1 Power-Down Mechanisms

As mentioned in the hint, we only focus on a single idle period because if we know that our algorithm is $c$-competitive for any idle period, we also know that it is $c$-competitive for the complete busy sequence.

a) Analogously to the 2-competitive ski-rental online algorithm, we consider an algorithm $\text{Alg}$ that powers down after $D$ time units. To see that $\text{Alg}$ is 2-competitive, we distinguish two cases for the length of the current idle period $T$:

- $T < D$: The energy consumed by both algorithms is $c_{\text{Alg}} = c_{\text{Opt}} = T$, hence the competitive ratio is $c = T/T = 1$.
- $T \geq D$: We have $c_{\text{Alg}} = D + D$ since $\text{Alg}$ waits $D$ time units and then powers-down and $c_{\text{Opt}} = D$ because $\text{Opt}$ powers down immediately. Hence we get $c = 2$.

b) Let $\text{Alg}$ be any deterministic power down algorithm. Then the time $t_{\text{Alg}}$ after which it powers down in an idle period is known in advance. The “worst” idle period ends immediately after $\text{Alg}$ has powered down, that is we have $T = t_{\text{Alg}} + \varepsilon$. Again, we distinguish two cases with respect to the time $t_{\text{Alg}}$ when $\text{Alg}$ powers down.

- $t_{\text{Alg}} < D$: We have $c_{\text{Alg}} = t_{\text{Alg}} + D$ and $c_{\text{Opt}} = t_{\text{Alg}} + \varepsilon$, hence
  \[ c = \frac{t_{\text{Alg}} + D}{t_{\text{Alg}} + \varepsilon} = 1 + \frac{D - \varepsilon}{t_{\text{Alg}} + \varepsilon} > 2 \quad \text{for } \varepsilon \to 0 \]
  since $t_{\text{Alg}} < D$.
- $t_{\text{Alg}} \geq D$: We have $c_{\text{Alg}} = t_{\text{Alg}} + D$ again and $c_{\text{Opt}} = D$, hence
  \[ c = \frac{t_{\text{Alg}} + D}{D} = 1 + \frac{t_{\text{Alg}}}{D} \geq 2 \quad \text{for } \varepsilon \to 0 \]
  since $t_{\text{Alg}} \geq D$.

Hence, $\text{Alg}$ cannot be better than 2-competitive.

c) Let $\text{Alg}$ be a randomised algorithm that powers down at time $\frac{2}{3}D$ with probability $\frac{1}{2}$ and at time $D$ otherwise. Let $C_{\text{Alg}}$ be a random variable for the cost incurred by the algorithm. We again consider an arbitrary idle period of length $T$. We distinguish three cases:

- $T < \frac{2}{3}D$: The energy consumption of both algorithms is $c_{\text{Alg}} = c_{\text{Opt}} = T$, hence $c = T/T = 1 < 2$. 

\[ \text{Discrete Event Systems} \]
\[ \text{Solution to Exercise Sheet 14} \]
• $\frac{2}{3} D \leq T < D$: The expected energy consumption of $\text{Alg}$ is

$$E[C_{\text{Alg}}] = \frac{1}{2} \left( \frac{2}{3} D + D \right) + \frac{1}{2} T = \frac{5}{6} D + \frac{1}{2} T$$

and further $c_{\text{Opt}} = T$. Hence we get

$$c = \frac{\frac{2}{3} D + \frac{1}{2} T}{T} = \frac{1}{2} + \frac{5}{6} \cdot \frac{D}{T} \leq \frac{1}{2} + \frac{5}{6} \cdot \frac{D}{\frac{2}{3} D} = \frac{1}{2} + \frac{5}{4} = \frac{7}{4} < 2.$$  

• $T \geq D$: We have for the expected energy consumption of $\text{Alg}$

$$E[C_{\text{Alg}}] = \frac{1}{2} \left( \frac{2}{3} D + D \right) + \frac{1}{2} (D + D) = \frac{5}{6} D + D = \frac{11}{6} D$$

and further $c_{\text{Opt}} = D$. Hence we get

$$c = \frac{\frac{11}{6} D}{D} = \frac{11}{6} < 2.$$  

Hence, the randomised algorithm is $\frac{11}{6}$-competitive which is better than any deterministic algorithm.

*Note:* This result, however, is not optimal yet. The best randomised algorithm uses a continuous probability distribution for the shutdown time and thereby achieves a competitive ratio of $e/(e-1) \approx 1.58$.

### PhD-Scheduling [Exam]

**a)** (i) $\text{SmallLoad}$ distributes the tasks as follows:

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

$\text{Opt}$ uses the following distribution (or another one with the same cost):

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

$\text{SmallLoad}$ thus distributes the tasks with cost $\text{Alg}(\sigma) = 13$ while $\text{Opt}(\sigma) = 11$. 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 $\text{SmallLoad}$. 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. $\sigma = 1, \ldots, 1, m$. After assigning the first $m(m-1)$ tasks, $\text{SmallLoad}$
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 \(m(m-1)\) tasks 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 \(\sigma\). 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

\[
\text{Opt}(\sigma) \geq \frac{m\cdot E + w}{m} = E + \frac{w}{m}.
\]

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

\[
\text{Alg}(\sigma) = w + E \\
\leq w + \text{Opt}(\sigma) - \frac{w}{m} \\
= \text{Opt}(\sigma) + \left(1 - \frac{1}{m}\right)w \\
\leq \text{Opt}(\sigma) + \left(1 - \frac{1}{m}\right)\text{Opt}(\sigma) \\
= \left(2 - \frac{1}{m}\right)\text{Opt}(\sigma)
\]