Anàlisi estadístic d’algorismes “pathfinding”

Ferran Benéitez - David Morais - Ada Peña

21 Desembre 2023

Índex

1. Resum

1.1 Objectius

L'objectiu d'aquest treball és esbrinar mitjançant l'anàlisi estadístic, si existeix una diferència en la velocitat de resolució de laberints entre els algorismes BFS i DFS. Concretament, els posarem a prova amb laberints de dimensions 100 x 100 i 1000 x 1000 cel·les. Si trobéssim una diferència en el temps d'execució significativa, quantificaríem aquest resultat per poder obtenir una clara mesura de la millora en un interval de confiança raonable.

1.2 Mètodes

Hem utilitzat un codi generador de laberints aleatoris implementat en el llenguatge de programació C++, i dos algorismes de cerca, concretament el Breadth-First Search (BFS) i Depth-First Search (DFS).

1.3 Resultats

Després de dur a terme l'anàlisi de les dades obtingudes en la fase de recollida, vam observar que cap dels dos algorismes presentava una superioritat en temps de computació o en nombre de nodes visitats. Simplement, era causa de l'atzar i depenia d'on quedés la sortida respecta de l'entrada.

2. Introducció

Tots coneixem els laberints, però no tots sabem com trobar la solució per sortir d'ells. Per aquest motiu, hem volgut aprofitar els coneixements proporcionats per l'assignatura d'Estructures de Dades i Algorismes (EDA), en la qual hem estudiat i treballat amb dos algorismes de cerca, Breadth-First Search (BFS) i Depth-First Search (DFS).

En aquest treball, hem utilitzat aquests dos algorismes per trobar la solució a diferents laberints generats de manera aleatòria. Posteriorment, hem analitzat el seu comportament per veure si existeix alguna diferència en la velocitat de la resolució dels laberints o en el nombre de nodes visitats fins a trobar la solució, donant a lloc a la conclusió de que un dels dos és més eficient que l'altre.

3. Mètodes

Per la recollida de dades hem de fer servir el següent mètode. En primer lloc, hem generat laberints aleatòriament de dues dimensions diferents (100x100 i 1000x1000) per obtenir dues mostres tant amb laberints petits com laberints grans. Assegurant que tot laberint creat té una solució, també hem determinat que la casella de sortida de la qual partim sigui una "S" i la sortida o tresor que estem buscant una "t".

També hem de tenir en compte que la densitat del laberint és fixa, això vol dir que la quantitat de nodes o parets també ho és. Això pot causar una variabilitat en la quantitat de nodes visitats fent que aquesta mai arribi a ser el màxim possible en cada cas depenent del laberint que estem visitant. Això s'ha fet per facilitar la generació dels laberints, buscant una densitat mitjana que no afavoreixi a cap algorisme.

Després de generar els laberints, hem llançat una moneda per saber l'ordre en el qual executarem els algorismes (DFS i BFS), per augmentar l'aleatorietat. Seguidament, després de l'execució de cada algorisme hem fet servir la comanda "time" que ens proporciona la terminal per poder veure el temps d'execució de cada un. D'aquesta manera, per cada execució tenim tant el temps que ha necessitat per trobar la sortida/tresor, com el nombre de nodes que ha visitat.
Seguint aquesta metodologia hem fet un total de 200 mostres, 100 als laberints 100x100 i 100 als laberints 1000x1000.

Al final de cada execució guardem les dades en un Excel que trobareu adjunt al document. Ordenant les dades per categories, depenent de quin laberint han recorregut, quin algorisme han utilitzat i si mirem el temps o els nodes que ha visitat.

Per fer l'estudi hem fet servir un ordinador amb les següents característiques:

- CPU: AMD Ryzen 7 5800X3D 8-Core processor 3,4GHz
- RAM: 16BG 3200MHz
- SO: Ubuntu 22.04.3

