Zombies

Claudia Alvarez Martínez
Roger Moreno Gutiérrez
Jan Carlos Pérez González

Cuarto año, Ciencias de la Computación.
Facultad de Matemática y Computación, Universidad de La Habana, Cuba

1 Problema:

En una ciudad infestada de zombis, los pocos supervivientes están intentando escapar. La ciudad está representada por un conjunto de intersecciones conectadas por calles, y cada camino tiene una longitud.

Roger, uno de los supervivientes propone buscar el camino más corto en un mapa que encontró tirado por ahí. Cuando el grupo está a punto de partir, Jan se levanta y dice: "Tengo una idea mejor. En lugar de ir por el camino más corto, busquemos uno de los más largos, de al menos tamaño k. Así sorprenderemos a los zombies, nunca dejes que sepan tu próximo movimiento".

El grupo de brillantes supervivientes comenzó a aplaudir asombrado, mientras que Roger se quedaba frío pensando "¿de cuando acá los zombies pueden razonar?"

Ayude al grupo encontrando, dado un mapa de intersecciones y calles, si existe un camino simple de tamaño al menos k.

(PD: De más está decir que todo el grupo murió entre terribles sufrimientos).

2 Nuestra redefinición del Problema:

Sea $G = (V, E)$ el grafo que representa con V el conjunto de intersecciones, y con las aristas en E las conexiones entre dichas intersecciones. Se desea saber si para un grafo cualquiera G, existe un camino simple de tamaño al menos k.

Camino simple: Es un camino sobre G que no repite vértices, excepto quizás, el primero y el último.

3 NP-Completitud:

Saber si en G existe un camino simple de tamaño al menos k, es fácil de responder si sabemos la respuesta al siguiente problema de decisión.

***K-PATH*:** Determinar si existe en G un camino simple de tamaño k.

Demostremos que *K-PATH* es NP-Completo:

Necesitamos demostrar que ***K-PATH*** $\in$ NP y ***K-PATH*** $\in$ NP-Hard.

- ***K-PATH*** $\in$ NP:
  Supongamos que tenemos un grafo no dirigido $G = (V, E)$ y un entero k; comprobar que una secuencia de vértices $V_k = v_1, v_2, ..., v_k$ es un camino de tamaño k en G, se logra asegurando que $V_k \subseteq V$, que todo vértice $v_i \in V_k$ aparece solo una vez en la secuencia y que, efectivamente, es posible hacer la transición $v_i \to v_{i+1}$ mediante una arista existente en E. Esta comprobación se realiza en tiempo polinomial, por lo que podemos decir que ***K-PATH*** es NP.

- ***K-PATH*** $\in$ NP-Hard:
  Definamos como ***HAM-CYCLE*** al problema de decisión de determinar si en un grafo existe un ciclo de Hamilton, el cual sabemos que es uno de los 21 problemas NP-Completo de Karp.

  Vamos a demostrar que ***K-PATH*** es NP-Hard a partir de que ***HAM-CYCLE*** es menos o igual de duro (hard) que ***K-PATH***, o sea:

  $$\textit{HAM-CYCLE} \leq_p \textit{K-PATH}$$

  Para ello hagamos una reducción de ***HAM-CYCLE*** a ***K-PATH***.

  El algoritmo de reducción T toma como entrada una instancia $\langle G \rangle$ del problema ***HAM-CYCLE*** y produce como salida la instancia $\langle G', k \rangle$, donde $G' = G$ y $k = |V|$, o sea:

  $$T : \langle G \rangle \to \langle G, |V| \rangle$$

  Es evidente que esta transformación se computa en tiempo polinomial, por lo que solo restaría demostrar que dicha transformación es de hecho una reducción: el grafo G tiene un ciclo de Hamilton $\iff$ G tiene un camino simple de longitud $|V|$.

  $\Rightarrow$

  > G tiene un ciclo de Hamilton $\Rightarrow$ tiene un camino simple que comienza y termina en el mismo vértice, y los contiene a todos $\Rightarrow$ tiene un camino simple de longitud $|V|$.

