Optimal Flow Control in Acyclic Networks with Uncontrollable Routings and Precedence Constraints*

Spyros Reveliotis

School of Industrial & Systems Engineering,

Georgia Institute of Technology

spyros@isye.gatech.edu

Theologos Bountourelis

Department of Industrial Engineering

University of Pittsburgh

thb28@pitt.edu

Abstract

This paper introduces a novel optimal flow control problem that seeks to convey a specified amount of fluid to each of the nodes of an acyclic digraph with a single source node, while minimizing the total amount of fluid inducted into the network. Two factors complicating the aforementioned task are (i) the presence of nodes with uncontrollable routing of the traversing flow and (ii) a set of precedence constraints regarding the satisfaction of the nodal fluid requirements. It is shown that the considered problem can be naturally formulated as a continuous-time optimal control problem that can be reduced to a hybrid optimal control problem with controlled switching. This property subsequently enables the solution of the considered problem through a Mixed Integer Programming formulation. Additional results in the paper establish the NP-hardness of the considered problem, highlight its affinity to some well known scheduling problems, and offer guidelines that can alleviate the increased problem complexity.

*This work was partially supported by the NSF grant CMMI-0619978.

1 Introduction

Networks and network flows have been a powerful modeling abstraction for a very broad array of applications. Hence, for instance, water, oil, gas, power, telecommunication and data networks are important elements in our contemporary technological infrastructure that present naturally the structure and the topological properties of a network. But network flows can also abstract more complex and maybe less tangible activity, allowing us to talk about production and distribution networks, workflow networks, and financial networks. More generally, a network abstraction can provide an effective representation of the existing dependencies among a set of entities, while the flow conveyed through the network can represent some sort of activity among these entities that is observant to the expressed dependencies.

From a methodological standpoint, the investigation of network flow problems can be classified to those addressing *static* network flows and those addressing flows with more *dynamic* attributes. Static network flow problems assume a time-invariant structure for the studied network and consider the flow taking place through it at an equilibrium or "steady state" mode. As a result, time is not an explicit parameter in the context of static network flow problems, and the flow traversing the different arcs of the network is perceived as a "lump sum" of material transferred through these arcs in an instant manner. The resulting class of problems falls into the broader area of combinatorial optimization, and their study has led to very strong analytical results and powerful algorithms. An excellent comprehensive exposition of the relevant theory can be found in [1]. On the other hand, dynamic network flow problems allow for time dependency for some of the network attributes, and they also consider explicitly the flow rates materialized at different parts of the network. Therefore, this class of network flow problems are more appropriate for investigating the transient behavior of the networks, and they can be perceived as real-time control problems that seek the shaping of this behavior in order to support certain performance objectives. But the increased representational capability of dynamic network flow problems comes with an increase of the relevant computational cost. In particular, while most of the classical static network flow problems admit algorithms of polynomial complexity with respect to the size of the underlying network, the majority of dynamic network flow problems are NP-hard. In fact, for many dynamic flow problems formulated in continuous time, even the effective computation of a solution can be a challenge, and, in practice, these problems can be addressed only through some *ad hoc* discretization of time.

An introductory exposition to the theory of dynamic network flow problems and its intricacies can be found in [2], while [3, 4, 5] offer additional surveys of the area in terms of analytical results and applications.

The particular dynamic network flow problem we consider in this work is defined by the following attributes: (i) The underlying network has an *acyclic* structure with a single source node. (ii) Flow enters the network from the source node, at a *constant rate* of one unit of flow per unit of time, and exits the network from its terminal nodes. Furthermore, there is no fluid concentration at any of the nodes of the network. (iii) The routing of the flow at certain nodes of the network is *controllable*, i.e., at each of these nodes, the allocation of the total incoming flow to the arcs emanating from them is decided by an applied control policy and it varies over time based on some representation of the network "state". On the other hand, the flow allocation taking place in the remaining nodes is *uncontrollable*, i.e., the incoming flow to these nodes is distributed to the arcs emanating from them according to an externally determined pattern that is fixed over time. (iv) Every node presents a *fluid requirement*, i.e., a certain (possibly zero) amount of fluid that must be traversed through it. Furthermore, there is a set of *precedence constraints* that must be observed during the satisfaction of these requirements. In particular, the satisfaction of the posed requirement at a certain node cannot be initiated until the fluid requirement at some other node(s) has been fully satisfied. In order to concretize the exposition of our results, and also for reasons relating to the application that motivated this problem and that are further discussed below, in the rest of this work we shall consider the particular precedence constraint that stipulates that the satisfaction of the fluid requirement of a certain node cannot be initiated until the fluid requirements of all of its children have been satisfied. The problem objective is to synthesize a routing policy for the controllable nodes that minimizes the total amount of fluid that must be induced in the network in order to satisfy all the nodal fluid requirements. For the needs of the modeling approach pursued in the later parts of this document, it is also important to realize that, under the problem Assumption (ii), the aforestated objective is equivalent to the objective that seeks to satisfy all the nodal fluid requirements in minimum time.

Obviously, the problem described in the previous paragraph admits a natural, straightforward interpretation in material flow networks. But the primary motivation of this problem in the context of our work comes from the need to provide a deterministic relaxation for a discrete (token) routing problem that takes place in a repetitive manner on an acyclic stochastic digraph.

Indeed, the last 15-20 years have seen an extensive use of dynamic network flow problems as pertinent approximations for discrete stochastic scheduling and routing problems. This class of problems is typically known as the (deterministic) "fluid" relaxations of their discrete stochastic counterparts, and their modeling and analytical power derives from the following two facts: (i) In many cases, the study of the abstracting fluid model can resolve important properties of the original stochastic network [6, 7]. (ii) Optimized solutions for problems formulated on the fluid model can lead to efficient solutions for the corresponding problems formulated on the original stochastic network [8, 9, 10, 11]. On the other hand, many of the flow control problems defined in the context of the abstracting fluid networks can be quite challenging themselves, giving rise to some very interesting optimal control problems [12, 13]. The recent publication of [14] offers an excellent treatment of the role of fluid models in the control of complex stochastic networks, and it defines the state-of-art in this area.

