2016 European Control Conference
June 29 - July 1, 2016. Aalborg, Denmark

Minimum controller substructure for generic arbitrary pole placement: multicommodity flow and TSP based formulations*

Atreya Kotoky[1], Ashutosh Mahajan[1], Ashwin Arulselvan[2], Madhu N. Belur[1], Rachel K. Kalaimani[3]

Abstract— **This paper deals with finding a 'least interaction' controller that generically achieves pole placement, given a actuator-sensor interaction possibility for the controller. The structure of the plant and the controller are modeled as an undirected and bipartite graph. Assuming that the plant/controller structures are specified, we find a minimum controller *substructure* within the specified controller structure such that the controller substructure allows generic eigenvalue assignability. The minimum is in the sense that the bipartite graph consisting of the proposed controller has the minimum number of edges. The complexity of a brute-force algorithm to identify a minimum controller substructure would be exponential and hence we propose two formulations for solving this problem, using recent results about equivalence of generic pole-assignability and covering of plant edges using cycles. The first uses multi-commodity flow networks to include all plant edges in some cycle using the least number of controller edges. We show that an integer solution to this formulation gives a minimum controller substructure for arbitrary pole placement problem. Since the problem of finding a feasible integer flow in multicommodity networks is NP-complete, there is ample reason that identifying a minimum controller substructure is NP-hard. The second formulation uses the framework of travelling salesman with profits (TSP with profits) to cover all vertices of the bipartite graph by cycles using the least number of controller edges. The TSP-with-profits problem too belongs to the class of NP-hard problems. We show that our formulation is equivalent to the so-called Generalized Travelling Salesman Problem (GTSP) thus allowing branch-cut algorithms developed for GTSP problems.**

Index Terms— **bipartite graph, behaviors, DAE systems, vertex cover, multi-commodity flow, travelling salesman with profits, generalized travelling salesman problem (GTSP)**

I. Introduction

In this paper, we address the problem of finding a minimum controller structure within a specified structure. We assume dynamical systems are represented by linear differential-algebraic equations (DAEs) of order possibly higher than one. We consider a structured system of DAEs for both the to-be-controlled plant and the controller and model the structure of the plant and the controller as an undirected and bipartite graph. We assume that the controller has *structural constraints*, i.e., the controller equations are constrained by a structure that specifies which system-variables occur in each controller equation. Such a constraint is motivated by an actuator-sensor constraint that occurs in practice, i.e., often a controller-designer is specified with constraints about the exact sensors, feedback from which each actuator can avail. The controller-designer is allowed to choose a controller to achieve desired control specifications, like arbitrary pole placement, but these sensor-actuator interaction constraints have to be obeyed for practical reasons. We call this a 'specified controller structure', and the problem addressed in this paper is to choose a 'minimum interaction' controller from within this structure. The minimum here is in the number of edges of the bipartite graph; in terms of equations-variables, this amounts to minimizing the number of terms across all controller equations. We assume the to-be-controlled system, i.e. the plant, too is specified in terms of a bipartite graph: i.e. an equation-variable interaction graph. This is made precise later below in Section II. In [8] necessary and sufficient conditions on this graph have been formulated for a given structured controller to generically achieve arbitrary pole placement.

For such structured systems, we pursue the problem of finding the *minimum* controller structure for generic arbitrary pole placement. We assume that the plant and controller structures are already specified and we need to find a minimum controller structure from within this specified structure: minimum in the sense that the graph which portrays the controller structure has the least number of edges. This problem is motivated by minimality in design of a sensor-actuator network. A minimum controller structure corresponds to minimum interaction between sensors and actuators, reducing the complexity of design and cost of network.

While arbitrary pole placement with a specified controller structure which is in output feedback has been addressed for state space systems in [13], the techniques in this paper apply to more general models of dynamical systems: linear differential-algebraic equations (DAEs) of order possibly higher than one. Similarly, a generic solution to the minimum controller structure problem is provided in [12], but only for first order descriptor systems.

We propose two formulations for solving the minimum controller structure problem, one using multicommodity flows and the other using Travelling salesman with profits (TSP with profits): these are described briefly below.

