21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Decentralized stabilization, graphs and fat graphs

A. Kirkoryan[1] and M.-A. Belabbas[2]

***Abstract*—A sparse matrix space is a vector space of matrices with entries that are either fixed zeros or arbitrary real. Letting the pattern of zeros and arbitrary entries define an adjacency matrix (by setting the arbitrary entries to one), we can attach a graph to such vector spaces and think of it as describing the decentralization topology of a control system. We want to determine whether this topology can sustain stable dynamics, or equivalently whether the corresponding sparse matrix space contains stable (Hurwitz) matrices. We present in this extended abstract a necessary and sufficient condition for a symmetric sparse matrix space to contain stable matrices.**

I. Introduction

The study of multi-agent systems for which a communication topology is described by a directed graph has been the subject of an increased interest in the past decade [1], [2], [3], [4], [5]. The work presented here is related to this line of work but addresses a different question than the ones usually addressed, namely we ask whether the communication topology can sustain stable dynamics. Precisely, we present new results that address the following issue: to a directed graph, we can associate an adjacency matrix [6] A and, furthermore, to this adjacency matrix A we associate a vector space of matrices as follows: we consider all matrices obtained from A by letting the entries $a_{ij} = 1$ (denoting the presence of a edge between vertices i and j) be arbitrary real, and the entries $a_{ij} = 0$ be fixed at zero. We are interested in the general issue of whether the vector space so-obtained, which we termed a **sparse matrix space** or **zero pattern**, contains stable (Hurwitz) matrices. In terms of graphs, we rephrase the issue as whether the communication topology allows the multi-agent system to stabilize around an equilibrium. This problem was studied in the general case of non-symmetric patterns in [7]. In the present abstract, we focus on the case of **symmetric graphs**, that is directed graphs where every edge has a reciprocal edge (i.e. if (i, j) is an edge, so is (j, i)). Of course, a bidirectional or symmetric topology as described does not require that agents at both ends of a given link have a similar gain applied to their observation. Said otherwise, while the zero-pattern is symmetric, we allow matrices in it to be assymmetric. Precisely, we look at vector spaces of matrices such that

$$a_{ij} = 0 \Leftrightarrow a_{ji} = 0.$$

We give necessary and sufficient conditions to characterize symmetric graphs which allow stable dynamics. We call such graphs **symmetric stable graphs**. Moreover, we show that the conditions provided can be checked in polynomial time.

The approach taken in this work is mostly graph theoretic. In addition to providing insights that are sometimes lacking using an algebraic approach, graph theoretic methods are also well-suited for finding polynomial time algorithms. The use of such methods in control theory is not new, and we mention here the work of [8], [9], [10] among others. Our work and approach differ from these in the sense that while these works are concerned with questions relating to controllability, which can be reduced to a rank condition as is well known [11], we are concerned with stability. Because of this, we need to introduce a new set of tools and we have shown in [7] that the presence of simple cycles in an underlying graph was strongly correlated with its stability properties. We have furthermore given polynomial time algorithms [12], [13] to construct stable graphs recursively.

While the above results dealt with the general case of asymmetric patterns, in many practical situations communication links come with their reciprocal; translated in the language of this paper, such situations correspond to symmetric zero-patterns.

We also introduced in [7] the notion of **minimal stable graphs**: these are stable graphs for which the removal of any edge precludes stabilization. While minimal graphs appear to be hard to characterize in the general case, the other contribution of this paper is to provide a complete characterization of symmetric minimal stable graphs.

The paper is organized as follows: in the following section, we provide the necessary background from graph theory and present some known results for sparse stable systems. In Section III, we provide a necessary and sufficient condition for the stability of a symmetric sparse matrix space and show that the condition can be checked in polynomial time. In Section IV, we describe minimal symmetric graphs and we briefly summarize the results in V.

