algorithm to minimise $\text{Vol}(\mathcal{D})$ over the powerset of the nodes could be applied directly assuming perfect arithmetic.

The difficulty of applying the greedy algorithm directly arises in two ways. The first difficulty is that storing all potential contributions of each driver node requires np^2 double precision variables which, if p is large, could be prohibitive. The second difficulty is computing $\text{Vol}(\mathcal{D})$ for the first few driver node sets as the Gramian is known to have extremely small (below double precision accuracy) eigenvalues when the number of target nodes is large relative to the number of driver nodes (Klickstein

et al., 2017). A proposed method (Summers et al., 2016) to handle the first few driver nodes replaces the evaluation of $\text{Vol}(\mathcal{D})$ with $-\text{rank}_{\text{num}}(\mathcal{D})$ where the function $\text{rank}_{\text{num}}(\cdot)$ computes the numerical rank of the output Gramian $\bar{W}_{\mathcal{D}}(t_f)$ (Sun & Motter, 2013). This substitute is used until enough driver nodes have been added by the greedy algorithm to ensure the output controllability Gramian is of full numerical rank. Algorithm 1 in Appendix A.1 shows this modified version where a flag is used to perform the switch from computing the rank to the determinant.

4.2 L_0 -constrained projected gradient method

To minimise the expected energy cost in Equation (17), a continuous relaxation step is introduced so that the previous restrictions on B are removed, that is, now $B \in \mathbb{R}^{n \times m}$. The main result in L. Gao et al. (2018) that allows a gradient descent method to be used is the derivative of Equation (17) with respect to B .

$$\frac{\partial E(B)}{\partial B} = -2 \int_0^{t_f} e^{A^T t} C^T \bar{W}_B^{-1}(t_f) C X_f \times C^T \bar{W}_B^{-1}(t_f) C e^{A t} dt B. \quad (42)$$

With information about the gradient, a projected gradient method can be used such that at each iteration the next B matrix

moves in the steepest descent direction until a local minimum is found. A probabilistic projection is used, $\mathcal{P} : \mathbb{R}^{n \times m} \mapsto 2^{\mathcal{V}}$, that finds a set of nodes of cardinality m from a dense matrix, which is the solution returned for the original optimisation problem. Details of the algorithm can be found in Algorithm 2 in Appendix A.2.

5. Comparison

5.1 Comparison with greedy algorithm

To compare the FLP formulation described above and the greedy algorithm, we create 1000 graphs and compute the set of driver nodes returned by the greedy algorithm in Algorithm 1 and by solving the FLP in Equation (40). In Figure 5, 1000 graphs of the following types are used to make the comparison; 5(A) a k -regular graph with $\kappa = 10$, 5(B) an Erdős-Rényi graph with $\kappa_{av} = 10$, 5(C) a Watts-Strogatz graph with average degree $\kappa_{av} = 8$ and 5(D) a graph with a power law degree distribution with exponent $\gamma = 3$ and average degree $\kappa_{av} = 10$ created using the configuration model. Each graph is undirected and is constructed with $n = 50$ nodes and $p = 20$ nodes are chosen randomly to be in the target node set $\mathcal{T}$. We look for a set of $m = 10$ driver nodes, $\mathcal{D}$, such that the cost function in Equation (14) is minimised. The set of driver nodes returned using the FLP