A multi-commodity flow problem involves a collection of several networks whose flows must independently satisfy conservation of flow constraints, but are coupled through some other constraints or the cost function. The problem

*This work was supported in part by SERB, DST and BRNS, India.

[1]Atreya Kotoky and Madhu N. Belur are in the Department of Electrical Engineering, and Ashutosh Mahajan is in the Industrial Engineering and Operations Research, at the Indian Institute of Technology Bombay, India. Email: belur@iitb.ac.in and amahajan@iitb.ac.in.

[2]Ashwin Arulselvan is in the Department of Management Science, University of Strathclyde Glasgow, UK. Email: ashwin.arulselvan@strath.ac.uk.

[3]Rachel K. Kalaimani is in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Email: rachelpsg@gmail.com.

of producing an integer flow satisfying all constraints is NP-complete [3] even for only two commodities and unit capacities.

Travelling salesman problems with profits (TSPs with profits) are a generalization of the travelling salesman problem (TSP), where it is not necessary to visit all vertices. A profit is associated with each vertex and a cost with each edge. The overall goal is the simultaneous optimization of the collected profit and the travel costs. TSPs with profits belong to the class of NP-hard problems because they trivially belong to NP and because a TSP instance can be stated as a TSP with profits instance by defining very large profits on vertices [4].

The rest of the paper is organized as follows. The following section formulates more precisely the problem considered in this paper. It is straightforward to see that a brute force approach to obtaining a minimum controller has exponential runtime complexity. Section III formulates this problem as a multicommodity flow problem, while Section IV formulates this problem as a variant of the Travelling Salesman problem. Section V concludes this paper.

II. Problem formulation

We first briefly describe how to associate a dynamical system: both the plant and the controller with a bipartite graph. Consider Figure 1, in which the left-side nodes are associated to equations, and the right side nodes are associated to variables. Since the equations are either plant equations or controller equations, we use 'plant nodes' and 'controller nodes' when referring to the nodes on the left side. In the same vein, the nodes on the right are often called 'variable nodes'. Suppose the plant equations are described by a system of p linear, constant-coefficient ordinary differential equations of the form:

$$R_0 w + R_1 \frac{d}{dt} w + R_2 \frac{d^2}{dt^2} w + \cdots + R_N \frac{d^N}{dt^N} w = 0 \qquad (1)$$

with $R_i \in \mathbb{R}^{p\times q}$, and $w \in \mathscr{C}^\infty(\mathbb{R},\mathbb{R}^q)$, the space of infinitely often differentiable trajectories from $\mathbb{R}$ to $\mathbb{R}^q$. Associate the polynomial matrix $R(\xi) \in \mathbb{R}^{p\times q}[\xi]$ with p rows and q columns and the indeterminate ξ with the above differential equation: $R(\xi) := R_0 + R_1\xi + R_2\xi^2 + \cdots R_N\xi^N$, and use a short-hand notation for the above differential equations $R(\frac{d}{dt})w = 0$.

We consider further the polynomial matrix $R(\xi) \in \mathbb{R}^{p\times q}[\xi]$, in order to construct a bipartite undirected graph. Associate p nodes in the left-part corresponding to the rows (i.e. the equations describing the system) of $R(\xi)$ and q nodes in the right-part with the columns (i.e. the variables constituting the system equations) of $R(\xi)$. The edges in this bipartite graph correspond to the nonzero entries in $R(\xi)$: each nonzero entry in $R(\xi)$ results in an edge from the corresponding row-node and column-node.

This is undertaken for the *plant* equations: which is typically under-determined, i.e. more columns than rows in $R(\xi)$. In the behavioral framework, to design and implement a (feedback or open-loop) *controller*, means to introduce additional laws on the same variables, so that we together have a 'square' system of equations, i.e. a determined system of equations. Suppose the additional laws themselves are a system of linear, constant-coefficient, ordinary differential equations: associate a polynomial matrix $C(\frac{d}{dt})w = 0$ to these equations, with $C(\xi) \in \mathbb{R}^{(q-p)\times q}[\xi]$: such that $\begin{bmatrix} R(\xi) \\ C(\xi) \end{bmatrix}$ is square and nonsingular. It can be found in [14], about how the roots of the determinant of $\begin{bmatrix} R(\xi) \\ C(\xi) \end{bmatrix}$ are exactly the 'closed loop' poles. A detailed exposition on the behavioral approach and the view that control is nothing but introduction of additional laws can also be found in [14].