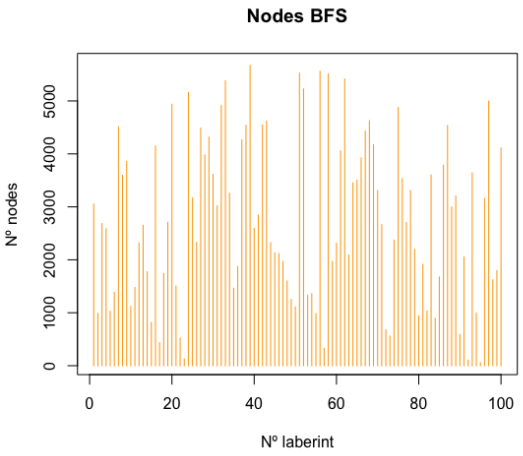
4. Resultats

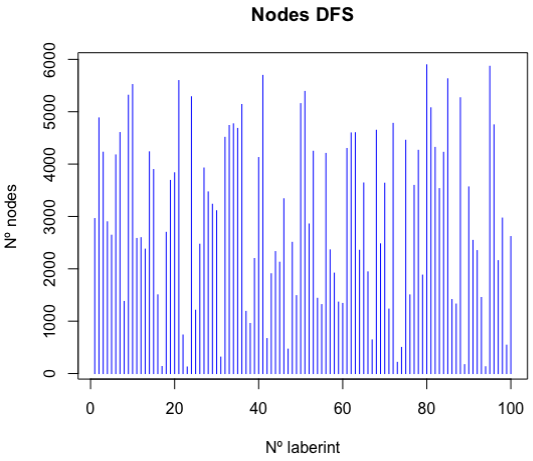
Donant un cop d'ull ràpid a les dades obtingudes podem observar el següent:

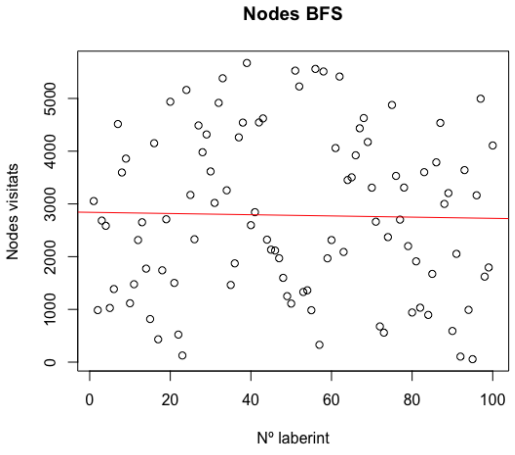
4.1 Laberints 100x100

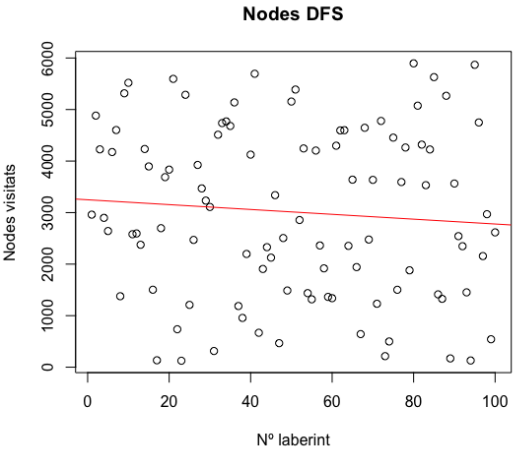
Tenint en compte que els nodes màxims que es poden visitar són 10.000, podem observar que els nodes visitats per l'algorisme BFS fluctuen entre [56, 5672], amb una mitjana mostral de 2782,06 nodes visitats per laberint, variància mostral de 2358161 i desviació típica mostral de 1535,63.

En canvi, amb l'algorisme DFS els nodes visitats es troben entre [125, 5896], amb una mitjana mostral de 3010,44, variància mostral de 2700720 i desviació típica mostral de 1643,39.

