SCHEDULING UNIT PROCESSING TIME ARC SHUTDOWN JOBS TO MAXIMIZE NETWORK FLOW OVER TIME: COMPLEXITY RESULTS

N.Boland, R.Kapoor, S.Kaur
University of Newcastle, Australia

T. Kalinowski
University of Rostock, Germany

Motivation: Maintenance and system throughput

Many real world complex problems can be viewed as networks with arc capacities, for example, network for trains, supply chain etc. in which system throughput needs to be maximized.

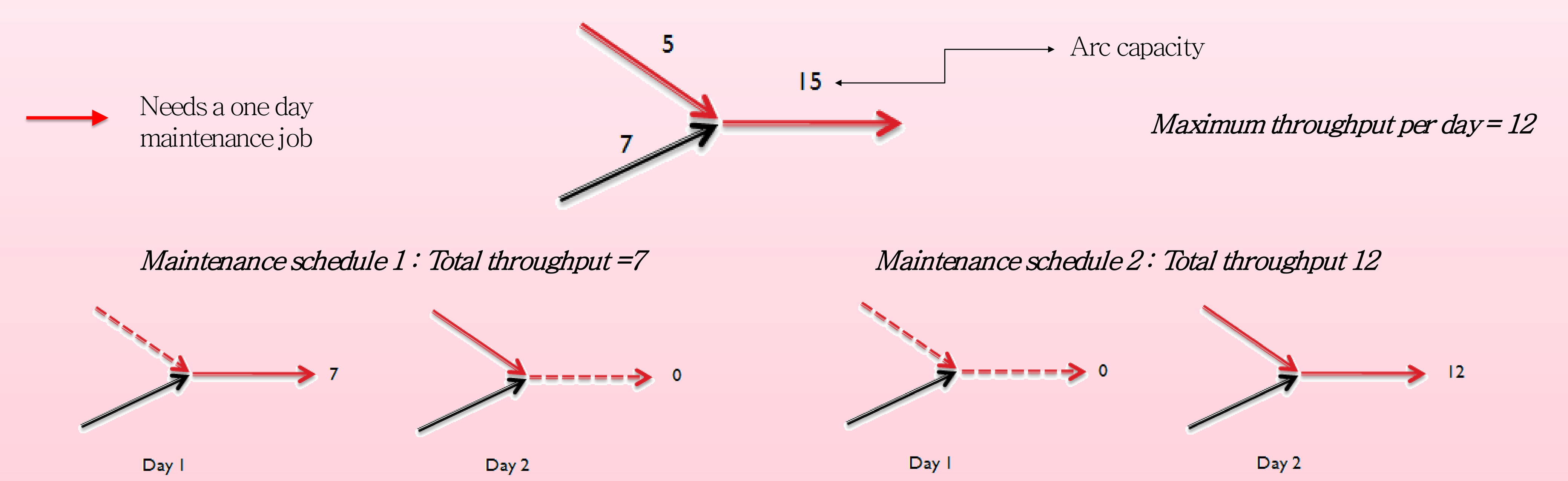
Arcs of a network represent important components of the corresponding system that may degrade over time.

Maintenance of these components (arcs of the network) is important to keep them in good working condition and to maintain their productivity.

But every maintenance activity incurs some loss of productivity as the arc will be unavailable during its maintenance.

How can maintenance be scheduled so as to maximize the throughput of the network?

Example: How the maintenance schedule can affect throughput?



Mathematical model of a special case

Given: a network with a multiset of capacitated arcs, a subset of arcs that have to go on maintenance exactly once for one time unit and a specified time horizon T.

The objective is to find a maintenance schedule so that maximum throughput is obtained from the network over the time horizon T, i,.e. the flow in the network summed over all time periods is maximized.

MIP Formulation

$$\max \ z = \sum_{i=1}^{T} \sum_{a\in\delta^+(s)} x_{ai}$$

$$\begin{aligned}
\text{s.t.} \quad & x_{ai} \le u_a && a \in A \setminus J,\ i \in [T],\\
& x_{ai} \le u_a y_{ai} && a \in J, i \in [T],\\
& \sum_{i=1}^{T} y_{ai} = T-1 && a \in J,\\
& \sum_{a\in\delta^-(v)} x_{ai} = \sum_{a\in\delta^+(v)} x_{ai} && v \in N \setminus \{s,t\},\ i \in [T],\\
& x_{ai} \ge 0 && a \in A,\ i \in [T],\\
& y_{ai} \in \{0,1\} && a \in J,\ i \in [T]
\end{aligned}$$

x_{ai} = the flow on arc a in time period i, $y_{ai} = 0$ in the period i in which the outage for arc a is scheduled, and u_a = capacity of arc a.

The problem in general is ***strongly NP-hard*** (reduction from 3-Partition problem). In the proof, the reduction gave rise to a network with a single transhipment node, which was not balanced (capacity in = capacity out at all transhipment nodes), and in which not all arcs needed to be shut down. This left open the complexity of the cases that all arcs have an associated outage, or the network is balanced.

Results

All arcs have an associated outage

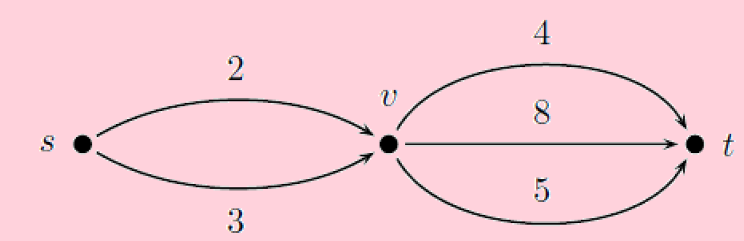
- If all arcs in a network must have a maintenance then it is optimal to schedule all jobs in the same time period, i.e. schedule all jobs together.

The network is balanced

capacity in = capacity out at all transhipment nodes

- If the network is series-parallel and balanced then it is optimal to schedule all jobs at the same time period.
- In general the problem is strongly NP-hard for balanced networks. (reduction from 3-Partition problem)

The network is not balanced but consists of a single transshipment node



- Suppose M1 and M2 denote the total capacity of arcs coming into and going out of the transshipment node v respectively and C1 , C2 are the total capacity of arcs that do not have a maintenance job associated coming in and going out of the v respectively. Without loss of generality we can assume that M1 < M2. Then if C1 ≤C2, it is optimal to schedule all jobs together.
- For an unbalanced network with one transhipment node on a time horizon of two periods it is NP-hard (reduction from 2-Partition problem) to decide if it is optimal to schedule all jobs at time 1.

Future Direction

The above results show either the problem is strongly NP-hard, or it is optimal to schedule all jobs at the same time. Other network features that may distinguish cases that are not strongly NP-hard, but in which it is also not optimal to have all jobs at the same time, are of interest.