The particular fluid relaxation that is studied in this paper has been motivated by some of our past work that seeks to determine an optimal disassembly policy for products that are processed at the emerging remanufacturing facilities. The intention is to maximize the expected value that is retrieved from each such product unit and re-introduced in the corresponding supply chain, while disposing in an systematic and environmentally friendly manner whatever material remains from this re-processing. In principle, this problem can be formulated as a stochastic dynamic programming (DP) problem on the underlying disassembly graph, where the state of the decision making process is determined by the set of the currently extracted items and their quality status. The latter also determines the set of possible actions that are available for the disposition of the obtained items. But the probability distributions that determine the quality of the extracted items, and eventually govern the dynamics of the disassembly process, typically are not known a priori.[1] Hence, the optimal disassembly policy must be determined through techniques of incremental, real-time dynamic programming (also known as "reinforcement learning"). The relevant modeling of the optimal disassembly problem as a reinforcement learning problem and the application of the Q-learning algorithm on it can be found in [15]. However, in [16] it was also shown that by being defined on an acyclic state space, the reinforcement learning problem corresponding to the optimal disassembly problem is amenable to customized learning algorithms of PAC (Probably Approximately Correct) nature,

[1] These distributions are determined by the consumer behavior with respect to the considered product, which is an uncontrollable and unobservable part of the entire process.

with stronger convergence and performance guarantees than Q-learning. This new class of algorithms seek to learn the optimal policy by performing an adequate sampling of the outcomes of each of the actions that are available at each decision node. In particular, the amount of sampling that must be performed at each node is determined so that an actual optimal action will be selected with a certain probability. However, the sampling at a certain node can be performed only after the selected action at each of its children nodes has been determined, and therefore, the sampling process employed by the algorithms of [16] presents a precedence structure[2] that is similar to the precedence structure discussed in the above description of the network flow problem to be considered in this work. The implementation of the resulting sampling process is further complicated by the fact that the set of disassembly states – or the decision nodes – that are materialized during the processing of any single product unit is not fully controllable, since it is affected by the randomness in the quality of the items that are extracted during the performed disassembly steps. Hence, it is possible that the disassembly of a certain product unit does not contribute to the sampling effort, because the extracted artifacts might correspond to decision nodes that are already sufficiently sampled. When combined with the precedence requirements to be observed by the applied sampling process, this last effect gives rise to an interesting stochastic scheduling / routing problem that seeks the satisfaction of the entire set of the posed sampling requirements in a way that observes the aforestated precedence constraints, and minimizes, in expectation, the number of product units that must be disassembled in the process. The detailed formulation and a thorough study of this problem is presented in [17].

In the light of the above discussion, the main objectives and contributions of this work are as follows: First, it provides a complete, natural formulation of the dynamic flow control problem described above as a continuous-time optimal control problem. Second, it establishes that the initial formulation can be recast as a hybrid optimal control problem with controlled switching [18] and it exploits this finding in order to develop an *effective* solution approach to the problem that takes the form of a mixed integer programming (MIP) formulation [19]. Finally, the last part of the paper provides a complexity analysis of the problem, establishing its NP-hardness [20] and offering additional guidelines that can alleviate this increased complexity. In addition, this last part of the work highlights the affinity of the considered problem to some well known scheduling problems in the literature.

[2] or a sampling schedule

2 Problem definition and formulation

Problem definition A general description of the optimal flow control problem considered in this work is as follows: We are given a network, modeled by an acyclic, connected digraph $G = (V, E)$, where V and E denote respectively the sets of the graph nodes and edges. Furthermore, V is partitioned to two node classes, V^c and V^u. The sets of source and leaf nodes of graph G are respectively denoted by ${}^\bullet V$ and $V^\bullet$, and it is further assumed that ${}^\bullet V = \{v_0\}$.[3] In addition, in the following, $E^\bullet(v)$ denotes the set of edges emanating from node v, ${}^\bullet E(v)$ denotes the set of edges leading to node v, and ${}^\bullet E^\bullet(v)$ denotes the entire set of edges incident on v. Finally, $v^\bullet$ denotes the set of the immediate successors of node v, i.e., $v^\bullet$ collects all the terminating nodes of the edges $e \in E^\bullet(v)$. A fluid is pumped into this network through its source node v_0, at a constant rate of one unit of fluid per unit of time. Flow reaching a node $v \in V^u$ is distributed to its outgoing edges according to an uncontrollable and time invariant distribution $d_v =< d_v(e),\ e \in E^\bullet(v) >$. On the other hand, the distribution of the flow reaching a node $v \in V^c$ to its emanating edges is controllable and it can be varied over time. Finally, each node $v \in V$ has a fluid requirement $\bar{F}(v)$ associated with it, and it is also stipulated that node v can begin accumulating the incoming flow in order to fulfill its requirement $\bar{F}(v)$ only after all of its successor nodes in graph G, $v' \in V$, have fulfilled their own requirements, $\bar{F}(v')$. A node v that can proceed to the accumulation of its fluid requirement, $\bar{F}(v)$, is characterized as *activated*. An activated node is further characterized as *completed*, when the accumulated amount of fluid reaches the designated level $\bar{F}(v)$. The control problem considered in this work is the determination of a possibly time-varying routing policy for nodes $v \in V^c$, that will enable the completion of all the nodal requirements $\bar{F}(v),\ v \in V$, in *minimal time* (or equivalently, while pumping the minimal amount of fluid into the network).

Example An example problem instance is presented in Figure 1. In the depicted digraph, uncontrollable nodes are represented by black circles. The nodal fluid requirements are reported by the numbers in bold, on the right side of each node, and the distributions characterizing the routing pattern at the uncontrollable nodes are reported by the numbers on the left of the edges emanating from these nodes.[4]

[3]The assumption $|{}^\bullet V| = 1$ can also be satisfied for problem instances with $|{}^\bullet V| > 1$ through the addition of a new dummy node; the details are left to the reader.

[4]We also notice that when the network of Figure 1 is perceived in the context of the stochastic DP problem that motivated this work, the controllable (white) nodes correspond to decision nodes, and the arcs emanating

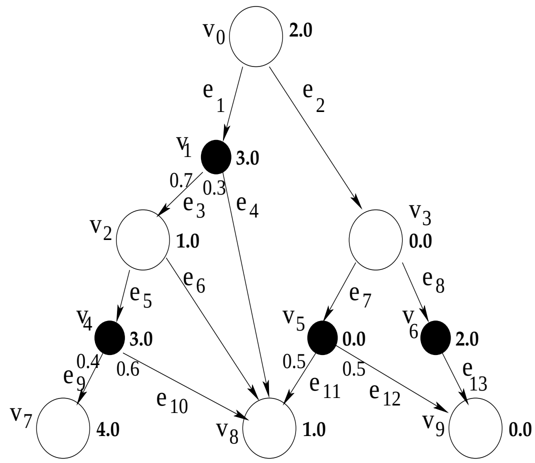

Figure 1: An example problem instance

Problem formulation as a continuous-time optimal control problem The considered problem can be naturally formulated as a continuous-time optimal control problem. In this modeling framework, the key "decision variables" are the functionals $f_e(t),\ t \in [0, \infty)$, defining a flow profile for each edge $e \in E$. We shall also use the functionals $F_v(t),\ t \in [0, \infty)$, to denote the evolution of the fluid accumulation at node $v \in V$, and the notation $I_{\{H\}}$ to represent the indicator variable defined by some predicate H. Then, the considered problem can be succinctly expressed as follows:

$$\min \int_0^\infty I_{\{F_{v_0}(t) \leq \bar{F}(v_0)\}} dt \tag{1}$$

s.t.