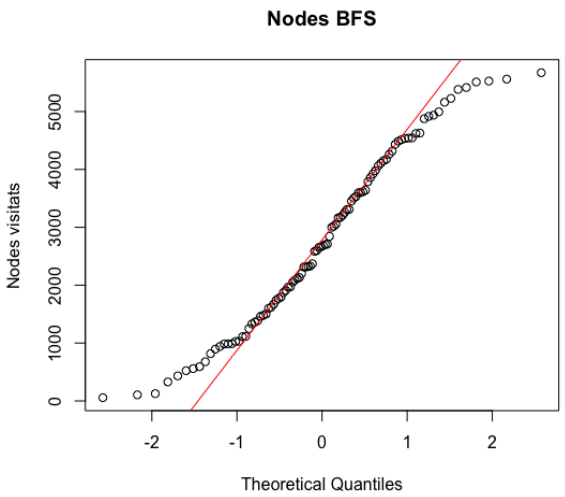


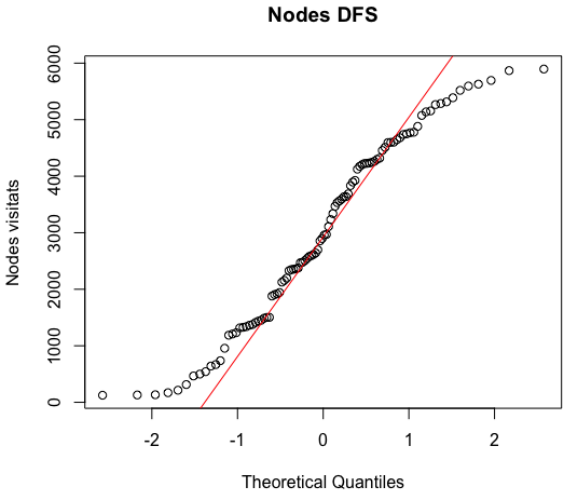






Observant els següents gràfics també podem veure com els nodes visitats pels dos algorismes es comporten similar a una normal, encara que difereix una mica als extrems.





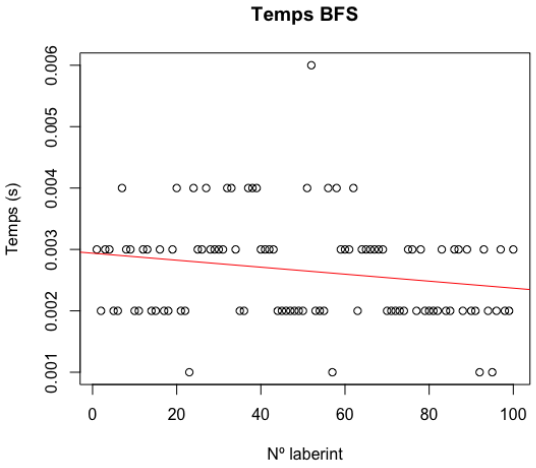
Amb les dades fent la diferència entre nodes visitats (DFS-BFS) podem estimar si existeix una millora en la qual destaqui algun d'aquests algorismes. Per fer aquesta estimació farem l'interval de confiança del 95% sobre la mitjana de la diferència que hem establert com la resta dels nodes DFS menys la resta dels nodes BFS.

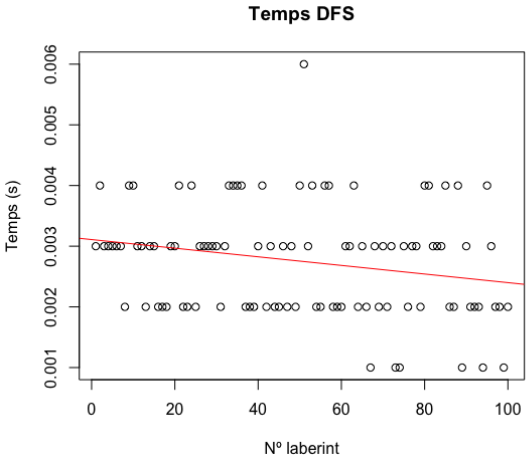
Fent servir el programari R-studio, podem fer una estimació d'aquesta mitjana diferència poblacional fent servir la comanda t.test i usant el paràmetre paired = TRUE perquè són mostres aparellades, ja que cada laberint ha passat pels 2 algorismes. Observem els següents resultats:

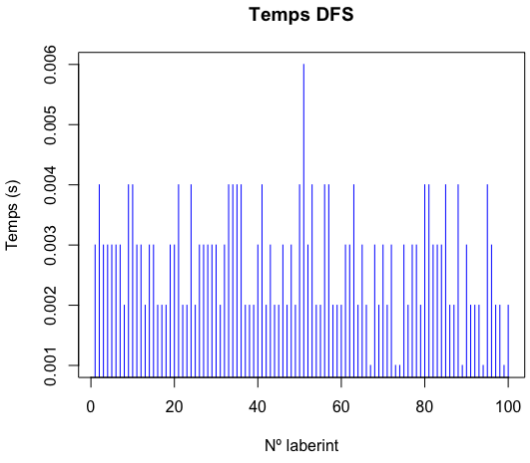
```
> t.test(nodes_DFS, nodes_BFS, paired = TRUE)

        Paired t-test

data:  nodes_DFS and nodes_BFS
t = 1.0338, df = 99, p-value =
0.3037
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -209.9453  666.7053
sample estimates:
mean difference
         228.38
```

