Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

1 Descripción del problema

Sea M el conjunto módulos disponibles, sea J el conjunto de los jefes encargados de módulos y sea T el conjunto de los técnicos. Cada técnico y cada jefe tiene módulos disponibles y solo podrá trabajar en dichos módulos, mientras que en cada módulo debe haber una cantidad máxima de C técnicos trabajando y debe tener un jefe de módulo asignado al mismo módulo, mientras que cada técnico debe trabajar un total de un módulo a la semana. El problema consiste en buscar la mejor asignación de jefes y técnicos a cada módulo, de modo de usar la mayor cantidad de técnicos disponibles.

2 Modelación matemática

2.1 Conjuntos

- $M = \{0, 1, ...\}$: Módulos disponibles semanales.
- J: El conjunto de encargados de módulo disponibles.
- T : El conjunto de técnicos

2.2 Parámetros

- C: Cantidad máxima de técnicos que pueden haber en un módulo cualquiera
- C_m : Cantidad de cupos disponibles que quedan en un módulo (Es decir se C - técnicos asignados por default)
- t_{im} : 1 si el jefe i tiene disponible para trabajar en el módulo m, 0 en otro caso.
- a_{im} : 1 si el técnico i tiene disponible para trabajar en el módulo m, 0 en otro caso.
- h : Máximo de módulos en los que puede estar un jefe.

2.3 Variables

- x_{im}: Variable binaria. 1 si el técnico i es asignado al módulo m. 0 en otro caso.
- y_{im} : Variable binaria. 1 si el jefe i es asignado al módulo m. 0 en otro caso.

2.4 Función Objetivo

Se busca maximizar la cantidad de módulos usados por los ténicos, por lo tanto la función objetivo queda como:

$$max \quad \sum_{m \in M} \sum_{i \in T} x_{im}$$

2.5 Restricciones

1. Solo se pueden agregar jefes o técnicos a módulos en los que tengan disponibles:

$$x_{im} \leq a_{im} \quad \forall i \in T, m \in M$$

$$y_{im} \leq t_{im} \quad \forall i \in J, m \in M$$

2. Solo se puede asignar un técnico a un módulo, si hay un jefe previamente asignado:

$$x_{im} \leq \sum_{j \in J} y_{jm} \quad \forall i \in T, m \in M$$

3. Máximo de módulos en los que puede estar un jefe

$$\sum_{m \in M} y_i m \leq h \quad \forall i \in J$$

4. Máximo de módulos en los que puede estar un técnico

$$\sum_{m \in M} x_{im} \leq 1 \quad \forall i \in T$$

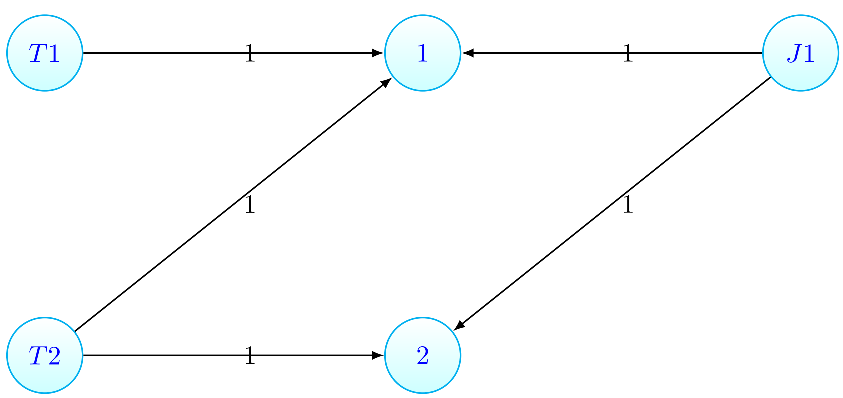
5. Naturaleza de variables

$$x_{im}, y_{im} \in \{0,1\} \forall i \in T, J, m \in M$$

La matriz asociada a este problema tendría una dimensión cercana a $|T| \cdot |M| + |J| \cdot |M| \times 3(|T| \cdot |M|) + 2(|J| \cdot |M|) + |J| + |T$
(Número de variables $\times$ Número de restricciones)

2.6 Reformulación del problema

El problema puede ser llevado a un problema de flujo en redes, donde cada técnico y cada jefe se conectan a los nodos de módulos que tengan disponibles. La demanda viene dada por parte de los nodos de módulos (el máximo de cada módulo) y la oferta será 1 en el caso de los técnicos, mientras que en los nodos de los jefes será igual al máximo de módulos que pueden estar a la semana.



