Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Scheduling maintenance jobs in networks

Fidaa Abed [a], Lin Chen [b], Yann Disser [c], Martin Groß [d,*,1], Nicole Megow [e], Julie Meißner [f], Alexander T. Richter [g], Roman Rischke [h]

[a] University of Jeddah, Al-Sharafeyah, Jeddah 23218, Saudi Arabia
[b] University of Houston, 4800 Calhoun Rd, Houston, TX 77004, USA
[c] TU Darmstadt, Karolinenpl. 5, 64289 Darmstadt, Germany
[d] University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada
[e] University of Bremen, Bibliothekstraße 1, 28359 Bremen, Germany
[f] TU Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany
[g] TU Braunschweig, Universitätsplatz 2, 38106 Braunschweig, Germany
[h] TU München, Arcisstraße 21, 80333 München, Germany

ARTICLE INFO

Article history:
Received 1 September 2017
Received in revised form 24 January 2018
Accepted 21 February 2018
Available online xxxx

Keywords:
Scheduling
Maintenance
Connectivity
Complexity theory
Approximation algorithm

ABSTRACT

We investigate the problem of scheduling the maintenance of edges in a network, motivated by the goal of minimizing outages in transportation or telecommunication networks. We focus on maintaining connectivity between two nodes over time; for the special case of path networks, this is related to the problem of minimizing the busy time of machines.

We show that the problem can be solved in polynomial time in arbitrary networks if preemption is allowed. If preemption is restricted to integral time points, the problem is NP-hard and in the non-preemptive case we give strong non-approximability results. Furthermore, we give tight bounds on the power of preemption, that is, the maximum ratio of the values of non-preemptive and preemptive optimal solutions.

Interestingly, the preemptive and the non-preemptive problem can be solved efficiently on paths, whereas we show that mixing both leads to a weakly NP-hard problem that allows for a simple 2-approximation.

1. Introduction

Transportation and telecommunication networks are important backbones of modern infrastructure and have been a major focus of research in combinatorial optimization and other areas. Research on such networks usually concentrates on optimizing their usage, for example by maximizing throughput or minimizing costs. In the majority of the studied optimization models it is assumed that the network is permanently available, and our choices only consist in deciding which parts of the network to use at each point in time.

* Corresponding author.
E-mail addresses: fabed@uj.edu.sa (F. Abed), chenlin198662@gmail.com (L. Chen), disser@mathematik.tu-darmstadt.de (Y. Disser), martin.gross@mailbox.org (M. Groß), nicole.megow@uni-bremen.de (N. Megow), jmeiss@math.tu-berlin.de (J. Meißner), a.richter@tu-bs.de (A.T. Richter), rischke@ma.tum.de (R. Rischke).
[1] Present address: RWTH Aachen, Templergraben 55, 52062 Aachen, Germany.

https://doi.org/10.1016/j.tcs.2018.02.020

Practical transportation and telecommunication networks, however, can generally not be used non-stop. Be it due to wear-and-tear, repairs, or modernizations of the network, there are times when parts of the network are unavailable. We study how to schedule and coordinate such maintenance in different parts of the network to ensure connectivity.

While network problems and scheduling problems individually are fairly well understood, the combination of both areas that results from scheduling network maintenance has only recently received some attention [1–5] and is theoretically hardly understood.

Problem definition. In this paper, we study connectivity problems which are fundamental in this context. In these problems, we aim to schedule the maintenance of edges in a network in such a way as to preserve connectivity between two designated vertices. Given a network and maintenance jobs with processing times and feasible time windows, we need to decide on the temporal allocation of the maintenance jobs. While a maintenance on an edge is performed, the edge is not available. We distinguish between MINCONNECTIVITY, the problem in which we minimize the total time in which the network is disconnected, and MAXCONNECTIVITY, the problem in which we maximize the total time in which it is connected.

