Network and Integer Optimization - Problem 2.2

Sarp Sümer

May 1, 2025

Night shift planning

(a) We construct the directed graph $G = (V, A)$, where:

$$
\begin{aligned}
V &= \{s, t\} \cup P \cup (P \times [k]) \cup N \\
A &= (\{s\} \times P) \\
&\quad \cup \{(p, (p, i)) \mid p \in P, i \in [k]\} \\
&\quad \cup \{((p, i), n) \mid p \in P, i \in [k], n \in N_i\} \\
&\quad \cup (N \times \{t\})
\end{aligned}
$$

along with a capacity function $u : A \to \mathbb{Z}_{\geq 0}$ defined as:

$$
\begin{aligned}
\forall j \in [q] \cdot u(s, p_j) &= \alpha_j \\
\forall p \in P, \forall i \in [k] \cdot u(p, (p, i)) &= 2 \\
\forall j \in [q], \forall i \in [k], \forall n \in N_i \cdot u((p_j, i), n) &= \begin{cases} 1 & \text{if } n \in W_j \\ 0 & \text{otherwise} \end{cases} \\
\forall n \in N \cdot u(n, t) &= 2
\end{aligned}
$$

To solve the problem, we compute an integral maximum $s - t$ flow f in G with capacities u. A feasible assignment exists iff $\nu(f) = 2n$. To retrieve the actual assignments, we use f as follows.

A physician p_j is assigned to a night $n \in N_i$ iff $f((p_j, i), n) = 1$.

We now show correctness. The first kind of arc capacities ensure that the limits on the number of night shifts physicians can be assigned to are respected. The second kind of arc capacities ensure that no physician may be assigned to more than 2 night shifts per block. The third kind of capacities ensure that physicians may only be assigned to nights on which they are available. Finally, the fourth class of constraints ensure that to each night, at most 2 physicians are assigned. In a maximum flow, these arcs have to be saturated, so a maximum flow will correspond to a feasible assignment.

The graph clearly has size polynomial in the input parameters, and one can use an algorithm such as Edmonds-Karp to compute the maximum flow in polynomial time.

(b) Let $k \in \mathbb{Z}_{\geq 0}$. Replace the numbers α_j by $\alpha'_j = \min(k, \alpha_j)$, and run the above algorithm. We use binary search on the range $[n]$ to determine the smallest possible k for which a feasible assignment exists with these modified parameters. Clearly, this increases the running time by a logarithmic factor, so the procedure is still efficient.

(c) See accompanying notebook.