ManjaBe

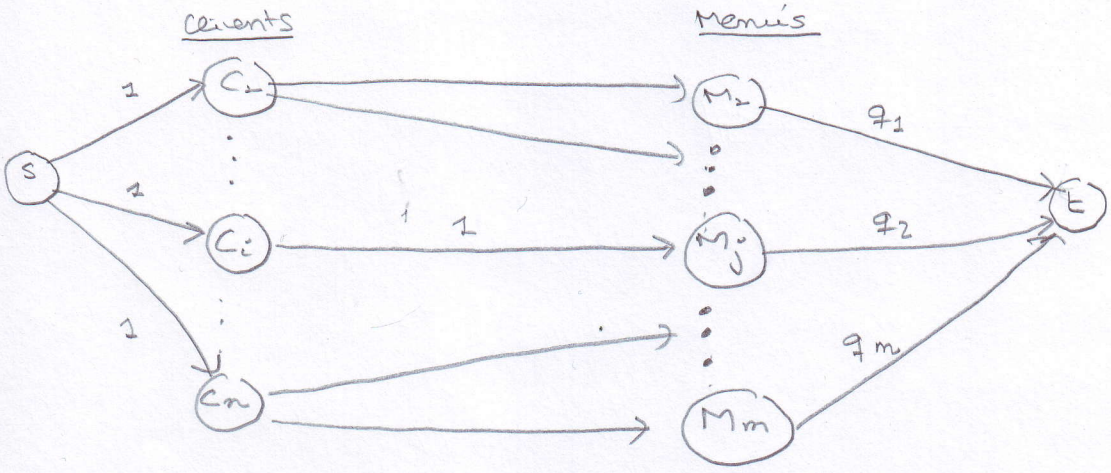
Resoldrem el problema via Max-Flow.

Definim la xarxa següent $G=(V,E)$ amb capacitats $c: E \to \mathbb{Z}^+ \cup \{0\}$

* Vèrtex $V = \{s, t, C_1, \dots, C_n, M_1, \dots, M_m\}$

on $\{C_1, \dots, C_n\}$ representen el conjunt dels n clients

i $\{M_1, \dots, M_m\}$ " " " dels m menus



$(C_i, M_j) \in E \Leftrightarrow M_j \in A_i$ (menú j és a la llista de preferències del client C_i

* Arestes i capacitat:

i) Per cada client C_i, $1 \leq i \leq n$,

$(s, C_i) \in E$ i $c(s, C_i) = 1$

ii) Per cada client C_i tal que $1 \leq i \leq n$ i per cada menú M_j tal que $M_j \in A_i$ (llista de preferències del client)

$(C_i, M_j) \in E$ i $c(C_i, M_j) = 1$

iii) Per cada menu M_j, $1 \leq j \leq m$

$(M_j, t) \in E$ i $c(M_j, t) = q_j$ (quantitat de menús de tipus M_j que ManjaBe pot servir

- Un cop construida la xarxa, apliquem un algorisme de Max-Flow de manera que si f correspon a un flux màxim calculat per l'algorisme podem definir l'assignació òptima

$si\_f : E \to \mathbb{N}$ i el flux màxim calculat per l'algorisme podem definir l'assignació òptima

$$assig : \underbrace{\{c_1, \dots, c_n\}}_{\text{Clients}} \longrightarrow \underbrace{\{M_1, \dots, M_m\}}_{\text{Mans}} \cup \{\text{"val"}\}$$

- $f(s, c_i) = 1 \Rightarrow \exists j$ $M_j \in A$ i $f(c_i, M_j) = 1$

  En aquest cas definim $assig(c_i) = M_j$

  el cost per a MenjaBe, $cost(c_i) = 0$

- $f(s, c_i) = 0 \Rightarrow \forall j$ $M_j \in A$ i $f(c_i, M_j) = 0$

  En aquest cas $assig(c_i) =$ "val"

  el cost per a MenjaBe, $cost(c_i) = 5$

- $f(M_j, t) \leq q_j \Rightarrow$ Mai se serviran més mans dels que es disposa

- L'algorisme de Max-Flow maximitza $|\{c_i \mid f(s, c_i) = 1\}|$

  i per tant maximitza $|\{c_i \mid cost(c_i) = 0\}|$

  i això vol dir que minimitza $|\{c_i \mid cost(c_i) = 5\}|$

  minimitza el nombre de vals que ha de repartir

Així tenim demostrada la correctesa

Eficiència

Quin algorisme seria més ràpid d'entre

Ford-Fulkerson i Edmons-Karp?