$$\sum_{e \in E^\bullet(v_0)} f_e(t) = 1.0 \tag{2}$$

$$\forall v \in V^c \backslash (V^\bullet \cup^\bullet V),\ \forall t \in [0, \infty), \quad \sum_{e \in E^\bullet(v)} f_e(t) = \sum_{e' \in ^\bullet E(v)} f_{e'}(t) \tag{3}$$

$$\forall v \in V^u \backslash V^\bullet, \quad \forall e \in E^\bullet(v),\ \forall t \in [0, \infty), \quad f_e(t) = d_v(e) \cdot \sum_{e' \in ^\bullet E(v)} f_{e'}(t) \tag{4}$$

$$\forall v \in V,\ \forall t \in [0, \infty), \quad F_v(t) = \int_0^t (\sum_{e \in ^\bullet E(v)} f_e(\tau)) \cdot I_{\{\forall v' \in v^\bullet,\ F_{v'}(t) \geq \bar{F}(v')\}} d\tau \tag{5}$$

from them are the available decisions. On the other hand, the uncontrollable (black) nodes model the impact of the randomness that might determine the outcome of some of the applied decisions. Obviously, in the context of such an interpretation of the considered problem and its flow dynamics, uncontrollable nodes should have a zero fluid requirement (since the sampling process takes place only at the decision nodes).

$$\forall e \in E,\ \forall t \in [0,\infty), \quad f_e(t) \geq 0 \tag{6}$$

Constraint 2 in the above formulation expresses the finiteness of the ingress capacity of the considered network and establishes the equivalence between the cumulative amount of fluid entering this network and the passage of time. Constraints 3 and 4 impose a flow balance requirement for the set of nodes $V \backslash ({}^{\bullet}V \cup V^{\bullet})$, with Constraint 4 further communicating the uncontrollable nature of the flow routing that takes place at nodes in V^u. Constraint 5 characterizes the accumulation of the fluid passing through the different nodes $v \in V$ towards the satisfaction of the corresponding nodal fluid requirements, and it ensures that the dynamics of this accumulation are in agreement with the precedence constraint defined in the introductory section. In particular, the presence of the indicator function $I_{\{\forall v' \in v^{\bullet}, F_{v'}(t) \geq \bar{F}(v')\}}$ in the integral of Equation 5 ensures that any amount of fluid passing through node v will be accounted towards the satisfaction of the corresponding fluid requirement, only if the fluid requirements for all the successor nodes of v have been fully satisfied. Finally, the problem objective function seeks the completion of all the nodal fluid requirements in minimal time (or in the light of Constraint 2, with a minimal amount of fluid induced into the network).

Reformulating the considered problem as a hybrid optimal control problem with controlled switching While the formulation of Equations 1-6 offers a succinct characterization of the considered problem, it is very cumbersome from a computational standpoint. However, next we present a structural property that will enable its transformation to a mixed integer program and will render it solvable through "off-the-shelf" optimization software [19]. The essence of this property is that the restriction of the original problem to flows $< f_e(t),\ e \in E,\ t \in [0,\infty) >$ that maintain a constant distribution at all nodes $v \in V$ between two consecutive completions of some nodal fluid requirements, does not compromise the optimality of the derived solution. This result can be stated and proven as follows:

Proposition 1 *Let $< f_e(t),\ e \in E;\ F_v(t),\ v \in V;\ t \in [0,\infty) >$ denote a feasible solution for the formulation of Equations 1-6, and consider a time interval $[t_1, t_2]$ such that*

$$\forall v \in V, \quad I_{\{F_v(t_1) < \bar{F}(v)\}} = I_{\{F_v(t_2) < \bar{F}(v)\}} \tag{7}$$

Then, there exists a solution $< f'_e(t),\ e \in E;\ F'_v(t),\ v \in V;\ t \in [0,\infty) >$ such that

$$\forall e \in E,\ \forall t \in [t_1, t_2],\ f'_e(t) = c_e \tag{8}$$

and

$$\int_0^\infty I_{\{F'_{v_0}(t)\leq \bar{F}(v_0)\}}dt = \int_0^\infty I_{\{F_{v_0}(t)\leq \bar{F}(v_0)\}}dt \tag{9}$$

Proof: Consider the flow $< f'_e(t),\ e \in E;\ t \in [0,\infty) >$ that is defined by flow f as follows:

$$\forall e \in E, \quad f'_e(t) = \begin{cases} \frac{1}{t_2-t_1}\int_{t_1}^{t_2} f_e(\tau)d\tau, & \text{if } t \in [t_1,t_2] \\ f_e(t), & \text{otherwise} \end{cases} \tag{10}$$

Clearly, the flow f' defined by Equation 10 satisfies the condition of Equation 8 and it also satisfies Constraint 6. Furthermore, the definition of f', together with the linearity of the integral, imply that f' is also feasible with respect to Constraints 2-4. Next we consider the fluid accumulations $< F'_v(t),\ v \in V,\ t \in [t,\infty) >$, that are induced by f' through the integral of Equation 5, and we establish that

$$\forall t \in \{t_1,t_2\},\ \forall v \in V, \quad F'_v(t) = F_v(t) \tag{11}$$

The validity of Equation 11 for $t = t_1$ follows immediately from the definitions of f' and F' (c.f., Equations 10 and 5). The validity of Equation 11 for $t = t_2$ can be established inductively as follows: First consider the set of leaf nodes and notice that for any such node $v \in V^\bullet$, $I_{\{\forall v' \in v^\bullet,\ F_{v'}(t)\geq \bar{F}(v')\}} = 1,\ \forall t \in [0,\infty)$, since these nodes have no successors. Hence,

$$\begin{aligned} \forall v \in V^\bullet, \quad F'_v(t_2) &= \\ F'_v(t_1) + \int_{t_1}^{t_2} \sum_{e\in {}^\bullet E(v)} f'_e(\tau)d\tau &= \\ F_v(t_1) + \int_{t_1}^{t_2} [\sum_{e\in {}^\bullet E(v)} \frac{1}{t_2-t_1}\int_{t_1}^{t_2} f_e(s)ds]d\tau &= \\ F_v(t_1) + \frac{1}{t_2-t_1}\int_{t_1}^{t_2} d\tau \int_{t_1}^{t_2} \sum_{e\in {}^\bullet E(v)} f_e(s)ds &= \\ F_v(t_1) + \int_{t_1}^{t_2} \sum_{e\in {}^\bullet E(v)} f_e(s)ds &= \\ F_v(t_2) & \end{aligned} \tag{12}$$