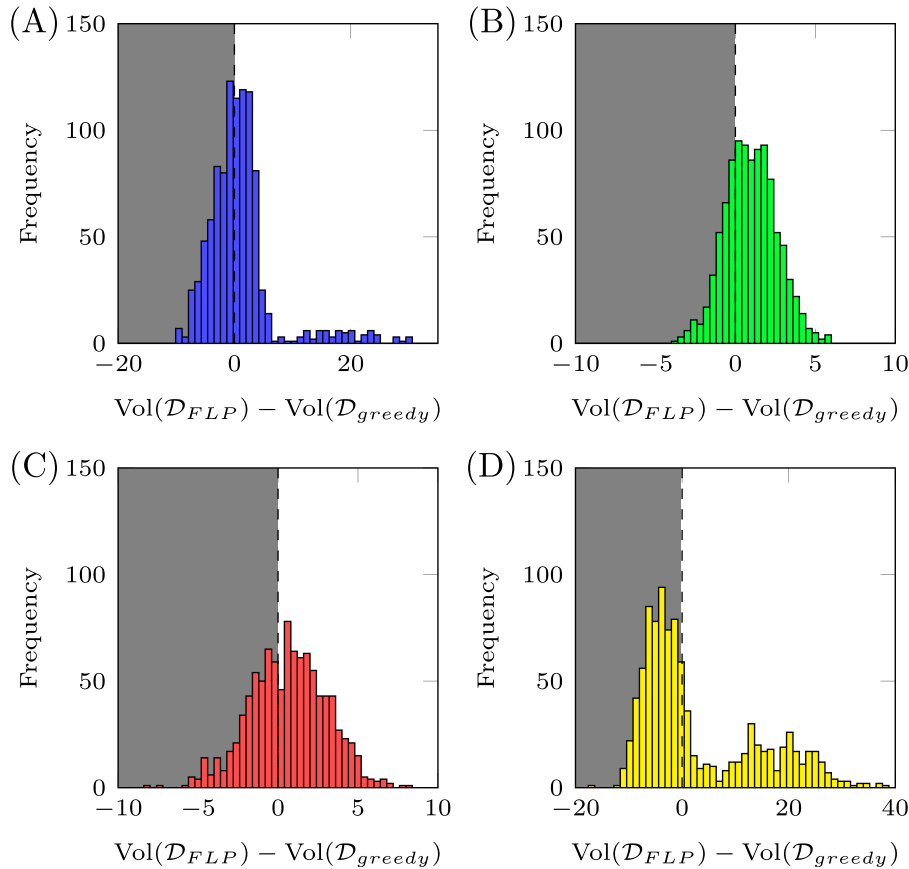


Figure 5. Comparison of the performance of the FLP formulation with the greedy algorithm. Each panel computes the difference between $\text{Vol}(\mathcal{D}_{\text{FLP}})$ and $\text{Vol}(\mathcal{D}_{\text{greedy}})$ defined in Equation (15). The four types of graphs used are (A) Erdős-Rényi graphs with $\kappa_{av} = 6$, (B) k -regular graphs with $\kappa = 5$, (C) Watts-Strogatz graphs (Watts & Strogatz, 1998) with $p = 5\%$, and (D) graphs with a power-law distribution with exponent $\gamma = 3$ and $\kappa_{av} = 6$. Each graph has $n = 50$ nodes, $p = 20$ targets and $m = 10$ driver nodes are selected. All graphs are undirected. Each panel finds the set of driver nodes returned by the FLP formulation and the greedy algorithm and compares the returned cost. The grey background represents cases when the FLP formulation performs better and the white background represents cases when the greedy algorithm performs better.

formulation, denoted $\mathcal{D}_{\text{FLP}}$, and the set of driver nodes returned by the modified greedy algorithm, denoted $\mathcal{D}_{\text{greedy}}$, are found for each graph and their costs are computed. The difference of their costs,

$$D_{\text{greedy}} = \log \det(\bar{W}_{\text{FLP}}^{-1}) - \log \det(\bar{W}_{\text{greedy}}^{-1}), \quad (43)$$

is taken so that if $D_{\text{greedy}} < 0$, $\mathcal{D}_{\text{FLP}}$ is more efficient while if $D > 0$, $\mathcal{D}_{\text{greedy}}$ is more efficient. In Figure 5, the cases when $\mathcal{D}_{\text{FLP}}$ is more energy efficient are shown with a gray background while the cases when $\mathcal{D}_{\text{greedy}}$ is more energy efficient are shown with a white background. We see that for some graph types (panels 5(A) and 5(D)), the FLP method performs better than the greedy algorithm more often, while for other graph types, the greedy algorithm performs better more often. Also, especially for the graphs with a power-law degree distribution in Figure 5(D), the FLP method may not perform well as seen by the second peak in the section of the plot with a white background.

Despite the mixed results in Figure 5, the main benefit is that our approach avoids the difficulty of computing the determinant of an ill-conditioned matrix. Also, in Appendix A.2, we discuss how the greedy algorithm's computational complexity scales as $\mathcal{O}(nmp^3 + n^4)$. To estimate the complexity of the FLP method,

we use the number of nonzero entries in the constraint matrix, which is $(n + 3np)$. This difference is seen in the computation times for the two methods, with the FLP solved considerably faster than the greedy algorithm. This means one can use both the FLP method and the greedy algorithm and take whichever solution returned has a smaller cost without increasing the amount of computational time appreciably while preserving the approximation guarantee of the greedy algorithm.