Since the focus in this paper is more on the graph-theoretic implications, we do not delve further along this direction. The essential results can be found in [8]. The key intuitive idea, that has been formalized and proved in [8] is that, after the plant and controller are interconnected, i.e. the system variables have to satisfy laws of both the plant and the controller, then the bipartite graph has equal number of equation nodes and variable nodes, with the equation nodes partitioned into plant laws and controller laws. 'Generic' investigation allows studying just the graph and the exact numerical values that comprise the coefficient matrices R_i in equation (1), do not play a role as long as the values are 'generically' chosen. More precisely, the set of values for which the generic conclusions do *not* hold form a 'thin' set, or a set of measure zero. This can be made more precise: we refer the reader to [12], [8].

The closed loop system, i.e. the interconnected system, is required to be autonomous: generic nonsingularity of $\begin{bmatrix} R(\xi) \\ C(\xi) \end{bmatrix}$ translates to existence of a *perfect* matching in the bipartite graph: i.e. a subgraph of the given graph in which each node has degree one in the subgraph.

An example of a polynomial matrix that gives rise to the bipartite graph shown in Figure 1 is as follows.

$$\begin{array}{lrl} \text{equation } P_1: & a_1(\frac{d}{dt})V_1 + a_2(\frac{d}{dt})V_2 & = 0 \\ \text{equation } P_2: & a_3(\frac{d}{dt})V_2 + a_4(\frac{d}{dt})V_3 & = 0 \\ \text{equation } C_1: & b_1(\frac{d}{dt})V_1 + b_2(\frac{d}{dt})V_2 + b_3(\frac{d}{dt})V_3 & = 0 \end{array}$$

Here equations P_1 and P_2 are plant equations, i.e. $a_1,\ldots,a_4$ are *given* nonzero polynomials, while equation C_1 is a *to-be-designed* controller equation, and hence b_1, b_2, b_3 are to-be-chosen polynomials such that the determinant of the corresponding square 3×3 polynomial matrix has desired roots.

Consider the bipartite graph $G = (R, C; E)$, depicting the interconnection of the plant and the controller. The two sets R and C denote the set of equations and variables respectively. The node set $R := R_p \cup R_c$ is further partitioned into plant and controller nodes. Similarly the edge set E is partitioned into plant and controller edges ($E := E_p \cup E_c$), depending on whether the edge is incident on R_p or R_c. The subgraph $G^P(R_p, C; E_p)$ of G represents the structure of the plant, and $G^C(R_c, C; E_c)$ represents the structure of the controller.

Assumption 2.1: Using the notation above, assume that the graph G satisfies the following properties.

A: Every edge $e \in E$ is part of a perfect matching.

B: The graph $G^P(R_p, C; E_p)$ is a tree.

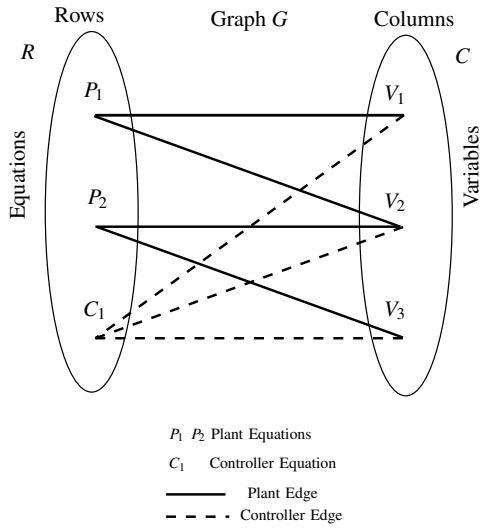

Fig. 1. Bipartite graph for system

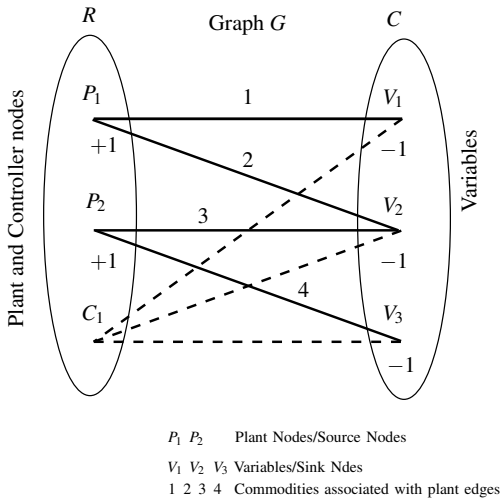

Fig. 2. Bipartite graph with source, sink and commodities

C: Every plant edge E_p is in some cycle containing controller edges E_c.

