Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Electronic Notes in Discrete Mathematics 36 (2010) 1073–1080

Electronic Notes in DISCRETE MATHEMATICS

www.elsevier.com/locate/endm

Structural Analysis for Differential-Algebraic Systems : Complexity, Formulation and Facets

Mathieu Lacroix[1], A. Ridha Mahjoub[1], Sébastien Martin [1]

Université Paris-Dauphine, LAMSADE
Paris, France

Abstract

In this paper we consider the structural analysis problem for differential-algebraic systems with conditional equations. This consists, given a conditional differential algebraic system, in verifying if the system is well-constrained for every state and if not in finding a state for which the system is bad-constrained. We first show that the problem reduces to the perfect matching free subgraph problem in a bipartite graph. We then show the NP-completeness of this problem and give a formulation as an integer linear program. We also discuss the polytope of the solutions of this problem and propose a Branch-and-Cut algorithm.

Keywords: Differential algebraic system, structural analysis, bipartite graph, matching, polytope, facet.

1 Introduction

Differential-algebraic systems (DAS) are used for modeling complex physical systems such as electrical networks or dynamic movements. Such a system can be given as $F(z, \dot{z}, t) = 0$, where z is the variable vector, t is time and

[1] Email: {lacroix,mahjoub,martin}@lamsade.dauphine.fr

doi:10.1016/j.endm.2010.05.136

$\dot{z}$ is the partial derivative of z with respect to time. A DAS is *structurally solvable* if it admits a unique solution [2]. A necessary (but not sufficient) condition for a DAS to be structurally solvable is that there are as many equations as variables, and there exists a mapping between the equations and the variables in such a way that each equation is related to only one variable and each variable is related to only one equation. If this is satisfied, then we say that the system is *well-constrained*. Otherwise, the system is said to be *bad-constrained*. A bad-constrained system is also refered as *structurally singular system*. Object-oriented modeling langages like Modelica [1] enforce this as simulation is not possible if the system is bad constrained. The *structural analysis problem* (SAP) of a DAS consists in checking whether or not the system is bad-constrained.

The structural analysis problem for DASs has been proved to be polynomial-time solvable by Murota [2]. Given a DAS, one can associate a bipartite graph $G = (U \cup V, E)$, called *incidence graph*, where U corresponds to the equations, V to the variables, and there is an edge $uv \in E$ between a node $u \in U$ and a node $v \in V$ if the variable corresponding to v appears in the equation corresponding to u. Murota [2] proved that the DAS is bad constrained if and only if its incidence graph does not contain a perfect matching.

In many practical situations, the form of an equation of a DAS, especially the variables that appear in it, may depend of some conditions such as temperature changements in hydraulic systems. Such equation is called *conditional*. Therefore, from a conditional equation, we can obtain different (non-conditional) equations with respect to the values of the conditions associated with it. A DAS containing conditional equations is called *conditional DAS*. An assignment of the values *true* and *false* to the conditions of a DAS will be called a *state* of the system. Hence each state yields a non-conditional DAS and, therefore, verifying if a conditional DAS is well constrained reduces to verifying whether for any state, the incidence graph of the corresponding DAS contained a perfect matching.

A first and preliminary study of SAP with conditional equations is given in [3]. In this paper we show the NP-completeness of the problem, introduce a new and stronger model and discuss the associated polytope.

We consider conditional DASs such that every conditional equation has exactly two different forms, depending on the true/false value of one condition associated with this equation. Morevoer, we suppose that all the conditions are independant.

The paper is organized as follows. In the following section we give a formulation of the SAP for conditional DAS in terms of matching in bipartite graphs. In section 3 we show the NP-completeness of the problem. In section 4, we give a model for the problem as an integer linear program and discuss the associated polytope. Some concluding remarks are given in section 5.