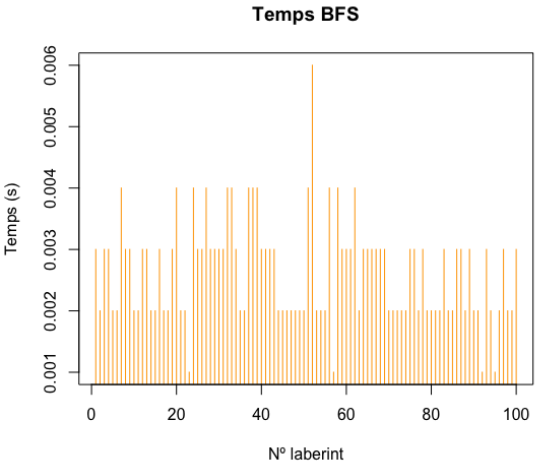
Podem observar que l'IC és [-209,9453; 666,7053]. Veiem que a l'interval de confiança hi apareix el 0 i també que el valor del p-value és major a 0,05. Per aquests motius, podem afirmar que no existeix cap diferència notable entre els dos algorismes en els laberints de dimensions 100x100. Existint la possibilitat que el 0 estigui a la mitjana diferència poblacional amb un 95% de seguretat.

En funció al temps utilitzat per cada algorisme, hem de tenir en compte que en el cas dels laberints 100x100 no podem treure moltes conclusions degut a que els valors són molt propers al 0 màquina tant per un algorisme com per l'altre. Per aquest motiu és millor avaluar aquest paràmetre als laberints 1000x1000.







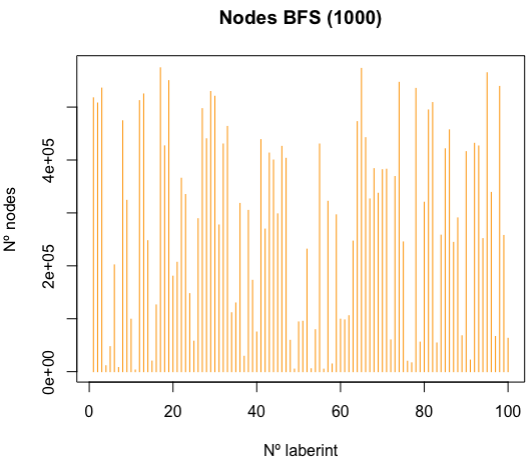


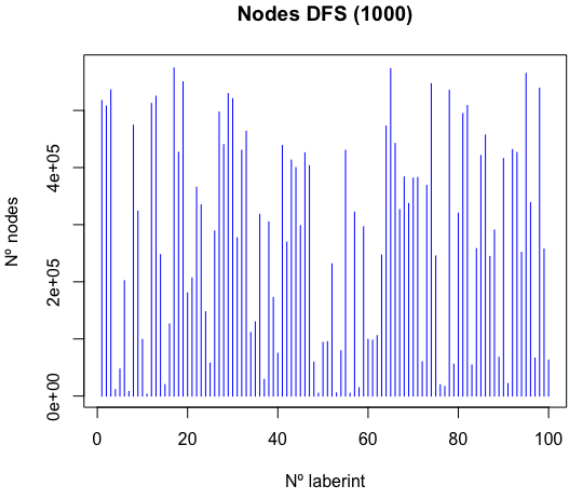
4.2 Laberints 1000x1000

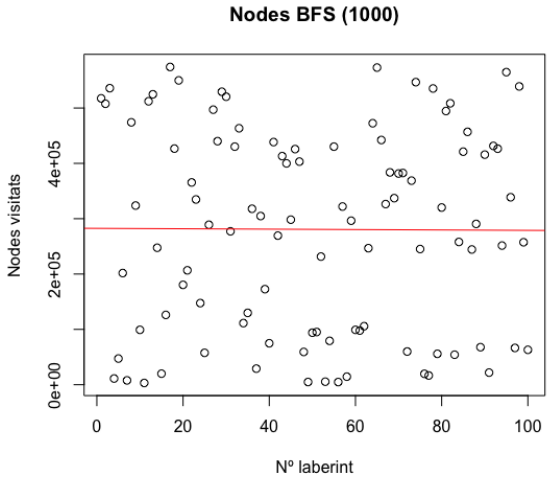
Quan passem a utilitzar laberints més grans podem avaluar els dos camps, tant els nodes visitats com el temps que necessita cada algorisme per trobar la sortida/tresor, tenint en compte que ara la quantitat màxima de nodes que es pot visitar és d'1.000.000.

