

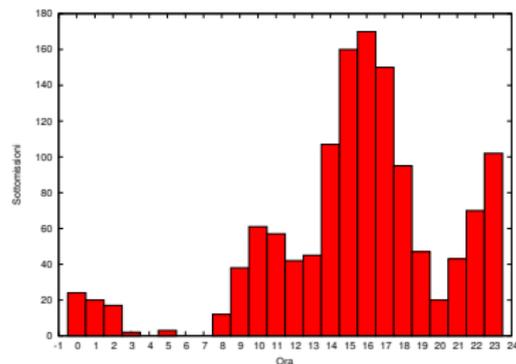
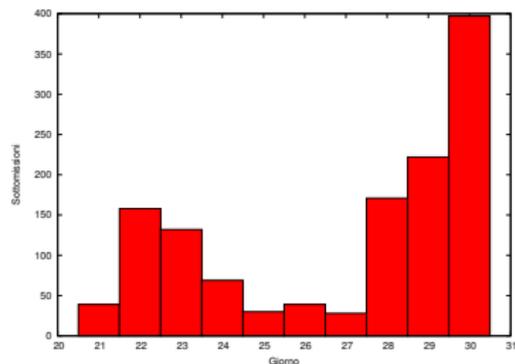
Soluzione Progetto 3 ASD a.a. 2018/2019



Alpinocalypse Now

Alessio Guerrieri e Lorenzo Ghiro
30 maggio 2018

Numero sottoposizioni: 1285



- ▶ 82 gruppi iscritti;
- ▶ 187 studenti;
- ▶ 4 ore di ricevimento (compresi i laboratori);
- ▶ 6 mail ricevute;

Definizione del problema

Undirected Feedback Vertex Set (Undirected FVS)

Dato un grafo non direzionato, individuare l'insieme minimale di nodi la cui rimozione rende il grafo aciclico.

Definizione del problema

Undirected Feedback Vertex Set (Undirected FVS)

Dato un grafo non direzionato, individuare l'insieme minimale di nodi la cui rimozione rende il grafo aciclico.

- ▶ Il problema è noto per essere NP-COMplete;

Definizione del problema

Undirected Feedback Vertex Set (Undirected FVS)

Dato un grafo non direzionato, individuare l'insieme minimale di nodi la cui rimozione rende il grafo aciclico.

- ▶ Il problema è noto per essere NP-COMplete;
- ▶ Pertanto non è possibile individuare la soluzione ottima in tempo polinomiale; (se $P \neq NP$)

Definizione del problema

Undirected Feedback Vertex Set (Undirected FVS)

Dato un grafo non direzionato, individuare l'insieme minimale di nodi la cui rimozione rende il grafo aciclico.

- ▶ Il problema è noto per essere NP-COMPLETE;
- ▶ Pertanto non è possibile individuare la soluzione ottima in tempo polinomiale; (se $P \neq NP$)
- ▶ Per affrontare il problema esistono diversi approcci euristici. Alcune tecniche proposte sono state utilizzate, oltre che in letteratura, anche dai più bravi tra voi.

Soluzione banale

Osservazione

In un grafo non direzionato un ciclo, per esistere, deve coinvolgere almeno 3 nodi.

Soluzione banale

Osservazione

In un grafo non direzionato un ciclo, per esistere, deve coinvolgere almeno 3 nodi.

- ▶ \Rightarrow Una soluzione banale ma valida è pertanto rimuovere dal grafo $N - 2$ vertici. Nel grafo rimangono così 2 soli nodi e il grafo risulterà necessariamente aciclico.

Soluzione banale

Osservazione

In un grafo non direzionato un ciclo, per esistere, deve coinvolgere almeno 3 nodi.

- ▶ \Rightarrow Una soluzione banale ma valida è pertanto rimuovere dal grafo $N - 2$ vertici. Nel grafo rimangono così 2 soli nodi e il grafo risulterà necessariamente aciclico.
- ▶ Questa soluzione è molto lontana dall'individuare un FVS minimo.

Approcci euristici

Si può fare di meglio! Ad esempio si può provare con:

Approcci euristici

Si può fare di meglio! Ad esempio si può provare con:

1. **Riduzioni**: ridurre il grafo in input rimuovendo nodi/archi non significativi per il problema;

Approcci euristici

Si può fare di meglio! Ad esempio si può provare con:

1. **Riduzioni**: ridurre il grafo in input rimuovendo nodi/archi non significativi per il problema;
2. **Greedy**: Scegliere con un criterio più intelligente i nodi da rimuovere, rimuovendo per primi i nodi che si suppone facciano parte di molti cicli;

Approcci euristici

Si può fare di meglio! Ad esempio si può provare con:

1. **Riduzioni**: ridurre il grafo in input rimuovendo nodi/archi non significativi per il problema;
2. **Greedy**: Scegliere con un criterio più intelligente i nodi da rimuovere, rimuovendo per primi i nodi che si suppone facciano parte di molti cicli;
3. **Metaeuristiche**: La ricerca della soluzione ottima all'interno dello spazio delle soluzioni...si può fare in tantissimi modi! Proseguendo la lettura troverete alcune tecniche metaeuristiche tra cui quella del Simulated annealing.