2 Graph representation and matchings

In this section, we shall discuss a graph based model for the problem. Consider a conditional DAS with n conditional equations, say $eq_1, \ldots, eq_n$, n' non-conditional equations $eq_{n+1}, \ldots, eq_{n+n'}$ and $n+n'$ variables, say $z_1, \ldots, z_{n+n'}$. With this system, we associate a bipartite graph $G = (U \cup V, E)$ where $U = \{u_1, ..., u_{n+n'}\}$ (resp. $V = \{v_1, ..., v_{n+n'}\}$) is associated with the equations (resp. variables). For $i = 1, \ldots, n$, we consider an edge $\{u_i, v_j\}$ between a vertex $u_i \in U$ and a vertex $v_j \in V$, called *true edge* (resp. *false edge*, *true/false edge*), if the variable z_j appears in equation eq_i, when the condition of equation eq_i is supposed true (resp. false, both true and false). For $i = 1, \ldots, n'$, we associate an edge $\{u_{n+i}, v_j\}$ between a vertex $u_{n+i} \in U$ and a vertex $v_j \in V$ if the variable z_j appears in the non-conditional equation eq_{n+i}. These edges are called *non-conditional edges*. Let E^{tf} be the set of the true/false and non-conditional edges in G. For $i = 1, \ldots, n$, let E_i^t (resp. E_i^f) be the set of true (resp. false) edges incident to u_i which are not true/false. Hence, the sets E_i^t, E_i^f, $i = 1, \ldots, n$ and E^{tf} are disjoint, and we have $E = \bigcup_{i=1,\ldots,n}(E_i^t \cup E_i^f) \cup E^{tf}$. We set $\mathcal{E} = \{E_i^t, E_i^f : i = 1, \ldots, n\}$.

Now, the SAP reduces to finding a perfect matching free subgraph $G' = (U \cup V, E')$ of G. Such that :
- $E^{tf} \subseteq E'$
- at most one edge set among E_i^t, E_i^f is contained in E', and
- the number of edge sets of $\mathcal{E}$ in E' is maximum.

Clearly the number of edge sets of $\mathcal{E}$ in G' is at most n. If this number is equal to n, this means that we have found a state which yields a bad-constrained system, and hence our conditional DAS is also bad-constrained. Otherwise there exists a perfect matching in the incidence graph for any state of the system and, thus, the system is well-constrained. We will refer to the problem of determining the graph G' as the *free perfect matching subgraph problem (FPMSP).*

3 NP-completeness of the FPMSP

In order to show that the FPMSP is NP-complete, we are going to show first that a related problem is NP-complete.

Let $G = (V^1 \cup V^2 \cup V^3, E)$ be a tripartite graph, where $|V^1| = |V^2| = |V^3| = k$, $V^z = \{v_1^z, ..., v_k^z\}, z = 1, 2, 3$, and such that the sets V^1 and V^2 are connected by a perfect matching. Consider the following problem : Does there exist a stable set in G of size $k + 1$? We will call this problem the *tripartite stable set problem with perfect matching (TSSPPM).* In what that follows, we will show that this problem is NP-complete. Afterwards, we will prove that FPMSP equivalent to TSSPPM.

Theorem 3.1 *The TSSPPM is NP-complete.*

Proof. It is easy to see that $TSSPPM \in NP$. In what follows, we will show that the one-in-three 3SAT reduces to TSSPPM. Let $L = \{l_1, l_2, ..., l_n\}$ be a set of variables and $C = \{c_1, c_2, ..., c_m\}$ a set of clauses, where $|c_j| = 3$ for $j = 1, ..., m$. The *one-in-three 3SAT* problem consists in finding a truth assignment for L such that each clause in C has exactly one true literal. We shall construct a tripartite graph $G = (V^1 \cup V^2 \cup V^3, E)$, where $|V^1| = |V^2| = |V^3| = k$, with a perfect matching between V^1 and V^2. We will show that G has a stable set of size $k + 1$ if and only if the one-in-three 3SAT problem admits a truth assignment. Let x_i be a variable which represents either the literal l_i or the literal $\bar{l}_i$. If $x_i = l_i$ (resp. $x_i = \bar{l}_i$), then $\overline{x}_i = \bar{l}_i$ (resp. $\overline{x}_i = l_i$).