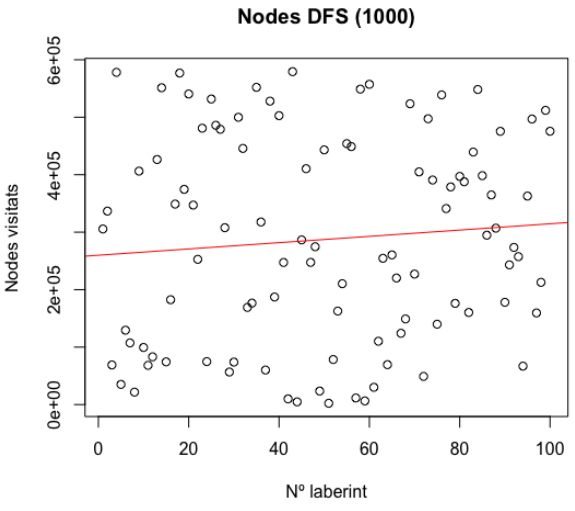
En quant als nodes, pel cas de l'algorisme BFS trobem que el número de nodes visitats varia entre els valors [3149, 574226]. A partir d'aquí podem treure la mitjana mostral dels nodes visitats pel BFS que correspon a un total de 280663 nodes, amb una variància mostral 33262632661 i una desviació típica mostral 182380,5.

En canvi, els nodes visitats per l'algorisme DFS fluctuen entre els valors [2296, 579319]. Partint de les dades, tenim una mitjana mostral 287507,5, una variància mostral de 32045286115 i una desviació típica mostral 179012.

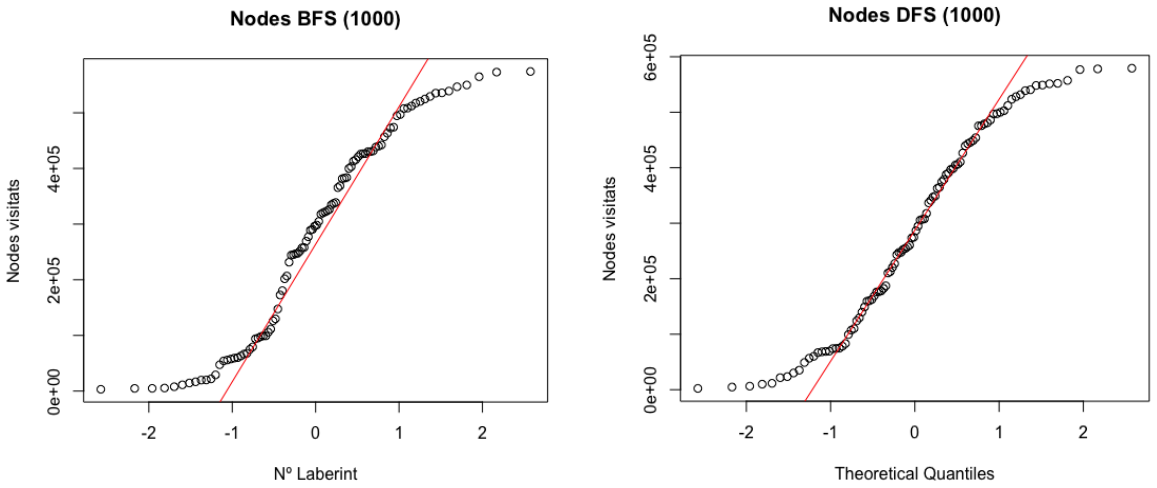








En aquests laberints podem tornar a afirmar que les dades dels nodes segueixen una normal que difereix una mica un altra vegada als extrems



Ara bé per avaluar si algun d'aquests algorismes és millor ara que estem treballant amb laberints més grans hem de tornar a fer l'interval de confiança de la diferència entre els nodes visitats dels dos algorismes, la tornarem a establir com nodes DFS - nodes BFS. Tornarem a fer servir la comanda t.test amb el paràmetre paired = TRUE, ja que encara són mostres aparellades perquè tots dos passen pel mateix laberint.