5.2 Comparison with LPGM

A comparison of the performance of the FLP method and the LPGM heuristic for sets of four types of graphs is shown in Figure 6. As in Figure 5, bars in front of the gray background represent cases where the FLP algorithm returns more energy efficient driver node sets than the LPGM algorithm and vice versa for the bars with a white background. For the four types of graphs examined, we see that neither the FLP method nor the LPGM heuristic perform better than the other, with some slight bias towards one or the other depending on the graph. The benefit of the FLP method is that it scales to larger problems better than the LPGM heuristic and it does not suffer from the same overflow/underflow issues as discussed in Appendix 2.

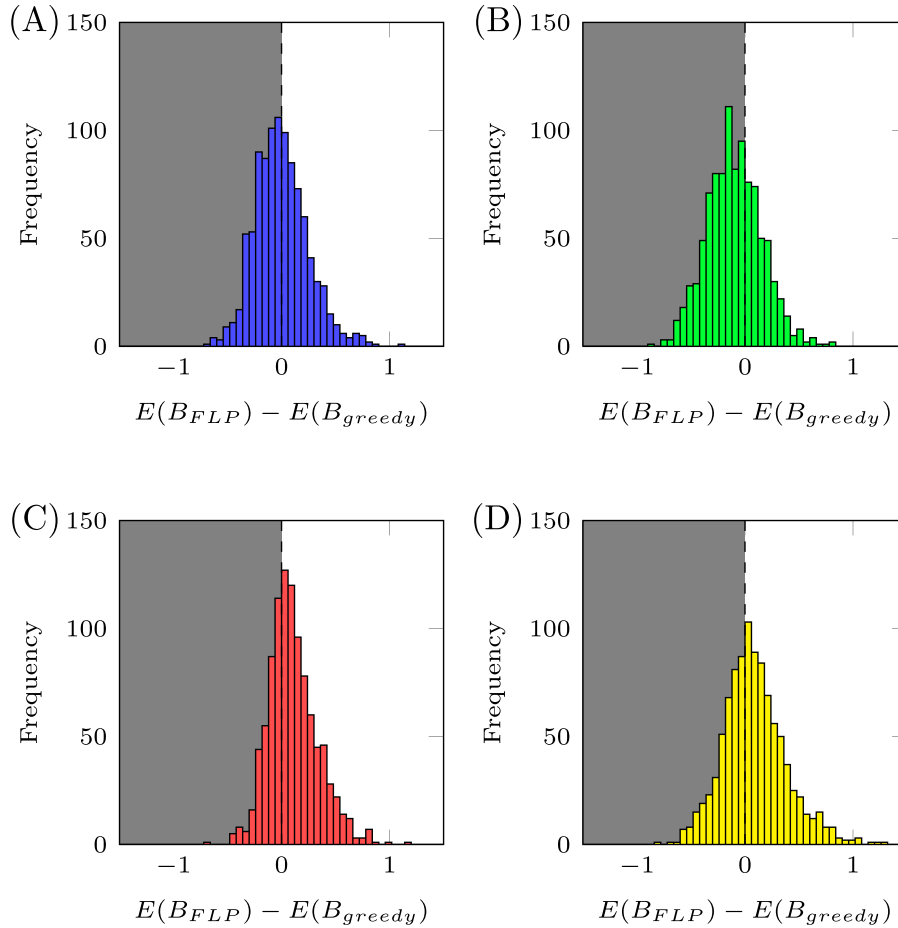


Figure 6. A comparison of the performance between LPGM and FLP methods for the expectation energy cost in Equation (17) for 1000 realisations of the following four types of graphs. The four types of graphs used are (A) Erdős-Rényi graphs with $\kappa_{av} = 6$, (B) k -regular graphs with $\kappa = 5$, (C) Watts-Strogatz graphs (Watts & Strogatz, 1998) with $p = 5\%$, and (D) graphs with a power-law distribution with exponent $\gamma = 3$ and $\kappa_{av} = 6$. All graphs have $n = 50$ nodes, $p = 20$ target nodes selected randomly, and $m = 10$ driver nodes are selected.