In both of these problems, we are given an undirected graph $G = (V, E)$ with two distinguished vertices $s^+, s^- \in V$. We assume w.l.o.g. that the graph is simple; we can replace a parallel edge $\{u, w\}$ by a new node v and two edges $\{u, v\}, \{v, w\}$. Every edge $e \in E$ needs to undergo $p_e \in \mathbb{Z}_{\geq 0}$ time units of maintenance within the time window $[r_e, d_e]$ with $r_e, d_e \in \mathbb{Z}_{\geq 0}$, where r_e is called the release date and d_e is called the deadline of the maintenance job for edge e. An edge $e = \{u, v\} \in E$ that is maintained at time t, is not available at t in the graph G. We consider preemptive and non-preemptive maintenance jobs. If a job must be scheduled non-preemptively then, once it is started, it must run until completion without any interruption. If a job is allowed to be preempted, then its processing can be interrupted at any time and may resume at any later time without incurring extra cost.

A *schedule* S for G assigns the maintenance job of every edge $e \in E$ to a single time interval (if non-preemptive) or a set of disjoint time intervals (if preemptive) $S(e) := \{[a_1, b_1], \dots, [a_k, b_k]\}$ with

$$r_e \leq a_i \leq b_i \leq d_e, \text{ for } i \in [k] \text{ and } \sum_{[a,b] \in S(e)} (b - a) = p_e.$$

If not specified differently, we define $T := \max_{e \in E} d_e$ as our *time horizon*. We do not limit the number of simultaneously maintained edges.

For a given maintenance schedule, we say that the network G is *disconnected at time t* if there is no path from s^+ to s^- in G at time t, otherwise we call the network G *connected at time t*. The goal is to find a maintenance schedule for the network G so that the total time where G is disconnected is minimized (MINCONNECTIVITY). We also study the maximization variant of the problem, in which we want to find a schedule that maximizes the total time where G is connected (MAXCONNECTIVITY).

Our results. For *preemptive* maintenance jobs, we show that we can solve both problems, MAXCONNECTIVITY and MINCONNECTIVITY, efficiently in arbitrary networks (Theorem 1). This result crucially requires that we are free to preempt jobs at arbitrary points in time. Under the restriction that we can *preempt* jobs only at *integral points in time*, the problem becomes NP-hard. More specifically, MAXCONNECTIVITY does not admit a $(2 - \epsilon)$-approximation algorithm for any $\epsilon > 0$ in this case, and MINCONNECTIVITY is inapproximable (Theorem 4), unless $\mathsf{P} = \mathsf{NP}$. By inapproximable, we mean that it is NP-complete to decide whether the optimal objective value is zero or positive, leading to unbounded approximation factors.

This is true even for unit-size jobs. This complexity result is interesting and may be surprising, as it is in contrast to results for standard scheduling problems, without an underlying network. Here, the restriction to integral preemption typically does not increase the problem complexity when all other input parameters are integral. However, the same question remains open in a related problem concerning the busy-time in scheduling, studied in [6,7].

For *non-preemptive* instances, we establish that there is no $(c\sqrt[3]{|E|})$-approximation algorithm for MAXCONNECTIVITY for some constant $c > 0$ and that MINCONNECTIVITY is inapproximable even on disjoint paths between two nodes s and t, unless $\mathsf{P} = \mathsf{NP}$ (Theorems 5, 6). On the positive side, we provide an $(\ell + 1)$-approximation algorithm for MAXCONNECTIVITY in general graphs (Theorem 8), where ℓ is the number of distinct latest start times (deadline minus processing time) for jobs.

We use the notion *power of preemption* to capture the benefit of allowing arbitrary job preemption. The power of preemption is a commonly used measure for the impact of preemption in scheduling [8–11]. Other terms used in this context include *price of non-preemption* [12], *benefit of preemption* [13] and *gain of preemption* [14]. It is defined as the maximum ratio of the objective values of an optimal non-preemptive and an optimal preemptive solution. We show that the power of preemption is $\Theta(\log |E|)$ for MINCONNECTIVITY on a path (Theorem 9) and unbounded for MAXCONNECTIVITY on a path (Theorem 12). This is in contrast to other scheduling problems, where the power of preemption is constant, e.g. [9,10].

On paths, we show that *mixed* instances, which have both preemptive and non-preemptive jobs, are weakly NP-hard (Theorem 13). This hardness result is of particular interest, as both purely non-preemptive and purely preemptive instances can be solved efficiently on a path (see Theorem 1 and [15]). Furthermore, we give a simple 2-approximation algorithm for mixed instances of MINCONNECTIVITY (Theorem 14).

Please cite this article in press as: F. Abed et al., Scheduling maintenance jobs in networks, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.02.020