Riduzioni valide

Alcune possibili riduzioni valide e che si possono eseguire in tempo polinomiale:

- ▶ Rimuovere le foglie, cioè nodi con grado 1: sicuramente non fanno parte di alcun ciclo e sono irrilevanti per il problema;
- ▶ Dato un nodo n con grado 2 e vicini u, v , rimuovere il nodo n e inserire un arco tra u, v .

Fonte: *Dehne, Frank, et al. "An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem." International Computing and Combinatorics Conference, 2005.*

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

- ▶ Grado

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

- ▶ Grado
- ▶ PageRank

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

- ▶ Grado
- ▶ PageRank
- ▶ Centralità (ad esempio *Betweenness Centrality*[Brandes 2011])

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

- ▶ Grado
- ▶ PageRank
- ▶ Centralità (ad esempio *Betweenness Centrality*[Brandes 2011])
- ▶ ...

Approcci greedy

Intuizione

È una buona idea rimuovere per primi i nodi *più importanti*, quelli presumibilmente coinvolti nel maggior numero di cicli.

Per fare questo bisogna ordinare i nodi secondo un criterio di importanza! Si potrebbero rimuovere i nodi in ordine per:

- ▶ Grado
- ▶ PageRank
- ▶ Centralità (ad esempio *Betweenness Centrality*[Brandes 2011])
- ▶ ...

Rimozione probabilistica

Se l'ordine di rimozione è totalmente deterministico proverete UNA SOLA sequenza di rimozione dei nodi. Rimuovete i nodi con probabilità proporzionale al criterio scelto! Ripetendo il processo probabilistico più volte testerete più sequenze di rimozioni, scoprendo magari una sequenza migliore di quella iniziale.

Metaeuristiche

UFVS è un problema di ottimizzazione, in particolare di minimizzazione. Proponiamo 3 tecniche di ottimizzazione: la prima è stata utilizzata da alcuni di voi, la seconda è stata utilizzata per implementare la soluzione ufficiale e la terza, non testata, potrebbe rivelarsi ancora più efficiente.

Metaeuristiche

UFVS è un problema di ottimizzazione, in particolare di minimizzazione. Proponiamo 3 tecniche di ottimizzazione: la prima è stata utilizzata da alcuni di voi, la seconda è stata utilizzata per implementare la soluzione ufficiale e la terza, non testata, potrebbe rivelarsi ancora più efficiente.

1. Algoritmi genetici

Metaeuristiche

UFVS è un problema di ottimizzazione, in particolare di minimizzazione. Proponiamo 3 tecniche di ottimizzazione: la prima è stata utilizzata da alcuni di voi, la seconda è stata utilizzata per implementare la soluzione ufficiale e la terza, non testata, potrebbe rivelarsi ancora più efficiente.

1. Algoritmi genetici
2. Simulated Annealing

Metaeuristiche

UFVS è un problema di ottimizzazione, in particolare di minimizzazione. Proponiamo 3 tecniche di ottimizzazione: la prima è stata utilizzata da alcuni di voi, la seconda è stata utilizzata per implementare la soluzione ufficiale e la terza, non testata, potrebbe rivelarsi ancora più efficiente.

1. Algoritmi genetici
2. Simulated Annealing
3. Tabu Search

Metaeuristiche

UFVS è un problema di ottimizzazione, in particolare di minimizzazione. Proponiamo 3 tecniche di ottimizzazione: la prima è stata utilizzata da alcuni di voi, la seconda è stata utilizzata per implementare la soluzione ufficiale e la terza, non testata, potrebbe rivelarsi ancora più efficiente.

1. Algoritmi genetici
2. Simulated Annealing
3. Tabu Search

La soluzione proposta è ispirata da questo articolo:

Fonte: *S.M. Qin et al. "Solving the undirected feedback vertex set problem by local search." The European Physical Journal, 2014.*

Algoritmi genetici: Considerano le soluzioni come cromosomi e simulano un processo evolutivo, ad esempio: i) incrociando soluzioni (*crossover*), ii) generando *mutazioni*. Una opportuna funzione di *selezione naturale* guida l'evoluzione dei cromosomi verso una soluzione ottima.

Simulated Annealing: Si ispira alla metallurgia dove *annealing* indica il processo di eliminazione di difetti tramite riscaldamento seguito da lento raffreddamento. Per cercare i punti di minimo di una funzione si definisce allora una funzione *energia* f_e (anche detta funzione-obiettivo) sullo spazio delle soluzioni. Quindi si parte da una soluzione con alta energia e si utilizza f_e per guidare la ricerca del minimo verso soluzioni con energia via via minore (*raffreddamento*). Si individuano così i punti di minimo (locali) di f_e e si ripete il processo sperando di individuare il minimo globale.

Tabu Search: Nel Simulated Annealing è permesso solo far decrescere f_e : la ricerca del minimo globale può finire intrappolata in un minimo locale. Per ovviare a questo problema nelle Tabu Search sono ammesse ricerche peggioranti (con funzione obiettivo crescente), purché le ultime esplorazione effettuate siano considerate vietate, Tabu appunto, altrimenti si ricadrebbe velocemente nei minimi locali appena visitati.