II. Background for Sparse Matrix Spaces

A. *Sparse matrix spaces and graphs*

We start by introducing some useful vocabulary. Let $n > 0$ be a positive integer, we call a *sparse matrix space* (SMS) a vector space of matrices in $\mathbb{R}^{n\times n}$ with entries either free or zero. Precisely, let α be a set of pairs of integers between 1 and n, that is $\alpha \subset \{1, \ldots, n\} \times \{1, \ldots, n\}$ and denote by E_{ij} the $n \times n$ matrix with zero entries except for the ijth entry, which is one. We define Σ_α to be the vector space of matrices of the form $A = \sum_{(i,j)\in\alpha} a_{ij}E_{ij}$ for some $a_{ij} \in \mathbb{R}$. For example, if

[1]A. Kirkoryan is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign kirkory2@illinois.edu

[2]M.-A. Belabbas is with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign belabbas@illinois.edu

ISBN: 978-90-367-6321-9

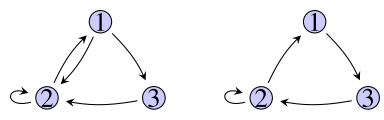


Fig. 1: The graph on the left is stable, whereas the graph on the right is unstable, even though both are strongly connected and have a node with a self-loop. Theorem 1 below allows to decide the stability of these graphs.

$n = 3$ and $\alpha = \{(1, 2), (1, 3), (2, 1), (2, 2), (3, 2)\}$, then Σ_α is the subspace of matrices of the form

$$A = \begin{bmatrix} 0 & * & * \\ * & * & 0 \\ 0 & * & 0 \end{bmatrix} \quad (1)$$

where $*$ are arbitrary real values. We also refer to this space of matrices as a **zero pattern**.

A sparse matrix space Σ_α can be uniquely represented as a graph G with vertex set $V = \{1, 2, \dots, n\}$ and edge set $E = \alpha$. Hence the graphical representation of the SMS (1) is given by the graph in Figure 1-left. We say that a *graph is stable* if its associated SMS contains stable matrices.

Some rather intuitive necessary conditions for stability are easily expressed in the graphical representation. The first is that the graph has a vertex with a self-loop: indeed, a stable matrix necessarily has a negative trace and since self-loops correspond to diagonal elements in the sparse matrix space, at least one free variable on the diagonal is needed. A less obvious necessary condition is that every vertex needs to be *strongly connected*[1] to a vertex with a self-loop [7], a physically intuitive reason being that only vertices with a self-loop can dissipate energy. These conditions are far from sufficient however, as the graph in Figure 1-right illustrates: every vertex is strongly connected to vertex 2, but this graph is not stable. We provide in Section II-C results that allows to decide the stability properties of the two graphs above.

B. Notions from graph theory

We present in this section the necessary background from graph theory. Let $G = (V, E)$ be a directed graph, also called digraph. If there is an edge $e = (v_i, v_j) \in E$ we say that v_i and v_j are adjacent; moreover, we say that e is *incident* to the vertices v_i and v_j. Recall that an edge e which is incident to only one vertex (that is $e = (v_i, v_i)$) is called a *self-loop*.

By convention, we take the vertex set of a graph with n vertices to be $V = \{1, 2, \dots, n\}$. A **cycle** of length k in G is a sequence of vertices $(v_1, v_2, \dots, v_k, v_1)$, such that $(v_i, v_{i+1}) \in E$ for $1 \le i \le k$. A cycle is **simple** (also called a **circuit**) if it does not visit the same vertex twice, that is if $v_i \ne v_j$ for $1 \le i \ne j \le k$. A vertex with a self-loop is a simple cycle of length 1. We say that a set of cycles *covers* G if every vertex of G appears in at least one cycle and we say that two cycles are *disjoint* if they do not have any vertices in common. We call a **Hamiltonian decomposition** of G a set of pairwise disjoint simple cycles that covers G. We call a **k-decomposition** of G a set of disjoint simple cycles in G whose union covers k vertices. Hence a n-decomposition is a Hamiltonian decomposition. We use the notation $(v_1 v_2 \dots v_k)$ to refer to the simple cycle $(v_1, v_2, \dots, v_k, v_{k+1} = v_1)$ and write a k-decomposition as the product of its constituent cycles. For example $(12)(3)$ refers to the 3-decomposition containing the cycle $(1, 2, 1)$ and the loop $(3, 3)$. We illustrate some of these notions in Figure 2.

