

RELAZIONE TERZO ELABORATO

Algoritmi euristici e metaeuristici

Autore: Marco Carlo Cavalazzi

N° Matricola: 0000644460

Il problema scelto è quello dell'OpenShop che dati un numero di macchine e un numero di lavori, deve distribuire questi ultimi tra le macchine minimizzando il tempo totale, processandoli una sola volta. La scelta del tipo di problematica da affrontare è stata presa dal sito "<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>". Nella documentazione presente viene data una porzione di codice relativo alla generazione delle matrici delle macchine e dei costi, utile nello svolgimento dell'algoritmo

Il progetto è stato affrontato attraverso l'implementazione di due algoritmi metaeuristici, il genetico e il GRASP.

Algoritmo genetico

Viene richiesto all'utente di inserire un numero di iterazioni (preferibili valori alti, esempio con valori superiori o pari 1000 si hanno delle soluzioni migliori), la dimensione della popolazione, il numero di figli per generazione (che dovrà essere un valore pari) e la probabilità di mutazione (compresa tra 0 e 1).

La prova è stata effettuata con 10 macchine e 10 lavori, 2000 iterazioni, 700 popolazione, 400 figli e una probabilità di 0,5.

Complessità:

$O(\text{iterazioni} * (\text{probabilità_mutazione} + (\text{figli} * \text{dimensione_soluzione}) + O(\log n)) + (\text{popolazione} * \text{dimensione_soluzione} + (\text{iterazioni} - 1) * (\text{figli} * \text{dimensione_soluzione})))$.

Dove $\log n$ è la complessità dovuta alla scelta casuale dei genitori.

Algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*)

Viene richiesto all'utente di inserire un numero di iterazioni (preferibili valori alti, esempio con valori superiori o pari 1000 si hanno delle soluzioni migliori) e un valore di greediness (compresa tra 0 e 1).

La prova è stata effettuata con 10 macchine e 10 lavori, 2000 iterazioni, e un livello di greediness pari a 0,1.

Complessità: $O(\text{iterazioni} * (\text{dimensione_soluzione} + (n-1)^2 + ((n-1) * (n-2) + 1) * n))$

Dove $(n-1)^2$ corrisponde alla complessità per ricerca dei vicini e $((n-1) * (n-2) + 1) * n$ corrisponde alla complessità per il calcolo della soluzione.

Conclusioni

Con i dati raccolti è stato effettuato un confronto tra i 2 algoritmi attraverso il testo di rudità 3. Dai grafi ottenuti si nota che le distribuzioni non sono normali e attraverso Wilcoxon si ottiene che i due algoritmi non sono significativamente diversi poiché hanno un'efficacia equivalente.