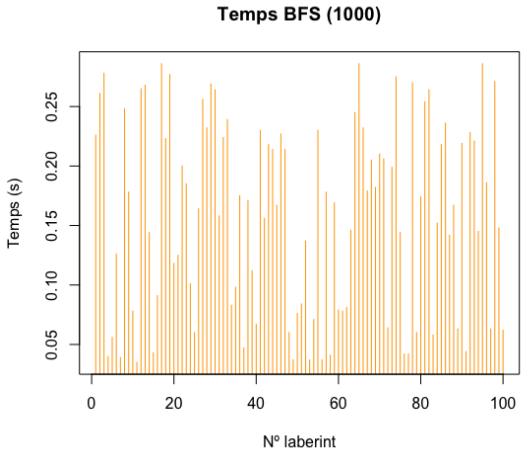
Observem els següents resultats:

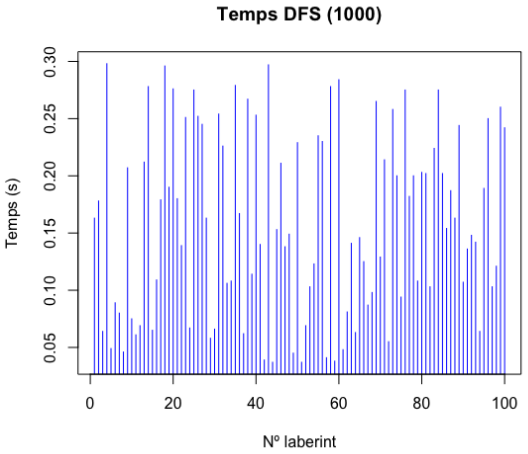
```
> t.test(nodes_D1000, nodes_B1000, paired = TRUE)

        Paired t-test

data:  nodes_D1000 and nodes_B1000
t = 0.27275, df = 99, p-value = 0.7856
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -42948.60  56637.76
sample estimates:
mean difference
        6844.58
```

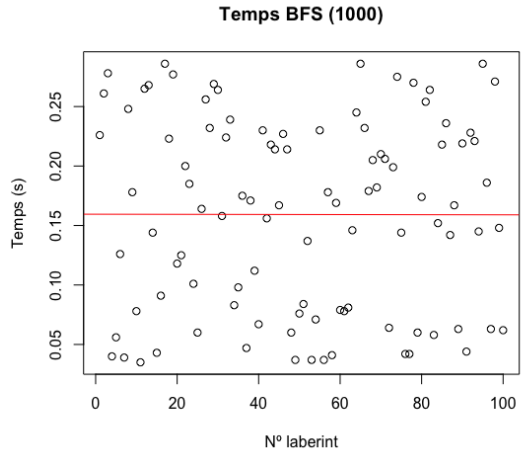
Un cop més tornem a observar que l'interval de confiança del 95% conte el 0 i que el paràmetre t.test torna a ser major que 0,05. Tenint en compte aquests dos

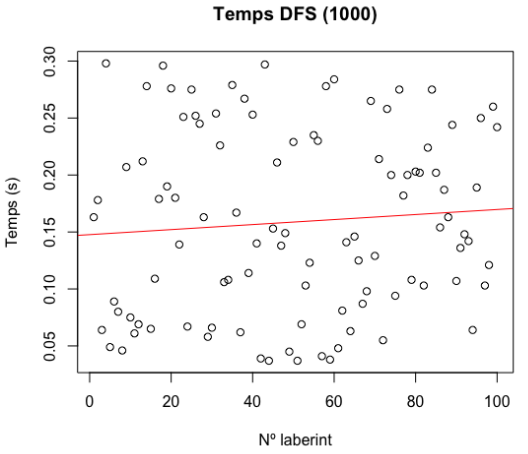
paràmetres podem tornar a afirmar que un cop més els dos algorismes són quasi similars amb la quantitat de nodes que visiten, ho podem afirmar amb una seguretat del 95%.

En qüestió del temps, ara sí que podem avaluar el temps dels dos algorismes per què tenim temps molt més variable, però que encara són propers al 0.
Analitzant les dades podem veure que els temps que fa servir el BFS per trobar la sortida/tresor ronda els valors [0,035; 0,286], a partir d'aquestes dades podem calcular la mitjana mostral 0,15919, variància mostral 0,006484681 i la desviació típica mostral 0,08052752.
Si parlem del temps que necessita l'algorisme DFS per trobar la sortida/tresor els valors oscil·len entre [0,037; 0,298], tenint en compte les dades podem considerar una mitjana mostral de 0,1588, variància mostral 0,006343152 i desviació típica mostral 0,0796439.

