Computers & Operations Research 78 (2017) 278–289

Contents lists available at ScienceDirect

Computers & Operations Research

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

A constraint-programming-based approach for solving the data dissemination problem

Ronan Bocquillon, Antoine Jouglet*

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

ARTICLE INFO

Article history:
Received 21 July 2015
Received in revised form
4 September 2016
Accepted 7 September 2016
Available online 9 September 2016

Keywords:
Combinatorial optimization
Constraint programming
Lower bounds
Symmetry-breaking techniques
Systems of systems
Data transfer problem

ABSTRACT

Systems of mobile Systems are intermittently connected networks that use store-carry-forward routing for data transfers. Independent systems collaborate and exchange data to achieve a common goal. Data transfers are only possible between systems that are close enough to each other, when a so-called *contact* occurs. During a contact, a sending system can transmit to a receiving system a fixed amount of data held in its interna then assume it holds at a til buffer. We assume that the trajectories of component systems are predictable, and consequently that a sequence of contacts may be considered. This dissemination problem is aimed at finding a transfer plan such that a set of data can be transferred from a given subset of source systems to all the recipient systems. In this paper, we propose an original constraint-programming-based algorithm for solving this problem. Computational results show that this approach is an improvement on the integer-linear-programming-based approach that we proposed in a previous paper.

1. Introduction

Mo Jamshidi has defined *Systems of Systems* (SoS) as large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together to achieve a common goal [1]. To reach this goal, the systems must collaborate and share data. When mobile or deployed in extreme environment, systems may suffer from a lack of continuous end-to-end connectivity. They must opportunistically make use of contacts that arise when two systems are close enough to each other. This may become critical when contact durations are relatively short with respect to the volume of information being routed through the SoS. Via a series of separate contacts, elements of data are transferred from one system to another until their final destination is reached. This routing scheme is based on a *store-carry-forward* paradigm.

Data transfer problems within SoS are addressed in the literature, in both opportunistic [2] and delay/disruption-tolerant networking (DTN) [3]. In general, the mobility of systems is assumed to be random [4,5], although realistic predictions can be made in many applications. The latter include satellite networks [6] (since the trajectory of the component systems obey straightforward dynamic laws), and networks over public transportation systems [7]. In this paper, we will look at how knowledge about node mobility and collaboration opportunities may be harnessed when routing data from a given set of systems to another inside a given time horizon. The fundamental question is: "Which elements of information should be transferred during each contact so that the dissemination length is minimized? ". We assume the physical network is perfect, and failures are ignored. A solution to this problem is termed a *transfer plan*.

Below we propose a centralized approach to compute an optimal transfer plan. This is a first step towards a complete solution. In a near future, we plan to investigate techniques for solving the problem in an uncertain (more realistic) context. This implies studying robust optimization in order to find transfer plans which remain valid when not all contacts are successful. We hope to cover a wide range of applications in this way. Such an approach is unfortunately not suitable for all situations (e.g. when new nodes may arrive), and online reopitmization techniques should also be developed to help the systems to react quickly to unpredictable and unexpected changes (not to systematically compute a new solution from scratch). Note that distributed approaches should be developed too.

For real-world examples, we refer to projects *Bluespots* [8] (where a small computer on a bus serves as a bluetooth content distribution station in a university public transit scenario), *DakNet* [9] (some buses carry a mobile access point to provide a disconnected Internet access to isolated villages), and *Disaster Monitoring Constellation* [6] (a multi-satellite Earth-imaging low-Earth–

* Corresponding author.
E-mail addresses: ronan.bocquillon@hds.utc.fr (R. Bocquillon), antoine.jouglet@hds.utc.fr (A. Jouglet).

http://dx.doi.org/10.1016/j.cor.2016.09.004

orbit sensor network where captured image swaths are stored onboard each satellite and later downloaded from the satellite payloads to a ground station).

2. Related works

The problem of routing in deterministic delay-tolerant networks has attracted an increasing number of researchers over the last decade.

Jain et al. [7] proposed four oracles to compare performances of routing protocols in terms of the amount of knowledge of network topology that they require (e.g. 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 then extended by Zhao et al. [10] to take multicasting protocols into account (e.g. the *membership oracle* can answer questions about group dynamics). Below we consider that all the oracles are available.

Handorean et al. [11] assumed that contacts are *atomic*, which means that contact durations – as opposed to inter-contact durations – are assumed to be instantaneous (propagation and transmission delays are both disregarded). Hay and Giaccone [12] proposed a graph-based model, termed the *event-driven graph*, which enables numerous problems where contacts are atomic to be solved straightforwardly (by using basic tools from graph theory). For example, the authors solve shortest-path or max-flow subproblems to minimize the delay or to maximize the network throughput. Below we also assume that contacts are atomic.

Many solutions were proposed to model delay-tolerant networks. Merugu et al. [13] proposed the so-called *space–time graph*, i.e. a graph composed of several snapshots (instantaneous connectivity graphs) placed side by side, and interconnected by "temporal" edges. Ferreira [14] proposed the *evolving graph*, an effective model capturing the most significant characteristics of time-varying networks. This model will be described in more detail in Section 3.1.

In a previous paper [15], we introduced several dominance rules, different deduction procedures, a promising preprocessing algorithm, and an integer-linear-programming model for solving the so-called *dissemination problem*. Here we extend this previous work and propose a constraint-programming-based algorithm to solve the problem more efficiently.

