Juan Carlos Soriano Valle
1493037
13/01/2021

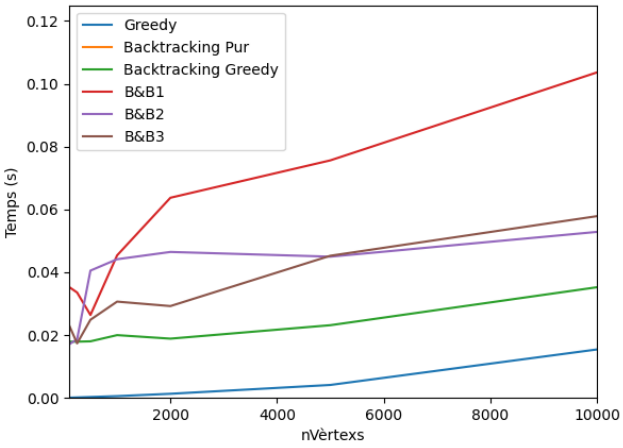
Informe pràctica Salesman

Anàlisi dels diferents algorismes

1. Com varia el temps d'execució dels algorismes en variar el nombre de vèrtexs?

Vèrtexs	100	250	500	1000	2000	5000	10000
Greedy	65.53us	149.9us	288.83us	568.3us	1.31ms	4.12ms	15.42ms
Backtracking pur	23,89s	29,67s	70,36s	116s	1028,95s	-	-
Backtracking Greedy	17.89ms	17.9ms	18.01ms	19.98ms	18.86ms	23.14ms	35.22ms
B&B 1	35.3ms	33,54ms	26.39ms	45.37ms	63,71ms	75.61ms	103.6ms
B&B 2	17.05ms	18.68ms	40.52ms	44.14ms	46.46ms	44.96ms	52.84ms
B&B 3	23,12ms	17.36ms	24.87ms	30.65ms	29.23ms	45.25ms	57.85ms

Els paràmetres de les mesures han estat mantenir el nombre d'arestes dintre del mínim que es pot establir amb els vèrtexs seleccionats i mantenint el nombre de visites a 10.



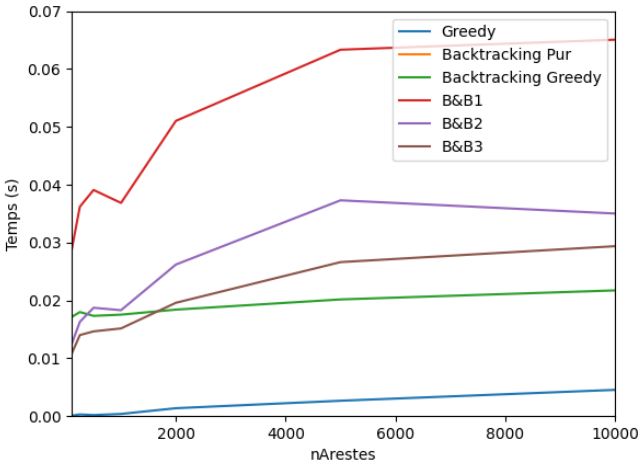
Veient els resultats obtinguts, veiem com en qüestió de temps-nvèrtexs el més òptim és l'algorisme greedy, sense tenir en compte que segurament no obtingui la solució òptima. La majoria d'algorismes tenen una forma més semblant a la lineal, menys el B&B2 que té un cert aspecte de logarítmica i el cas del Backtracking pur, que no surt a la gràfica per les altes dimensions dels resultats però si s'observen els resultats a la taula veiem com tenim una forma exponencial.

