88. ✎ El *problema de la evacuación* tiene la siguiente definición. Nos dan un grafo dirigido $G = (V, E)$ que representa una red de carreteras. Una cierta colección de nodos $X \subseteq V$ se designan como *nodos habitados*. Otra colección de nodos S son designados como *nodos seguros*. Suponemos que S y X son disjuntos. En el caso de una emergencia, queremos conocer rutas de evacuación desde los nodos habitados a los seguros. Un conjunto de *rutas de evacuación* se define como un conjunto de caminos en G tales que: (i) cada nodo en X aparece como el inicio de un camino, (ii) el último nodo en un camino aparece en S, (iii) los caminos no comparten aristas.

Cada ruta de evacuación proporciona una ruta a travéss de la cual los habitantes de un nodo poblado pueden escapar a un nodo seguro sin compartir ninguna parte del camino con habitantes de otros nodos.

Un conjunto de *rutas de evacuacin mixtas* se define como un conjunto de caminos en el que (i) cada nodo en X aparece como el inicio de un camino, (ii) el último nodo en un camino aparece en S.

Este tipo de rutas proporciona rutas de escape en las que parte del camino es compartido entre habitantes de distintas ciudades.

Supongamos que además, debido a las restricciones de tráfico, cada arco $e \in E$ tiene asignada una capacidad de tráfico $c(e)$ que corresponde al número máximo de vehículos que pueden atravesarlo. Una petición de tráfico $r(x)$, para un nodo habitado x, representa el número de vehículos que se quieren evacuar desde x.

(a) Dados $G = (V, E, c)$ y $S, X \subseteq V$, muestra como decidir en tiempo polinómico si un conjunto de rutas de evacuación existe o no. En caso de que si que existan, el algoritmo debe proporcionar, para cada nodo poblado, la capacidad del arco de capacidad mínima de la ruta que se inicia en él.

(b) Dados $G = (V, E, c)$, $S, X \subseteq V$, y una petición de tráfico $r(x)$, para cada $x \in X$, muestra como decidir en tiempo polinómico si existe un conjunto de rutas mixtas de evacuación que permitan enviar los vehículos solicitados desde X a nodos seguros y que, además, para cada arco, se cumpla la condición de que el número de vehículos que lo atraviesan no supere la capacidad de tráfico del arco. En caso de que sea posible la evacuación con rutas mixtas, el algoritmo tiene que devolver además, para cada nodo $x \in X$, un plan de evacuación indicando: para cada vehículo saliendo de x, un camino con origen en x y final en un nodo seguro. El total de tráfico asignado entre todos los caminos del plan de escape de todos los nodos $x \in X$ tiene que cubrir la totalidad de las peticiónes de tráfico y cumplir las restricciones de capacidad de los arcos.

Una solución.

Dados $G = (V, E, c)$ y $S, X \subseteq V$, muestra como decidir en tiempo polinómico si un conjunto de rutas de evacuación existe o no. En caso de que si que existan, el algoritmo debe proporcionar, para cada nodo poblado, la capacidad del arco de capacidad mínima de la ruta que se inicia en él.

Lo plantearemos como un problema de asignación con restricciones en una red de flujo en la que una unidad de flujo de s a t represente un camino de un nodo habitado a un nodo seguro.

La red $\mathcal{N}$ tiene

- Nodos: s, t, V, $|V| = n$.
- Aristas y capacidades:

$$\begin{array}{rl} \{(s,u) \mid u \in X\} & \text{capacidad } 1 \\ \{(u,t) \mid u \in S\} & \text{capacidad } \infty (\text{o } |X|) \\ \{(u,v) \in E\} & \text{capacidad } 1 \end{array}$$

Un camino de s a t tiene la forma $s \to u \to \cdots \to v \to t$, con $u \in X$ y $v \in S$ y representa una ruta de un nodo habitado a un nodo seguro. Si transporta una unidad de flujo, debido a la capacidad, las aristas en el camino no se pueden usar en ningún otro camino.

Un nodo en S puede ser el final de más de una camino, por eso la capacidad en (u, t) es ilimitada.

Si el problema tiene solución podremos asignar a todos los nodos poblados un camino a un nodo seguro.

Si traducimos cada asignación en una unidad de flujo en el camino asociado, obtenemos un flujo válido. Con valor de flujo n que es el máximo posible.

De la misma forma, si el flujo máximo en la red es n, reinterpretando el flujo como asignación, podemos asignar a cada nodo habitado un camino (en aristas con flujo 1) hasta un nodo seguro.