Note that, under mild assumptions on the plant and after suitable simplifications which are without loss of generality, conditions A, B and C are necessary and sufficient conditions for generic arbitrary pole placement. This is shown and elaborated in [8], [9]: we include here only a brief justification for each of the three conditions. Condition A translates to the closed loop system being *autonomous*; as explained earlier in this section, generic nonsingularity of a polynomial matrix is equivalent to existence of a perfect matching. Condition B is an assumption that is without loss of generality: cycles, if any, in the plant graph can be 'merged' into a new node; this is elaborated in [9]. It is also shown there that condition C (together with assumptions A and B) is equivalent to ability of the controller to achieve (generic) pole placement.

For simplicity, we classify the edges into plant edges and controller edges, depending on the left-node that the edge is incident on. We are now ready to state the minimum pole-placement controller substructure identification problem.

Problem 2.2 *Minimum pole-placement controller substructure problem:* Suppose a plant and controller structure represented by the graph G is given and suppose conditions A, B and C are satisfied. Identify a minimum controller structure from the specified structure. More precisely, find a subset $E_c^{\min} \subseteq E_c$ such that the number of edges in $E_c^{\min}$ is *minimum* amongst all controller edge subsets $E_c{}' \subseteq E_c$ satisfying conditions A, B and C above.

Note that the plant satisfies Assumption 2.1 and hence feasibility is guaranteed: E_c already satisfies conditions A, B and C. A minimum cardinality subset of E_c search is possible by brute-force with at most an exponential running-time complexity. The following sections pursue alternative formulations that help obtain faster heuristic methods and suggest NP-hardness of the above problem.

III. Multi-commodity flow based formulation

This section proposes a multicommodity flow formulation of the minimum controller substructure problem above. We first describe the notion of a multicommodity network flow and recall a line from [2]: "Multicommodity network flow problems involve several flow types or commodities, which simultaneously use the network and are coupled through either the arc flow bounds, or through the cost function." This model finds application in diverse domains such as communications, logistics, manufacturing, transportation, urban housing and food grain export-import. For example, in communication networks the commodities are the streams of different classes of traffic (telephone, data, video, etc.) that involve different origin-destination pairs. These commodities must each satisfy not only their own conservation of flow constraints, but also coupling constraints due to communication capacity threshold of the network arcs.

We formulate the minimum pole placement problem Section II using multi-commodity flows. Associate a commodity $k \in K (K = \{1,2,...,|E_p|\})$ with each plant edge E_p. For each of the k commodities, the plant node is designated as the source node (with divergence (div) = +1) and the variable node as the sink node (with divergence = -1). Each undirected edge $e = (i,j) \in E$ from node i to j permits flow of commodities in both directions (i to j) and (j to i). Hence, given the undirected graph G, a bi-directed graph $B = (N,A)$ can be obtained by replacing each undirected edge $e \in E$ with arcs (i,j) and (j,i). For each arc $a = (i,j) \in A$, the variable x_{ij}^k denotes the flow of commodity k from node i to j. The integer variable y_{ij} is defined for each controller arc $a = (i,j) \in A_c$, where $A_c \subset A$ is the set of controller arcs.

