

問題 1

貪欲法を用いて、解を出しました。

解としては、なるべく資源量を少ないものを出力できるようになっています。割り当てるジョブを基準として、ジョブごとに、資源量が最小のものとして番目に最小のものの差を意味し、おき、その差が大きいジョブを割り当てるという操作をくり返します。

(直感的には、ジョブごとにこの割り当てる量を少なくすれば、大きく資源量が増加してしまうものを選ぶ)

以下に、アルゴリズムを示す。

① $J := \{1, 2, \dots, n\}$ (ジョブの集合) と、

$c_i = c_i$ ($i=1, \dots, m$) (残り利用可能資源量) を定義

$f(i, j) = a_{ij}$ とする。(今回は基準を資源量とした)

② $j \in J$ に対して, $F_j = \{i : a_{ij} \leq b'_j\}$ とする。
(F_j : 零)当可能なエッジの集合に注意)

$\rightarrow F_j \neq \emptyset$ ならば infeasible

$i_j : \arg \min_{i \in F_j} f(i, j)$ ($j \in J$) ($f(i, j)$ を最小とするエッジ)

$\rho_j : \min_{\substack{s \in F_j \\ s \neq i_j}} f(s, j) - f(i_j, j)$ (2番目と最小の差)

③ $\hat{j} = \arg \max_{j \in J} \rho_j$ (ρ_j が大きいものを選ぶ)

best sol を更新

$$b'_j = b'_j - a_{i_j j}$$

$$J = J - \{\hat{j}\}$$

④ $J = \emptyset$ ならば解を出力して終了。

そうでなければ ② に戻る。

(フロリダ大学の H. Edwin Romeijn の論文を参考にしました。)