For the inductive step, consider a node $v \in V \backslash V^\bullet$, and suppose that

$$\forall v' \in v^\bullet, \quad F'_{v'}(t_2) = F_{v'}(t_2) \tag{13}$$

Then, if there exists a node $v' \in v^\bullet$ with $F'_{v'}(t_2) = F_{v'}(t_2) < \bar{F}(v')$, Constraint 5 implies that

$$F'_v(t_2) = F_v(t_2) = 0 \tag{14}$$

In the opposite case, $F'_{v'}(t_2) = F_{v'}(t_2) \geq \bar{F}(v')$, $\forall v' \in v^\bullet$, which combined with Equation 7 and the established validity of Equation 11 for $t = t_1$, further implies that

$$\begin{aligned} \forall t \in [t_1, t_2], \quad & I_{\{\forall v' \in v^\bullet,\ F'_{v'}(t) \geq \bar{F}(v')\}} = \\ & I_{\{\forall v' \in v^\bullet,\ F_{v'}(t) \geq \bar{F}(v')\}} = 1 \end{aligned} \tag{15}$$

But then, the equality of $F'_v(t_2)$ and $F_v(t_2)$ can be established through a computation similar to that presented in Equation 12. Finally, Equation 9 follows from the definition of f' (c.f. Equation 10) and the application of Equations 7 and 11 to node v_0. □

Proposition 1 implies that we can restrict the search for an optimal control law into the class of control laws that allow for a switch of the applied routing scheme only at the time points corresponding to the completion of some nodal fluid requirement. A more technical restatement of this result is that the original problem is reduced to a *hybrid optimal control problem with controlled switching* [18] where the control *"modes"* are defined by the completion status of the fluid requirements that are posed by the different nodes. A complete, formal characterization of the hybrid automaton that defines the aforementioned hybrid optimal control problem would be of a rather pedantic nature, and therefore, it is omitted. Instead, in the next section we capitalize upon the structure that is implied by this hybrid automaton, in order to motivate and introduce an alternative MIP formulation for the considered problem.

3 A MIP formulation for the considered problem

As indicated in the previous section, the search for an optimal control law for the considered flow control problem can be restricted to the class of control laws that allow for a switch of the applied routing scheme only at the time points corresponding to the completion of some nodal fluid requirement. In order to model the flow dynamics generated by this restricted class of control laws, we define the set of "*control modes*", $\mathcal{V}$, that contains all the possible partitions of the node set V, that satisfy the following two requirements: (i) They split V into two subsets, one containing the nodes that have their flow requirements completed, and its complement. (ii) The completions indicated at each mode are consistent with the precedence constraints

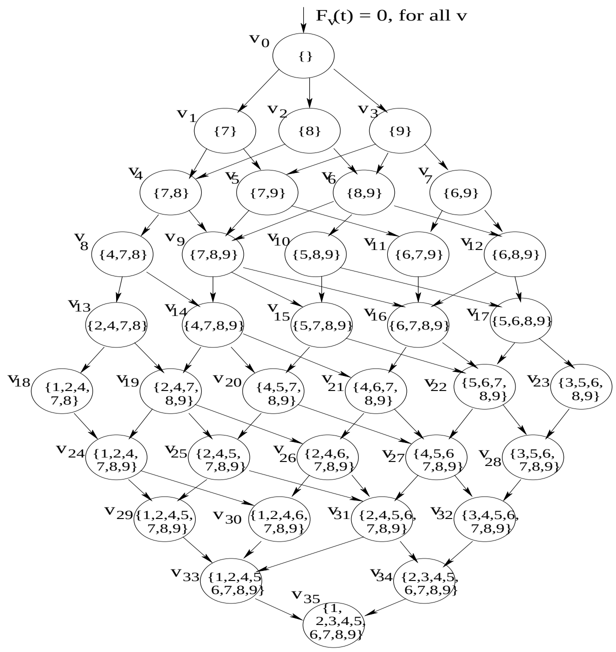

Figure 2: The control modes and interconnecting transitions of the graph $\mathcal{G}$ corresponding to the example problem instance of Figure 1

expressed by Equation 5. A systematic enumeration of the mode set $\mathcal{V}$ can be obtained by a search process that starts from the initial control mode ν_0, where all nodes have their fluid requirements uncovered, and subsequently, it reaches out to the remaining control modes by flagging one node at a time as completed, while observing the precedence constraints that are imposed by the structure of graph G. The resulting graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has an acyclic structure, in which the different modes are layered according to their number of completed nodes; for further reference, we shall characterize these layers of $\mathcal{G}$ as $L_0,\ L_1, \ldots, L_{|V|-1}$. The mode graph $\mathcal{G}$ corresponding to the problem instance of Figure 1 is depicted in Figure 2.

Given the mode graph $\mathcal{G}$ defined in the previous paragraph, any solution belonging to the restricted space of control laws characterized at the beginning of this section, can be effectively represented by the following two elements: (i) a directed path from the source to the sink node of $\mathcal{G}$, and (ii) the nodal fluid accumulations that take place at each visited mode. In order to characterize the aforementioned paths of $\mathcal{G}$, we introduce the binary variables $\delta_\nu, \nu \in \mathcal{V}$, and we stipulate that $\delta_\nu = 1$ *iff* mode ν belongs on the path followed by the considered solution. Obviously, the pricing of the variables $\delta_\nu,\ \nu \in \mathcal{V}$, must be restricted by an additional set of constraints, which will ensure that they express meaningful paths from the source to the sink node of graph $\mathcal{G}$. Letting ${}^{\bullet}\nu$ denote the immediate predecessors of any mode $\nu \in \mathcal{V}$, such a constraint set can be structured as follows:

$$\forall i \in \{0, 1, \ldots, |V| - 1\}, \quad \sum_{\nu \in L_i} \delta_\nu = 1 \tag{16}$$

$$\forall \nu \in \mathcal{V} \backslash \{\nu_0\}, \quad \delta_\nu \leq \sum_{\nu' \in {}^{\bullet}\nu} \delta_{\nu'} \tag{17}$$

$$\forall \nu \in \mathcal{V}, \quad \delta_\nu \in \{0, 1\} \tag{18}$$

The validity of the above constraint set can be established by noticing that the combination of Constraints 16 and 18 expresses the fact that any path from the source to the sink mode of $\mathcal{G}$ has exactly one node belonging to each of the layers of $\mathcal{G}$, while Constraint 17 enforces the path feasibility with respect to the connectivity of $\mathcal{G}$.