The multicommodity flow formulation is as follows: the flow of commodity k is not allowed on the edge connecting its source and sink nodes. The flow therefore has to find an alternate path for each commodity k from source to sink. Since the plant is a tree, every alternate path must include controller edges. The multicommodity flow formulation

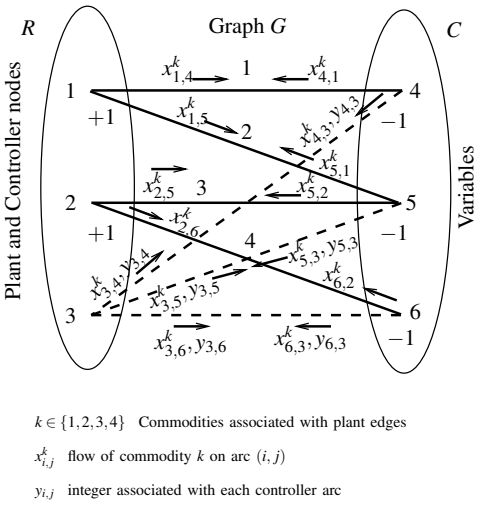

Fig. 3. Multicommodity flow formulation

finds an alternate path connecting the two ends of every plant edge $E_p \in E$ using minimum number of controller edges. This is equivalent to including all plant edges in some cycle using least number of controller edges.

Sets arising in the formulation:

N: set of nodes
A: set of arcs
A_c: set of controller arcs ($A_c \subset A$)
K: set of commodities ($K = \{1,2,...,|E_p|\}$)

Variables introduced:

x^k_{ij}: $k \in K, (i,j) \in A$: flow of commodity k on arc (i,j)
y_{ij}: $(i,j) \in A_c$: integer associated with controller arc (i,j)

$$\text{Performance objective:} \quad \min \sum_{(i,j)\in A_c} y_{ij} \tag{2}$$

Constraints:

$$\sum_{j:(i,j)\in A} x^k_{ij} - \sum_{j:(j,i)\in A} x^k_{ji} = \begin{cases} 1 & \text{if } i \text{ = source node for commodity } k, \\ -1 & \text{if } i \text{ = sink node for commodity } k, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$x^k_{ij} = \begin{cases} 1 & \text{if one unit of flow goes from node } i \text{ to } j \text{ for commodity } k, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$x^k_{ji} = 0$ if div(j) = +1 and div(i) = -1 are the source and sink nodes for commodity k respectively (5)

$$\sum_{(i,j)\in A_c} y_{ij} \geqslant 1 \qquad \text{for all } (i,j) \in A_c \tag{6}$$

$$x^k_{ij} \leqslant y_{ij} \qquad \text{for all } (i,j) \in A_c \tag{7}$$

$$x^k_{ij} \geqslant 0 \qquad \text{for all } k \in K, (i,j) \in A \tag{8}$$

$$y_{ij} \text{ is integer} \qquad \text{for all } (i,j) \in A_c \tag{9}$$

1) The mass balance constraints in equation (3) state that for a commodity k, the flow out minus the flow in at a node must equal the supply/demand of that node for that commodity.
2) The flow constraint in equation (5) restricts flow of commodity k on the edge connecting its source and sink nodes. The flow therefore has to find an alternate path for each commodity $k \in K$ from source to sink.
3) The constraint in equation (9) states that y_{ij} is a integer variable indicating whether or not arc $(i,j) \in A_c$ is included in the solution.
4) The constraint in equation (7) indicates that flow on a controller arc $a = (i,j) \in A_c$ for commodity $k \in K$ is allowed if the controller arc is included in the solution.
5) The constraint on y_{ij} in equation (6) forces controller arcs to be used in the solution.
6) The objective function in equation (2) minimizes the number of controller arcs required to find alternate paths for each commodity $k \in K$.

   An integer flow x^k_{ij} for all $(i,j) \in A$ and $k \in K$ gives integer values of y_{ij} for all $(i,j) \in A_c$. The controller arcs $a = (i,j) \in A_c$ for which $y_{ij} > 0$ give $E_c^{\min}$.

   Solution methods available: Specialized adaptations of linear programming algorithms [10] are available for solving multi-commodity flow problems. The problem of producing an integer flow satisfying all constraints in multi-commodity flow problems is known to be NP-complete [3]: hence the claim that there is ample reason for the minimum pole placement controller substructure problem to be NP-hard.

