

# Lab. 3 - Il Commesso Viaggiatore

Ballarin Simone, Gobbo Alessio, Rossi Daniel

May 2019

## Domanda 1

Istanza	Held-Karp			Euristica costruttiva			2-approssimato		
	soluzione	tempo (s)	$\epsilon$ (%)	soluzione	tempo (s)	$\epsilon$ (%)	soluzione	tempo (s)	$\epsilon$ (%)
burma14.tsp	3323.0	8.33E-01	0.00	3346.0	1.55E-04	0.69	4062.0	1.87E-04	22.24
ulysses22.tsp	7418.0	1.20E+02	5.77	7183.0	3.07E-04	2.42	8132.0	3.45E-04	15.96
eil51.tsp	1041.0	1.20E+02	144.37	468.0	1.61E-03	9.86	607.0	1.57E-03	42.49
kroD100.tsp	148184.0	1.20E+02	595.90	22931.0	5.72E-03	7.69	27340.0	5.52E-03	28.39
gr229.tsp	182455.0	1.20E+02	35.55	150495.0	2.76E-02	11.81	170205.0	2.28E-02	26.45
d493.tsp	112276.0	1.20E+02	220.77	38592.0	1.14E-01	10.26	45827.0	8.24E-02	30.93
dsj1000.tsp	552579267.0	1.20E+02	2861.35	20820213.0	6.31E-01	11.58	25729081.0	3.69E-01	37.89

## Domanda 2

Descriviamo il comportamento di ciascun algoritmo al variare delle istanze di seguito.

### Held-Karp

L'algoritmo esatto data la sua elevata complessità, trova la soluzione esatta entro il tempo limite di 120 secondi solamente nell'istanza più piccola (burma14). Sempre con il limite di 120 secondi vediamo come il risultato di ulysses22 possa essere considerato accettabile, mentre per le altre istanze del problema, in generale, l'errore è più significativo. Nonostante quanto appena detto, in alcuni casi, l'errore di istanze più grandi risulta inferiore a quello calcolato per istanze più piccole (kroD100.tsp, gr229.tsp). Questa peculiarità trova spiegazione, secondo noi, nella particolare configurazione dei pesi di queste istanze: l'ordine con cui iteriamo i circuiti possibili è deterministico. In questi due particolari problemi, casualmente una soluzione buona si trova nei primi circuiti analizzati, portando ad avere soluzioni difficilmente migliorabili aumentando di poco il tempo.

### Euristica costruttiva

L'euristica costruttiva scelta è stata la *Random Insertion*, questa euristica prevede la costruzione di un circuito attraverso l'aggiunta iterativa di un nodo. Questo viene scelto casualmente ed inserito nella posizione che minimizzi la deviazione necessaria.

I risultati ottenuti sono sorprendentemente buoni. Tranne per il caso burma14 è l'algoritmo che trova sempre la soluzione migliore nel tempo più breve. Adirittura nell'istanza più grande l'errore risulta pari a 11.58% calcolato in un tempo di 6.31E-01 secondi.

Al fine di ottenere una maggiore comprensione dei risultati elaborati in relazione alla natura casuale dell'algoritmo, abbiamo deciso di eseguire per ciascuna istanza del problema **10000** volte l'algoritmo. Abbiamo quindi estrapolato l'errore minimo e massimo per ciascuna istanza, riportando il tutto nella tabella a seguire:

Istanza	$\epsilon_{min}(\%)$	$\epsilon_{max}(\%)$
burma14.tsp	0.00	16.58
ulysses22.tsp	0.00	15.67
eil51.tsp	0.00	18.78
kroD100.tsp	1.16	21.55
gr229.tsp	3.93	23.25
d493.tsp	6.23	15.88
dsj1000.tsp	8.48	17.56

Dalla tabella notiamo come gli errori massimi siano simili, mentre gli errori minimi sono lineari all'aumentare del numero di nodi dell'istanza.

## 2-approssimato

L'algoritmo basato sull'albero di cammino minimo sembra non presentare nessuna particolare correlazione tra l'errore e la grandezza del problema.

## Considerazioni generali

In generale si nota che l'istanza kroD100.tsp in relazione alla sua grandezza limitata è sempre il problema che negli algoritmi approssimati dà i risultati peggiori. L'algoritmo in genere migliore è sempre l'euristica costruttiva in termini di errore percentuale. Per quanto riguarda l'efficienza notiamo come l'euristica costruttiva per istanze piccole sia migliore mentre per istanze più grandi risulti più efficiente l'algoritmo 2-approssimato.

## Domanda 3

In allegato alla consegna sono stati inseriti i codice sorgente Python.