Tutorial 7 Answer sketches

Q1: Let X, Y denote the cut obtained by the algorithm and U, W be the optimal cut. Let *alg* denote the size of the cut as given by the algorithm, and *opt* denote the optimal size cut.

Let $V_1 = X \cap U$ and $V_2 = X \cap W$, $V_3 = Y \cap U$ and $V_4 = Y \cap W$.

For $i, j \in 1, 2, 3, 4$, $i \neq j$, let $e_{i,j}$ denote the number of edges from V_i to V_j.

Thus, $alg = e_{1,3} + e_{1,4} + e_{2,3} + e_{2,4}$.

$opt = e_{1,2} + e_{1,4} + e_{2,3} + e_{3,4}$.

As the algorithm is not able to find a vertex to switch from X to Y (or vice versa), we have that for any vertex $v \in V_1$, the number of edges from v to V_2 is at most the number of edges from v to $V_3 \cup V_4$. It follows that

$$e_{1,2} \leq e_{1,3} + e_{1,4}$$

Similarly

$$e_{3,4} \leq e_{1,3} + e_{2,3}$$

Thus, we have that

$opt = e_{1,2} + e_{1,4} + e_{2,3} + e_{3,4} \leq (e_{1,3} + e_{1,4}) + e_{1,4} + e_{2,3} + (e_{1,3} + e_{2,3}) \leq 2alg$.

Q2. Note that the load on the processor 1 is highest. Furthermore, if we drop a_1 from processor 1, then its load will become the least.

Thus, based on the schedule given by the algorithm, maximum load *maxalg* is at most

$$\ell(a_1) + [\sum_{i=2}^{n} \ell(a_i)]/m$$

Let *optmax* denote the maximum processor load given by optimal schedule. Then,

$$\ell(a_1) \leq optmax$$

and

$$[\sum_{i=2}^{n} \ell(a_i)]/m \leq \sum_{i=1}^{n} \ell(a_i)/m \leq optmax$$

Thus, we immediately have that

$maxalg \leq 2optmax$.

Q3. Let $CLeft_{end}$ denote the value of $CLeft$ at the end of the algorithm. Whenever a literal ℓ is set true in the loop, let n_ℓ denote the number of clauses which get satisfied due to ℓ being set true in the corresponding while loop. Thus, if $\overline{\ell}$ occurs in $CLeft_{end}$, then the number of occurences of $\overline{\ell}$ in $CLeft_{end}$ is at most n_ℓ. (Since this is the property ensured by the algorithm when making ℓ true). It follows that the number of clauses which are satisfied is at least the number of literals left in $CLEFT_{end}$. If each clause has at least k literals, then $k * |CLEFT_{end}| \leq$ number of clauses satisfied. Thus, the number of clauses satisfied is at least $k/(k+1)$ of the number of clauses. As the optimal algorithm can at best satisfy all the clauses, we have our result.