European Journal of Operational Research 256 (2017) 713–728

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Modeling elements and solving techniques for the data dissemination problem

Ronan Bocquillon, Antoine Jouglet*

Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, Compiègne cedex 60203, France

ARTICLE INFO

Article history:
Received 2 July 2015
Accepted 8 July 2016
Available online 1 August 2016

Keywords:
Combinatorial optimization
Dominance rules
Preprocessing procedures
Systems of systems
Data transfer problem

ABSTRACT

Systems of Systems (SoS) are collections of non-homogeneous, independent systems that interact to provide services. These systems can opportunistically share data during contacts that arise whenever two entities are close enough to each other. It is assumed in this paper that all contacts can be reliably predicted, *i.e.* the mobility of every system can be reliably estimated. A datum is split into several identified datum units to be delivered to a subset of recipient systems. During a contact, a given emitting system can transmit to a given receiving system one of the datum units that it possesses. The dissemination problem consists in finding a transfer plan which enables all the datum units to be transmitted from the sources (the systems that possess datum units from the beginning) to all the recipient systems. In this paper, we propose dominance-rule-based techniques for solving the data dissemination problem. In particular, we describe preprocessing procedures and some integer-linear-programming formulations to solve the problem.

1. Introduction

Systems of systems (or SoS) have been defined in many ways. One practical definition is that systems of systems are "*supersystems*" comprising other elements that are themselves complex, independent operational systems, all interacting to achieve a common goal (Jamshidi, 2008). If subsystems are not permanently connected, they must opportunistically make use of *contacts* that arise when entities are close enough to each other. These exchanges enable them to collaborate and route information from a subset of sources to a subset of recipient systems. This collaboration may be necessary, for instance, when contact durations are relatively short with respect to the volume of information to be disseminated. Here the information needs to be split up and possibly routed through non-recipient messenger systems whose role is to carry and forward data. Existing works have already looked at this kind of environment, in both opportunistic (Belblidia, De Amorim, Costa, Leguay, & Conan, 2011) and delay-tolerant networking (Fall, 2003). Most of the time, no assumptions are made about the contacts that occur between the systems, although for many applications it is quite possible to make realistic predictions about node mobility and contacts. Such applications include satellite networks (where the trajectories of subsystems depend on straightforward physics), public transportation systems (Pentland, Fletcher, & Hasson, 2004), and fleets of drones.

The present paper addresses this problem of making use of knowledge about possibilities of collaboration (Hay & Giaccone, 2009; Merugu, Ammar, & Zegura, 2004) when information needs to be routed from sources to destinations within a given time horizon. The fundamental question is which elements of the information should be transferred from which system to which system when contacts occur.

This problem has exercised an increasing number of researchers over the last decade.

Alonso and Fall (2003), for instance, proposed a linear formulation for computing a minimum delay transfer plan with respect to a set of nodes, a set of contacts and a set of messages. Available links need to be assigned to data such that every message can travel through the network from its sender to its receiver. The formulation incorporates constraints that are to be found in real applications, such as transmission delays, propagation delays and buffer capacities. As in most of the other works presented below, data transfers are modeled by unidentified numbers of bytes to be transmitted through a dynamic transportation graph. The problem can therefore be seen as a *dynamic* multi-commodity flow problem (Even, Itai, & Shamir, 1976) in which messages are the commodities and edge capacities are time-varying. The main drawback here is that flow constraints implicitly forbid duplication of data, making such approaches unsuitable for multicast and multisource situations. In the present paper, we instead consider

* Corresponding author. Fax: +33 03 44 23 44 77.
E-mail addresses: ronan.bocquillon@hds.utc.fr (R. Bocquillon), antoine.jouglet@hds.utc.fr (A. Jouglet).