Desde este punto de vista el problema se puede fraccionar para cada técnico y para cada jefe, en particular el problema de asignación se transforma en buscar los módulos con mayor disponibilidad horaria para los técnicos y que abarquen la mayor diversidad de técnicos, de modo de asignar a los jefes en dichos módulos. Para los técnicos, dado los módulos en que los jefes estarán disponibles, se busca asignarlos en un módulo *"poco restrictivo"*, es decir, buscar para cada caso el módulo en el que su asignación no implique directamente dejar a un técnico fuera, o lo que es lo mismo, asignar primero a los técnicos con menor flexibilidad horaria.

3 Algoritmo alternativo

3.1 Asignación de técnicos

Primero son eliminados todos aquellos nodos que no puedan formar parte de la solución factible, es decir:

- Nodos de módulos en que ningún jefe pueda asistir.
- Nodos de módulos a los que ningún técnico pueda asistir
- Nodos de técnicos que solo tengan disponibles módulos a los que ningún jefe puede asistir
- Nodos de jefes que tengan disponibles módulos a los que ningún técnico puede asistir

Sean M el conjunto de módulos que quedó tras realizar la etapa anterior. Para cada módulo $m \in M$, se asignarán los técnicos en relación a una *prioridad*. Un técnico i tiene prioridad n respecto a un módulo m, si este tiene disponible al módulo m y $n - 1$ módulos disponibles. El algoritmo consiste en:

1. Obtener el módulo que tenga mayor cantidad de técnicos con " 1.
2. Agregar técnicos hasta que se llene la capacidad, respetando la capacidad del nodo, o bien hasta que no queden más técnicos con el módulo disponible.
3. Quitar a los técnicos que ya fueron asignados y los módulos que completaron su capacidad. Esto implica que la prioridad de cada técnico que queda debe ser calculada nuevamente.
4. Repetir hasta que todos los módulos estén llenos, o bien se agoten los técnicos.

Sea x una asignación no vacía obtenida por este algoritmo, el conjunto de los técnicos que no tienen una asignación en x, el conjunto D está formado por los siguientes subconjuntos:

- Técnicos que no son asignados se debe a que los módulos están totalmente ocupados (El algoritmo se detiene hasta que todos los módulos estén totalmente disponibles)
- Técnicos que no pueden ser asignados debido a falta de disponibilidad horaria.

3.2 Soluciones múltiples

Sea $i, j \in T$, donde i es un técnico que es asignado a un módulo, mientras que j no. Ambos técnicos son conmutativos si al asignar i a un módulo, j también podía ser asignado y tenía la misma prioridad que i, en este caso x_2, una solución idéntica a x, solo que asigna a j en vez de a i.
Esto permite afirmar que cualquier solución que genera la misma cantidad de técnicos asignados se puede obtener al conmutar un técnico de x con uno de D.
Demostración:
Por contradicción sea z una solución que usa la misma cantidad de técnicos de x, pero que no conmuta elementos de x con los de D (es decir, z es una solución que no puede ser obtenida a partir del algoritmo presentado).

- z no puede estar formado solo por técnicos que están en x (de lo contrario es conmutable con D).
- z no puede ser formado solo con elementos de D. Pues por una parte implicaría que $|x| = |D|$ lo que no siempre se debe cumplir, pero además z es conmutable con x y luego con D, por lo que si puede ser obtenida a partir del algoritmo
- z debe entonces estar formados por elementos de x y de D.

Luego, por inducción fuerte:

1. $|D| = 0$, entonces $|x| = |T|$ y z = x y se genera una contradicción, pues z si puede ser obtenida a partir del algoritmo.

2. $|D| = 1$. Entonces todo i que esté en x, en caso de ser conmutable lo es con j.Entonces z tiene a j y a otros elementos de x. Luego por casos, si ningún elemento de z conmuta con j, puedo seleccionar uno de los que no están en z que conmuten con j y conmutar z, esto hace que z esté formado solo por elementos de x y por ende se puede obtener mediante el algoritmo.

