# Lab. 3 - Il Commesso Viaggiatore

---

**Componenti gruppo:**

- Federico Caldart , matricola: 1211144
- Stefano Panozzo, matricola: 1211143
- Davide Zago, matricola: 1211260

---

## Domanda 1

| | Held Karp | | | Closest Insertion | | | MST-approssimato | | |
|---|---|---|---|---|---|---|---|---|---|
| Istanza | Soluzione | Tempo (s) | Errore | Soluzione | Tempo (ms) | Errore | Soluzione | Tempo (ms) | Errore |
| burma14.tsp | 3323 | 0.46 | 0.0% | 3336 | 0.1 | 0.39% | 3702 | 0.5 | 11.41% |
| ulysses22.tsp | 7105 | 180 | 1.31% | 7390 | 0.28 | 5.38% | 8363 | 1.16 | 19.25% |
| eli51.tsp | 1011 | 180 | 137.32% | 459 | 1.59 | 7.75% | 594 | 5.8 | 39.44% |
| kroD100.tsp | 146819 | 180 | 589.49% | 24999 | 7.55 | 17.39% | 28027 | 24.03 | 31.62% |
| gr229.tsp | 176680 | 180 | 31.26% | 154089 | 70.39 | 14.48% | 177614 | 131.69 | 31.95% |
| d493.tsp | 111743 | 180 | 219.25% | 41248 | 701 | 17.84% | 45190 | 715.67 | 29.11% |
| dsj1000.tsp | 551354888 | 180 | 2854.79% | 22722300 | 4.37e3 | 21.77% | 25034577 | 3142.31 | 34.16% |

## Domanda 2

Di seguito vengono commentati i risultati e il comportamento per ciascuno dei tre algoritmi implementati.

### Held-Karp

Come ci si aspetta, l'algoritmo esatto Held-Karp non si comporta bene se applicato a grafi con elevato numero di nodi.
Osserviamo come in breve tempo si riesca a trovare la soluzione ottima per il grafo avente 14 nodi e in un tempo accettabile (3 minuti) si riesca ad ottenere una stima della soluzione molto vicina all'ottimo per il grafo a 22 nodi; per i successivi grafi, tuttavia, non è possibile ottenere una buona stima della soluzione in tempi brevi: anche con solamente 51 nodi, la soluzione si discosta molto dall'ottimo.
In linea generale dunque possiamo affermare che l'errore dell'algoritmo esatto aumenta con l'aumentare del numero di nodi, ma è possibile incontrare delle

eccezioni; per esempio, per il grafo avente 229 nodi l'errore non è eccessivamente elevato (rimane comunque non accettabile) e ciò accade per una questione di "*fortuna*": come si può vedere dal file gr229.tsp, il grafo è composto da nodi aventi coordinate che aumentano o diminuiscono gradualmente (dunque poco distanti tra loro) e dunque, considerando che i nodi vengono visitati in un ordine non casuale, i primi circuiti trovati sono composti da archi con pesi *fortunatamente* poco elevati.

**Closest Insertion**

L'euristica costruttiva "Closest Insertion" offre un buon compromesso in termini di prestazioni e risultati ottenuti. Come si può notare, per grafi con uno scarso numero di nodi, si riesce a calcolare un percorso minimo in tempi molto brevi (<1ms) con un errore rispetto alla soluzione ottima soddisfacente. Riguardo a questo, si può notare come all'aumentare del numero di nodi l'errore sia via via crescente, fino ad arrivare al 21.77%, accettabile considerando la velocità d'esecuzione dell'algoritmo e comunque ottenendo una soluzione 2-approssimata.
I test sono stati eseguiti scegliendo il primo nodo in maniera casuale, così da apprezzare la diversa approssimazione nei vari casi. Di seguito sono elencati i nodi di partenza con cui si sono ottenuti i risultati riportati in tabella:

| Istanza | Nodo iniziale |
|---|---|
| burma14.tsp | 13 |
| ulysses22.tsp | 3 |
| eli51.tsp | 22 |
| kroD100.tsp | 95 |
| gr229.tsp | 202 |
| d493.tsp | 424 |
| dsj1000.tsp | 177 |

**MST-Approssimato**

Senza dubbio l'algoritmo MST-approssimato implementato creando un albero di copertura minimo con l'algoritmo di kruskal è il più efficiente con grafi di grandi dimensioni.
Questa implementazione infatti permette di trovare una soluzione approssimata in pochi secondi anche per grafi di grandi dimensioni come nel caso di dsj1000.
L'approssimazione ottenuta è abbastanza buona, dai risultati trovati si nota che l'errore relativo non dipende dalla dimensione del grafo e non supera in nessun caso il 40%, quindi la soluzione rimane sempre ben sotto al caso peggiore trovato teoricamente nel quale la soluzione approssimata è due volte superiore alla soluzione ottima.
E' possibile inoltre ottenere soluzioni migliori provando a utilizzare nodi diversi come radici dell'albero di copertura minimo, tuttavia non sembra esserci una regola per la selezione del nodo che minimizzi l'errore per ogni grafo.

**Considerazioni finali**

*Rispetto all'errore di approssimazione*, l'algoritmo che utilizza l'euristica *Closest Insertion* è sempre migliore degli altri due, indipendentemente dalla grandezza del grafo, ma richiede un tempo d'esecuzione maggiore per grafi di grandi dimensioni rispetto a *MST-approssimato*.

Possiamo dunque suggerire l'utilizzo di *Held-Karp* per grafi molto piccoli, di un algoritmo che utilizzi un'euristica costruttiva per grafi di media-grande dimensione e dell'algoritmo *MST-approssimato* per grafi di elevata dimensione.