http://dx.doi.org/10.1016/j.ejor.2016.07.013
0377-2217/

a set $\mathcal{N} = \{1, 2, \dots, q\}$ of q interacting mobile nodes (systems) and a single set $\mathcal{D} = \{1, 2, \dots, u\}$ of u identified "datum units", representing unitary, indivisible fragments of data, *e.g.*each datum unit might be a block of pixels corresponding to one high-resolution satellite picture. Initially, $\mathcal{D}$ (that we will sometimes refer to as "the datum") is distributed over different nodes $i \in \mathcal{N}$, the data sources, each of which holds a subset $\mathcal{O}_i \subseteq \mathcal{D}$ of datum units. The whole datum must then be transmitted within the allotted time to the subset $\mathcal{R} \subseteq \mathcal{N}$ of recipient nodes. The recipient nodes are required to obtain all the datum units. To our knowledge, no paper has so far addressed the multi-source case, despite its relevance if resulting algorithms can be executed on-line, such as when routing tables need to be refreshed dynamically following new predictions on node mobility or connectivity.

Alonso's and Fall's works were subsequently extended by Jain, Fall, and Patra (2004), who in particular proposed four oracles to compare the performances of routing procedures in terms of the amount of knowledge of network topology that they require. For example, the *contacts oracle* can answer any question regarding the contacts. Computational tests showed as expected that the greater the available knowledge, the better the performances. These oracles were extended by Zhao, Ammar, and Zegura (2005) to take multicasting protocols into account (*e.g.*the *membership oracle* answers questions about group dynamics). In the present paper we consider that all the oracles are available.

In order to address higher-dimensional problems certain assumptions have been proposed. For example, Handorean, Gill, and Roman (2004) defined *atomic contacts*, where contact durations (as opposed to inter-contact durations) are assumed to be instantaneous (both propagation and transmission delays are therefore disregarded). Hay and Giaccone (2009) made the same assumption, and proposed a particularly interesting model that they called the *event-driven graph*. As the graph is *time-independent* and polynomial in size with respect to the number of contacts in the instance, very basic tools from graph theory can be used to solve numerous problems straightforwardly. So, for example, the authors solve shortest-path or max-flow subproblems to minimize the delay or to maximize the network throughput. In the present paper we also use the idea of atomic contacts. We define a sequence of contacts $\sigma = [\sigma_1, \sigma_2, \dots, \sigma_m]$ of m ordered pairs $\sigma_c \in \mathcal{N}^2$ of nodes. During a contact $(s, r) \in \sigma$, the receiving node r can receive from the sending node s at most one datum unit $k \in \mathcal{D}$ that is held in s's buffer at that time (either from the outset or as a result of previous contacts). r is assumed to be in possession of k following the contact. In the following, s_c and r_c denote the sending and the receiving nodes at each contact $\sigma_c = (s_c, r_c) \in \sigma$. Buffers are assumed to be infinite and network failures are disregarded.

The *dissemination problem* is finding a *transfer plan* (a routing scheme) that minimizes the dissemination length, *i.e.* the number of contacts used to transmit the datum to all the recipient nodes. Instances of this problem are defined by one set of nodes $\mathcal{N} = \{1, 2, \dots, q\}$, one datum $\mathcal{D} = \{1, 2, \dots, u\}$, q sets $\mathcal{O}_i \subseteq \mathcal{D}$ $(i \in \mathcal{N})$ corresponding to the datum units possessed by the nodes from the outset, one sequence σ of m contacts $(s_c, r_c) \in \mathcal{N}^2$, and one subset of recipients $\mathcal{R} \subseteq \mathcal{N}$. Every instance can be represented with an *evolving graph* (Ferreira, 2004), a time-dependent graph model proposed by Ferreira and described in Section 2.1. The concept of transfer plan will also be formulated in Section 2.1. A transfer plan is a solution to the dissemination problem.

For those who want to go further – we refer to the *delay-tolerant networking research group* The delay-tolerant networking research group, Voyiatzis' survey (Voyiatzis, 2012), and Zhang's survey (Zhang, 2006) for their extensive review of the literature. The large number of papers that they reference, reflects a high level of interest in the problem of routing in intermittently connected networks.

