The lockmaster's problem : a computational study

W. Passchyn, S. Coene, F.C.R. Spieksma
KULeuven, Research group Operations Research and Business Statistics (ORSTAT)
{ward.passchyn, sofie.coene, frits.spieksma}@econ.kuleuven.be

G. Vanden Berghe
KAHO Sint-Lieven, CoDeS,Computer Science
Greet.VandenBerghe@kahosl.be

D. Briskorn
Siegen University, Department of Operations Research
dirk.briskorn@uni-siegen.de

J. L. Hurink
University of Twente, Applied mathematics
j.l.hurink@utwente.nl

Inland waterways form a natural network that is an existing, congestion free infrastructure with capacity for more traffic. Transportation of goods by ship, over sea as well as over waterways, is a promising alternative for transport over land. Reasons are its reliability, its efficiency (a ship of 1200 tons can transport as much as 40 train wagons or 60 trucks), its relatively low operating cost and its environmental friendliness. Hence, the relative importance of this mode of transport is rising. Here, we focus on transport by ships over inland waterways and transport by sea ships entering a harbor/waterway network by passing through one or multiple locks. Locks are needed to control the water level so that large and heavy ships can continue to access the corresponding waterways. At several waterway networks congestion is expected to increase, yielding extra pressure on the locks. Locks managing the water level on waterways and within harbors sometimes constitute bottlenecks for transportation over water. The lockmaster's problem concerns the optimal strategy for operating such a lock.

We now continue with the description of a very basic situation that will act as our core problem : the lockmaster's problem. Consider a lock consisting of a single chamber. Ships that are traveling downstream arrive at the lock at given times. Other ships traveling upstream arrive at the lock, also at given times. Let A represent the set of all ships. Let $t(a)$ be the given arrival time of ship $a \in A$ and $p(a)$ the arrival position of ship $a \in A$ with $p(a) = 0$ for a downstream arrival and $p(a) = 1$ for an upstream arrival. Let T denote the *lockage duration* : this is the time between closing the lock for entering ships, and opening the lock so that ships can leave. We assume that all data are integral and the lockage duration is independent of the lock position and the number of ships in the lock. Our goal is to find a feasible lock-strategy that minimizes total waiting time of all ships. The waiting time of a ship is the time that passes between the ship's arrival time at the lock and the moment in time when the ship enters the lock. Thus, we need to determine at which moments in time the lock should start to go up (meaning those moments when ships that are downstream enter the lock and are lifted), and at which moments in time the lock should start to go down (meaning those moments when ships that are upstream enter the lock and are being lowered). Clearly for such a strategy to be feasible : (i) going-up moments and going-down moments (referred to as moments) should alternate, and (ii) consecutive moments should be at least T time-units apart.

A dynamic programming algorithm (DP) is proposed that solves the lock-master's problem in polynomial time. The exact solution is obtained by finding a shortest path in a graph. We first note that in an optimal schedule, all lock moves start at the arrival of a ship or at the arrival of the lock. We define a block as a sequence of consecutive up and down movements followed by a waiting period of less than $2T$ time units. It is shown that an optimal schedule exists that is a sequence of blocks. We build the graph by adding an edge for each of these possible blocks with a weight equal to the waiting time of the ships in this block. This gives an acyclic graph with $O(n^2)$ edges where $n = |A|$ is the total number of ships. The shortest path from s to t in this graph corresponds to a solution with the lowest possible waiting time. A straightforward implementation of this graph yields an algorithm with $O(n^3)$ time complexity. By using some specific properties of the graph, a speed-up of the algorithm can be achieved to $O(n^2)$.

This algorithm can also be used to solve a single batching machine scheduling problem more efficiently than the current algorithms from literature do. Relevant extensions that take into account the lock's capacity, the different priorities of ships, the water usage of a lock, and the possibility of multiple chambers, can be dealt with by slightly modifying the basic algorithm. Further, we prove that the problem with multiple non-identical chambers is strongly NP-hard when the number of chambers is part of the input.

A computational study is performed to evaluate the performance of some basic heuristics for the lockmaster's problem and the optimal solution algorithm developed here. We use a set of problem instances from literature for similar lock scheduling problems with size restrictions on the lock and ships. Arrivals in these instances follow a Poisson distribution. The heuristics used for comparison are a number of straightforward strategies to operate the lock, assuming near complete lack of information. A lock operator knows only the number of ships waiting on either side of the lock, no other specific information about the problem instance is available.