IV. Travelling salesman based formulation

We propose that the *minimum* number of controller edges $E_c^{\min} \subseteq E_c$ required to satisfy conditions of Section II can be determined by covering all vertices of the bipartite graph by cycles using least number of controller edges. Consider the following ways in which vertex cover is possible:

1) *Cover by Hamiltonian cycle:* If a Hamiltonian cycle exists, then all plant edges are in some cycle with the controller edges required to complete the Hamiltonian cycle. The number of controller edges required is then the minimum.
$$E_c^{\min} = 2R_c = 2(C - R_p)$$
where R_c, R_p and C are the number of controller nodes, plant nodes and variable nodes respectively. In this case, all plant edges will be a part of some perfect matching.
2) *Cover by vertex disjoint cycles:* The vertices of the graph are covered by multiple vertex disjoint cycles. Since plant is a tree, at least one controller node is required to form a cycle; to cover all the vertices we can have $2, \ldots, R_c$ number of disjoint cycles. However the number of controller edges required will again be a minimum i.e. $E_c^{\min} = 2R_c$ irrespective of the number of disjoint cycles.

   We now consider the case when there is only a single plant edge connecting two disjoint cycles (see edge

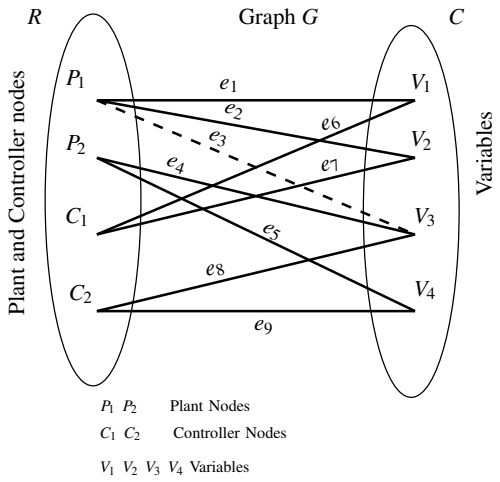

Fig. 4. Vertex cover by disjoint cycles

e_3 in Fig. 4). Such a plant edge will not be in a cycle with the selected controller edges. However we note that these plant edges will not be part of any maximum matching and hence are inadmissible by the first condition of Section II.

Thus if vertex cover with disjoint cycles is possible,

- All admissible plant edges are in a cycle with the selected controller edges.
- $E_c^{\min} = 2R_c$ which is the minimum possible.

3) *Cover by non-disjoint cycles:* The vertices of the graph are covered by non-disjoint cycles using the least number of controller edges i.e. some of the vertices are covered by multiple cycles. As cycles share vertices, plant edges connecting different cycles will be part of a cycle with the selected controller edges. In this case, the minimum number of controller edges required, $E_c^{\min} \geqslant 2R_c$.

Similarly vertex cover by a combination of vertex disjoint cycles and non-disjoint cycles will also satisfy conditions of problem described in Section II as only inadmissible plant edges will not be in a cycle with the selected controller edges.

Thus our problem is equivalent to covering all vertices of the bipartite graph by cycles using least number of controller edges. We propose a *Travelling Salesman Problem (TSP)* based formulation to find $E_c^{\min}$. We find a Hamiltonian cycle, if it exists, in the graph; else the vertices are covered by a set of cycles, using the least number of controller edges.

A. Travelling salesman with profits

Travelling salesman problems with profits (TSPs with profits) are a generalization of the traveling salesman problem (TSP), where it is not necessary to visit all vertices. A profit is associated with each vertex and a cost with each edge. The overall goal is the simultaneous optimization of the collected profit and the travel costs. These two optimization criteria appear either in the objective function or as a constraint. TSPs with profits may be seen as bicriteria TSPs with two opposite objectives, one to visit more vertices to collect profit and the other to minimize travel costs (with the right to drop vertices). Viewed in this light, "solving TSPs with profits should result in finding a non-inferior solution set, i.e., a set of feasible solutions such that neither objective can be improved without deteriorating the other" [4].

TSPs with profits are a useful model for problems involving simultaneous selection and sequencing decisions, e.g., as in location-routing problems. They find practical applications in warehouse order picking with multiple stock locations, routing of welfare clients through governmental agencies, sequencing computer files, flexible manufacturing scheduling, postal routing, airport selection and routing for courier planes, and the design of ring networks. See [5].