For each variable $l_i \in L$, we associate the nodes $v_i^1,\ \overline{v}_i^1 \in V^1,\ v_i^2,\ \overline{v}_i^2 \in V^2$and $v_i^3,\ \overline{v}_i^3 \in V^3$. These are called *variable nodes.* For each clause $c_j = \{x_r,\ x_s,\ x_t\}$, we associate the nodes $w_{j,r}^1 \in V^1,\ w_{j,s}^2 \in V^2,\ w_{j,t}^3 \in V^3$. These are called *clause nodes.* Finally, we add the nodes $f_q^1 \in V^1,\ f_q^2 \in V^2,\ f_q^3 \in V^3$, for $q = 1, ..., n-1$. These are called *fictitious nodes.*

For each variable $l_i \in L$, we consider the edges $(v_i^1,\ \overline{v}_i^2)$, $(\overline{v}_i^2,\ v_i^3)$, $(v_i^3,\ \overline{v}_i^1)$, $(\overline{v}_i^1,\ v_i^2)$, $(v_i^2,\ \overline{v}_i^3)$, $(\overline{v}_i^3,\ v_i^1)$ in E. These are called *variable edges.* Note that these edges form a cycle, which we will denote by Γ_i, for $i = 1, ..., n$. For each clause $c_j = \{x_r,\ x_s,\ x_t\}$, we add in E the edges $(w_{j,r}^1,\ w_{j,s}^2)$, $(w_{j,s}^2,\ w_{j,t}^3)$, $(w_{j,t}^3,\ w_{j,r}^1)$. These are called *clause edges.* Note that these edges form a triangle, which we will denote by T_j, for $j = 1, ..., m$. For $q = 1, ..., n-1$, we add in E the edge (f_q^1, f_q^2). We remark that we have a perfect matching between V^1 and V^2 given by the edges $(v_i^1,\ \overline{v}_i^2)$, $(\overline{v}_i^1,\ v_i^2)$, $i = 1, ..., n$, $(w_{j,r}^1,\ w_{j,s}^2)$, $j = 1, ..., m$, and (f_q^1, f_q^2), $q = 1, ..., n-1$. Now according to the values of the literals, we add edges in E as follows. For every clause $\{x_r,\ x_s,\ x_t\}$

- if $x_r = l_r$, add the edges $(w_{j,r}^1, \overline{v}_r^3)$, $(w_{j,s}^2, v_r^3)$, $(w_{j,t}^3, v_r^1)$, $(w_{j,t}^3, v_r^2)$,
- if $x_r = \overline{l_r}$, add the edges $(w_{j,r}^1, v_r^3)$, $(w_{j,s}^2, \overline{v}_r^3)$, $(w_{j,t}^3, \overline{v}_r^1)$, $(w_{j,t}^3, \overline{v}_r^2)$,
- if $x_s = l_s$, add the edges $(w_{j,r}^1, v_s^3)$, $(w_{j,s}^2, \overline{v}_s^3)$, $(w_{j,t}^3, v_s^1)$, $(w_{j,t}^3, v_s^2)$,
- if $x_s = \overline{l_s}$, add the edges $(w_{j,r}^1, \overline{v}_s^3)$, $(w_{j,s}^2, v_s^3)$, $(w_{j,t}^3, \overline{v}_s^1)$, $(w_{j,t}^3, \overline{v}_s^2)$,

- if $x_t = l_t$, add the edges $(w^1_{j,r}, v^3_t)$, $(w^2_{j,s}, v^3_t)$, $(w^3_{j,t}, \overline{v}^1_t)$, $(w^3_{j,t}, \overline{v}^2_t)$,
- if $x_t = \overline{l_t}$, add the edges $(w^1_{j,r}, \overline{v}^3_t)$, $(w^2_{j,s}, \overline{v}^3_t)$, $(w^3_{j,t}, v^1_t)$, $(w^3_{j,t}, v^2_t)$,

These are called *satisfiability edges*. For each fictitious node in $V^1 \cup V^2$, we add edges to connect all nodes in V^3 and for each fictitious node in V^3, we add edges to connect all non fictitious nodes in $V^1 \cup V^2$. Thus from an instance of the one-in-three 3SAT with n variables and m clauses, we obtain a tripartite graph with $9n + 3m - 1$ nodes and $33n + 19m - 3$ edges.