For those who want to go further, we refer to the "delay-tolerant networking research group" [16], Voyiatzis' survey [17], and Zhang's survey [18] for their extensive review of the literature. The large number of references in those papers reflects a high level of interest in the problem of routing in intermittently connected networks.

The remainder of the paper is organized as follows. The statement of the problem is recalled in Section 3. In Sections 4 and 5 we propose a new constraint-programming-based algorithm for solving it. In Section 6 we assess this approach and compare it with the one we proposed in [15]. The paper ends with a conclusion and prospects.

3. The dissemination problem

In this section we first recall the statement of the dissemination problem, then summarize the dominance rules we proposed in [15]. Dominance rules yield conditions that allow a subset of the search space to be ignored when solving the problem.

3.1. Formal description

First of all, we consider a set $\mathcal{N} = \{1, 2, \ldots, q\}$ of q interacting mobile systems, the *nodes*, and one *datum* $\mathcal{D} = \{1, 2, \ldots, u\}$ of u datum units. Each *datum unit* represents a unitary, indivisible fragment of data. Each node $i \in \mathcal{N}$ possesses a subset $O_i \subseteq \mathcal{D}$ of datum units from the outset. Subset $\mathcal{R} \subseteq \mathcal{N}$ defines the nodes wishing to obtain the datum $\mathcal{D}$ (all the datum units) inside the given time horizon. In this paper the term *source nodes* will refer to the nodes $i \in \mathcal{N}$ such that $O_i \neq \varnothing$. The nodes in $\mathcal{R}$ are termed the *recipients*. A node can be both a source and a recipient.

Remark 3.1. To simplify the formulation of this problem, we assume that the recipient nodes need to obtain all the datum units. Note, however, that the results described in this manuscript can all be generalized (with minor updates) to the case where the recipients only need a subset of the datum units.

To ensure the dissemination of the datum $\mathcal{D}$, nodes may exchange datum units whenever they are close enough to communicate. A communication opportunity is known as a *contact*. We assume that these contacts are perfectly known, or easily predictable at any time, because the trajectory of each node is deterministic. Thus, we consider a *sequence of contacts* $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ of m ordered pairs of $\mathcal{N}^2$. During contact $(s, r) \in \sigma$, the sending node s can transmit to the receiving node r at most one datum unit that it already possesses (either from the outset or as a result of previous contacts). If node s transmits a datum unit $k \in \mathcal{D}$ to node r, then node r possesses unit k afterwards. Below, nodes s_c and r_c denote the sender and the receiver in contact $\sigma_c = (s_c, r_c) \in \sigma$.

Remark 3.2. To represent an undirected contact $[i, j]$, we can consider a first directed contact (i, j), followed by a reverse contact (j, i). Moreover, to represent a longer contact (i, j), during which several datum units might be sent, we can duplicate the contact (one contact per possible transfer).

A *transfer plan* ϕ: $\{1, 2, \ldots, m\} \mapsto \{\varnothing, \{1\}, \{2\}, \ldots, \{u\}\}$ is a function where $\phi(c)$ designates the datum unit received by node r_c during contact σ_c. If $\phi(c) = \varnothing$, then nothing is transferred during contact σ_c. In the following, T_ϕ denotes the target set $\{\varnothing, \{1\}, \{2\}, \ldots, \{u\}\}$ of ϕ. A transfer plan ϕ has a corresponding set of states $O_i^t \subseteq \mathcal{D}$, defined for each time index $t \in \{0, 1, \ldots, m\}$ and each node $i \in \mathcal{N}$. The states are defined in the following recursive way:

$$\begin{aligned}
&(1)\ \forall\, i \in \mathcal{N}, \quad O_i^0 = O_i, \\
&(2)\ \forall\, c \in \{1, \ldots, m\}, \quad O_{r_c}^c = O_{r_c}^{c-1} \cup \phi(c), \\
&(3)\ \forall\, c \in \{1, \ldots, m\}, \quad \forall\, i \in \mathcal{N} \backslash \{r_c\}, \quad O_i^c = O_i^{c-1}
\end{aligned} \tag{1}$$

Each state O_i^t therefore contains the datum units received by node i during the first t contacts of sequence σ (in addition to the datum units that node i had from the start). The transfer plan is *valid* when nodes transmit only datum units that they possess, i.e.

$$\forall\, \sigma_c \in \sigma, \quad \phi(c) \in \{\varnothing\} \cup \{\{k\} \,|\, k \in O_{s_c}^{c-1}\} \tag{2}$$

A valid transfer plan ϕ has a *delivery length* $\lambda_i(\phi)$ for each node $i \in \mathcal{N}$, corresponding to the smallest contact index t after which node i possesses every datum unit $k \in \mathcal{D}$, i.e. $\lambda_i(\phi) = \min\{t \in \{0, 1, \ldots, m\} | O_i^t = \mathcal{D}\}$. If this index does not exist, then it is assumed in the following that $\lambda_i(\phi) = \infty$. The *dissemination length* $\lambda(\phi)$ of the transfer plan corresponds to the smallest index t at which all the recipient nodes are delivered, i.e. $\lambda(\phi) = \max_{i \in \mathcal{R}}\{\lambda_i(\phi)\}$. In this paper we address the problem of finding a valid transfer plan minimizing $\lambda(\phi)$.