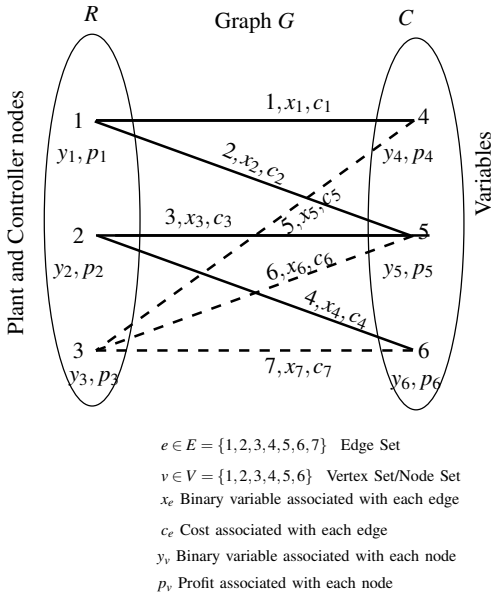

Fig. 5. TSP with profits formulation

We formulate the minimum pole placement problem described in Section II as Travelling Salesman Problem (TSP) with profits [7, Chapter 13, page 611] and [1]. Let $G = (V,E)$ be the graph where the node set $V := R \cup C$ and $|V| = 2n$. Associate a cost c_e with each edge $e \in E$ and a profit p_v with each node in $v \in V$. For any $S \subseteq V$, let $\delta(S)$ denote the set of edges with exactly one node in S:

$$\delta(S) := \{(i,j) \in E : i \in S, j \notin S\}$$

Solve the following bivariate TSP in the first iteration:

variables

x_e = 1, if the edge belongs to the optimal cycle
= 0, otherwise

c_e = 0, if it is a plant edge
= 1, if it is a controller edge

y_v = 1, if the node belongs to the optimal cycle
= 0, otherwise

p_v profit associated with each node

$$\text{Objective} \qquad \min \quad \sum_{e \in E} c_e x_e - \sum_{v \in V} p_v y_v \qquad (10)$$

Constraints:

$$x_e \in \{0,1\}, \quad y_v \in \{0,1\} \tag{11}$$

$$\sum_{e \in \delta(v)} x_e = 2y_v \quad \text{for all} \quad v \in V \tag{12}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2(y_i + y_j - 1) \text{ for all } S \subset V,\ i \in S,\ j \in V \backslash S \tag{13}$$

First iteration

1) The objective function in equation (10) states that we are looking for an optimal cycle that simultaneously satisfies the following two criteria :
   - Covers the most number of vertices.
   - Uses the least number of controller edges.

   The solution algorithm therefore either finds a Hamiltonian cycle, if such a cycle exists in the graph, or an optimal cycle on a subset of the vertices.
2) The degree constraints in equation (12) state that the number of edges incident on a vertex v is either 2 (if v is visited) or 0 (otherwise).
3) The constraints in equation (13) are connectivity constraints which state that each cut separating two visited nodes (i and j) must be crossed at least twice.

Subsequent iterations: Suppose the graph does not contain a Hamiltonian cycle. Let m nodes be covered by the cycle in the first iteration. Let $\{E_{c1}, E_{c2}, \ldots, E_{cm}\} \in E_c$ be the controller edges used in the optimal cycle. Do the following:

1) Set $c_e = 0$ for $e \in E_{c1}, E_{c2}, \ldots, E_{cm}$.
2) Associate profits p_1 to already covered vertices and p_2 to uncovered vertices such that:
$$\frac{p_2}{p_1} = \frac{m}{2n-(m+1)}$$
   This is to ensure that the new optimal cycle uses vertices not covered in the previous iteration.
3) Force the new optimal cycle to start and end at an uncovered vertex by setting
$$y_k = 1 \quad \text{where } y_k \text{ is an uncovered vertex.} \tag{14}$$

The problem is now reformulated with the objective function described in equation (10) the additional constraint equation (14) and the re-adjusted costs and profits. The above procedure is repeated till all vertices of the graph are covered.

