



Universidade Federal da Paraíba

Centro de Informática

Análise e Projeto de Algoritmos

Projeto Final

Relatório dos Resultados

André Iarley Soares da Cruz
Diogo Cantuaria Da Silva Hiebert
Luiz Paulo de Souza Fontes Junior

Grupo:
Mat. 20230145258
Mat. 20200001121
Mat. 20230146291

Professor:
Bruno Petrato Bruck

17 de setembro de 2025

Tabela dos resultados

		Heurística construtiva			VND		
	ótimo	valor da solução	tempo	gap	valor da solução	tempo	gap
n12_q20	15700	44100	0.05914	180.8%	28400	0.14005	80.8%
n12_q30	14600	44100	0.04657	202.0%	28400	0.13695	94.5%
n12_q30	16900	47600	00.0564	181.6%	42500	0.09911	151.4%
n14_q12	13500	17100	0.05516	26.6%	15500	0.10302	14.8%
n14_q30	12600	14900	0.01898	18.2%	13400	0.04114	6.3%
n17_q10	31443	69022	0.03485	119.5%	39179	0.09159	24.6%
n17_q20	29259	69022	0.02711	135.9%	39179	0.08770	33.9%
n20_q20	91619	116843	0.04063	27.5%	107739	0.31071	17.5%
n20_q30	76999	87670	0.03673	13.8%	80317	0.34176	4.3%
n26_q20	31100	67700	0.05822	117.6%	46200	0.27689	48.5%
n26_q30	30300	67700	0.05616	123.4%	45900	0.26756	51.4%
n40_q20	59493	175332	0.15802	194.7%	133379	0.43026	124.1%
n40_q30	57476	175332	0.12707	205.0%	133379	0.65912	132.0%
n54_q30	120277	550598	0.30998	357.7%	408853	3.6526	239.9%
n58_q30	65669	123550	0.29910	88.1%	94291	1.8555	43.5%
n74_q20	48829	105207	0.37599	115.4%	70373	4.16317	44.1%
n79_q30	39979	54718	0.42180	36.8%	53315	1.59262	33.3%
n81_q10	388680	663158	0.39565	70.6%	613489	5.74253	57.8%
n115_q20	157115	692103	0.86896	340.5%	408795	32.399	160.1%

Instruções de Compilação

Na raiz do projeto execute o comando no terminal:

```
make
```

Instruções de Execução

Para executar uma instância basta passar o caminho como argumento no terminal:

```
./build/prog <caminho_do_arquivo>
```

Para executar todas as instâncias e calcular o gap basta executar sem argumentos:

```
./build/prog
```

Organização do projeto

./input - Instâncias de teste.

./output - saída das instâncias de teste.

./include - Arquivos de cabeçalho.

./src - Implementações do código.

Principais Arquivos:

Algoritmo guloso:

LowerCost

A classe *LowerCost* implementa um algoritmo guloso do **vizinho mais próximo**, utilizando uma **matriz de adjacência** ordenada para selecionar a próxima estação com o menor custo que ainda não tenha sido visitada, garantindo que a capacidade do caminhão não seja ultrapassada e que a carga não se torne negativa.

RouteStep

A classe *RouteStep* representa um movimento na rota construída, armazenando as informações relevantes de um trecho entre duas estações, como o ID da estação de origem, o ID da estação de destino, a carga, o custo e o custo acumulado até aquele ponto.

VND:

./src/VND.cpp - A lógica de execução do VND utilizando os algoritmos de vizinhança descritos a seguir.

Algoritmos de vizinhança:

Swap - *./src/swap.cpp* - Implementado conforme demonstrado na vídeo aula, realiza as permutações em cada rota.

2-Opt - *./src/TwoOpt.cpp* - Também implementado conforme demonstrado na vídeo aula. Realiza as trocas em cada rota.

Swap2 - *./src/swap2.cpp* - Implementado conforme demonstrado na vídeo aula, com a modificação que permite não apenas trocar nós dentro de uma mesma rota, mas também realizar permutações entre diferentes rotas.