First observe that a stable set in G cannot contain more than $3n+m$ nodes (three nodes from each cycle Γ_i corresponding to a variable $l_i \in L$ and one node from each triangle T_j corresponding to a clause C_j). Suppose that we have a stable set S of G that is of size exactly $3n + m$. From S, we will construct a solution I of the one-in-three 3SAT such that each clause has exactly one true literal. We first remark that S does not contain any fictitious node since, by construction, the cardinality of any stable set with at least one fictitious node cannot exceed $3n + m - 1$. As $|S| = 3n + m$ then from each cycle Γ_i exactly three nodes are in S, and these nodes are either v^1_i, v^2_i, v^3_i or $\overline{v}^1_i$, $\overline{v}^2_i$, $\overline{v}^3_i$. Also from each triangle T_j, exactly one node is in S. Now we construct a solution I of the one-in-three 3SAT as follows : if $v^z_i \in S$, $z = 1, 2, 3$ (resp. $\overline{v}^z_i \in S$, $z = 1, 2, 3$) then we associate the value true (resp. false) to the variable l_i. In what follows we will show that for each clause $c_j = \{x_r,\ x_s,\ x_t\}$ we have exactly one literal with value true. For this it suffises to show that a clause node of T_j is taken in S if and only if the corresponding literal is of value true. Indeed suppose that $w^1_{j,r} \in S$. We may suppose that $x_r = l_r$, the case where $x_r = \overline{l}_r$ is similar. By construction of G, if $w^1_{j,r}$ is in S then, as $\overline{v}^3_r$ is adjacent to $w^1_{j,r}$, $\overline{v}^3_r \notin S$. By the remark above, it follows that v^3_r and hence v^1_r,v^2_r belong to S. This implies that l_r has been assigned the value true. Thus x_r is of value true. Conversely, if $x_r = \mathit{true}\ (= l_r)$, then by definition of I, v^1_r, v^2_r, v^3_r must belong to S. As in this case, by construction of G, $w^2_{j,s}$ and $w^3_{j,t}$ are adjacent to v^3_r and v^2_r, respectively, S must contain $w^1_{j,r}$. Consequently, a literal of clause c_j is true with respect to I if and only if the corresponding node of T_j is in S. As S contains exactly one node from each T_i, it then follows that each clause has exactly one true literal.

Using similar argument, we can prove that if I admits a truth assignment, then G contains a stable set of cardinality $3n + m$, which ends the proof. □

Theorem 3.2 *FPMSP is equivalent to TSSPPM.*

Proof. We will outline the proof. Let $G = (U \cup V, E)$ and $G' = (V^1 \cup V^2 \cup V^3, E')$ be the graphs on which the problems FPMSP and TSSPPM are considered respectively. We will show how an instance of TSSPPM can be transformed to an instance of FPMSP and vice versa. For an edge (v_i^1, v_i^2) where $v_i^1 \in V^1$, and $v_i^2 \in V^2$, we consider a node $u_i \in U$. And for a node $v_i^3 \in V^3$ we consider a node $v_i \in V$. Moreover if the edges (v_i^1, v_k^3) and (v_i^2, v_k^3) are in E', for some i, k, then add the edge $(u_i, v_k) \in E^{tf}$. Otherwise if edge $(v_i^1, v_k^3) \in E'$ (and $(v_i^2, v_k^3) \notin E'$), then add (u_i, v_k) in E_i^t and if edge $(v_i^2, v_k^3) \in E'$ (and $(v_i^1, v_k^3) \notin E'$), then add (u_i, v_k) in E_i^f, we have $E = (\cup_i E_i^t) \cup (\cup_i E_i^f) \cup E^{tf}$. In order to reduce a FPMSP instance to a TSSPPM one, we can just use the reverse transformation. Using this, we can easily show that FPMSP in G is equivalent to TSSPPM in G'. □

Theorem 3.2 implies that there exists a polynomial reduction from TSSPPM into the FPMSP. As the TSSPPM has been proved NP-complete in Theorem 3.1, we then deduce the following result.

