

Contents

El algoritmo de Dinitz	1
idea básica de Dinitz	1
Esquema básico de Dinitz	1
Flujos bloqueantes	2
Definición:	
.	2
Algoritmos tipo Dinic	2
Layered Networks	2
network “por niveles”.	2
Definición:	
.	2
Network auxiliar,	2
vértices	2
Lados y capacidades:	3
Otra forma de pensar esto	3
Cuyos lados son:	
.	3
Construcción	3
Observaciones	4

El algoritmo de Dinitz

idea básica de Dinitz

“guardar” todos los posibles caminos aumentantes de la misma longitud (mínima) en una estructura auxiliar.

esta primera parte se hace, al igual que con Edmonds-Karp, con BFS, pero guardamos toda la información y no sólo la necesaria para construir un camino.

Esquema básico de Dinitz

1 Construir un network auxiliar (usando BFS). 2 Correr Greedy con DFS en el network auxiliar hasta no poder seguir.

3 Usar el flujo obtenido en el network auxiliar para modificar el flujo en el network original.

4 Repetir [1] con el nuevo flujo, hasta que, al querer construir un network auxiliar, no llegamos a t.

En el network auxiliar, como se usa Greedy, nunca se des-satura un lado. los lados siguen pudiendo des-saturarse, es sólo en el network auxiliar que no se des-saturan.

Flujos bloqueantes

Definición:

Llamaremos a un flujo en un network si todo camino DIRIGIDO desde s a t tiene al menos un lado saturado. (es decir con $c(- \rightarrow xy) = f(- \rightarrow xy)$).

En otras palabras, si cuando queremos usar Greedy en el network, no llegamos a t .

Algoritmos tipo Dinic

- 1 Construir un network auxiliar (usando BFS).
- 2 Encontrar un flujo bloqueante en el network auxiliar. 3 Usar ese flujo bloqueante del network auxiliar para modificar el flujo en el network original.
- 4 Repetir [1] con el nuevo flujo, hasta que, al querer construir un network auxiliar, no llegamos a t .

Layered Networks

network “por niveles”.

Definición:

Un Network por niveles es un network tal que el conjunto de vértices esta dividido en subconjuntos V_i (los “niveles”) tales que sólo existen lados entre un nivel y el siguiente. Es decir, $- \rightarrow xy \in E \quad i : x \in V_i, y \in V_{i+1}$

Network auxiliar,

vértices

el conjunto de vértices es $V = \bigcup_{i=0}^r V_i$, donde los V_i son:

Sea $r = df(s, t)$ donde df es la función definida en la prueba de Edmonds-Karp.

Es decir, r es la distancia entre s y t usando caminos aumentantes.

Para $i = 0, 1, \dots, r - 1$, definimos $V_i = \{x : df(s, x) = i\}$.

Observar que entonces $V_0 = \{s\}$.

Definimos $V_r = \{t\}$

Lados y capacidades:

$u \rightarrow v$ es un lado del network auxiliar si:

$u \in V_i$, $v \in V_{i+1}$ y:

1 $u \rightarrow v$ es un lado del network original con $f(u \rightarrow v) < c(u \rightarrow v)$. o:

2 $v \rightarrow u$ es un lado del network original con $f(v \rightarrow u) > 0$.

En el caso de [1], la capacidad de $u \rightarrow v$ en el network auxiliar será $c(u \rightarrow v) - f(u \rightarrow v)$, y en el caso de [2], la capacidad del lado $u \rightarrow v$ en el network auxiliar será $f(v \rightarrow u)$.

Otra forma de pensar esto

es que construimos primero un “network residual”

Cuyos lados son:

los lados originales, con capacidad igual a $c - f$ y los reversos de los lados originales, con capacidad f .

Y luego, de ese network residual nos quedamos con los lados que unan vertices de distancia i con vertices de distancia $i + 1$.

Construcción

la forma de construirla es tomar como V_0 a $\{s\}$.

Y luego ir construyendo una cola a partir de s al estilo Edmonds-Karp.

Y si x agrega a z y x está en V_i , entonces z está en V_{i+1} .

si z ya está agregado, si bien z no se $u \rightarrow v$ vuelve a agregar, el lado $u \rightarrow v$ si se agrega al network auxiliar, siempre y cuando la distancia de z a s sea uno mas que la distancia de x a s .

Si en algún momento llegamos a t , no paramos inmediatamente, pues podria haber mas lados que lleguen a t .

Pero borramos todos los vertices que ya hubieramos incluido en el mismo V_r en el cual estamos poniendo a t

Y de ahi en mas no agregamos mas vertices, sólo lados entre vertices de V_{r-1} y t .

Observaciones

Como el network auxiliar es un network por niveles, **todos** los caminos de un mismo network auxiliar deben tener **la misma longitud**.

Si obtienen dos caminos de distinta longitud dentro de un mismo NA, lo tienen mal.