Journal of Process Control 64 (2018) 7–14

Contents lists available at ScienceDirect

Journal of Process Control

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

Network decomposition for distributed control through community detection in input–output bipartite graphs☆

Wentao Tang, Prodromos Daoutidis*

Department of Chemical Engineering and Materials Science, University of Minnesota, MN 55455, USA

ARTICLE INFO

Article history:
Received 18 June 2017
Received in revised form
27 November 2017
Accepted 27 January 2018

Keywords:
Distributed control
Network decomposition
Control architecture design

ABSTRACT

This paper addresses the decomposition of network systems for distributed control. We construct a novel weighted input–output bipartite graph representation of control systems, in which the input–output edge weights capture topological connectivity and short-time response sensitivities. We then introduce community detection as a network-theoretic tool to generate a decomposition with strong intra-subsystem interactions and weak inter-subsystem interactions. A modularity-based graph bisection procedure is applied recursively to determine the optimal decomposition. The proposed method is applied to a chemical process network example.

1. Introduction

Large-scale systems with complex interconnections, including biomolecular reactions [56], electric grids [13], cyber-physical systems [27], and chemical and energy systems [4], typically exhibit a networked structure. In recent years, a large volume of literature has emerged on the control and control-relevant analysis of network systems (see e.g. [3,34,18]). Distributed control plays a central role in the control of such systems [2]. It is based on a decomposition of a large-scale network into constituent subsystems, each controlled by a local controller with some degree of communication between controllers. Distributed control has been pursued in the context of typical network control problems such as consensus and formation control [45,57,44], for which there are well-identified agents to be coordinated. For more general linear network systems under linear control, significant effort has been put on the design of well-posed and stabilizing distributed control laws [10,29], robust stability analysis [1,65] and optimal control [37,30].

A related direction of research of particular importance to process systems is the design of distributed model predictive control (MPC) schemes (see e.g. [50,9,39]). Such controllers retain the inherent advantages of MPC (e.g. flexibility of using different control objective functions, potential of optimizing process economics, and direct handling of constraints) while addressing its key limitation when applied to large-scale nonlinear systems, namely the prohibitive cost of the repeated on-line solution of the underlying dynamic optimization problem. In distributed MPC, the plant is decomposed into constituent subsystems, and the optimal control problem is solved in a distributed manner, with some information sharing between the controllers [31,14,59]. In tightly interconnected plants with an underlying networked structure the decomposition of the system into the distributed architecture is both important and difficult to determine due to the strong and often hidden interactions among the process variables. It is well recognized that the optimal decomposition for nonlinear distributed MPC, i.e. the optimal allocation of the process variables into the distributed control subsystems in order to reduce the computational effort without compromising control performance, is an open and challenging problem [50,9].

Large-scale system decomposition has a long history, especially in the context of decentralized control. Early efforts seek to detect the underlying hierarchical (acyclic) pattern and permute the system into a hierarchy of subsystems, so that the subsystems become controllable in sequence [54,53]. This decomposition is achieved through graph-theoretic search of strongly connected states [35,60] or acyclic input–output reachable systems [46], with sufficiently small elements in the coefficient matrix neglected [51]. Later studies [64,52] considered the situation when the system has a small number of complicating components between otherwise separable subsystems and thus exhibit bordered block diagonal interaction pattern. However, the special graph structures required for these rearrangements are usually restrictive for process systems, due to

☆ This work is supported by NSF-CBET.
* Corresponding author.
E-mail address: daout001@umn.edu (P. Daoutidis).

https://doi.org/10.1016/j.jprocont.2018.01.009
0959-1524/

the complex interactions between the system states and the existence of recycle structures.

On a different vein, several papers considered the design of decentralized control architecture as an optimal graph partitioning problem. In [19], the number of interconnecting edges is to be minimized, with size restrictions in the corresponding subsystems. In [42,43,12], algorithms to partition networks into subnetworks of approximately equal sizes with minimized interconnections were developed. The work in [6] sought to partition the graph with a trade-off among subsystem sizes, interconnections, pairwise distances and information relevance. The work in [24] proposed using a system digraph representation with sensitivity weights with a merging scheme based on LQG regulator metrics. In [62], a computationally expensive genetic algorithm was applied to optimize the effect of decompositions on the control performance for linear systems. Several other papers have considered application-specific decompositions [8,28,38].

From the perspective of network theory, network decomposition can be considered as a *community detection* problem. Community detection aims to divide the vertices of a graph into subsets (communities), such that the links inside communities significantly exceed the links between communities [15,41]. The use of community detection in the context of control-relevant decomposition has only recently been pursued. A hierarchical agglomerative clustering approach has been proposed for the generation of input–output clusters with strong intra-cluster interactions (in an input–output connectivity sense) and weak inter-cluster coupling; agglomerative clustering [21] and divisive clustering approaches have been followed for this purpose [20] to produce candidate structures with different extents of decentralization. Graph theoretic formulations of these methods and a-posteriori assessment of optimality of the resulting clusters using modularity were proposed in [25], and extended to PDE-governed systems [36]. Recently, community detection in the system digraph based on modularity maximization was proposed to generate well-decoupled subsystems containing inputs, states and outputs [23]. This approach is based on pure connectivity considerations.