Corollary 3.3 *The FPMSP is NP-complete.* □

4 Integer programming formulation

In this section, we give a formulation of the FPMSP as an integer linear program. With every edge set F of $\mathcal{E}$, we associate a binary variable x_F which takes 1 if F is contained in E' and 0 otherwise. The FPMSP is then equivalent to the following integer linear program.

$$\max \sum_{F \in \mathcal{E}} x_F$$

$$x_{E_i^t} + x_{E_i^f} \leq 1 \qquad \forall\ i = 1, \ldots, n, \tag{1}$$

$$\sum_{F \in \mathcal{E} : F \cap M \neq \emptyset} x_F \leq n - 1 - |M \cap E^{tf}| \qquad \forall\ M \in \mathcal{M}, \tag{2}$$

$$x_F \in \{0, 1\} \qquad \forall\ F \in \mathcal{E}, \tag{3}$$

where $\mathcal{M}$ is the set of perfect matchings of G. We will call inequalities (1) *incidence inequalities* and inequalities (2) *matching inequalities*. Inequalities (1) express the fact that at most one edge set among E_i^t and E_i^f may be taken in E'. And inequalities (2) ensure that, given a perfect matching M of G, all the edge sets intersecting M cannot be contained in E', since the edges of these sets, together with the edges of E^{tf}, induce a subgraph containing M as a perfect matching.

Let $P_{FPMSP}(G)$ be the convex hull of the solution of the above program, that is,

$$P_{FPMSP}(G) = conv(\{x \in \{0,1\}^{2n} | x \text{ satisfies } (1), (2)\}).$$

Without loss of generality, we may suppose that any subgraph $\tilde{G} = (U \cup V, \tilde{E})$ where $\tilde{E} = E^{tf} \cup F$, with $F \in \mathcal{E}$, does not contain a perfect matching. Indeed, suppose that there exists $i \in \{1, \dots, n\}$ such that for instance, the graph $(U \cup V, E^{tf} \cup E_i^t)$ contains a perfect matching. Then one can transform the instance of the FPMSP into another one by removing E_i^t and adding E_i^f to E^{tf}. This corresponds to considering that equation eq_i is no more conditional. A similar transformation can be done if F belongs to $\{E_i^f : i = 1, \dots, n\}$. This assumption leads us to give the following.

Theorem 4.1 *$P_{FPMSP}(G)$ is full dimensional.* □

Theorem 4.2 *1) Inequality $x_F \geq 0$ defines a facet of $P_{FPMSP}(G)$.*
2) Inequality $x_F \leq 1$ defines a facet of $P_{FPMSP}(G)$ if and only if the graph $\tilde{G} = (U \cup V, E^{tf} \cup F \cup F')$, is perfect matching free for all $F' \in \mathcal{E} \setminus \{F\}$. □

Theorem 4.3 *Inequality (1) associated with some $i \in \{1, \dots, n\}$ defines a facet of $P_{FPMSP}(G)$ if and only if for all $F \in \mathcal{E} \setminus \{E_i^t, E_i^f\}$, either the graph $\tilde{G} = (U \cup V, E^{tf} \cup E_i^t \cup F)$ or $\tilde{G} = (U \cup V, E^{tf} \cup E_i^f \cup F)$ does not contain a perfect matching.* □

Theorem 4.4 *Let M be a perfect matching and $F_M = \{F \in \mathcal{E} : F \cap M \neq \emptyset\}$. Inequality (2) associated with M, defines a facet of $P_{FPMSP}(G)$ if and only if*

(i) *for all E_i^t (resp. E_i^f) $\in F_M$, the graph induced by $(E^{tf} \cup E_i^f) \cup (\cup_{F \in F_M} F \setminus E_i^t)$ (resp. $(E^{tf} \cup E_i^t) \cup (\cup_{F \in F_M} F \setminus E_i^f)$) is perfect matching free,*

(ii) *for all E_k^t (resp. E_k^f) with $(\{E_k^t, E_k^f\}) \cap F_M = \emptyset$, there exists $F^* \in F_M$ such that the graph induced by $(E^{tf} \cup E_k^t) \cup (\cup_{F \in F_M} F \setminus F^*)$ (resp. $(E^{tf} \cup E_k^f) \cup (\cup_{F \in F_M} F \setminus F^*)$) is perfect matching free.*