6. Conclusion

The energy efficient driver node selection problem is addressed in this paper in a novel way. As it has previously been shown to be NP-hard, while $P \neq NP$, we cannot hope to find the optimal solution but rather we must search for ‘good’ solutions, defined to be a solution better than one which could be reasonably expected to be found during a random search. While previous heuristics developed to find good solutions to this discrete optimisation problem required the repeated calculation of the controllability Gramian, the method we have developed here uses the well known facility location problem with a cost matrix designed using values derived from a simple graph model. The benefits of our method are twofold. The first is the fact that our method can provide better solutions than the previously published methods in some situations. The second is the fact that it is efficient so it can be used in tandem with either of the previous methods without significantly increasing the computational cost.

The method presented here also provides a proof of concept that finding energy efficient sets of driver nodes can be done by using graph structure alone, rather than using properties of the controllability Gramian directly which has been shown to often be ill-conditioned or singular. While the cost matrix we design uses the single target single driver cost, this choice ignores the scaling of the control energy for a single driver with multiple targets. Future work will improve the method presented here by including terms in the cost matrix associated with a single driver assigned to multiple targets.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Science Foundation (NSF) through grant numbers 1727948 and CRISP-1541148.

ORCID

Isaac Klickstein  <http://orcid.org/0000-0002-6830-8154>

Francesco Sorrentino  <http://orcid.org/0000-0001-8899-1176>

References

- Arianos, S., Bompard, E., Carbone, A., & Xue, F. (2009, March). Power grid vulnerability: A complex network approach. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(1), 13119. <https://doi.org/10.1063/1.3077229>
- Barabási, A. L., Albert, R., & Jeong, H. (2000). Scale-free characteristics of random networks: The topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1–4), 69–77. [https://doi.org/10.1016/S0378-4371\(00\)00018-2](https://doi.org/10.1016/S0378-4371(00)00018-2)
- Bartels, R. H., & Stewart, G. W. (1972, September). Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, 15(9), 820–826. <https://doi.org/10.1145/361573.361582>
- Bateman, H. (1954). In A. Erdélyi (Ed.), *Tables of integral transforms* (Vol. I). McGraw-Hill Book Company, Inc.
- Bovet, A., & Makse, H. A. (2019, December). Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, 10(1), 7. <https://doi.org/10.1038/s41467-018-07761-2>
- Commault, C., Van Der Woude, J., & Boukhobza, T. (2017). On the fixed controllable subspace in linear structured systems. *Systems & Control Letters*, 102(3), 42–47. <https://doi.org/10.1016/j.sysconle.2017.01.002>
- Fisher, M. L., Nemhauser, G. L., & Wolsey, L. A. (1978). An analysis of approximations for maximizing submodular set functions–II. In M. L. Balinski, A. J. Hoffman (Eds.), *Polyhedral combinatorics. Mathematical programming studies* (Vol. 8). Berlin, Heidelberg: Springer. <https://doi.org/10.1007/BFb0121195>
- Gao, J., Liu, Y. Y., D’Souza, R. M., & Barabási, A. L. (2014, December). Target control of complex networks. *Nature Communications*, 5(1), 5415. <https://doi.org/10.1038/ncomms6415>
- Gao, L., Zhao, G., Li, G., Deng, L., & Zeng, F. (2018, November). Towards the minimum-cost control of target nodes in directed networks with linear dynamics. *Journal of the Franklin Institute*, 355(16), 8141–8157. <https://doi.org/10.1016/j.jfranklin.2018.08.011>
- Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., & Lazer, D. (2019, January). Fake news on Twitter during the 2016 U.S. presidential election. *Science*, 363(6425), 374–378. <https://doi.org/10.1126/science.aau2706>
- Guo, J., Ji, Z., & Liu, Y. (2021). Sufficient conditions and limitations of equivalent partition in multi-agent controllability. In *Science China Information Sciences*. Science China Press. <https://www.sciengine.com/publisher/scp/journal/SCIS/doi/10.1007/s11432-020-3159-9?slug=full-text>
- Iudice, F. L., Sorrentino, F., & Garofalo, F. (2019). On node controllability and observability in complex dynamical networks. *IEEE Control Systems Letters*, 3(4), 847–852. <https://doi.org/10.1109/LCSYS.7782633>
- Jain, K., Mahdian, M., & Saberi, A. (2002). A new greedy approach for facility location problems. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing – STOC ’02* (p. 731). ACM Press. <http://portal.acm.org/citation.cfm?doid=509907.510012>
- Ji, Z., Lin, H., Cao, S., Qi, Q., & Ma, H. (2020). The complexity in complete graphic characterizations of multiagent controllability. *IEEE Transactions on Cybernetics*, 51(1), 64–76. <https://doi.org/10.1109/TCYB.6221036>
- Kailath, T. (1980). *Linear systems*. Prentice Hall.
- Klickstein, I., Kafle, I., Bartaula, S., & Sorrentino, F. (2018, May). Energy scaling with control distance in complex networks. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–5). IEEE. <https://ieeexplore.ieee.org/abstract/document/8351828>
- Klickstein, I., Shirin, A., & Sorrentino, F. (2017a, April). Energy scaling of targeted optimal control of complex networks. *Nature Communications*, 8(1), 15145. <https://doi.org/10.1038/ncomms15145>
- Klickstein, I., Shirin, A., & Sorrentino, F. (2017b, December). Locally optimal control of complex networks. *Physical Review Letters*, 119(26), 268301. <https://doi.org/10.1103/PhysRevLett.119.268301>
- Klickstein, I., & Sorrentino, F. (2018a, December). Control energy of lattice graphs. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 6132–6138). IEEE. <https://ieeexplore.ieee.org/abstract/document/8619267>
- Klickstein, I. S., & Sorrentino, F. (2018b). Control distance and energy scaling of complex networks. *IEEE Transactions on Network Science and Engineering*, 7(2), 726–736. <https://doi.org/10.1109/TNSE.2018.2887042>
- Lauri, J., & Scapellato, R. (2016). *Topics in graph automorphisms and reconstruction*. Cambridge University Press.
- Li, G., Deng, L., Xiao, G., Tang, P., Wen, C., Hu, W., & Stanley, H. E. (2018). Enabling controlling complex networks with local topological information. *Scientific Reports*, 8(1), 4593. <https://doi.org/10.1038/s41598-018-22655-5>
- Li, G., Ding, J., Wen, C., & Huang, J. (2018). Minimum cost control of directed networks with selectable control inputs. *IEEE Transactions on Cybernetics*, 49(12), 1–10. <https://doi.org/10.1109/TCYB.2018.2868507>
- Li, G., Ding, J., Wen, C., & Pei, J. (2016, September). Optimal control of complex networks based on matrix differentiation. *EPL (Europhysics Letters)*, 115(6), 68005. <https://doi.org/10.1209/0295-5075/115/68005>
- Li, G., Hu, W., Xiao, G., Deng, L., Tang, P., Pei, J., & Shi, L. (2016). Minimum-cost control of complex networks. *New Journal of Physics*, 18(1), 13012. <https://doi.org/10.1088/1367-2630/18/1/013012>
- Li, Y. H., Wang, P. P., Li, X. X., Yu, C. Y., Yang, H., Zhou, J., & Zhu, F. (2016, November). The human kinome targeted by FDA approved multi-target drugs and combination products: A comparative study from the drug-target interaction network perspective. *PLoS ONE*, 11(11), e0165737. <https://doi.org/10.1371/journal.pone.0165737>