3. Suponer que si para todo $k < n$, tal que $|D| = k$ entonces z es obtenible a partir del algoritmo. Por demostrar que si $|D| = n$, entonces z también se obtiene del algoritmo:
   Sea $D_k \subset D$, tal que $|D_k| = k < n$. Sea z una solución tal que todo elemento de D, pertenece también a D_k, por hipótesis de inducción entonces se sabe que z puede ser obtenida a partir del algoritmo. En caso contrario, si $D_k = D$, por casos:

   - $|D_k| = |x|$: Entonces z está formada solo por elementos de D y se demostró que puede ser obtenida a partir del algoritmo.
   - $|D_k| = n \leq |x|$: En este caso, z tiene $m = |x| - |D_k|$ elementos de x. Si $m < n$, entonces esto implica que existe un i que se asigna en x que conmuta con más de un elemento en D. Esto implica que necesariamente z debe estar formado por i y todos los elementos de D que conmutan con i (de lo contrario se puede conmutar y obtener x). Sin embargo, en z no pueden estar los otros compañeros de i (de lo contrario se supera la capacidad del módulo), por lo que tanto i como sus conmutables de D son conmutables con estos técnicos, finalmente se pueden conmutar y volver a obtener x. Si $m = n$ que todos los técnicos de x deben a lo más ser conmutable con uno único de D, luego z necesariamente entonces superará la capacidad del módulo.
   - $D_k = n > |x|$ Entonces z no es una solución equivalente con x.

Un alcance que no se vio en esta demostración es cuando dos técnicos que están en x puede ser mutados (intercambiar módulo), la calidad de la solución se mantiene, sin embargo esta solución también puede ser obtenida a partir del algoritmo si se tiene en cuenta que pueden conmutar de esta manera.
Con esto se demuestra que z no puede tener elementos de x ni de D, por lo tanto z es vacío, lo que genera nuevamente una contradicción, pues se había mencionando que z era una solución que asignaba la misma cantidad de técnicos que x

3.3 Optimalidad

Por contradicción, si existe una solución óptima que genera una mejor asignación que la de este algoritmo, implica que se están usando $|x|$ técnicos más un número mayor de técnicos.Sea x una solución del algoritmo y sea x^* una solución mejor que x. Por casos:

- Si para todo técnico que se asignó en x, está en x^*, entonces x^* excede la carga máxima de un módulo.

- Si todo técnico que está en D, está en x^*, sea x^*_D parte de esta asignación (que contiene exclusivamente a técnicos de D, se demostró que puede ser obtenida mediante el algoritmo, por lo que todo técnico que no esté en x^*_D se debe a que su asignación pasa a llevar una de las restricciones del problema, por lo tanto x^* pasa a llevar restricciones del problema.

- Si x^* se forma por una combinación de técnicos de x y D, entonces sea $x^*_x \subset x^*$ tal que x^*_x es obtenible a partir del algoritmo, se genera nuevamente una contradicción pues implica que x^* asigna técnicos que pasan a llevar ciertas restricciones.

Finalmente x^* debe ser vacío, lo que genera una contradicción, pues se supuso que era una mejor solución que x. Con lo anterior se demuestra que el algoritmo es capaz de generar múltiples soluciones óptimas para los técnicos.

3.4 Asignando Jefes

Dado que se asignaron los técnicos, los jefes deben ser asignados con el fin de lograr maximizar la cantidad de técnicos, en particular, deben ser asignados primero los módulos con mayor cantidad de técnicos. Por contradicción, sea $m_1, m_2, ...m_n$ la cantidad de técnicos asignados por este procedimiento a cada módulo (donde se asignó un jefe) y sea $n_1, n_2, ...n_n$ una asignación mejor que esta, se tiene entonces que:

$$m_1 + m_2 + ...m_n \leq n_1 + n_2 + ...n_n$$

Pero por definición del algoritmo $m_i \geq n_i \forall i$. Por lo que:

$$m_1 + m_2 + ...m_n \geq n_1 + n_2 + ...n_n$$

3.4.1 Múltiples soluciones

En caso de un jefe, completar su capacidad, pero los módulos que no se usan tienen la misma cantidad de técnicos asignados, entonces se pueden conmutar y se mantiene la solución óptima.

4 Asignando técnicos por default

Para añadir técnicos por default a ciertos nodos, simplemente se quitan del registro y se dejan asignados previamente a un módulo, de modo que la capacidad de ese nodo se ve reducida al momento de ejecutar el algoritmo.