$\Leftarrow$

> G tiene un camino simple de longitud $|V|$, y como un camino simple no repite vértices, excepto, quizás el primero y el último, dicho camino tiene que ser un ciclo de longitud $|V| \Rightarrow G$ tiene un ciclo de Hamilton.

De esta forma queda demostrado que T es una reducción válida y que por tanto ***K-PATH*** $\in$ NP-Hard.

Luego, ***K-PATH*** $\in$ NP $\wedge$ ***K-PATH*** $\in$ NP-Hard $\Rightarrow$ ***K-PATH*** $\in$ NP-Completo.

4 Acercamiento práctico:

Dada la pertenencia de ***K-PATH*** al conjunto de problemas NP-Completos, no existe una solución conocida que funcione para el caso general de forma polinomial. Veamos a continuación alguna implementación práctica.

4.1 Fuerza bruta para el caso general

Una primera solución con este enfoque pasa por comprobar si existe un camino entre todas las posibles permutaciones de subconjuntos de $k+1$ vértices de V, o si hay un ciclo simple que involucra k vértices. Veámoslo con mayor deteniemiento. Sea $G = (V, E)$ un grafo no dirigido, y sea un k, $0 \leq k \leq |V|$. Identifiquemos dos tipos de caminos simples, dígase **camino simple completo** al camino simple que no repite vértice en lo absoluto, y llamémosle **camino simple de ciclo** al camino que determina un ciclo simple, donde solo se repite un vértice, el primero y el último. Dicho esto, el algoritmo propuesto intenta determinar si existe alguno de estos dos caminos en el grafo, para un k dado. Si $k \leq 2$, no comprobaremos la existencia de *camino simple de ciclo*; y si $k = |V|$, no comprobaremos la existencia de *camino simple completo*.

4.1.1 Análisis de correctitud

Hagamos un análisis a partir de los distintos valores que puede tomar k, $(0 \leq k \leq |V|)$. Si $k = 0$, por definición podemos decir que todo grafo no vacío tiene un camino de tamaño 0. Nos quedaría por analizar los valores de k, tales que $0 < k \leq |V|$. Si $k = |V|$, el único camino simple posible de longitud k es si el grafo es de Hamilton. El algoritmo comprueba en ese caso la existencia del *camino simple de ciclo* que contenga todos los vétices de V. Si $k \leq 2$, entonces no es posible encontrar un ciclo simple con esos vértices en un grafo no dirigido, dada la propia definición de ciclo simple, por lo que el algoritmo comprueba solo los *caminos simples completos*. Finalmente quedaría por comprobar los valores de k que cumplen $2 < k < |V|$, para los que es posible encontrar ambos tipos de caminos simples, por lo que el algoritmo hace ambas comprobaciones.

4.1.2 Análisis de Complejidad Temporal

Al ser un algoritmo que comprueba de cierta forma todos los posibles caminos de tamaño k, se espera que su complejidad temporal sea exponencial en cuanto a la cantidad de caminos.
Analicemos el algoritmo con más detalle. Para comprobar si hay un *camino simple completo*, primeramente se obtienen todos los subconjuntos de tamaño $k+1$, y para cada uno se realizan las permutaciones de esos vértices, en las que se comprueba la existencia de aristas en el grafo que conecten dichos vértices; esto se computa en:

$$O(\binom{n}{k}*k!*k) = O(\frac{n!}{k!(n-k)!}*k!*k) = O(n*n-1*...*n-k+1*k!*k) = O(n^k*k!*k)$$

De forma análoga se determina la complejidad de comprobar si hay *caminos simples de ciclo*, lo que nos lleva a una complejidad total $O(2*n^k*k!*k)$ que sabemos que es $O(n^k*k!*k)$

4.1.3 Análisis de Complejidad Espacial

En cuanto a la complejidad espacial, notemos que se almacenan todas las combinaciones $\binom{n}{k}$, y por cada una las $k!$ permutaciones. Estas últimas se analizan sin costo espacial alguno, por lo que se concluye que analizar los *caminos simples completos* es espacialmente $O(n^k * k!)$. Tras incluir las comprobaciones de *caminos simples de ciclo*, tenemos una complejidad espacial total de $O(2 * n^k * k!) = O(n^k * k!)$.