- Liu, Y. Y., & Barabási, A. L. (2016). Control principles of complex systems. *Reviews of Modern Physics*, 88(3), 35006. <https://doi.org/10.1103/RevModPhys.88.035006>
- Liu, Y. Y., Slotine, J. J., & Barabási, A. L. (2011, May). Controllability of complex networks. *Nature*, 473(7346), 167–173. <https://doi.org/10.1038/nature10011>
- Lo Iudice, F., Garofalo, F., & Sorrentino, F. (2015, December). Structural permeability of complex networks to control signals. *Nature Communications*, 6(1), 8349. <https://doi.org/10.1038/ncomms9349>
- Makhorin, A. (2018). *GNU linear programming kit*. Free Software Foundation.
- McKay, B. D., & Piperno, A. (2014, January). Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60(3), 94–112. <https://doi.org/10.1016/j.jsc.2013.09.003>
- Mirchandani, P. B., & Francis, R. L. (1990). *Discrete location theory*. John Wiley & Sons, Inc.
- Olshevsky, A. (2014). Minimal controllability problems. *IEEE Transactions on Control of Network Systems*, 1(3), 249–258. <https://doi.org/10.1109/TCNS.2014.2337974>
- Pagani, G. A., & Aiello, M. (2013). The power grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, 392(11), 2688–2700. <https://doi.org/10.1016/j.physa.2013.01.023>
- Qu, J., Ji, Z., & Shi, Y. (2020). The graphical conditions for controllability of multiagent systems under equitable partition. *IEEE Transactions on Cybernetics*, 51(9), 4661–4672. <https://doi.org/10.1109/TCYB.2020.3004851>
- Sporns, O. (2013). Structure and function of complex brain networks. *Dialogues in Clinical Neuroscience*, 15(3), 247–262. doi: 10.31887/DCNS.2013.15.3/osporns. <https://doi.org/10.31887/DCNS.2013.15.3/osporns>
- Summers, T. H., Cortesi, F. L., & Lygeros, J. (2016). On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1), 91–101. <https://doi.org/10.1109/TCNS.2015.2453711>
- Summers, T. H., & Lygeros, J. (2014). Optimal sensor and actuator placement in complex dynamical networks. *IFAC Proceedings Volumes*, 47(3), 3784–3789. <https://doi.org/10.3182/20140824-6-ZA-1003.00226>
- Sun, J., & Motter, A. E. (2013, May). Controllability transition and non-locality in network control. *Physical Review Letters*, 110(20), 208701. <https://doi.org/10.1103/PhysRevLett.110.208701>
- Tzoumas, V., Rahimian, M. A., Pappas, G. J., & Jadbabaie, A. (2015, July). Minimal actuator placement with optimal control constraints. In *2015 American Control Conference (ACC)* (pp. 2081–2086). IEEE.
- Tzoumas, V., Rahimian, M. A., G. J. Pappas, & Jadbabaie, A. (2016). Minimal actuator placement with bounds on control effort. *IEEE Transactions on Control of Network Systems*, 3(1), 67–78. <https://doi.org/10.1109/TCNS.2015.2444031>
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684), 440–442. <https://doi.org/10.1038/39918>
- Yan, G., Ren, J., Lai, Y. C., Lai, C. H., & Li, B. (2012). Controlling complex networks: How much energy is needed? *Physical Review Letters*, 108(21), 218703. <https://doi.org/10.1103/PhysRevLett.108.218703>
- Yan, G., Tsekenis, G., Barzel, B., Slotine, J. J., Liu, Y. Y., & Barabási, A. L. (2015, September). Spectrum of controlling and observing complex networks. *Nature Physics*, 11(9), 779–786. <https://doi.org/10.1038/nphys3422>
- Yuan, Z., Zhao, C., Di, Z., Wang, W. X., & Lai, Y. C. (2013, December). Exact controllability of complex networks. *Nature Communications*, 4(1), 2447. <https://doi.org/10.1038/ncomms3447>
- Zhang, X., Wang, H., & Lv, T. (2017). Efficient target control of complex networks based on preferential matching. *PLoS ONE*, 12(4), e0175375. <https://doi.org/10.1371/journal.pone.0175375>
- Zhao, S., & Pasqualetti, F. (2018, June). Controllability degree of directed line networks: Nodal energy and asymptotic bounds. In *2018 European Control Conference (ECC)* (pp. 1857–1862). IEEE.