In order to characterize all the nodal fluid accumulations that can take place at any given mode $\nu \in \mathcal{V}$, we proceed as follows: First, we introduce the set of auxiliary variables $\{X_e^\nu\}$, which denote the total amount of fluid conveyed through the edges $e \in E$ during the network sojourn in the considered control mode. Clearly, $\{X_e^\nu, \ e \in E\}$ must satisfy the following balance constraints:

$$\forall v \in V^c \backslash (V^{\bullet} \cup^{\bullet} V), \sum_{e \in E^{\bullet}(v)} X_e^\nu = \sum_{e' \in {}^{\bullet}E(v)} X_{e'}^\nu \tag{19}$$

$$\forall v \in V^u \backslash V^{\bullet}, \ \forall e \in E^{\bullet}, \ X_e^\nu = d_v(e) \cdot \sum_{e' \in {}^{\bullet}E(v)} X_{e'}^\nu \tag{20}$$

$$\forall e \in E, \quad X_e^\nu \geq 0 \tag{21}$$

Second, we introduce the variables $\{\Delta F_v^\nu\}$ that denote the total amount of fluid accumulated at an activated node v during the network sojourn in the considered control mode. This new set of variables must satisfy the following constraints:

$$\forall \text{ non-activated or completed node } v \text{ in } \nu, \quad \Delta F_v^\nu = 0 \tag{22}$$

and

$$\forall \text{ activated but uncompleted node } v \text{ in } \nu, \quad \Delta F_v^\nu = \begin{cases} \sum_{e' \in {}^{\bullet}E(v)} X_{e'}^\nu, & \text{if } v \neq v_0 \\ \sum_{e' \in E(v)^{\bullet}} X_{e'}^\nu, & \text{otherwise} \end{cases} \tag{23}$$

In order to complete the characterization of the space of the control laws considered by the proposed formulation, we must also link the pricing of the variables $\delta_\nu, \nu \in \mathcal{V}$, to the pricing of the variables $X_e^\nu, e \in E, \nu \in \mathcal{V}$, that define the fluid accumulations at the different control modes. For this, consider a pricing of the variables $\delta_\nu,\ \nu \in \mathcal{V}$, according to any pattern that satisfies Constraints 16–18. Then, it should be clear from the above discussion, that any control law which is in agreement with this pricing, will engage only control modes ν with $\delta_\nu = 1$. Control modes ν with $\delta_\nu = 0$ will not contribute anything to the required fluid accumulations. In the light of Equations 19–21, this effect is communicated in the proposed formulation by setting

$$\forall \nu \in \mathcal{V}, \quad \sum_{e \in E^\bullet(v_0)} X_e^\nu \leq \delta_\nu \cdot M^\nu \tag{24}$$

The parameter M^ν appearing in the above equation is of the, so called, "big-M" type, and it must be adequately large to avoid any unintentional / artificial constraining of the left hand side of Equation 24, in the case that $\delta_\nu = 1$. In the considered problem context, a pertinent value for M^ν is provided by the combined fluid requirement of all the nodes that are activated but not completed in mode ν.

Equations 16–24 provide a complete characterization of the entire set of flows presenting the structure that was identified by Proposition 1. It remains to express the constraints arising by the nodal fluid requirements and the objective function that measures the performance of any such satisficing flow. The constraints imposing the satisfcation of the nodal fluid requirements can be succinctly expressed as follows:

$$\forall v \in V, \quad \sum_{\nu \in \mathcal{V}} \Delta F_v^\nu = \bar{F}(v) \tag{25}$$

On the other hand, the stated objective of minimizing the overall fluid losses can be expressed by setting the formulation objective to:

$$\min \sum_{\nu \in \mathcal{V}} \sum_{e \in E^\bullet(v_0)} X_e^\nu \tag{26}$$

The following theorem summarizes all the above discussion, by providing a formal statement for the validity of the MIP formulation of Equations 16–26 as a generator for an optimal control law for the flow control problem under consideration. It also articulates the construction of the optimal flow control law that is implied by the obtained optimal solution to the MIP formulation. Finally, we notice that the notation ${}^\bullet e$, that is used in the statement of the following result, implies the starting node for edge e.

Theorem 1 *Consider the MIP formulation defined by Equations 16–26 and let* $X_e^{\nu^*}$, $e \in E$, $\nu \in \mathcal{V}$, *denote the cumulative modal flows established by its optimal solution. Also, consider the flow functionals* $f_e(t)$, $e \in E$, $t \in [0, \infty)$, *that are obtained from* $< X_e^{\nu^*}, e \in E, \nu \in \mathcal{V} >$ *by setting* $\forall e \in E$,

$$f_e(t) = \frac{\sum_{\nu \in L_i} X_e^{\nu^*}}{\sum_{e' \in E^\bullet(^\bullet e)} \sum_{\nu \in L_i} X_{e'}^{\nu^*}} \tag{27}$$

if there exists an $i \in \{0, 1, \ldots, |V|-1\}$ *such that* $\sum_{j=0}^{i-1} \sum_{\nu \in L_j} \sum_{e' \in E^\bullet(v_0)} X_{e'}^{\nu^*} \leq t < \sum_{j=0}^{i} \sum_{\nu \in L_j} \sum_{e' \in E^\bullet(v_0)} X_{e'}^{\nu^*}$, *and*

$$f_e(t) = 0 \tag{28}$$

otherwise. Then, $< f_e(t), e \in E, t \in [0, \infty) >$ *constitutes an optimal flow for the original formulation of Equations 1–6. Furthermore,*

$$\int_0^\infty I_{\{F_{v_0}(t) \leq \bar{F}(v_0)\}} dt = \sum_{\nu \in \mathcal{V}} \sum_{e \in E^\bullet(v_0)} X_e^{\nu^*} \tag{29}$$

□

Example: When applied to the example problem instance of Figure 1, by means of the mode graph depicted in Figure 2, the aforementioned MIP formulation returned a solution that employed the mode sequence $< \nu_0, \nu_2, \nu_4, \nu_8, \nu_{13}, \nu_{19}, \nu_{24}, \nu_{29}, \nu_{33}, \nu_{35} >$ (c.f. Figure 2). Furthermore, modes ν_{13}, ν_{24}, and ν_{33} in this sequence were used just as transitional modes and they involved no fluid accumulation.[5] The cumulative flow conveyed through the different graph edges at each of the remaining modes is reported in Table I. It can be easily checked that the solution is consistent with the nodal fluid requirements communicated in Figure 1, and that the total volume of fluid induced in the network is equal to 27 units. Finally, we mention for completeness that the compilation and solution of the MIP formulation through CPLEX took 0.2 secs.

4 Complexity considerations

The MIP formulation developed in the previous section provides an effective methodology for the solution of the optimal flow control problem considered in this work. However, when

[5] An explanation of this effect can be found in the discussion provided at the end of the following section.