Notem que $|V| = n+m+2$

$$|E| = n + \sum_{i=1}^{n} |A_i| + m = O(nm)$$

- Ford-Fulkerson: $O(|E||f^*|) = O(n^2 \cdot m)$ $\quad (|f^*| \leq n)$
- Edmons-Karp: $O(|V||E|^2) = O(n^3m^2 + n^2m^3)$

En aquest cas el més eficient seria Ford-Fulkerson i caldria afegir el temps necessari per calcular l'assignació a partir del fluxo.

En total $T(n,m) = O(n^2 \cdot m)$

Menja Bé

apartat b) Examen Parcial

1.- Construir una xarxa com l'apartat a) considerant només els clients que tenen una llista de preferències (ci no menú sorpresa)

Calcular el Max-Flow $f^*_{\text{nosorpresa}}$

minimitzant així $|\{c_i \mid A_i \neq \text{sorpresa}\} \wedge \text{assig}(c_i) = \text{"val"}\}|$

és a dir minimitzant el nombre de vals de 5 euros dintre el conjunt $\{c_i \mid A_i \neq \text{sorpresa}\}$

Cost = 1€

2. Calculem $\emptyset = \sum_{i=1}^{m} q_j - |f^*_{\text{nosorpresa}}|$ menús que no han estat assignats encara

$S = |\{c_i \mid A_i = \text{sorpresa}\}|$ nombre de clients que desitgen menú sorpresa

si $\emptyset \geq S$ abastarem assignem un menú sorpresa als els clients que volem menú sorpresa cost mínim per $|\{c_i \mid A_i = \text{sorpresa}\}| : 3 \times S$

Si $\emptyset < S$ abastarem assignem tots els menus sorpresa possible, en total $\emptyset$, i assignem un val o bé roste, en total $S - \emptyset$ i els hores el cost mínim per $|\{c_i \mid A_i = \text{sorpresa}\}|$:

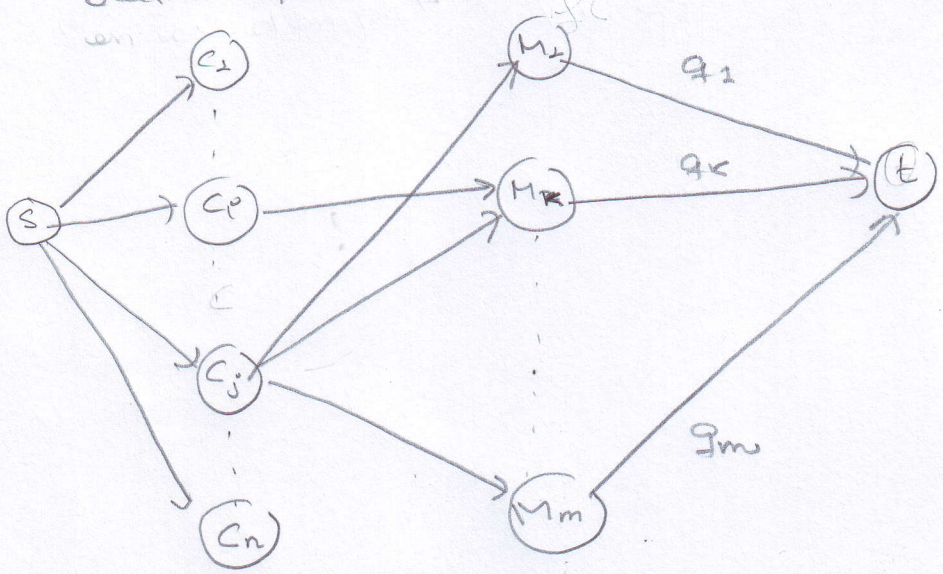
$3 \times S + 5 \times (S - \emptyset)$

3.- No os poden "barrejar" menus

clients c_i amb $A_i \subseteq \{M_1, \dots, M_m\}$ amb

clients c_j amb $A_j = \{$menu-sorpresa$\}$

Si a la xarxa tinguessim de manera temps aquests 2 tipus de clients, el client "sorpresa" accepta qualsevol menú.



Per exemple, si $q_k = 1$ ∀

$A_i = \{M_k\}$

$A_j = \{$sorpresa$\}$ $\Rightarrow (c_j, M_k) \in E$ $\forall k$ $1 \le k \le m$

maximitzar fluxe ≠ minimitzar cost

si $f(c_i, M_k) = 1 \Rightarrow$ cost = 0

si $f(c_j, M_k) = 1 \Rightarrow$ cost = 3 ?

Amb això tenim la correcció.

- Eficiència de l'algorisme proposat

Sobre l'apartat a) afegeix l'assignació de menus sorpresa ($O(n)$ addicional)

$T(n,m) = O(n^2 m)$