Appendices

Appendix 1. Alternative methods

Here we discuss some of the implementation details of the two alternative methods discussed in the text, namely, the greedy algorithm and the L_0 -constrained projected gradient method. In both methods, to compute the controllability Gramian, we use the SLICOT routine SB03TD which is an implementation of the Bartels-Stewart algorithm.

A.1 Greedy algorithm

Let $\mathcal{D}^{(k)}$ be the set of driver nodes after the k 'th greedy step. The first few greedy steps correspond to the situation when only a few driver nodes have been selected so far. If p is even of moderate size, the controllability Gramian for these first few steps will be numerically singular (Summers et al., 2016), or actually singular. To handle this situation, the first few steps make the greedy decision based on which node increases the rank of $\bar{W}_{\mathcal{D}^{(k)}}$ the most until at some step the new controllability Gramian is of full numerical rank. Then the algorithm switches to choosing driver nodes corresponding to which node increases $\text{Vol}(\mathcal{D})$ the most.

To compute the rank of the matrix, a rank revealing QR factorisation is performed using the SLICOT routine MB03OD. The determinant of a symmetric positive definite matrix is found from its Cholesky factor, $\bar{W} = LL^T$. Then, the determinant of $\bar{W}$ is

$$\det \bar{W} = (\det L)^2 = \prod_{j=1}^p L_{jj}^2 \quad (\text{A1})$$

To avoid overflow or underflow issues when taking this product, the logarithm of the determinant is computed instead.

$$\log \det \bar{W} = 2 \sum_{j=1}^p \log L_{jj} \quad (\text{A2})$$

We use the LAPACK routine DPOTRF to compute the Cholesky factor.