[1]We say that two nodes v_1 and v_2 are strongly connected if there exists a path from v_1 to v_2 and from v_2 to v_1.

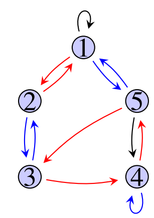


Fig. 2: The graph depicted above admits several Hamiltonian decompositions: one into the cycles (12) and (345), one into the cycles $(15), (23), (4)$ and one into the cycle (12345). The cycle (1) is a 1-decomposition and the cycle $(23)(15)$ is a 4-decomposition of G.

C. Known results on stable graphs

We now briefly recall results that have been established regarding the stability of sparse systems, we refer the reader to [7] for a detailed exposition. We call a sequence of k-decompositions for $1 \le k \le n$ *nested* if the i-decomposition covers all vertices covered by the $i-1$-decomposition plus one additional vertex. Observe that the edges used in each decompositions need not be the same. For example, the decompositions $(2), (12), (123)$ for the graph in Figure 1 left are nested.

Theorem 1 ([7])**.** *The following holds:*

1) *If a graph G is stable, then it contains at least one k-decompositions of every dimension and also every node of the graph is strongly connected to at least one self-loop.*
2) *If a graph G contains a sequence of nested k-decompositions for $k = 1, \dots, n$ then the associated sparse matrix space is stable.*

Using this Theorem, we can conclude that the graph in Figure 1 left is stable (using the second condition) and that the graph to the right is not (it is missing a 2-decomposition).

III. New Results on Symmetric Sparse Matrix Spaces

In this section, we present the main result of the paper: a symmetric sparse system is stable if and only if its associated graph contains a Hamiltonian decomposition and each of its nodes is connected to a self-loop.

Contrary to the proof of our earlier results in [13], which relied on perturbation of polynomials to prove stability and graph theoretic methods to provide polynomial time algorithms, the proof presented below is purely graph theoretic.

The idea is to show that the graph associated to a symmetric system always contains as a subgraph a type of planar graph we termed fat trees. The properties of such graphs, whose definition we give below, allow in particular to define an ordering of the nodes in the graph which in turn can be used to exhibit a nested k-decomposition, hence proving stability using Theorem 1.

A. Conditions for stability

We recall that a symmetric zero pattern does not require the values in symmetric entries to be the same. In fact, it is easy to see that if we restrict the matrices of a symmetric sparse matrix space to be themselves symmetric, then there cannot be any zero entries on the diagonal. This follows immediately from Sylvester's law of inertia [14]. We thus only consider the case of a symmetric zero-patterns with the understanding that the values in symmetric positions need not be the same.

We show that the necessary and sufficient conditions for stability of Theorem 1 coincide for symmetric patterns:

Theorem 2. *Let G be a graph corresponding to a symmetric sparse matrix space. Then G is stable* if and only if

1) *Every node in G is strongly connected to a self-loop.*
2) *The graph G contains a Hamiltonian decomposition.*

We make two remarks. First, the notion of strong connectedness is redundant in the symmetric case - indeed, if there exists a path from vertex u to vertex v in the graph, then there exists a path going in the other direction by symmetry.

Second, notice that the conditions in Theorem 2 are weaker than the necessary conditions of Theorem 1—namely, we do not require the existence of k-decompositions for every $1 \leq k \leq n$. We will show, however, that the conditions of Theorem 2 imply the sufficient conditions of Theorem 1, that is, we will prove the following result:

Proposition 1. *If the symmetric graph G contains a Hamiltonian decomposition, and if every node in G is connected to a self-loop, then G admits nested k-decompositions for $1 \leq k \leq n$.*

It is clear that Theorem 2 is a direct consequence of Proposition 1. We illustrate the result to the symmetric graphs shown below.

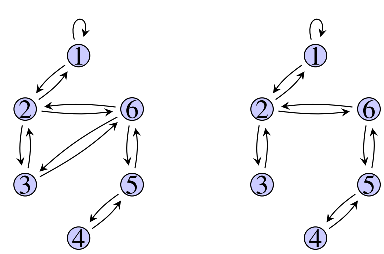


Fig. 3: Even though both graphs are connected and contain self-loops, the first one admits Hamiltonian decomposition - $(1), (236), (45)$, whereas the second one does not. Therefore the graph on the left is stable and the graph on the right is unstable.