2. Com varia el temps d'execució dels algorismes en variar el nombre d'arestes?

Arestes	100	250	500	1000	2000	5000	10000
Greedy	79.93us	292.27us	183.63us	379.6us	1.37ms	2.66ms	4.54ms
Backtracking pur	123s	1008s	-	-	-	-	-
Backtracking Greedy	17.13ms	17.99ms	17.33ms	17.54ms	18.41ms	20.17ms	21.73ms
B&B 1	28.69ms	36.19ms	39.09ms	36.85ms	51.03ms	63.33ms	65.06ms
B&B 2	12.42ms	16.3ms	18.73ms	18.31ms	26.2ms	37.3ms	35.02ms
B&B 3	10.77ms	14.01ms	14.65ms	15.16ms	19.59ms	26.62ms	29.37ms

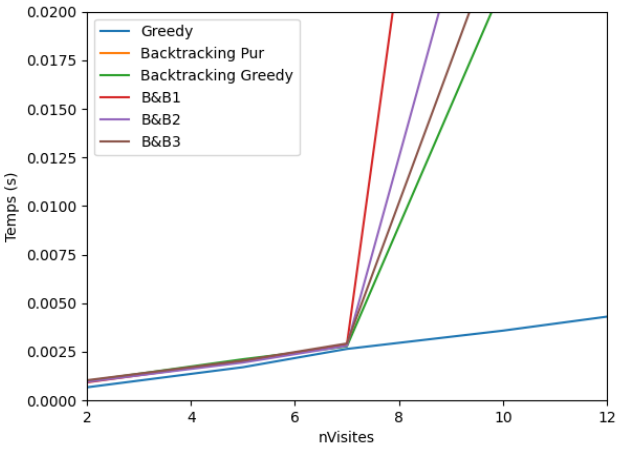
Si ara ens fixem en els resultats referents a la variació de les arestes, veiem com algunes tendències han canviat respecte la gràfica de variació dels vèrtexs.

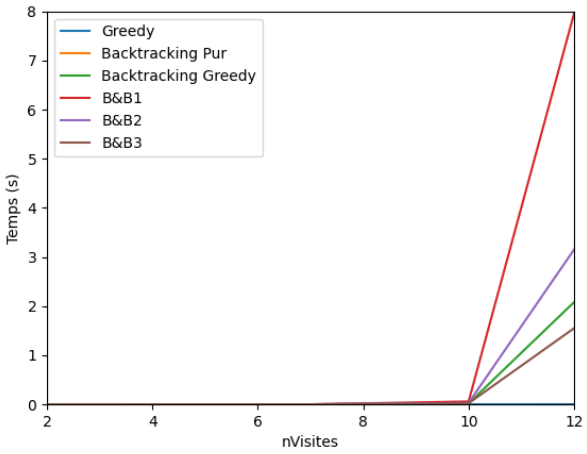
El cas dels dos algorismes Greedy segueixen en una forma lineal, mentre que els B&B tenen una forma logarítmica. Una altra vegada la línia referent al Backtracking Pur no apareix ja que com es pot observar a la taula superior la primera configuració amb 100 arestes ja s'enfila als 123 segons, pels 28,69ms del segon més lent.



3. Com varia el temps d'execució dels algorismes en variar el nombre de visites?

Visites	2	5	7	10	12
Greedy	675.4us	1.71ms	2.65ms	3.59ms	4.31ms
Backtracking pur	41.78s	-	-	-	-
Backtracking Greedy	998us	2.12ms	2.78ms	21.32ms	2.08s
B&B 1	929.7us	2.03ms	2.87ms	61.12ms	7.95s
B&B 2	960.83us	1.95ms	2.81ms	31.87ms	3.15s
B&B 3	1.04ms	2.05ms	2.93ms	24.63ms	1.55s





En aquesta relació temps-nVisites és on podem veure realment on està el condicionant del rendiment del nostre programa. Primerament descriurem la especificació de les proves. S'ha utilitzat el mateix graf per a totes les proves. Aquest graf es compon de 2000 vèrtexs i 5000 arestes, on només s'han anat canviant el nombre de visites depenent de les proves que s'estaven desenvolupant. També cal a dir que com a tots els anteriors experiments, el resultat individual, per exemple l'algorisme Greedy amb 2 visites, és el resultat de la mitja del temps en 3 execucions diferents on a cadascuna de les execucions es mantenia el graf però es canviaven la distribució de les visites (totes elles aleatòriament distribuïdes).