We include a flag so that the first iterations use the rank of the output controllability Gramian until at some iteration, the set of driver nodes selected so far ensures the output controllability Gramian has full numerical rank.

A.2 L_0 -constrained projected gradient method

A published algorithm proposed to solve the input selection problem to which we compare the FLP method is the $\mathcal{L}_0$ -constrained projected gradient method (LPGM) (L. Gao et al., 2018). The method combines the projected gradient method (PGM) (G. Li, Ding, et al., 2016; G. Li, Hu, et al., 2016) which assumes all values in the B matrix with a *probabilistic projection*. The probabilistic projection in Algorithm 2 appears as a step, denote $B^{L_0} \leftarrow \mathcal{P}(B)$ in the following gradient descent algorithm in Algorithm 3.

Appendix 2. Computational cost comparison

In the paper, namely Figures 3 and 4, we show that the FLP cost in Equation (39) used in the ILP formulation in Equation (40) can find competitive solutions to both the greedy algorithm with the volumetric cost in Equation (15) and the LPGM heuristic with the expected energy cost in Equation (17). While the FLP formulation does not clearly out-perform either of the other methods in all cases, it does avoid a numerical difficulty faced by both the greedy algorithm and the LPGM heuristic. In the greedy algorithm, we must first compute the output controllability Gramian for each potential driver nodes' contribution, which if every node is a viable candidate, using the Bartels-Stewart algorithm (Bartels & Stewart, 1972, September), requires $\mathcal{O}(n^4)$ work. At each step, k , for $k = 1, 2, \dots, m$, we must compute either the determinant (using a Cholesky decomposition) or the rank (using a rank revealing QR decomposition) for $(n - k + 1) p \times p$

Algorithm 1 Greedy Minimisation of a Set Function

Require: A desired set cardinality m , a state matrix A , a set of target nodes $\mathcal{T}$, a final time t_f (possibly ∞).
for $k = 1, \dots, n$ **do**
 Compute and store $CW_{\{k\}}(t_f)C^T$
end for
Initialise $\mathcal{D} \leftarrow \emptyset$, $\bar{W} \leftarrow O_p$, $k \leftarrow 0$.
flag $\leftarrow 1$
while $k < m$ **do**
 $f_{best} \leftarrow \infty$, $a_{best} \leftarrow -1$
 for $j \in \mathcal{V} \setminus \mathcal{D}$ **do**
 if flag **then**
 $f \leftarrow -\text{rank}(\bar{W} + CW_j(t_f)C^T)$
 else
 $f \leftarrow -\log \det(\bar{W} + CW_j(t_f)C^T)$
 end if
 if $f < f_{best}$ **then**
 $f \leftarrow f_{best}$, $a_{best} \leftarrow j$
 end if
 end for
 $\mathcal{D} \leftarrow \mathcal{D} \cup \{a_{best}\}$
 $\bar{W} \leftarrow \bar{W} + CW_{a_{best}}(t_f)C^T$
 if flag && $-C_{best} == p$ **then**
 flag $\leftarrow 0$
 end if
 $k \leftarrow k + 1$
end while
return $\mathcal{D}$

Algorithm 2 Probabilistic Projection (L. Gao et al., 2018)

Require: $B \in \mathbb{R}^{n \times m}$
Require: $m_0 > 0$
 $r_j \leftarrow \sum_{k=1}^m |B_{j,k}|$, $j = 1, \dots, n$
 $\mathcal{I}_{candidate} \leftarrow \{j \mid r_j \text{ is one of the } m + m_0 \text{ largest values}\}$
 $\mathcal{I}_{selected} \leftarrow \emptyset$
while $|\mathcal{I}_{selected}| < m$ **do**
 $p_j \leftarrow \begin{cases} r_j, & j \in \mathcal{I}_{candidate} \\ 0, & j \notin \mathcal{I}_{candidate} \end{cases}$, $j = 1, \dots, n$
 Choose j according to the probabilities in p .
 $\mathcal{I}_{selected} \leftarrow \mathcal{I}_{selected} \cup \{j\}$
 $\mathcal{I}_{candidate} \leftarrow \mathcal{I}_{candidate} \setminus \{j\}$
end while
 $B^{L0} \leftarrow O_{n \times m}$
for $j \in \mathcal{I}_{selected}$ ($k = 1, \dots, m$) **do**
 $B_{j,k}^{L0} \leftarrow mr_j$
end for
 $B^{L0} \leftarrow B^{L0} / \sum_{j \in \mathcal{I}_{selected}} r_j$
return B^{L0}

output controllability Gramians which both require $\mathcal{O}(p^3)$ work as we perform the comparison between each potential node to add to the set of driver nodes. Thus, the computational complexity of the whole greedy algorithm is $\mathcal{O}(nmp^3 + n^4)$.