Table 1: The optimal solution of the MIP formulation for the example problem instance of Figure 1

ν	0	2	4	8	19	29	35
X_1^ν	1.3889	12.8968	4.2857	1.4286	3.00		
X_2^ν						2.00	2.00
X_3^ν	0.9722	9.0278	3.00	1.00	2.10		
X_4^ν	0.4167	3.8690	1.2857	0.4286	0.90		
X_5^ν	0.9722	9.0278	3.00				
X_6^ν				1.00	2.10		
X_7^ν							2.00
X_8^ν						2.00	
X_9^ν	0.3889	3.6111	1.20				
X_{10}^ν	0.5833	5.4167	1.80				
X_{11}^ν							1.00
X_{12}^ν							1.00
X_{13}^ν						2.00	

viewed from the standpoint of its computational efficiency, this approach can be restricted by the general fact that the solution of MIP formulations requires a computational effort that grows exponentially with respect to the number of the involved variables [20]. In the considered case, things are further complicated by the fact that the number of variables and constraints involved in the MIP formulation of Section 3 are themselves exponentially related to the size of the elements that define the original problem. More specifically, as revealed by Equations 16–25, many of the variables and the constraints appearing in the proposed MIP formulation are determined by the structure of the modal graph $\mathcal{G}$, and it is easy to see that, in the worst case, the number of modes in $\mathcal{G}$, $|\mathcal{V}|$, will be equal to $2^{(|V|-1)}$, where $|V|$ is the number of nodes of the problem-defining graph G. Motivated by these remarks, in this section we take a closer look at the computational complexity of the original problem, defined by Equations 1–6. Along these lines, the key result of this section is that the decision version of the considered optimal flow control problem is NP-complete [20], a finding that renders the considered optimization problem itself NP-hard, and excludes the possibility of developing very efficient algorithms for it. Nevertheless, the last part of the section provides some additional remarks that can alleviate the increased complexity of the proposed MIP formulation, by taking advantage of some further structure that might be present in the problem defining graph G and its nodal fluid requirements.

Establishing the NP-hardness of the considered problem In order to study the computational complexity of the considered optimal control problem, we work with its *decision version*, which is defined as follows: *Given the problem defining graph G and a set of nodal fluid requirements associated with its nodes $v \in V$, does there exist a set of flow profiles $< f_e(t), e \in E, t \in [0, \infty) >$ that satisfies all the nodal fluid requirements in time less than or equal to some given constant Γ (or, equivalently, with the total flow injected into the graph not exceeding Γ)?* The next theorem establishes that this new problem version is NP complete [20].

Theorem 2 *The decision version of the optimal flow control problem considered in this work is NP-complete.*

Proof: The result of Theorem 2 will be obtained by providing a polynomial reduction to the considered problem of another scheduling problem that is known to be NP-complete. This scheduling problem is described as follows: We are given two processors, P_1 and P_2, and n tasks, $T_1, T_2, \ldots, T_n$, that must be executed by means of the two processors. More specifically, each task $T_j, j = 1, \ldots, n$, corresponds to a certain workload, W_j, and it can be executed either on processor P_1 or processor P_2. However, each processor executes each of the given tasks with a different speed that depends on, both, the task and the processor. In the following, we shall use the notation $R_{ij}, i = 1, 2, j = 1, \ldots, n$, to denote the processing speed of task T_j when executed by processor P_i. Furthermore, without loss of generality, in the following we shall assume that W_i and R_{ij} have been scaled so that $R_{ij} < 1$. Obviously, the amount of time that takes to execute task T_i on processor P_i, under the assumption that processor P_i is completely dedicated to the processing of task T_j, is equal to $t_{ij} = W_j / R_{ij}$. In addition, it is assumed that the execution of any task on any of the two processors can be *preempted* and tasks can be reassigned to processors in order to complete their remaining workload. A last problem feature is that the set of tasks $\{T_i, i = 1, \ldots, n\}$ is organized into *"chains"*, i.e., a partial order where every task has at most one predecessor and at most one successor. The question to be resolved is whether there exists a schedule for executing the n tasks on the two processors such that

Task	Immediate Predecessor	Immediate Successor	W_j	R_{1j}	R_{2j}
1	-	2	2	0.9	0.5
2	1	-	3	0.8	0.6
3	-	-	1	0.5	0.4

Table 2: An example instance of the scheduling problem considered in the proof of Theorem 2

the total time that is needed to complete the execution of all the tasks[6] is no more than a given constant C. According to [21], this problem is NP-complete. Table 2 provides a concrete instance of this scheduling problem which involves three tasks, T_1, T_2 and T_3, organized into two chains. The task workloads W_i and the relevant processing speeds R_{ij} are also reported in the provided table.

Next we show that, given an instance of the aforementioned scheduling problem, we can construct an instance of the decision problem considered in Theorem 2 such that (a) any solution for each of these two problems can be translated to a solution for the other problem, and (b) the ratio of the corresponding objective values for any such solution pair is equal to a constant. We start with the specification of the acyclic graph G. G is built from a modification of the chains that express the task precedence constraints in the original scheduling problem, which are further augmented with two additional nodes: (i) node v_0 that constitutes the single *"source"* node of the graph, and (ii) node v_d that represents a *"dumping"* node that absorbs the parts of the induced flow that correspond to processor idleness and inefficiencies in the original scheduling context. The "source" node v_0 is connected to the rest of the network through a number of subnets that represent all the possible allocations of the two processors, P_1 and P_2, to the contesting tasks. More specifically, the decision to load only one of the two processors, say processor $P_i, i = 1, 2$, with some task $T_j, j = 1, \ldots, n$, while idling the other processor, is modeled by the subnet depicted in Figure 3-(a). Similarly, the decision to load processor P_1 with task T_j and processor P_2 with task T_k is modeled by the subnet represented in Figure 3-(b). Furthermore, every edge belonging to some of the chains of the original scheduling problem is replaced by the subnet depicted in Figure 3-(c). The practical implication of this replacement is that any flow reaching the node corresponding to some task T_j through any of the subnets depicted in Figures 3-(a,b), can be used only locally, for the satisfaction of the flow requirements

[6] This time is known as the schedule makespan in the relevant theory.