B. Corollaries

Using Theorem 1, the algorithm for finding maximum Hamiltonian decompositions proposed in [12] and Dijkstra's algorithm for finding shortest paths in a directed graph [6] we obtain the following result:

Corollary 1. *There exists a polynomial time algorithm to determine whether given symmetric SMS is stable.*

Using the same algorithm from [12], we furthermore give a method for stabilizing a symmetric SMS by adding minimum number of self-loops:

Corollary 2. *There exists a polynomial time algorithm for finding the minimum number of self-loops needed to stabilize a symmetric SMS.*

IV. Minimal Stable Symmetric Sparse Matrix Spaces

We now focus our attention to minimal stable symmetric spaces, that is symmetric spaces such that changing any symmetric pair of non-zero entries (or diagonal entry) to a zero entry yields a non-stable SMS. Said otherwise, the minimal stable symmetric SMS are the minimal elements of the set of all stable symmetric SMS with respect to the subspace inclusion relation. We will extensively use below Theorem 2. We start by introducing some vocabulary.

A. Fat trees

We call a graph T **fat tree** if it can be constructed in the following manner:

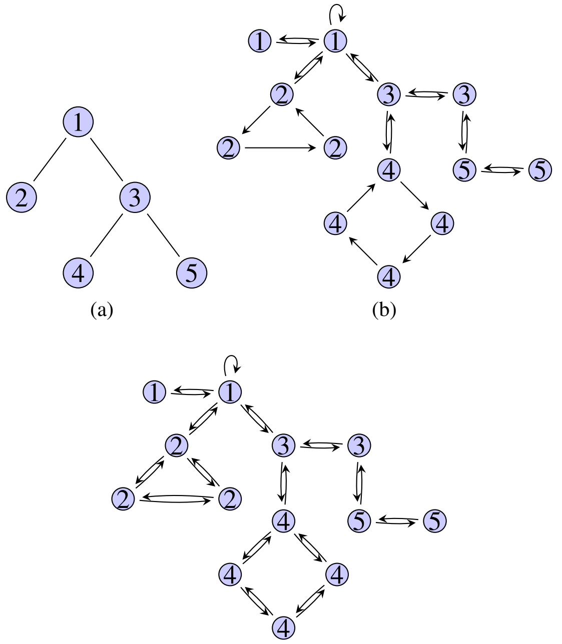
1) Start with an undirected tree T_0 with root r.
2) Replace the root r either with a 1-cycle or with a (directed) simple cycle with a larger number of nodes and with a self-loop attached to one of its nodes. We will call it **fat root** and denote it with H_r.
   Replace every node v different from r with a (directed) simple cycle H_v of size at least 2.
   Replace every edge (u, v) in T_0 with a pair of opposite (directed) edges connecting some vertex of H_u with a vertex of H_v.
3) Add all necessary edges to make the constructed directed graph symmetric.

We call the tree T_0 **skeleton** of the graph T, the simple cycles H_v along with the fat root **fat nodes**, and the edges connecting them — **connecting edges**.

We give an example of a fat tree T along with its construction steps on Fig. 4.

Since the definition we have given for a fat tree G is through construction, a natural question arises: is it possible to construct the same graph G in two different ways starting from the same or differenent skeletons? The answer to this question is negative and is given by the next Proposition 2.

Proposition 2. *There is a unique way to construct any fat tree G, i.e. if we are given a fat tree G, then its skeleton, fat nodes (up to orientation) and connecting edges can be determined uniquely.*



(a) (b) (c)

Fig. 4: The first graph represents the initial undirected tree - T_0 (the skeleton). Its nodes are labeled with the numbers from 1 to 5, where 1 denotes the root. On the second graph we see the replacement of the nodes and the edges in T_0 by fat nodes and connecting edges corresponding to them. The third graph represents the final fat tree T.