We note that pure variable connectivity such as the one used in [23] does not account for the intensity of input–output interactions. On the other hand, classical interaction measures based primarily on relative gain array [33,26] do not account for the network topology. A combination of connectivity and response sensitivities (the coefficients of the linearized system), in the form of a relative sensitivity array (RSA), was recently proposed [63] and used in the context of input–output pairing, yet not in the context of network decomposition for distributed control.

In this work, we define a new interaction measure combining connectivity and response sensitivities, which we term *input–output affinity*, and use it as the edge weight in an input–output bipartite graph. We establish that input–output affinities capture the intensity of short-time interactions and hence are suitable measures for decomposing networks for the application of nonlinear model predictive control, where, due to the need for repeated solution of dynamic optimization problems the prediction horizon over which the optimization problem is solved is usually short. To account for the network topology in the decomposition we focus on the problem of input–output partitioning such that intra-subsystem input–output interactions are significantly stronger than inter-subsystem interactions. This is formulated as a community detection problem on the weighted input–output bipartite graph. The method of [5], which allows forming communities from both independent vertex sets (containing the inputs and outputs in our case) of a bipartite graph using modularity maximization, is adopted for the solution of this problem. A chemical process network example is used to illustrate the application of the proposed method.

2. Barber's method for community detection

The problem of community detection in networks has been studied extensively in network science (see [16] for a recent review). Among the numerous algorithms of community detection, modularity-based methods stay as the mainstream, which consider the community detection problem as the maximization of a quality function, called *modularity*, defined for any partition of the nodes of the network. Modularity captures the difference between the number of intra-community edges in the network and its expected value in a randomized counterpart, thus characterizing the statistical significance of the existence of communities in the network [17]. The global optimization of modularity over all possible partitions is an NP-hard problem [7]. Approximate algorithms have been adopted instead to make the modularity maximization problem computationally tractable. Newman's spectral method for unipartite networks [40] performs the community detection by recursive bisections, in which each bisection is determined by the spectral decomposition of a modularity matrix and an additional adjustment (fine-tuning) step. Barber's method [5] is an extension of Newman's spectral method to bipartite graphs, aiming at detecting communities comprising of vertices from both independent vertex sets of such networks. This is the method that will be used in this paper. The procedure is described below.

Consider a bipartite graph in which $U=\{u_i|i=1,\ \ldots,\ p\}$ and $Y=\{y_j|j=1,\ \ldots,\ l\}$ are two independent sets of vertices such that an edge exists between a vertex in U and another vertex in Y. For unweighted graphs, let a_{ij} be the $(i,\ j)$th element of the bipartite adjacency matrix, such that $a_{ij}=1$ if there is an edge between u_i and y_j, and $a_{ij}=0$ otherwise. Let: $\beta(u_i)$ be the degree of u_i, namely the number of edges linking u_i and a vertex in Y; $\beta(y_j)$ be the degree of y_j, namely the number of edges linking a vertex in U and y_j; and β be the total number of edges, i.e.

$$\beta(u_i)=\sum_{j=1}^{l}a_{ij},\quad \beta(y_j)=\sum_{i=1}^{p}a_{ij},\quad \beta=\sum_{i=1}^{p}\sum_{j=1}^{l}a_{ij}. \tag{1}$$

As there are β edges in total, of which $\beta(u_i)$ are incident with u_i, the likelihood for an edge to be incident with u_i is $\beta(u_i)/\beta$. Similarly, the likelihood for an edge to be incident with y_j is $\beta(y_j)/\beta$. The product $\beta(u_i)\beta(y_j)/\beta^2$ can then be viewed as the likelihood for an edge to exist between u_i and y_j, or the expected fraction of edges connecting u_i and y_j. The difference between the fraction of edges between u_i and y_j and its expected value,

$$\begin{aligned} b_{ij} &= \frac{a_{ij}}{\beta}-\frac{\beta(u_i)\beta(y_j)}{\beta^2} \\ &= \frac{a_{ij}}{\sum_{i'=1}^{p}\sum_{j'=1}^{l}a_{i'j'}}-\frac{\sum_{i'=1}^{p}a_{i'j}\sum_{j'=1}^{l}a_{ij'}}{\left(\sum_{i'=1}^{p}\sum_{j'=1}^{l}a_{i'j'}\right)^2} \end{aligned} \tag{2}$$

is a modularity measure for a pair $(u_i,\ y_j)$ that captures the extent that the relation between u_i and y_j is closer than randomly expected. For a weighted graph where a_{ij} can take any real value on a certain interval, b_{ij} can be defined in the same way as above.

"Good" communities comprising of vertices from U and Y are those whose members u_i and y_j have a positive, large b_{ij}. For a partition $\mathcal{P}$ of $U\cup Y$ into disjoint communities C_k, $k=1,\ 2,\ \ldots$, its modularity $Q(\mathcal{P})$ is defined as the sum of all intra-community b_{ij}:

$$Q(\mathcal{P})=\sum_{C_k\in\mathcal{P}}\Big(\sum_{y_j\in C_k}\sum_{u_i\in C_k}b_{ij}\Big). \tag{3}$$