



ENZO VELO

LUCAS GOMES COLOMBO

RAFAEL BRUNINI pereira

**RELATORIO DA MACROENTREGA 1 DO PROJETO PRÁTICO DA
DISCIPLINA ALGORITMOS EM GRAFOS**

Manipulação de arquivos binários

LAVRAS – MG

2022

Aqui será apresentada a primeira etapa do projeto prático da disciplina algoritmos em grafos, no qual foram feitos códigos para solução de problemas em serviços de logística.

O programa conta com 3 structs para o gerenciamento das instâncias, dos nodes e dos caminhos que serão usados.

A função ler instancia é utilizada para fazer a leitura de dados, para isso foi utilizado um switch case que detecta quebras de linha, espaços, etc.

A função lerGrafo também é utilizada para fazer a leitura de dados, mas armazenando-os em uma matriz de adjacência e um vetor associado a struct node.

A função randomNumbers gerará números aleatórios com base no tamanho da instancia (serão usados para a solução aleatória).

Para a geração das soluções iniciais dos algoritmos de busca local, o subprograma verificaRestrição pedirá ao usuário que digite a quantidade de nodes em que ele deseja testar as restrições, após isso é chamada a função randomNumber que gerará uma quantidade de números aleatoriamente e setando o início e o fim do trajeto como a posição 0.

Ainda na função verificaRestrição será feita a verificação da capacidade do caminhão, caso ele chegue na capacidade máxima será uma mensagem de aviso, e caso ele não utilizar o total de seu espaço será exibida uma mensagem de quanto ele usou da carga, e logo após isso será feito o esvaziamento do caminhão além de a contagem de tempo gasto no ponto e se o tempo máximo foi excedido.

É importante lembrar que a cada interação da busca local será adicionado um novo caminhão.

A função main é basicamente usada para fazer a chamada das funções.

Abaixo encontra-se um pseudocódigo referente a solução final do trabalho onde é apresentado a base da ideia em que iremos aplicar as heurísticas de forma a achar uma solução viável.

A função recebe como parâmetros a semente usada para inicializar o gerador de números aleatórios, o número de vértices do grafo e a matriz de adjacência

Função SampleGreedy Prim (semente, n , p)

$S \leftarrow \emptyset$;

Inicialize o conjunto V das possíveis localizações de facilidades;

$q \leftarrow \lceil \log_2(n/p) \rceil$;

Enquanto ($|S| \neq p$) **Faça**

$k \leftarrow 0$, $Q \leftarrow \emptyset$, $V_{aux} = V$;

Enquanto ($k \neq q$) **Faça**

Selecione f aleatoriamente de V_{aux} ;

Avalie o acréscimo $C(f)$;

$Q \leftarrow Q \cup \{f\}$;

$V_{aux} \leftarrow V_{aux} \setminus \{f\}$;

$k \leftarrow k + 1$;

Fim-Enquanto

$f \leftarrow \operatorname{argmin}\{C(f), f \in Q\}$;

$S \leftarrow S \cup \{f\}$;

$V \leftarrow V \setminus \{f\}$;

Fim-Enquanto

Inicie com uma subárvore $T_1(S)$ consistindo de um terminal $f \in S$ escolhido aleatoriamente;

$k \leftarrow 1$, $X_k \leftarrow X_k \cup \{f\}$;

Enquanto ($k < p$) **Faça**

Determine um terminal $f \in S \setminus X_k$ mais próximo de $T_k(S)$;

Construa uma árvore $T_{k+1}(S)$ adicionando o caminho

mínimo entre f e $T_k(S)$;

$k \leftarrow k + 1$, $X_k \leftarrow X_k \cup \{f\}$;

Fim-Enquanto

$T(S) \leftarrow T_p(S)$;

Retorne S ;

Fim