Pràctica 1 || Aquæductus

Pablo Fraile Alonso

4 d'abril de 2021

Índex

Índex de figures

1 Funcionament algorisme

1.1 Primera idea: Backtracking

1.1.1 Per què no utilitzar Backtracking en aquest cas

cost de $O(n!)$.

1.2 Alternativa a backtracking: Principi d'optimitat

Abans de poder comentar la solució, hem d'entendre que és el principi d'optimitat:

> Principi d'optimitat: Una política òptima té la propietat que sigui quin sugui l'estat inicial i la decisió inicial, les decisions restants han de construir una política òptima respecte a l'estat resultat de la primera decisió.
>
> (Richard E.Bellman)

Per tant, seguint aquesta definició podem dir que un problema podrà ser resolt seguint el principi d'optimitat si la seva solució òptima pot ser construida eficientment a partir de les solucions òptimes dels seus subproblemes. En altres paraules, que podem resoldre un problema gran donades les solucions dels seus problemes petits.

1.2.1 Aplicació i funcionament en el nostre cas d'ús

En el nostre problema dels aqueductes, veiem que podem aplicar el principi d'optimitat per a trobar una solució òptima, ja que veiem que la solució és construida eficientment a partir de les solucions òptimes dels seus subproblemes. Per a explicar-ho millor he decidit resoldre un petit exemple.

Donada la següent entrada:

```
5 6 180 20
0 0
2 2
3 1
5 3
7 2
```

Figura 1: Entrada exemple

On podem veure que tenim 5 punts, una altura de aqueducte de 6, $\alpha = 180$ i $\beta = 20$. Si ho representem en un eix de coordenades, el perfil del sòl ens queda com la figura 2, en canvi, si ho volem veure en forma de digraf (ja descartant opcions que no són vàlides) ens queda com a resultat la figura 3

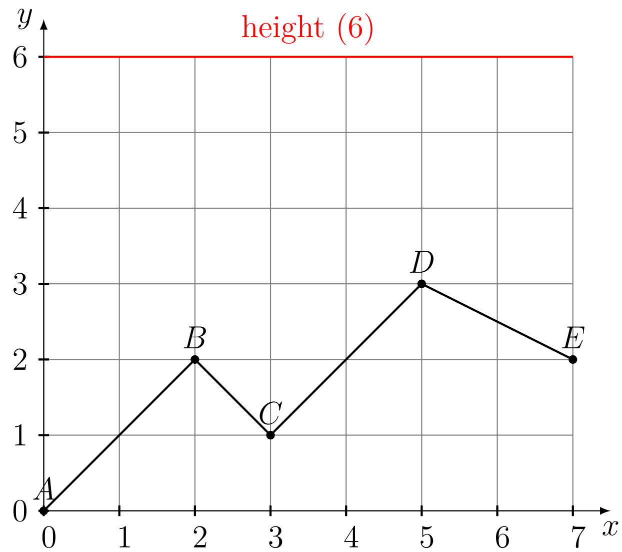

Figura 2: Exemple representat eix de coordenades

Finalment, anomenarem la funcio $f(x)$ com el minim cost per anar al node E. En el cas d'estar al propi node E, aquesta funció retornarà 0 (figura 4).

En el cas de $f(D)$, únicament té una opció possible, anar de D a F, per tant el cost minim sera aquest recorregut (figura 5).

En el cas de $f(C)$, té l'opció d'anar a D o d'anar a E. En aquest cas calcularem el cost de C a E i el cost de C a D + $f(D)$ i agafarem el mínim. Calculem cost de C a E i ens dona 1940, en canvi, el cost de C a D + $f(D)$ ens dona 2320. Per tant, el cost minim desde C sera anant de C a E (figura 6).

En el cas de $f(B)$, farem el mateix que amb $f(C)$. Calcularem quant val el cost de B a E, de B a C + $f(C)$ i de B a D + $f(D)$ i agafarem el minim. En aquest cas el minim es de B a E (1940) (figura 7).