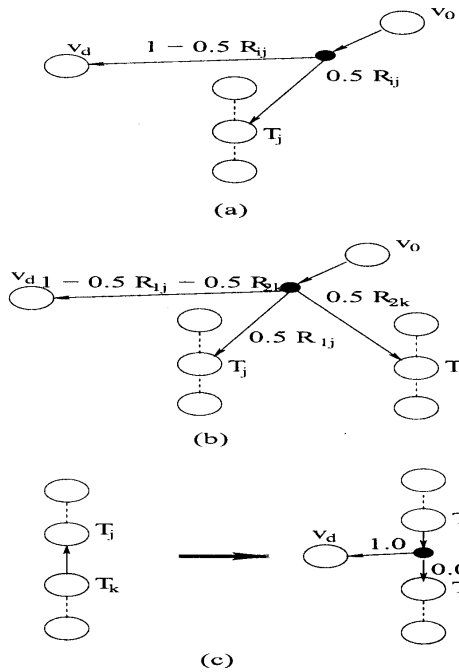

Figure 3: The main constructs employed in the reduction for the proof of Theorem 2

of that node. The flow requirements for the nodes corresponding to the tasks $T_j, j = 1, \ldots, n$, are set equal to the corresponding workloads W_j. The flow requirements for nodes v_0, v_d and all the uncontrollable nodes appearing in the subnets of Figures 3-(a,b,c) are set equal to zero. A more concrete example of this construction is provided in Figure 4, which depicts the flow control problem instance corresponding to the scheduling problem instance of Table 2.

It is obvious from the above discussion and from the provided example that the construction procedure for the induced flow control problem is of polynomial complexity with respect to the size of the departing scheduling problem instance. Furthermore, given a feasible schedule for the original scheduling problem, one can construct a satisficing flow profile for the induced flow control problem, by mapping each phase of the provided schedule to a flow volume that is equal to two times the duration of the considered phase and it is injected in the induced graph through the edge that emanates from node v_0 and models the processor loading pattern that corresponds to that phase. Similarly, every flow profile that satisfies the nodal fluid requirements in the induced graph G can be translated to a viable schedule for the original scheduling problem with a makespan equal to half the total volume of the induced flow. In particular, Proposition 1 implies that any flow routing pattern that is exerted at node v_0 and is of interest in this

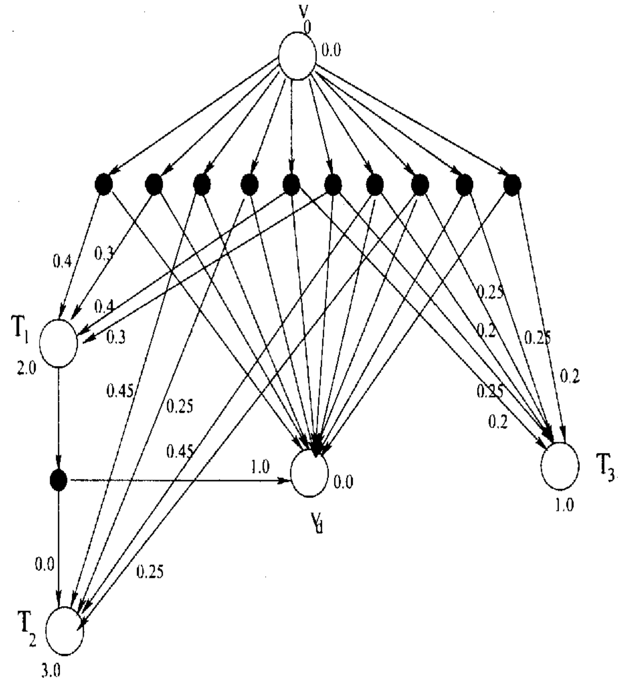

Figure 4: The flow control problem that is induced by the scheduling problem of Table 2

analysis, will consist of a sequence of flow distributions that remain constant over some time interval of finite non-zero length. Each of these flow distributions defines a phase in the schedule that is induced for the original problem, having a duration equal to half the duration of the application of the considered distribution (or, equivalently, to half the total flow conveyed during the application of this distribution). Furthermore, every such flow distribution that conveys all the injected flow to a single edge e in $E^{\bullet}(v_0)$ implies the processor loading pattern that corresponds to edge e according to the modeling logic delineated in the previous paragraph. On the other hand, a flow distribution that conveys flow into the graph G through more than one edges in $E^{\bullet}(v_0)$, let's say edges $e_1, e_2, \ldots, e_k$, implies the division of the corresponding phase in the schedule of the original problem into a number of k sub-phases; the i-th sub-phase in this set applies the loading pattern corresponding to edge e_i and its duration covers the f_{e_i}, percentage of the duration of the total phase. Thus, it is clear from the above, that the original scheduling problem will have a feasible schedule makespan of no more than C time units, if and only if the induced flow control problem has a flow profile that covers all the nodal fluid requirements of the induced graph G with a total flow volume no higher than $2C$.

The above results establish the NP-hardness [20] of the decision problem of Theorem 2. The problem NP-completeness is established by further noticing that the validity of any tentative solution for it can be verified with polynomial effort in terms of the size of the underlying problem defining elements. The details of this argument are straightforward and they are left to the reader. □

According to standard results presented in [20], Theorem 2 further implies that the optimization problem considered in this work is NP-hard. We state this result as a corollary.

Corollary 1 *The optimal flow control problem considered in this work is NP-hard.*

Alleviating the computational complexity of the proposed MIP formulation In the last part of this section we provide a series of observations that can alleviate the complexity of the MIP formulation presented in Section 3, by taking advantage of some additional structure that might be present in the problem defining graph G and its nodal visitation requirements.

The first of these observations concerns the dependence of the size of the mode graph $\mathcal{G}$ upon the precedence constraints for the satisfaction of the nodal fluid requirements, that are expressed by the structure of the problem defining graph G. In particular, it should be clear that, for any given number of nodes $|V|$ in G, the stricter the ordering imposed by G on V, the smaller the number of the possible modes, $|\mathcal{V}|$, that is encoded in $\mathcal{G}$. Hence, in many practical situations, the number of modes in graph $\mathcal{G}$ will be substantially smaller than $2^{(|V|-1)}$, which essentially corresponds to a very "flat" graph G where every node other than the "source" belongs to a single layer L_1.[7]

An additional reduction to the size of $\mathcal{G}$ can be obtained in the case where there are nodes $v \in V$ with zero fluid requirements. While we opted to include these nodes in the construction of the graph $\mathcal{G}$ of Figure 2 for simplicity and clarity of the relevant presentation, it should be obvious that nodes with zero fluid requirements do not contribute anything substantial in the computation of the optimal solution of the MIP formulation. Therefore they can be omitted during the specification of the underlying control modes.[8]

[7]Furthermore, it is easy to see that when defined on such a flat graph structure, the optimal control problem of Equations 1–6 has a very simple optimal solution, and therefore, there is no need to resort to the solution of the MIP formulation of Section 3 and the deployment of the mode graph $\mathcal{G}$.

