

TESINA RICERCA OPERATIVA

Realizzare una implementazione dell' algoritmo euristico di Clarke & Wright nelle sue versioni parallela e sequenziale, per problemi di CVRP(simmetrici) con Linehaul e Backhaul della quale si fornisce il pseudocodice(per problemi di soli linehaul):

function Clarke&Wright(file benchMarkIstance) - Output: file ResultBenchMarkIstance

1. Crea un set R di n routes (0,i,0) per $i = 1, \dots, n$
2. Calcola i savings $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ per $i, j = 1, \dots, n, i \neq j$
3. Ordina i savings in modo non-crescente
4. Per ogni saving s_{ij} ordinato al passo 3

Se in R esiste una route r_1 contenente l'arco (i,0) AND in R esiste un'altra route r_2 contenente l'arco (0,j)

Allora calcola la Σ delle domande servite da r_1 , la Σ delle domande servite da r_2 , e calcola la loro somma S

Se $S \leq C$

genera una nuova route $r_3 = [r_1 - \{i,0\}] \cup [r_2 - \{0,j\}] \cup \{i,j\}$ e aggiorna R inserendo r_3 al posto di r_1 ed r_2

end if

end if

Vincoli aggiuntivi.

1)Per ogni rotta : Prima si fanno le consegne(Linehaul), e poi si fanno i ritiri(Backhaul).

2)Non possono esserci rotte con soli Backhaul.

3)Il numero dei mezzi (k) richiesti nell'istanza deve essere rispettato. Modificate l'algoritmo C-W per risolvere questo problema

Scrivere una mappa strutturale più dettagliata per risolvere il dato problema usando anche i backhaul. (Vi consiglio di farci vedere, prima la mappa e dopo avuto il consenso da parte nostra, mettervi ad implementare il codice)

Tale implementazione dovrà dare possibilità all' utente di selezionare il file da elaborare direttamente dal filesystem. Tale funzionalità dovrà essere verificata utilizzando la fornita cartella contenente varie istanze di benchmark con le relative migliori soluzioni. Al fine di confrontare il proprio lavoro con i risultati in letteratura, è richiesta una breve presentazione dei risultati ottenuti.

Al termine dell' elaborazione, l' algoritmo deve produrre un file .txt contenente l' elenco delle route calcolate, e per ogni route il relativo costo. Deve essere inoltre salvato sul file il valore della funzione obiettivo, ottenuto dalla sommatoria dei costi delle singole route, e inoltre è richiesto di riportare nel file un parametro che indichi il tempo di elaborazione.

Tale implementazione dovrà essere realizzata in C++, utilizzando QtCreator, e realizzando una classe principale chiamata parallelCW, da sviluppare e documentare.