The computational complexity of the LPGM heuristic, on its face, is less than the greedy algorithm, but the use of finite precision instead is the main

Algorithm 3 Projected Gradient Descent (L. Gao et al., 2018)

Require: Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with adj. matrix $A \in \mathbb{R}^{n \times n}$
Require: $B_0 \in \mathbb{R}^{n \times m}$
Require: $\mathcal{T} \subseteq \mathcal{V}$ (and corresponding $C \in \{0, 1\}^{p \times n}$)
Require: $\eta > 0$, $t_f > 0$, $K > 1$
 $E_{best} \leftarrow \infty$
 $X_f \leftarrow e^{At_f} e^{A^T t_f}$
for $k = 0, \dots, K$ **do**
 $B_k^{L0} \leftarrow \mathcal{P}(B_k)$
 $W \leftarrow \text{Lyap}(A, B_k^{L0} B_k^{L0^T}, t_f)$
 $\bar{W} \leftarrow CWC^T$
 $E_k \leftarrow \text{Tr}(C^T \bar{W}^{-1} C X_f)$
 $R \leftarrow C^T \bar{W}^{-1} C X_f C^T \bar{W}^{-1} C$
 $W \leftarrow \text{Lyap}(A^T, R, t_f)$
 $\nabla E(B_k^{L0}) \leftarrow -2WB_k^{L0}$
 if $E < E_{best}$ **then**
 $E_{best} \leftarrow E$
 $B_{best}^{L0} \leftarrow B_k^{L0}$
 $B_{k+1} \leftarrow B_k^{L0} - \eta \nabla E(B_k^{L0})$
 else
 $B_{k+1} \leftarrow B_k - \eta \nabla E(B_k^{L0})$
 end if
end for
return $\mathcal{D} \leftarrow \mathcal{I}_{best}$

barrier to applicability. To compute the descent direction, we must solve the following Lyapunov equation:

$$O = A^T Y + Y A + R \quad (\text{A3})$$

for the square matrix Y where

$$R = C^T \bar{W}_B^{-1} C X_f C^T \bar{W}_B^{-1} C \quad (\text{A4})$$

can have extremely large values due to the inverse of the output controllability Gramian appearing twice. As Equation (A3) is a linear equation, it can also be written as $\bar{A} \cdot \text{vec}(Y) = -\text{vec}(R)$ where $\text{vec}(\cdot)$ stacks the columns of a matrix into a vector and $\bar{A} = A^T \otimes I_n + I_n \otimes A$. Let $\|\cdot\|$ be a vector norm, then we know that

$$\|\bar{A} \cdot \text{vec}(Y)\| = \|\text{vec}(R)\| \leq \|\bar{A}\| \cdot \|\text{vec}(Y)\| \quad (\text{A5})$$

where $\|\bar{A}\|$ is on the order of the maximum degree in the graph so that the norm of Y will be on the order of the norm of R , potentially very large and outside the ability of the finite precision used. Handling overflow issues requires care and accuracy is lost. The number of times this must be repeated is difficult to predict as the decay of E_{best} that appears in Algorithm 3 may plateau for many iterations before decreasing (L. Gao et al., 2018).

The FLP formulation as an ILP does not lend itself to an evaluation of the computational complexity directly as it depends strongly on the particular underlying algorithm and its implementation. An alternative metric that often correlates with the computational complexity of solving an ILP is the number of nonzero entries that appear in the constraint matrix. The constraint matrix that appears in our ILP contains $(n + 3np)$ nonzeros, which grows at worst quadratically in n if the number of targets p grows linearly with n , thus it grows more slowly than the greedy algorithm. Our implementation which uses the GNU Linear Programming Kit (Makhorin, 2018) to solve the ILP returns a set of driver nodes faster than the greedy algorithm every time it was compared. As for the LPGM, the FLP formulation does not suffer from overflow or underflow issues during the solution of the Lyapunov equation that appears in Algorithm 3 to which the LPGM heuristic is prone. The FLP method also performed considerably faster than the LPGM heuristic for all comparisons made.