Centrant-nos en els resultats, podem veure clarament la tendència exponencial del conjunt d'algorismes. L'única diferència es troba a l'algorisme Greedy, on sense trobar sempre la solució òptima, es troba una solució en un temps lineal.

4. Com empitjora el resultat de l'algorisme Greedy respecte la solució òptima en variar nombre de vèrtexs, arestes i visites? Per què no varia en modificar el nombre d'arestes i visites?

Per tal de mesurar l'error que ens genera l'algorisme Greedy envers la solució òptima ens fixarem en la longitud del camí que ens resulta de l'algorisme Greedy i el compararem amb el resultat òptim del problema. Per això, l'algorisme que considerem òptim serà el Branch & Bound versió 3, ja que com hem pogut veure en els 3 punts anteriors, és la que soluciona el problema en una mitjana de temps millor amb totes les configuracions testades fins ara.

Canviar Vèrtexs (1000 Arestes)	**Relació error Greedy/B&B3**
100 Vèrtexs	5/10 (4.77% delta)
250 Vèrtexs	8/10 (1.20% delta)
500 Vèrtexs	5/10 (3,03% delta)
Canviar Arestes (200 Vèrtexs)	
500 Arestes	8/10 (1.43% delta)
1000 Arestes	8/10 (1.13% delta)
2000 Arestes	7/10 (2,83% delta)
Canviar Visites (500V + 2500A)	
3 Visites	10/10 (0% delta)
5 Visites	5/10 (2,58% delta)
7 Visites	2/10 (7.83% delta)
10 Visites	1/10 (5.54% delta)

Aquesta taula és el resultat de les proves fetes. Per a cada configuració, s'ha posat a prova el mateix graf però canviant la disposició de les visites. Un exemple d'un cas particular és que per la prova del graf amb 1000 arestes i 100 vèrtexs s'han establert 4 visites que s'han anat canviat aleatòriament en 10 ocasions, per després treure el percentatge dels errors entre l'algorisme Greedy i el B&B3. El delta significa quant canvia la llargada dels conjunts de camins entre els dos algorismes.

Havent especificat la configuració de les proves, anem a observar-les. Si ens fixem tant en la variació d'arestes com la variació de vèrtexs, no veiem cap variació important en els resultats. Un punt estrany es troba en la secció de variació de vèrtexs, però segurament sigui pel fet que s'han executat 10 proves. En cas que s'hagessin executat més, aquest percentatge s'hauria normalitzat entre les diferents especificacions del graf.

On si veiem un canvi molt important és en el cas de la variació de les visites. Mentre que amb menys visites no hi ha diferència entre un algorisme o un altre, mentre es va pujant aquest número de visites aquesta bretxa es va fent més gran. L'explicació d'aquest fet la trobem en com funcionen aquests algorismes. Contra més visites intermèdies entre l'arribada i la sortida, més opcions té l'algorisme Greedy per trobar una solució que no sigui òptima, ja que aquest algorisme no troba la optimització global, sinó que va agafant la millor opció entre visites. Això vol dir que mentre amb 3 visites hi ha 2 camins a calcular, amb 10 visites n'hi ha 9 cosa que complica que l'algorisme Greedy trobi la solució òptima global.

5. Quant ha de podar un algorisme de Branch & Bound (1,2,3) per ser millor que un algorisme backtracking greedy?

Un algorisme B&B sempre serà millor que un algorisme basat en la metodologia Greedy. Aquest fet és gracies a que un algorisme Greedy troba sempre una solució subòptima, però pot ser que aquesta solució no sigui la òptima global. Aquest fet si que ens ho compleix l'algorisme Greedy sigui quina sigui la seva heurística.

Per tant, si es vol aconseguir un temps d'execució millor que un algorisme amb modalitat Greedy haurem d'aturar l'algorisme tan bon punt tinguem una solució completable, fent també una poda molt agressiva que ens elimini les branques que no ens garanteixin una solució subòptima.

Hem de recordar que amb aquesta poda tan agressiva el que podem aconseguir és un temps millor que en un algorisme Backtracking Greedy, però mai podem garantir que el camí resultant sigui l'óptim.