

Progetto Metaeuristiche

Tecnologia Utilizzata

Per la realizzazione di questo progetto si e' utilizzato il linguaggio haskell. Si e' scelto di utilizzare questa tecnologia unicamente come esercizio rispetto ad altre tecnologie come java o csharp dove ho gia' esperienza. Di conseguenza mi scuso per eventuali errori o performance scadenti proprio perche' si tratta del mio primo programma con questa tecnologia.

Strategia Comune e Altre Considerazioni

All'esecuzione del programma viene chiesto all'utente quale algoritmo si vuole utilizzare, e' possibile anche richiedere la terminazione del programma, e in seguito su quale/i istanza/e si vuole far eseguire il programma. Appena inseriti questi dati il programma procede con il parsing dell'istanza scelta, se piu' di una si parte dalla prima, poi la seconda e cosi via. Nel caso nell'istanza ci sia un limite nella capacita' del veicolo i nodi vengono suddivisi in sottogruppi e l'algoritmo viene eseguito per ognuno di questi, dando cosi piu' risultati per quella istanza, altrimenti se la capacita' e' sufficiente per tutti i nodi, questa suddivisione non avviene.

Importante, Visto che si tratta di un caso di minimizzazione si e' deciso, una volta calcolata la fitness di esporre i valori invertiti: quello con distanza minore avra' possibilita' maggiore in montecarlo o in calcoli di probabilita' di venire scelto rispetto a quello con distanza piu' grande. Semplicemente vengono calcolati normalmente e poi si scambiano il maggiore con il minore e via cosi, i valori centrali non vengono toccati.

Per brevità si e' deciso di escludere la descrizione del funzionamento dei singoli algoritmi, nel caso la si voglia leggere (da 1 pagina questo documento diventa di 2 pagine), specialmente perche' puo' risultare difficile ragionare sul codice senza conoscere la tecnologia, si apra il file markdown relativo(description.md). Si veda il file excel per le considerazioni statistiche

Esecuzione

Per eseguire il programma viene messo a disposizione un file .exe ipotizzando che il prof posseda una macchina windows nella root del progetto chiamato appunto **VRP.exe**. Affinche' sia possibile leggere le istanze, il programma avra' bisogno di sapere esattamente il percorso assoluto dei file. Quindi per semplicita' si e' deciso di compilare il tutto impostando come cartella dei file **C:\temp**. Se questa cartella non e' presente sulla macchina e' necessario crearla e copiare le istanze presenti in **src/files** all'interno di essa.

I parametri degli algoritmi sono visualizzabili nel file **src/Parameters.hs**