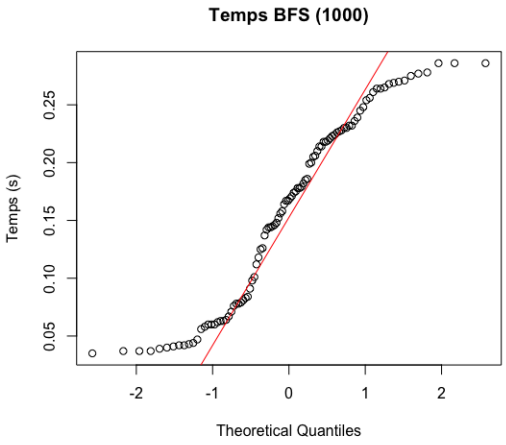


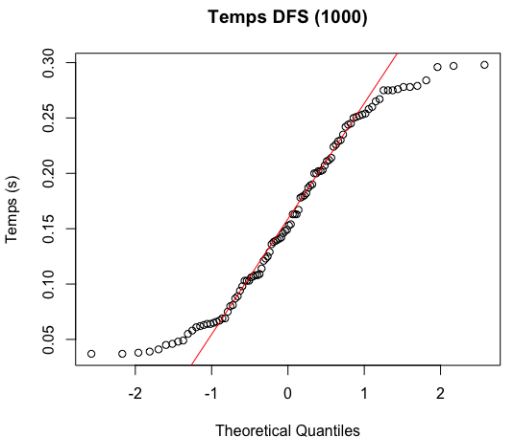






Un cop més podem observar una normalitat relativa a les dades, diferint als extrems.





Després d'exposar les dades ara podem fer la diferència entre els temps, per avaluar quin algorisme és millor en qüestió de temps. També podem estimar la mitjana de la diferència poblacional amb l'interval de confiança amb un 95% de confiança. Un cop més farem servir la comanda t.tes i el paràmetre paired = TRUE, que un cop més son aparellades perquè tots dos algorismes passen el mateix laberint un cop cadascun.

```
> t.test(temps_D1000, temps_B1000, paired=TRUE)

        Paired t-test

data:  temps_D1000 and temps_B1000
t = -0.034739, df = 99, p-value =
0.9724
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -0.02266601  0.02188601
sample estimates:
mean difference
       -0.00039
```

Contemplant la sortida de la comanda, ens adonem que un cop més a l'interval de confiança ens apareix el 0 i el p-value torna a ser un cop més major que 0,05. Per tant, podem considerar que cap dels dos algorismes sobresurt sobre l'altre fent així que són equivalents tant en temps com en nodes visitats.

5. Discussió

5.1 Conclusions

Una vegada realitzat l'estudi estadístic hem observat que cap dels dos algorismes és millor que l'altre. Tot i que presenten algunes diferències en temps i nodes aquestes són molt petites per considerar-se significatives. El p-value és molt gran en les quatre comparacions.

1. Nodes 100 x 100 : p-value = 0,3037
2. Nodes 1000 x 1000: p-value = 0,7856
3. Temps 100 x 100: p-value = 0,4138
4. Temps 1000 x 1000: p-value = 0,9724

Clarament aquests resultats ens indiquen que no existeix cap algorisme millor i les diferències son causa de l'atzar. Això confirma la nostra hipòtesi que no trobaríem cap superior a l'altre a causa de la naturalesa dels algorismes. Segueixen criteris diferents de cerca, però no estan basats en cap mètode matemàtic, simplement són diferents formes d'ordenar la selecció de nodes. Per aquest motiu, un serà millor que l'altre en funció de la situació.

5.2 Limitacions

Una de les limitacions que vam trobar va ser la capacitat de generar laberints de densitat variable. El problema que vam trobar era que amb densitats molt grans de murs no aconseguíem generar laberints amb solució.

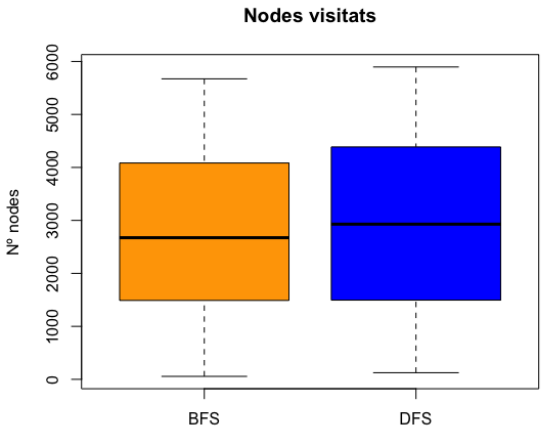
El generador el que fa és crear una matriu de mida n x n inicialitzada amb tot murs i després posa un nombre predefinit de punts. Una vegada fet això comprova si existeix un camí des de la sortida fins a l'entrada (també generats de forma aleatòria), si no el troba torna a començar. Amb densitats molt grans de murs, és a dir, pocs punts, trigava molt a generar un laberint amb solució. La solució per aquest problema seria programar un millor codi de generació, no obstant, no tenim ni els coneixements necessaris ni el temps per fer-ho.