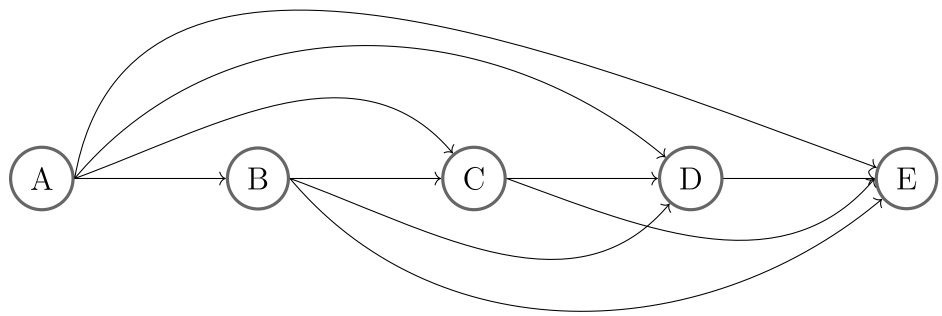

Figura 3: Exemple representat en forma de dígraf

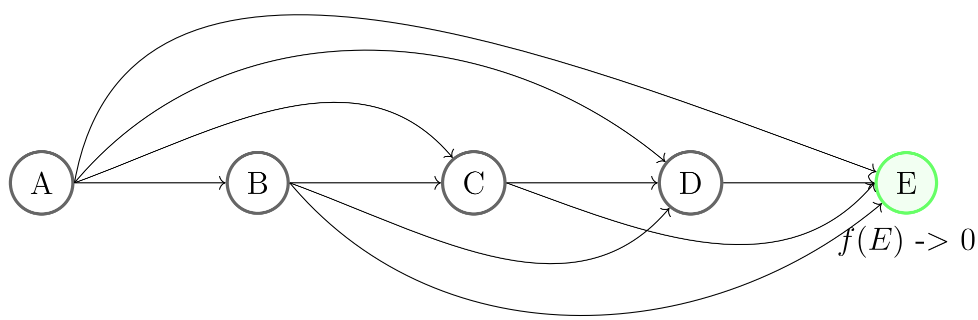

Figura 4: resultat de $f(E)$

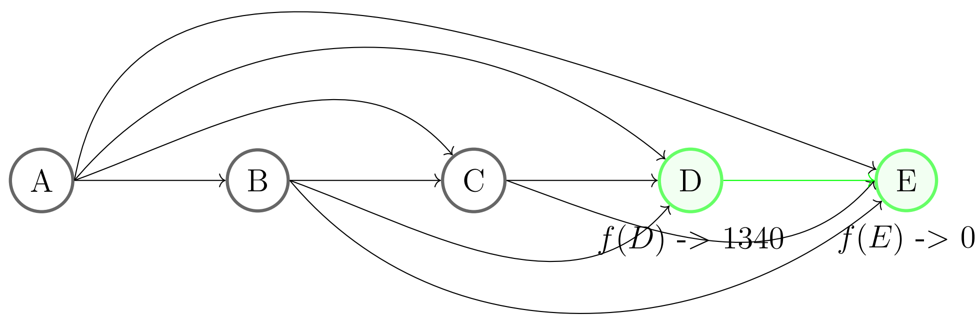

Figura 5: resultat de $f(D)$

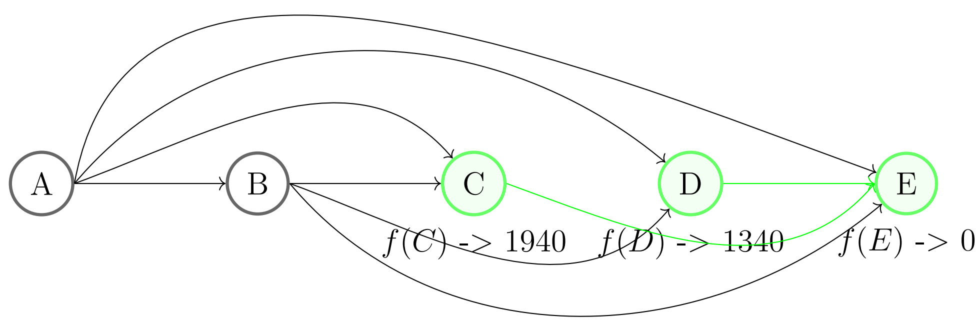

Figura 6: resultat de $f(C)$

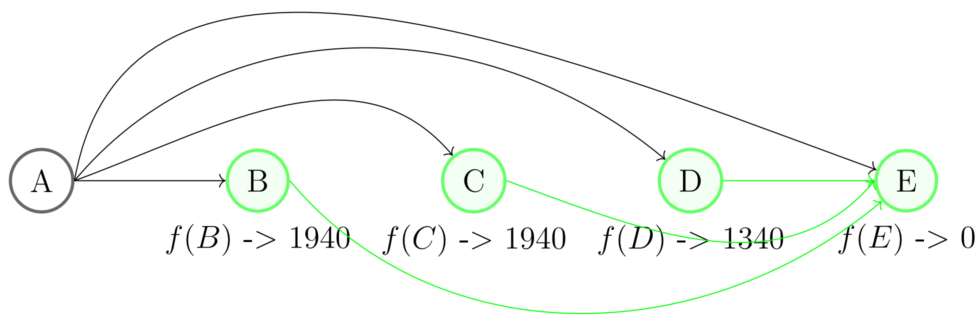

Figura 7: resultat de $f(B)$

Finalment, en el cas de $f(A)$, haurem de calcular el cost de A a E, de A a B + $f(B)$, de A a C + $f(C)$ i de A a D + $f(D)$ i agafar el minim cost. En aquest cas el minim es de A a E (2780) (figura 8)