We are now ready to specify a characterization of all minimal stable symmetric SMS. We present the condition in terms of the associated graph. Clearly, a (symmetric) graph is minimal stable if and only if each of its connected components is minimal stable. The next theorem gives us easy to check necessary and sufficient conditions for verifying this.

Theorem 3. *A connected symmetric graph G is minimal stable if and only if the following conditions hold:*

1) *G is a fat tree with fat nodes H_1, H_2, H_3, ... , H_m, such that every $H_i, i \neq 1$ is either a 2-cycle or an odd-cycle. The fat root H_1 is either a 1-cycle or a 2-cycle with a self-loop incident to one of its nodes.*
2) *There does not exist a path $(v_1, ..., v_{2l+2})$ in G such that:*
   - the vertex v_1 belongs to some odd-cycle $H_i, i \neq 1$;
   - the vertex v_{2l+2} belongs either to some odd-cycle $H_j, j \neq 1$ or to H_1. In the latter case, if H_1 is a 2-cycle, v_{2l+2} is not incident with e;
   - the edges (v_{2k}, v_{2k+1}) are the vertices of some 2-cycle H_i for every $1 \leq k \leq l$ and the edges (v_{2k-1}, v_{2k}) are connecting edges for every $1 \leq k \leq l+1$.

Furthermore, every connected symmetric minimal stable graph has a unique Hamiltonian decomposition (up to orientation of its cycles) and this decomposition represents the fat nodes of the graph's fat tree structure.

We call the paths described in condition 2 in the theorem **violating paths**.

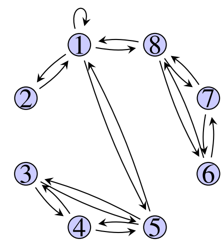


Fig. 5: Using Theorem 3, we can verify that the symmetric graph given above is minimal stable. First we find a Hamiltonian decomposition, given by the cycles $(1, 2)$, $(3, 4, 5)$ and $(6, 7, 8)$. The cycle $(1, 2)$ along with the self-loop attached to it represents the fat root and the other two 3-cycles represent the remaining fat nodes. The connecting edges are represented by the pairs $(1, 5), (5, 1)$ and $(1, 8), (8, 1)$. Finally, it is easy to see that a violating path does not exist. Therefore the symmetric graph is minimal stable.

V. Summary

Understanding which communication topologies allow for stabilization of a multi-agent systems requires the understanding of which sparse matrix space contains stable matrices. We have outlined in this extended abstract a series of results that completely characterize symmetric sparse stable matrices. We have used these results to furthermore characterize the minimal ones. All conditions presented can be checked in polynomial time, we provide more detail [15].

References

[1] J. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. Wiley, 2008.

[2] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

[3] Ali Jadbabaie, Jie Lin, and A. Stephen Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[4] B. Gharesifard and J. Cortes, "Distributed strategies for generating weight-balanced and doubly-stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539–557, 2012.

[5] M. Cao, B. Anderson, A. Morse, Yu, and C., "Control of acyclic formations of mobile autonomous agents," *Proc. of the 47th IEEE Conference on Decision and Control*, pp. 1187–1192, 2008.

[6] R. Diestel, *Graph Theory*. Springer, 2010.

[7] M.-A. Belabbas, "Sparse stable systems," *Systems & Control Letters*, vol. 62, pp. 981–987, 2013.

[8] K. J. Reinschke, "Multivariable control: A graph-theoretic approach," in *Lecture Notes in Control and Information Sciences*. Springer, 1988.

[9] C.-T. Lin, "Structural controllability," *Automatic Control, IEEE Transactions on*, vol. 19, no. 3, pp. 201–208, 1974.

[10] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[11] R. Brockett, *Finite Dimensional Linear Systems*. Addison-Wesley, 1970.

[12] M.-A. Belabbas, "Algorithms for sparse stable systems," in *Proc. of the 52nd IEEE Conference on Decision and Control*, 2013.

[13] A. Kirkoryan and M.-A. Belabbas, "Graph theory and the design of sparse distributed systems," *Submitted for publication*.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 1990.

[15] A. Kyrkorian and M.-A. Belabbas, "Symmetric sparse systems and fat graphs," *In preparation*, 2014.