B. Solution methods available

We show subsequently that the above formulation is equivalent to the Generalized TSP problem (GTSP), which is well studied in literature [6][7, Chapter 13, page 615]. In the GTSP, the vertex set V is partitioned into $m \geqslant 3$ clusters, $V_1, V_2, \ldots, V_m$, and the cycle is feasible if it visits each cluster at least once:

$$\sum_{v \in V_h} y_v \geqslant 1 \quad \text{for } h = 1, \ldots, m. \tag{15}$$

The additional constraint (15) is automatically satisfied by our TSP with profits formulation (10)–(11). In our formulation, the node set V can be partitioned into three clusters, plant vertices (R_p), controller vertices (R_c) and variables (C). Since the plant graph $G^P(R_p, C; E_p)$ is a tree, every cycle will contain at least one vertex from each of these three clusters. Thus our TSP with profits formulation (10)–(11) is equivalent to the GTSP. Hence algorithms developed for solution of GTSP can be applied directly to our formulation. Branch and cut algorithm for the exact solution of the symmetric GTSP with computational results is available (see [6]).

V. Conclusions

In this paper we considered the problem of finding minimum controller structure from a specified structure for generic arbitrary pole placement. We proposed two formulations for solving the minimum controller structure problem, one using multicommodity flows and the other using Travelling salesman with profits (TSP with profits). While existing techniques to address arbitrary pole-placement using structured controllers start from a *state space* representation of the system [13], the techniques in this paper apply to more general models of dynamical systems: linear differential-algebraic equations of order possibly higher than one.

The significance of showing the formulation of the minimum pole-placement controller substructure problem as a variant of the multicommodity flow problem and of the travelling salesman problem is to eventually prove the expectation of NP-hardness and, more importantly, utilize the well-developed faster heuristics (like branch-cut algorithms) for these standard problems for solving the minimum controller substructure problem.

References

[1] D. Bienstock, M.X. Goemans, D. Simchi-Levi and D. Williamson, "A note on the prize collecting traveling salesman problem", *Mathematical Programming*, vol. 59, no. 1-3, pages 413-420, 1993.

[2] P.B. Dimirti, "Network Optimization: Continuous and Discrete Models," *Athena Scientific, May*, 1998.

[3] S. Even, A. Itai and A. Shamir, "On the complexity of time table and multi-commodity flow problems", *Foundations of Computer Science, 1975, 16th Annual Symposium on IEEE*, pages 184-193, IEEE, 1975.

[4] D. Feillet, P. Dejax and M. Gendreau, "Traveling salesman problems with profits", *Transportation Sci.*, vol. 39, no. 2, pages 188-205, 2005.

[5] M. Fischetti, J.J. Salazar and P. Toth, "The Symmetric Generalized Travelling Salesman Polytope", *Networks*, vol. 26, no. 2, pages 113-123, 1995

[6] M. Fischetti, J.J. Salazar Gonzalez and P. Toth, "A branch-and-cut algorithm for the symmetric generalized traveling salesman problem", *Operations Research*, vol. 45, no. 3, pages 378-394, 1997.

[7] G. Gutin and A.P. Punnen, *The Traveling Salesman Problem and its Variations*, vol. 12, Springer Science & Business Media, 2002.

[8] R.K. Kalaimani, M.N. Belur and S. Sivasubramanian, "Generic pole assignability, structurally constrained controllers and unimodular completion", *Linear Algebra and its Applications*, vol. 439, no. 12, pages 4003-4022, 2013.

[9] R.K. Kalaimani and M.N. Belur, "Minimal controller structure for generic pole placement", *Proceedings of the IEEE European Control Conference (ECC)*, Zurich, Switzerland, pages 3446-3451, 2013.

[10] R.D. McBride, "Advances in solving the multicommodity-flow problem", *Interfaces* , vol. 28, no. 2, pages 32-41, 1998

[11] S. Pequito, S. Kar and G.J. Pappas, "Minimum cost constrained input-output and control configuration co-design problem: a structural systems approach," *Proceedings of the American Control Conference (ACC)*, pages 4099-4105, 1-3 July, Chicago, USA, 2015.

[12] M.E. Sezer, "Minimal essential feedback patterns for pole assignment using dynamic compensation", *The 22nd IEEE Conference on Decision and Control, San Antonio, USA*, pages 28-32, 1983.

[13] D.D. Šiljak, *Decentralized Control of Complex Systems*, Academic Press Inc, California, 1991.

[14] J.C. Willems, "On interconnection, control and feedback", *IEEE Transactions on Automatic Control*, vol. 42, pages 326-339, 1997.