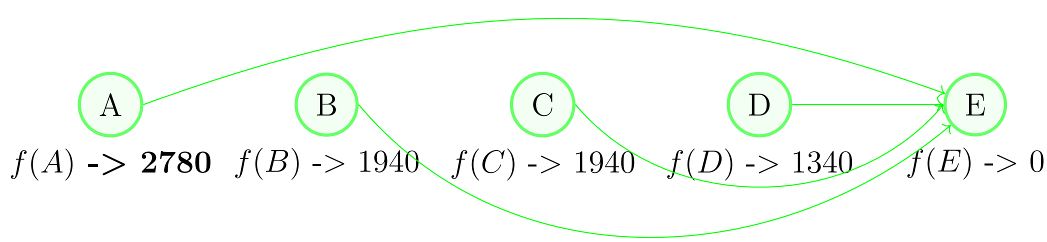

Figura 8: resultat del aqueducte minim ($f(A)$)

1.2.2 Demostració per reducció al absurd

Donat un aqueducte que va d'un punt A a un punt J i que té recorregut $R_{a...j}$ és el óptim (figura: 9).

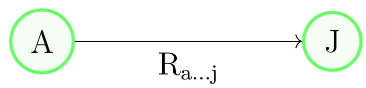

Figura 9: Aqueducte de punt A a punt J

Assumirem també que aquest recorregut passa per el punt K, per tant ara podem separar el recorregut com $R_{a...k}$ & $R_{k...j}$ (figura 10)

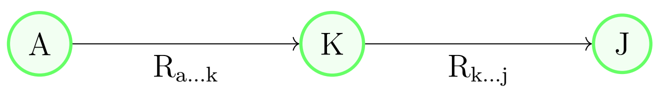

Figura 10: Aqueducte de punt A a punt J passant per K

A partir de la figura 10, considerarem que del punt A al punt K pot haver-hi un recorregut mes optim, que anomenarem $R'_{a...k}$.

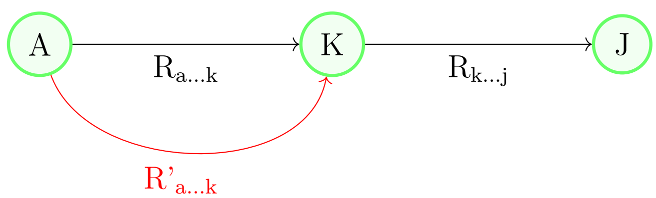

Figura 11: Graf que representa un posible $R'_{a...k}$

Si $R'_{a...k}$ és més òptim que $R_{a...k}$, llavors vol dir que:

$$R'_{a...k} < R_{a...k}.$$

Llavors:

$$\mathbf{R'_{a...k} + R_{k...j} < R_{a...k} + R_{k...j}}$$

Però aquesta afirmació NO pot ser certa! Ja que en un principi hem assegurat que $\mathrm{R_{a...k} + R_{k...j}}$ era la solució òptima i per tant no hi pot haver-hi cap més petita que aquesta.

1.2.3 Especificació formal

Un cop ja sabem el funcionament del algorisme i hem demostrat que el seu comportament es correcte, podem especificar-lo amb una formula molt similar a la de l'equacio de Bellman (ja que tal i com s'ha dit a la subseccio 1.2.1 aquest problema es resolt seguint el principi d'optimitat).

$$v(x_0) = min(f(x_0) + v(x_1)) \tag{1}$$

On $v(x)$ es la formula de la especificacio, x_0 es el primer pilar del aqueducte i x_1 es el resultat d'aplicar $v(x)$ al pilar que va despres de x_0

2 Cost algorisme

2.1 Iteratiu

$O(n^3)$

2.2 Recursiu

$O(n^3)$

3 Problemes/consideracions

3.1 Nombres en C++

Primerament es va pensar i elaborar l'algorisme en el llenguatje de programació python, i a continuació es va migrar el codi a C++. El problema va estar en no es va pensar que python al tindre tipus dinàmics el mateix intérpret assigna un tipus a cada variable de forma inteligent, mentres que a C++el propi programador és el que assigna els tipus. Això va generar un problema ja que, com els tests eren súmament grans i podíen arribar a fer operacions com per exemple 10000^2, feia que no fos suficient amb els tipus integer, i s'hagués d'utilitzar tipus com per exemple "long long int" o "unsigned long long int".

4 Conclusions

Appendix A: Pseudocodi algorisme iteratiu

Appendix B: Pseudocodi algorisme recursiu