CSC373 Winter 2015 Assignment # 1

Question 3 (Weidong An)
Algorithm: Suppose there are n CPOs(denoted by $c_1, ..., c_n$) and the examination periods last for k days(denoted by $d_1, ..., d_k$).

- Let P be the set of all examination periods and $P = \{d_1^m, ..., d_k^m\} \cup \{d_1^a, ..., d_k^a\} \cup \{d_1^e, ..., d_k^e\}$. d_j^m indicates "the morning of day d_j". d_j^a indicates "the afternoon of day d_j". d_j^e indicates "the evening of day d_j".
- Let P_j be the set of all examination periods which are available for c_j for $j = 1, ..., n$.
- Based on P_j, create set D_j which contains the dates on which c_j is available for at least one examination period for $j = 1, ..., n$. Each element in D_j has super script j. For example, if c_3 is available for days d_1, d_3, d_5, then $D_2 = \{d_1^2, d_3^2, d_5^2\}$.

Then, implement the following:

1. Create a network flow N with vertices $V = \{s, t\} \cup \{c_1, ..., c_n\} \cup P \cup (\bigcup_{i=1}^n D_i)$ and with edges
   - $E = (\{(s, c_1), ..., (s, c_n)\})$ (with $c(s, c_i) =$ maximum number of examinations that c_i can invigilate)
   - $\cup(\bigcup_{i=1}^n (\bigcup_{d \in D_i} \{(c_i, d)\}))$ (with $c(c_i, d) = 2$)
   - $\cup(\bigcup_{i=1}^n (\{(d, p) | d \in D_i, p \in P_i$ and d, p have the same subscript(i.e. the same date)$\}))$ (with $c(d, p) = 1$)
   - $\cup(\bigcup_{p \in P} \{(p, t)\})$ (with $c(p, t) = \lceil$(number of examinations in period p) $\times (1 + 10\%)\rceil$)
2. Find a maximum integer flow f in network N using Edmonds-Karp algorithm.
3. If there is an edge (p, t) with $p \in P$ and $f(p, t) < c(p, t)$, return NIL. Otherwise, set $C_i = \{p | f(d, p) = 1, d \in D_i, p \in P_i\}$ and return $C_1, ..., C_n$.

Runtime Analysis:

- Notice that $|V| \leq nk + 3k + 2$. $|E| \leq n + nk + 3nk + 3k$.
- Since Edmonds-Karp algorithm runs in $O(|V||E|^2)$, it takes $O((nk + 3k + 2)(n + nk + 3nk + 3k)^2) = O(n^3k^3)$ to run Edmonds-Karp algorithm on N.
- It takes $O(|V| + |E|) = O(nk)$ to build network N.
- It takes $O(|E|) = O(nk)$ to build C_i for $i = 1, ..., n$.
- Totally, the algorithm runs in $O(n^3k^3)$ which is in polynomial time.

Justification of Correctness

Claim 1. Every collection of valid sets of examination periods for CPOs $C_1, ..., C_n$ give rise to a flow f in N.

Since $C_1, ..., C_n$ are valid, we have the following:

(1) $f(s, c_i) = |C_i|$ (the number of examination periods that c_i will invigilate)

(2) For $d \in D_i, f(c_i, d)$ = number of examination periods that c_i will invigilate on day d_i and it is no more than 2.

(3) For $d \in D_i, p \in P_i, f(d, p) = 1$ if and only if $(d, p) \in C_i$

(4) For $p \in P, f(p, t)$ = number of CPOs in examination period $p = c(p, t)$

By (4), $|f|$ is maximized. Therefore, every valid collection of sets $C_1, ..., C_n$ gives rise of a maximum flow in N.

Claim 2. Every integer flow in N gives rise to a collection of sets of examination periods for CPOs $C_1, ..., C_n$ (or NIL if it is not possible).

- $C_i = \{p | f(d, p) = 1, d \in D_i, p \in P_i\}$
- Every CPO is within maximum availability because $c(s, c_i)$ = maximum number of examinations that c_i can invigilate
- Every CPO is assigned to no more than 2 examination periods in one day because $c(c_i, d) = 2, d \in D_i$.
- Every CPO is only assigned to examination periods that is available because $(d, p) \notin E$ for $d \in D_i, p \notin P_i$.
- Every examination period has enough CPOs if and only if $f(p, t) = c(p, t)$ for all $p \in P$. Therefore, $C_1, ..., C_n$ exist if and only if there is a flow such that $f(p, t) = c(p, t)$ for all $p \in P$. If such flow f exists, $|f|$ must be maximized.

Therefore, every maximized flow in N gives rise to a collection of sets of examination periods for CPOs $C_1, ..., C_n$ if $f(p, t) = c(p, t)$ for all $p \in P$ otherwise NIL.

By Claim 1 and Claim 2, the algorithm is correct.