Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2014 Prof. R. Wattenhofer / K.-T. Foerster, T. Langner, J. Seidel

Discrete Event Systems

Solution to Exercise Sheet 12

1 PhD-Scheduling

a) (i) SmallLoad distributes the tasks as follows:

PhD student 1:	2	4	7
PhD student 2:	5	3	

Opt uses the following distribution (or another one with the same cost):

PhD student 1:	2	5	3
PhD student 2:	4	7	

SmallLoad thus distributes the tasks with cost $\text{Alg}(\sigma) = 13$ while Opt incurs a cost of $\text{Opt}(\sigma) = 11$. Hence,

$$\rho(\sigma) = \frac{\text{Alg}(\sigma)}{\text{Opt}(\sigma)} = \frac{13}{11} \ .$$

(ii) The following sequence results in a larger competitive ratio: $\sigma = 1, 1, 2$. We have $\text{Alg}(\sigma) = 3$ and $\text{Opt}(\sigma) = 2$ and thus

$$\rho(\sigma) = \frac{\text{Alg}(\sigma)}{\text{Opt}(\sigma)} = \frac{3}{2} \ .$$

(iii) See **b)**.

(iv) No, finding the optimal solution offline corresponds to solving the Partition-problem, which is NP-complete, thus presumably no efficient algorithm exists for the problem.

b) We first show a lower bound of $(2 - \frac{1}{m})$ on the competitive ratio of SmallLoad. To this end, we choose an input sequence that consists of $m(m-1)$ tasks of size 1 concluded with a task of size m, i.e. $\sigma = \underbrace{1, \ldots, 1}_{m(m-1)}, m$. After assigning the first $m(m-1)$ tasks, SmallLoad has assigned $m - 1$ units to each of the m PhD students. The last task of size m incurs a load of $2m - 1$ for the student to whom it is assigned.

The optimal algorithm assigns the first $m(m-1)$ taks to only $m-1$ students and the last (heavy) task to the remaining student. This results in a maximal load of m and we get the following lower bound for the competitive ratio:

$$c \geq \frac{\text{Alg}(\sigma)}{\text{Opt}(\sigma)} = \frac{2m-1}{m} = 2 - \frac{1}{m}$$

Now we shall show a matching upper bound for the competitive ratio. Let $\sigma = (e_1, e_2, \ldots)$ be an arbitrary input sequence. Without loss of generality, we assume s_1 to be the student with the maximal load for σ. Furthermore, let w be the effort of the last task T assigned s_1 and E the load of s_1 before assigning its last task. The load of all other students must be at least E since s_1 was the student with minimal load when he was assigned task T (otherwise another student would have received T). Hence, the sum of the loads of all students is at least $m \cdot E + w$ and hence

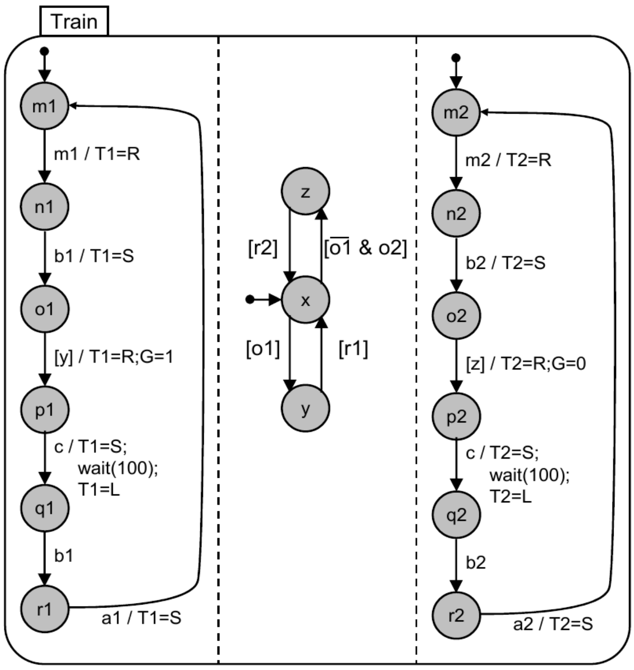
$$\textsc{Opt}(\sigma) \geq \frac{m \cdot E + w}{m} = E + \frac{w}{m} \ .$$

Using $\textsc{Opt}(\sigma) \geq w$, we get

$$\begin{aligned}
\textsc{Alg}(\sigma) &= w + E \\
&\leq w + \textsc{Opt}(\sigma) - \frac{w}{m} \\
&= \textsc{Opt}(\sigma) + \left(1 - \frac{1}{m}\right) w \\
&\leq \textsc{Opt}(\sigma) + \left(1 - \frac{1}{m}\right) \textsc{Opt}(\sigma) \\
&= \left(2 - \frac{1}{m}\right) \textsc{Opt}(\sigma)
\end{aligned}$$

2 The Winter Train Problem

We can model each train individually and combine the corresponding sub-states using an AND-super-state, see the figure below. Additionally, in order to "synchronize" the trains, a third sub-state is needed (shown in the middle) which implements a mutual exclusion: For instance, if there is no train between Stans and Engelberg and if train 1 is in state c1, T1 can enter the critical section and train 2 has to wait. (Notice that if both trains are in states c1 and c2 respectively, T1 has priority.)



- The trains start at their states m1 and m2. When m1 (m2) is pressed, then train 1 (2) moves to the right in n1 (n2), until it reaches the switch, where it stops in state o1 (o2).

- Now the "middle"-state can change its state to either y or z, depending on which train got there first. If train 1 (2) arrives first, then the state is changed to y (z) and train 1 (2) can move to state p1 (p2) while moving right.
- After arriving at the station Engelberg, the train waits for 100s, then moves to the left and switches to state q1 (q2) – until it hits the switch at b1 (b0), upon which the "middle"-state can change again – and the train continues to its original station, where it stops.

Positions of the trains (train 1 ; train 2):

- m1: Lucerne ; m2: Sarnen
- n1: Between Lucerne and the switch ; n2: Between Sarnen and the switch
- o1: At the left side of the switch ; o2: At the left side of the switch
- p1: Between the switch and Engelberg ; p2: Between the switch and Engelberg
- q1: Between Engelberg and the switch ; q2: Between Engelberg and the switch
- r1: Between the switch and Lucerne ; r2: Between the switch and Sarnen