Proof. Suppose there exists $E_i^t \in F_M$ (resp. $E_i^f \in F_M$) such that the graph $\tilde{G} = (U \cup V, ((E^{tf} \cup E_i^f) \cup (\cup_{F \in F_M} F \setminus E_i^t)))$ (resp. $\tilde{G} = (U \cup V, ((E^{tf} \cup E_i^t) \cup (\cup_{F \in F_M} F \setminus E_i^f))))$ contains a perfect matching. Therefore, inequality $\sum_{F \in \mathcal{E}: F \cap M \neq \emptyset} x_F + x_{E_i^f} \leq n - 1 - |M \cap E^{tf}|$ (reps. $\sum_{F \in \mathcal{E}: F \cap M \neq \emptyset} x_F + x_{E_i^t} \leq n - 1 - |M \cap E^{tf}|$) is valid for $P_{FPMSP}(G)$. Since this inequality dominates (2) and defines a face of $P_{FPMSP}(G)$ different from that defined by (2), this implies that this later cannot define a facet.

Suppose there exists E_k^t (resp. E_k^f) with $(\{E_k^t, E_k^f\}) \cap F_M = \emptyset$ such that the graph $\tilde{G} = (U \cup V, ((E^{tf} \cup E_k^t) \cup (\cup_{F \in F_M} F \setminus F^*)))$ (resp. $\tilde{G} =$

$(U \cup V, ((E^{tf} \cup E^f_k) \cup (\cup_{F \in F_M} F \setminus F^*)))$ contains a perfect matching, for all $F^* \in F_M$. Therefore, inequality $\sum_{F \in \mathcal{E}: F \cap M \neq \emptyset} x_F + x_{E^t_k} \leq n - 1 - |M \cap E^{tf}|$ (resp. $\sum_{F \in \mathcal{E}: F \cap M \neq \emptyset} x_F + x_{E^f_k} \leq n - 1 - |M \cap E^{tf}|$) is valid for $P_{FPMSP}(G)$. Since this inequality dominates (2) and defines a face of $P_{FPMSP}(G)$ different from that defined by (2), this implies that this later cannot define a facet.

Now suppose that (i) and (ii) are satisfied. To show that (2) defines a facet, it suffices to exhibit $2n$ solutions whose incidence vectors are affinely independent and satisfy inequality (2) with equality.

Consider the following sets : $S_i = (F_M \setminus \{E^t_i\}) \cup \{E^{tf}, E^f_i\}$, $S'_i = S_i \setminus \{E^f_i\}$, for i such that $M \cap E^t_i \neq \emptyset$, $S_j = (F_M \setminus \{E^f_j\}) \cup \{E^{tf}, E^t_j\}$, $S'_j = S_j \setminus \{E^t_j\}$, for j such that $M \cap E^f_j \neq \emptyset$. By condition (i), the sets S_i, S_j induce perfect matching free graphs. Hence the sets S'_i, S'_j so do. Consider also the sets $S_k = (F_M \setminus \{F^*\}) \cup \{E^{tf}, E^t_k\}$, $S'_k = (F_M \setminus \{F^*\}) \cup \{E^{tf}, E^f_k\}$, for k such that $M \cap (E^f_k \cup E^t_k) = \emptyset$. Note that F^* is the set introduced in condition (ii), of the theorem. By condition (ii) these sets also induce perfect matching free graphs. So we have $2n$ solutions of FPMSP. Moreover the incidence vectors of these solutions satisfy (2) with equality and are affinely independent. □

5 Concluding remarks

In this paper we have given a formulation of the SAP for conditional DAS in terms of matchings in bipartite graphs. We have shown that the problem is NP-complete. We have also presented an integer linear programming for this problem and discussed the associated polytope. We note that inequalities (2) can be separated in polynomial time. In fact, this reduces to the maximum matching problem in a bipartite graph. Using this, we have also developed a Branch-and-Cut algorithm for the problem which we have tested on real data.

References

[1] P. Fritzson. (2003). Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-Interscience.

[2] K. Murota. (1987). Systems Analysis by Graphs and Matroids. Springer-Verlag.

[3] M. Lacroix, A. R. Mahjoub and S. Martin. (2009). Structural analysis in Differential-Algebraic Systems and Combinatorial Optimization. Proceedings of 39th International Conference on Computers & Industrial Engineering (CIE39), pages 331-337 (2009).