[8]The insignificance of the nodes with zero fluid requirements for the determination of the optimal solution of the MIP formulation developed in Section 3, is manifested in the numerical example provided in that section, by the fact that modes ν_{13}, ν_{24} and ν_{33} correspond to zero total flows. These three modes correspond respectively

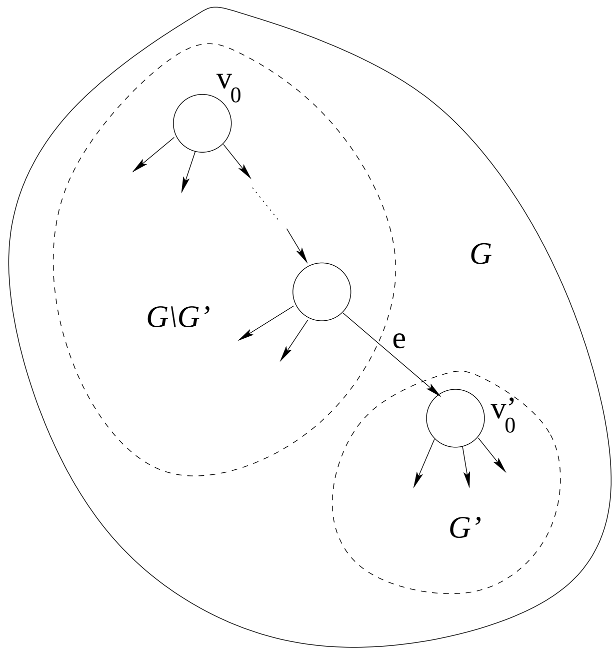

Figure 5: Special structure for the problem defining graph G that can enable efficient solution of the considered optimal control problem through decomposition

A third opportunity for controlling the increased computational complexity that results from the non-polynomial size of the mode graph $\mathcal{G}$ is provided by the appearance of a structure in the problem defining graph G similar to that depicted in Figure 5. Clearly, the total flow necessary for meeting the nodal fluid requirements of the part of the graph accessed through the depicted edge e depends only on the routing of the flow conveyed through e and this decision does not affect the satisfaction of the nodal fluid requirements in the remaining part of the graph. Hence, in the depicted case, the overall problem can be decomposed to the solution of the subproblem concerning the optimal satisfaction of the nodal requirements in subgraph G', and the subsequent solution of the optimal control problem for the original graph G, where, however, the sub-graph G' has been substituted by its source node v_0' with a corresponding fluid requirement equal to the value of the optimal solution for the first subproblem. Obviously, this decomposition can be applied iteratively on the derived subproblems and, when applicable, it can lead to significant reductions of the overall computational complexity.

Finally, we notice that, in certain cases, it might be possible to derive efficient solutions for the considered optimal flow control problem by leveraging results from the theory of parallel

to the completion of the (zero) fluid requirements for nodes 9, 5 and 3 in the graph G of Figure 1.

machine scheduling [21]. In particular, this theory contains a considerable number of cases that admit an optimal solution of polynomial complexity. If it is possible to establish that the problem instance under consideration reduces to such a parallel machine scheduling problem of polynomial complexity, then, this problem instance can be solved by solving the corresponding scheduling problem. The relevant reductions will employ constructs and arguments similar to those employed in the proof of Theorem 2.

5 Conclusion

This paper introduced a novel optimal flow control problem, provided a series of formulations for it that enabled its effective solution, and investigated the underlying computational complexity by establishing its NP-hardness. Future work will investigate the implications of the derived results for the stochastic routing problem of [17] and the optimal disassembly planning problem, that motivated the problem considered in this work.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[2] B. Kotnyek, "An annotated overview of dynamic network flows," INRIA, Tech. Rep. 4936, 2003.

[3] S. E. Lovetskii and I. I. Melamed, "Dynamic flows in networks," *Automation and Remote Control*, vol. 48, pp. 7– 29, 1987.

[4] J. E. Aronson, "A survey of dynamic network flows," *Annals of Operations Research*, vol. 20, pp. 1– 66, 1989.

[5] W. B. Powell, P. Jaillet, and A. Odoni, "Stochastic and dynamic networks and routing," in *Handbook in Operations Research and Management Science – Network Routings*, M. O. Ball et. al., Ed. Elsevier Science, 1995.

[6] A. N. Rybko and A. L. Stolyar, "Ergodicity of stochastic processes describing the operation of open queueing networks," *Problems of Information Transmission*, vol. 28, pp. 199–220, 1992.

[7] J. G. Dai, "On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models," *Annals of Applied Probability*, vol. 5, pp. 49–77, 1995.

[8] C. Maglaras, "Discrete-review policies for scheduling stochastic networks: trajectory tracking and fluid-scale asymptotic optimality," *Ann. Appl. Prob.*, vol. 10, pp. 897–929, 2000.

[9] D. Bertsimas and D. Gamarnik, "Asymptotically optimal algorithms for job shop scheduling and packet switching," *Journal of Algorithms*, vol. 33, pp. 296–318, 1999.

[10] D. Bertsimas and J. Sethuraman, "From fluid relaxations to practical algorithms for job shop scheduling: The makespan objective," *Mathematical Programming*, vol. 92, pp. 61–102, 2002.

[11] T. Bountourelis and S. Reveliotis, "Optimal node visitation in stochastic digraphs," *IEEE Trans. on Automatic Control*, vol. 53, pp. 2558–2570, 2008.

[12] D. Bertsimas, D. Gamarnik, and J. Tsitsiklis, "Stability conditions for multi-class queueing networks," *IEEE Trans. Autom. Control*, vol. 41, pp. 1618–1631, 1996.

[13] G. Weiss, "On the optimal draining of re-entrant fluid lines," Georgia Tech and Technion, Tech. Rep., 1994.

[14] S. Meyn, *Control Techniques for Complex Networks.* Cambridge, UK: Cambridge University Press, 2008.

[15] S. A. Reveliotis, "Uncertainty management in optimal disassembly planning through learning-based strategies," *IIE Trans.*, vol. 39, pp. 645–658, 2007.

[16] S. A. Reveliotis and T. Bountourelis, "Efficient PAC learning for episodic tasks with acyclic state spaces," *Journal of Discrete Event Systems: Theory and Applications*, vol. 17, pp. 307–327, 2007.

[17] T. Bountourelis and S. Reveliotis, "Optimal node visitation in acyclic stochastic digraphs with multi-threaded traversals and internal visitation requirements," *Journal of Discrete Event Systems: Theory and Applications*, vol. 19, pp. 347–376, 2009.

[18] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, pp. 31–45, 1998.

[19] W. L. Winston, *Introduction To Mathematical Programming: Applications and Algorithms, 2nd ed.* Belmont, CA: Duxbury Press, 1995.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York, NY: W. H. Freeman and Co., 1979.

[21] M. Pinedo, *Scheduling.* Upper Saddle River, NJ: Prentice Hall, 2002.