Relazione Implementazione di MCMinCut , Ferrari Luca s4784573

Nel codice py , applico 10^5 volte il MCminCut

---

for $i = n$ to 2

1. campiona un arco G con probabilità uniforme e identifica i suoi vertici, u e v, in un nuovo vertice uv (vedi Figura 3 con $u = a$ e $v = b$)
2. rimuovi tutti gli archi che univano i vertici u e v, incluso quello campionato, e mantieni tutti gli archi che incidono sul nuovo vertice uv
3. $i \leftarrow i - 1$

C è costituito dagli archi che uniscono gli ultimi due vertici rimasti di G

Di un multigrafo non orientato avente 9 vertici e 21 archi.

La frequenza empirica attesa dal seguente grafo è di p circa uguale a 2 /n^2 dove n è uguale al numero di vertici ovvero 9, quindi il valore atteso è di circa 0.0246.

Essendo un algoritmo randomizzato di tipo Monte Carlo non è sicuro che ad ogni run ci restituisca effettivamente il taglio minimo reale ma siamo sicuri che eseguendo un numero elevato di run(10^5volte) il valore che si presenterà più spesso sarà il taglio minimo effettivo e cosi è stato dai test eseguiti.

I risultati ottenuti:

```
Il taglio minimo è  4
la frequenza empirica  0.22868
```

Il taglio minimo naturale del grafo di Fritsch è 4, quindi il risultato ottenuto è conforme alla teoria, per quanto riguarda la frequenza empirica pur essendo di un ordine superiore è in linea alla stima soddisfando quindi il risultato