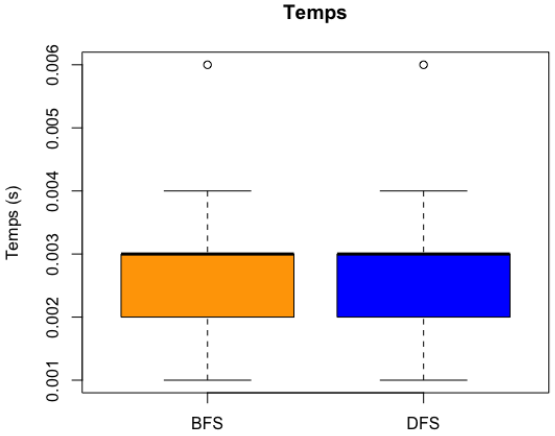
Una altra limitació, molt relacionada amb l'anterior, és la mida dels laberints, amb un ordinador més potent podríem haver generat laberints molt més grans per obtenir resultats més rellevants. Per exemple, el temps dels 100 x 100 ens donen molt poca informació, són molt propers als 0 màquina i no tenim cap eina per millorar la precisió.

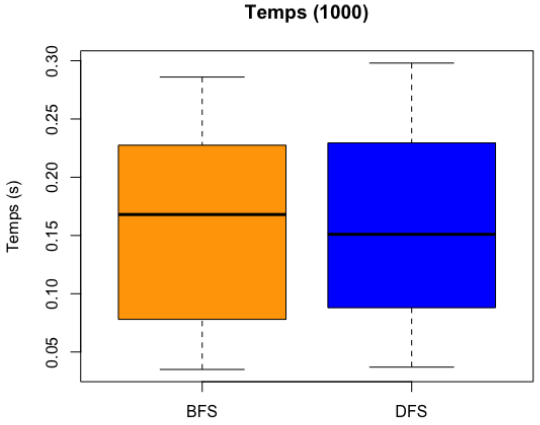
5.3 Estudis futurs

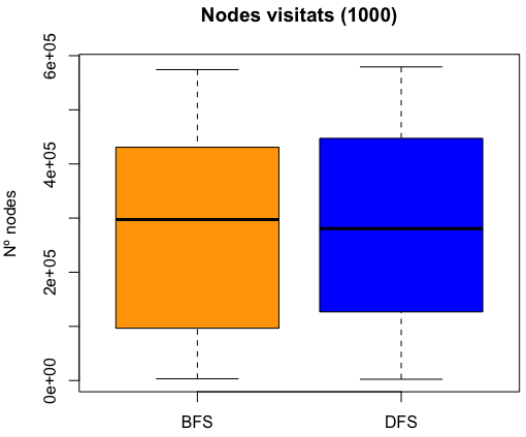
Si tornéssim a fer aquest treball realitzaríem alguns canvis per millorar la recollida de dades. Per una banda, programaríem un millor codi de generació de laberints per poder variar la seva densitat i augmentaríem la mida dels laberints per obtenir resultats més exactes. D'altra banda, compararíem els dos algorismes amb el A-Star, un altre algorisme de pathfinding, per intentar així trobar un algorisme superior. Tanmateix, creiem que els resultats del BFS i el DFS serien molt semblants i indicarien que son igual d'eficients. En conclusió, l'objectiu d'un nou treball no seria dedicat a millorar l'actual sinó a expandir la quantitat d'algorismes comparats per tal de trobar un superior.

6. Annex


Nodes visitats
Nº nodes
0
1000
2000
3000
4000
5000
6000
BFS
DFS


Temps
Temps (s)
0.001
0.002
0.003
0.004
0.005
0.006
BFS
DFS


Temps (1000)
Temps (s)
0.05
0.10
0.15
0.20
0.25
0.30
BFS
DFS


Nodes visitats (1000)
Nº nodes
0e+00
2e+05
4e+05
6e+05
BFS
DFS