Algorithmo:

Construir $\mathcal{N}$

$F = MaxFlow(\mathcal{N})$

if $|F| < n$ **then**

  return NO

else

  return F

El coste de la llamada a MaxFlow: Si $|V| = n$ y $|E| = m$, el número de vértices en la red es $N = n + 2$ y el número de aristas es $M \leq n + m$.

- Con FF el coste es $O(n(N + M)) = O(n(n + m))$
- Con EK el coste es $O(M^2 N) = O(n(n + m)^2)$

Utilizaremos FF.

El coste total del algoritmo es $O(n(n + m))$.

Una vez obtenido un flujo F con valor n tenemos que obtener caminos que nos permitan hacer la evacuación.

Algoritmo:

Repetir $|X|$ veces:

(a) Considerar solo las aristas en $\mathcal{N}$ con flujo 1.

(b) Con un BFS obtener un camino P de s a t.

(c) P contiene la ruta de evacuación del nodo poblado que sigue a s.

(d) Poner a cero el flujo en las aristas de P

Que tiene coste $O(n(n+m))$.

Dados $G = (V, E, c)$, $S, X \subseteq V$, y una petición de tráfico $r(x)$, para cada $x \in X$, muestra como decidir en tiempo polinómico si existe un conjunto de rutas mixtas de evacuación que permitan enviar los vehículos solicitados desde X a nodos seguros y que, además, para cada arco, se cumpla la condición de que el número de vehículos que lo atraviesan no supere la capacidad de tráfico del arco. En caso de que sea posible la evacuación con rutas mixtas, el algoritmo tiene que devolver además, para cada nodo $x \in X$, un plan de evacuación indicando: para cada vehículo saliendo de x, un camino con origen en x y final en un nodo seguro. El total de tráfico asignado entre todos los caminos del plan de escape de todos los nodos $x \in X$ tiene que cubrir la totalidad de las peticiónes de tráfico y cumplir las restricciones de capacidad de los arcos.

Lo plantearemos como un problema de asignación con restricciones en una red de flujo en la que una una unidad de flujoen un camino de s a t represente un camino de un nodo habitado a un nodo seguro por el que transitará un vehículo.

La red $\mathcal{N}$ tiene

- Nodos: $s, t, V, |V| = n$.
- Aristas y capacidades:

$$\begin{aligned} \{(s,u) \mid u \in X\} &\qquad \text{capacidad } r(u) \\ \{(u,t) \mid u \in S\} &\qquad \text{capacidad } \infty \\ \{(u,v) \in E\} &\qquad \text{capacidad } c(u,v) \end{aligned}$$

Un camino de s a t tiene la forma $s \to u \to \dots v \to t$, con $u \in X$ y $v \in S$ y representa una ruta de un nodo habitado a un nodo seguro. Si la arista con menor flujo tiene flujo k, esta ruta la pueden utilizar k vehículos.

Si el problema tiene solución podremos asignar a toda la demanda de nodos poblados a nodos seguros a través de la red, sin superar nunca la capacidad de las carreteras.

De la misma forma, si el flujo máximo en la red es $\sum_{x \in X} r(x)$, podemos asignar a cada vehículo en un nodo habitado un camino hasta un nodo seguro. Estos caminos pueden compartir tramos, pero nunca exceden la restricción de tráfico.

Algorithmo:

Construir $\mathcal{N}$
$F = MaxFlow(\mathcal{N})$
if $|F| < \sum_{x \in X} r(x)$ **then**
  return NO
else
  return F

El coste de la llamada a MaxFlow: Si $|V| = n$ y $|E| = m$, el número de vértices en la red es $N = n + 2$ y el número de aristas es $M \leq n + m$. Sea $R = \sum_{x \in X} r(x)$.

- Con FF el coste es $O(R(N + M)) = O(R(n + m))$
- Con EK el coste es $O(M^2 N) = O(n(n + m)^2)$

Utilizaremos EK ya que no tenemos información sobre el valor de R y podría ser exponencial en el tamaño de la entrada.

El coste total del algoritmo es $O(n(n + m)^2)$.

Una vez obtenido un flujo F con valor n tenemos que obtener caminos que nos permitan hacer la evacuación.

Algoritmo:

Mientras hay aristas con flujo positivo:

(a) Considerar solo las aristas en $\mathcal{N}$ con flujo ¿ 0.

(b) Con un BFS obtener un camino P de s a t.

(c) Sea b el flujo en la arista con flujo mínimo en P

(d) P contiene la ruta de evacuación del nodo poblado x que sigue a s, esta ruta se asigna a b vehículos de x.

(e) Restar b al flujo en las aristas de P

Que tiene coste $O((n+m)^2)$ ya que en cada iteración al menos una arista queda con flujo 0.