COMP9601 Assignment 4 Solution

Details are omitted, you should follow the proofs in the course to reconstruct them.

Problem 1

We analyze the MARK algorithm and OPT separately. (Please refer to Slide 13 on moodle)

- OPT (total cost) $\geq$ number of phases - 1.
- MARK (expected cost) $\leq H_k \sum n_i$, where $n_i = 1$ for all i.

Therefore, MARK is H_k competitive.

Problem 2

For any request sequence I, divide it into phases such that each phase contains $2k$ distinct items. Suppose there are m phases. Then,

- OPT: at least k miss in each phase, $OPT_k(I) \geq k(m-1)$.
- LRU: at most $2k$ miss in each phase, $LRU_{2k}(I) \leq 2km$.

Therefore, $LRU_{2k}(I) \leq 2k\frac{OPT_k(I)+k}{k} \leq 2OPT_k(I) + 2k$.

Problem 3

For any request sequence I, divide it into phases such that each phase contains k distinct items. Suppose there are m phases. Then,

- OPT: at least 1 miss in each phase, $OPT(I) \geq m - 1$.
- FIFO: at most k miss in each phase, $FIFO(I) \leq km$.

Therefore, $FIFO(I) \leq km \leq k(OPT(I) + 1) \leq kOPT(I) + k$

Problem 4

The following example shows that, from t=4 afterwards, the total work done by EDF is less than that of OPT:

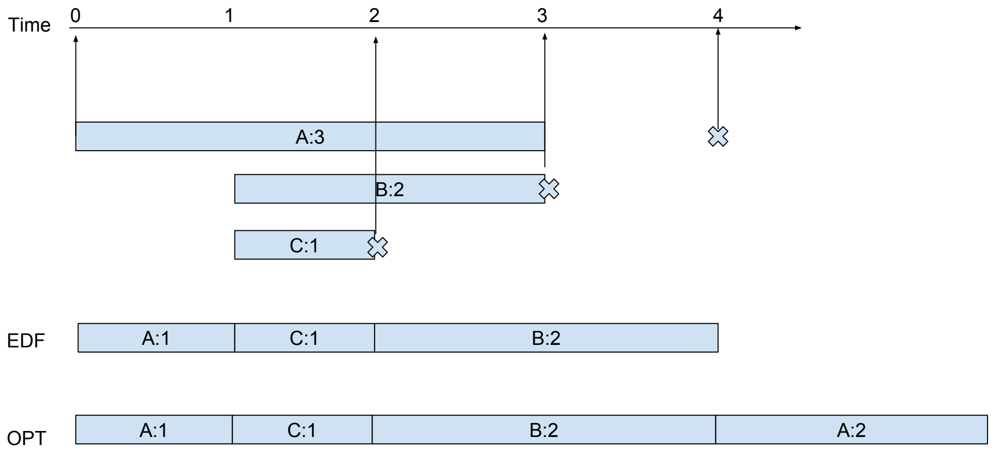
However, as we know, in the "hard deadline scheduling" scenario, EDF is optimal, i.e., EDF can schedule all jobs to meet the deadlines whenever some offline algorithm can do so.

So it won't happen that EDF requires to schedule a job after its deadline; in other words, this version of EDF is exactly the same as the original version.

Thus, this version of EDF is still optimal.

Problem 5

Let P be a power of 2. At time $r_i = 2P(1 - \frac{1}{2^i})$ for $i = 0, ..., \log P - 1$, three jobs are released. One job of processing time $\frac{P}{2^i}$, and two of $\frac{P}{2^{i+1}}$. And staring from time $2(P-1)$ and until time $P^2 + 2P - 3$, two jobs



of processing time 1 are released at each time unit.
Observe that $logP$ jobs are unfinished at time $2(P-1)$in the $SRPT$ schedule. Now, at each time $2(P-1)+k, k=0...P^2-1$, two jobs of processing time 1 are released. Therefore, $logP$ jobs are waiting to be processed for P^2 units of time, resulting in a total flow time of at least $P^2 \log P$.
On the other hand, an optimal solution finishes the three jobs presented at time r_i by time r_{i+1}. The job of processing time $\frac{P}{2^i}$ is scheduled on one machine at time ri, while the other machine is used for the two jobs of processing time $\frac{P}{2^{i+1}}$. This results in an increase of the flow time when we restrict the attention to the rest groups of jobs: the flow time of the jobs released at time r_i is $\frac{5}{2}\frac{P}{2i}$, and the total flow time of the $\log P$ groups of jobs is bounded by $5P$. But all jobs are finished in the optimal solution by time $2(P-1)$, and the flow time of the final P^2 groups of 2 jobs is just $2P^2$. We get an $o(\log P)$ competitive ratio. Then we can choose $P = 2^{100}$ to construct feasible instances.