In a previous paper (Bocquillon, Jouglet, & Carlier, 2015), we have shown that the dissemination problem is solvable in polynomial time when there is only one datum unit to be transferred (*i.e.* $u = 1$), or when there is only one recipient (*i.e.* $|\mathcal{R}| = 1$). Besides, a polynomial time algorithm has been provided for the case where the number of datum units and the number of recipients are both upper bounded by given constant numbers. The general problem is shown to be NP-Hard when the number of recipients and the number of datum units are greater than or equal to 2. Therefore, in this paper we propose practical procedures to solve the problem efficiently. To our knowledge, this is the first time such procedures are proposed.

We are first going to propose a few *dominance rules* for the problem. These yield conditions on which a subset of the search space considered to solve the problem can be ignored. Thereafter, we will propose algorithms which make use of these rules to deduce additional constraints, and this way, eliminate dominated solutions (solutions which can be ignored according to the dominance rules). The algorithms rely on a graph model, the *transfer graph*, aiming at capturing knowledge about admissible transfer plans. Finally, all of this will be tested and incorporated into a number of preprocessing procedures. These will aim at strengthening an *integer linear program* modeling the problem.

The remainder of the paper is organized as follows. In Section 2, we first formalize the notion of transfer plan and then introduce dominance rules for the dissemination problem. Sections 3 and 4 are devoted to algorithms that use dominance rules to detect irrelevant transfer plans. In Section 5 we propose a solving scheme that we then discuss and evaluate in Section 6.

2. Dominance rules

The solving techniques discussed in this paper are based on a number of dominance rules that dramatically improve the performance of enumeration algorithms (see the paper by Jouglet and Carlier (2011) for more details). These dominance rules are defined and discussed in this section. The results form the basis for additional constraints and deduction algorithms to be presented in the following sections. However, we will first formalize the notion of a transfer plan, which provides a means of describing solutions to the problem. Let us recall that a transfer plan defines a routing scheme, by indicating which units have to be transmitted during the different contacts.

2.1. Transfer plans

A transfer plan is an application $\phi : \{1, 2, \dots, m\} \mapsto \{\emptyset, \{1\}, \{2\}, \dots, \{u\}\}$, where $\phi(c)$ indicates the datum unit received by r_c during contact $\sigma_c \in \sigma$. Where $\phi(c) = \emptyset$, nothing is transferred during contact σ_c. Hereinafter, T_ϕ denotes the target set $\{\emptyset, \{1\}, \{2\}, \dots, \{u\}\}$ of ϕ. A transfer plan ϕ has a corresponding set of states $O_i^t \subseteq \mathcal{D}$, defined for each time index $t \in \{0, 1, \dots, m\}$ and each node $i \in \mathcal{N}$, such that:

(1) $\forall i \in \mathcal{N},\ O_i^0 = \mathcal{O}_i$,
(2) $\forall c \in \{1, 2, ..., m\},\ O_{r_c}^c = O_{r_c}^{c-1} \cup \phi(c)$,
(3) $\forall c \in \{1, 2, ..., m\},\ \forall i \in \mathcal{N} \setminus \{r_c\},\ O_i^c = O_i^{c-1}$

Thus, each state O_i^t contains the datum units obtained by node i during the first t contacts of sequence σ (in addition to the datum units held from the outset). The transfer plan is *valid* if nodes always send the units that they hold, *i.e.* $\forall \sigma_c \in \sigma$, we have $\phi(c) \in \{\emptyset\} \cup \{\{k\} \mid k \in O_{s_c}^{c-1}\}$.

A valid transfer plan ϕ has a *delivery length* $\lambda_i(\phi)$ for each node $i \in \mathcal{N}$, which corresponds to the smallest contact index t after which node i possesses every datum unit $k \in \mathcal{D}$, *i.e.* $\lambda_i(\phi) = \min\{t \in \{0, \dots, m\} \mid O_i^t = \mathcal{D}\}$. If this index does not exist, then it