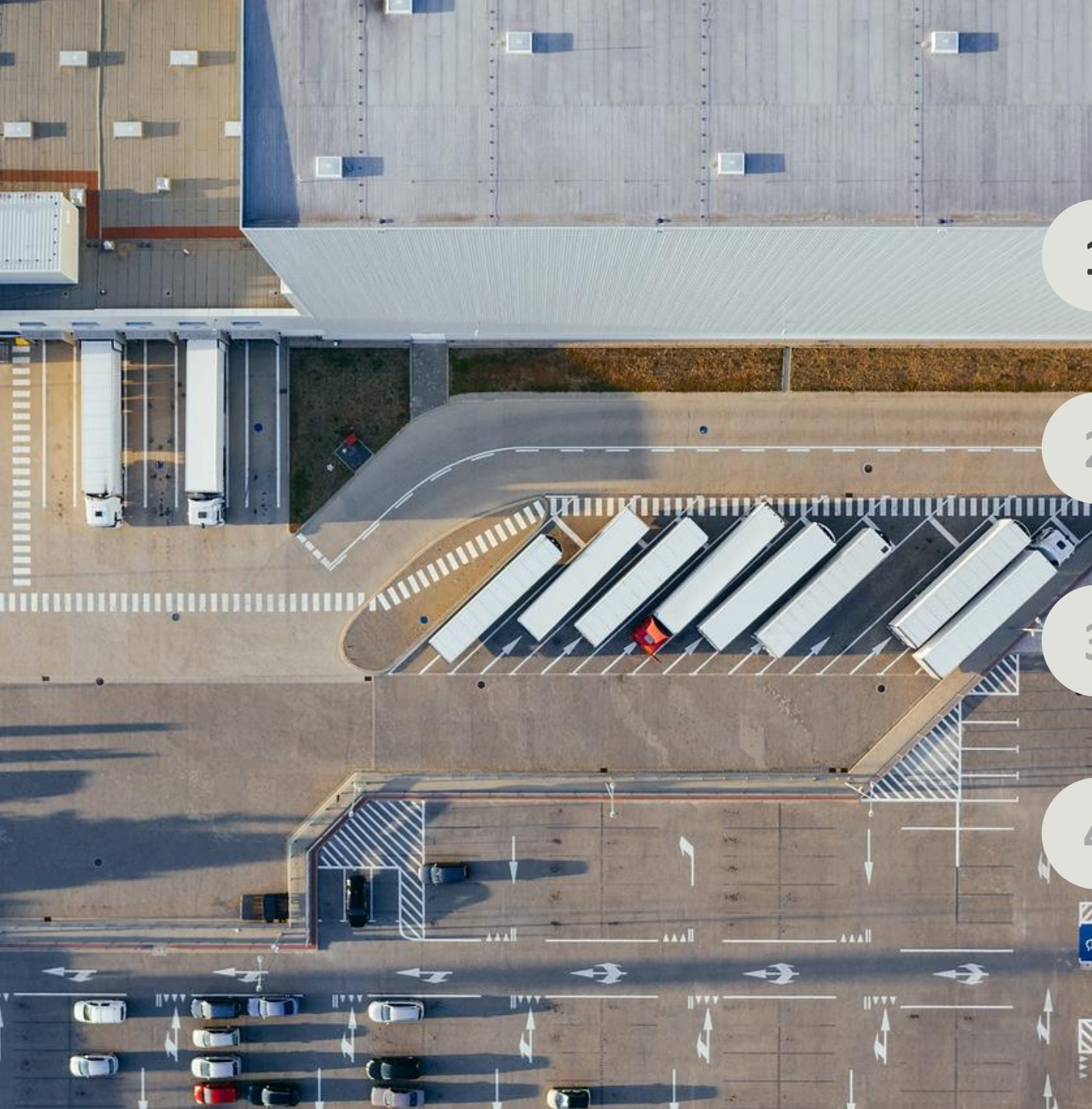


ATIVIDADE 01

PNV3512 - Planejamento e Operações de Sistemas Logísticos

Allan Gomes (11259396)

Guilherme Nalin (11262772)



1

Geração Exaustiva de Rotas & Permutações

2

Algoritmo de Clarke & Wright

3

Heurística Construtiva Probabilística

4

Tempos de Execução

Estratégias para redução do tempo de execução do algoritmo envolvem reduzir combinações e permutações avaliadas



Calcular máximo de clientes na rota:

Uma forma de reduzir a quantidade de combinações geradas é definir qual o máximo de clientes que uma dada capacidade comporta, Pode-se fazer isso com o seguinte método:

1. Ordenar lista de clientes considerados por demanda crescente
2. Iterar a lista, somando a demanda, até atingir a capacidade do veículo
3. Contabilizar quantos clientes foram necessários para atingir a demanda



Avaliar se a combinação respeita o limite de capacidade

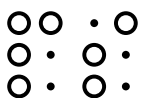
Para reduzir uso de memória, não se deve armazenar todas as combinações nem gerar as permutações para todas elas, O ideal é

- Avaliar se uma dada combinação de clientes respeita a capacidade
- Somente se a dada combinação respeitar, gerar as permutações desta combinação



Reduzir o número de permutações avaliadas

Dada a necessidade de definir a ordem de clientes com menor distância (Caixeiro Viajante), pode-se **reduzir pela metade** o número de permutações, Isso é possível pela condição simétrica das rotas

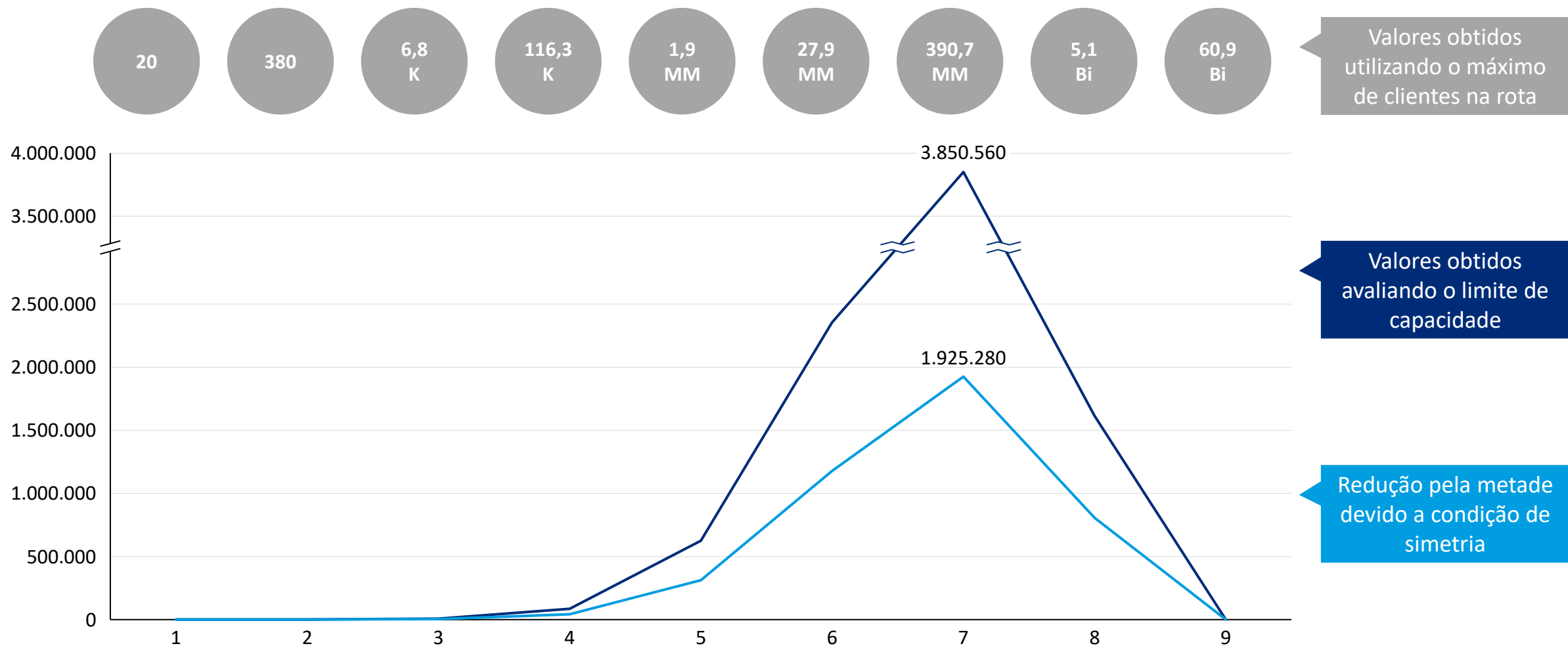


Gerar matriz de distâncias

Gerar uma matriz de distâncias $n \times n$ (onde n é número de clientes) para evitar o cálculo repetido dessas distâncias no momento de calcular a distância das rotas

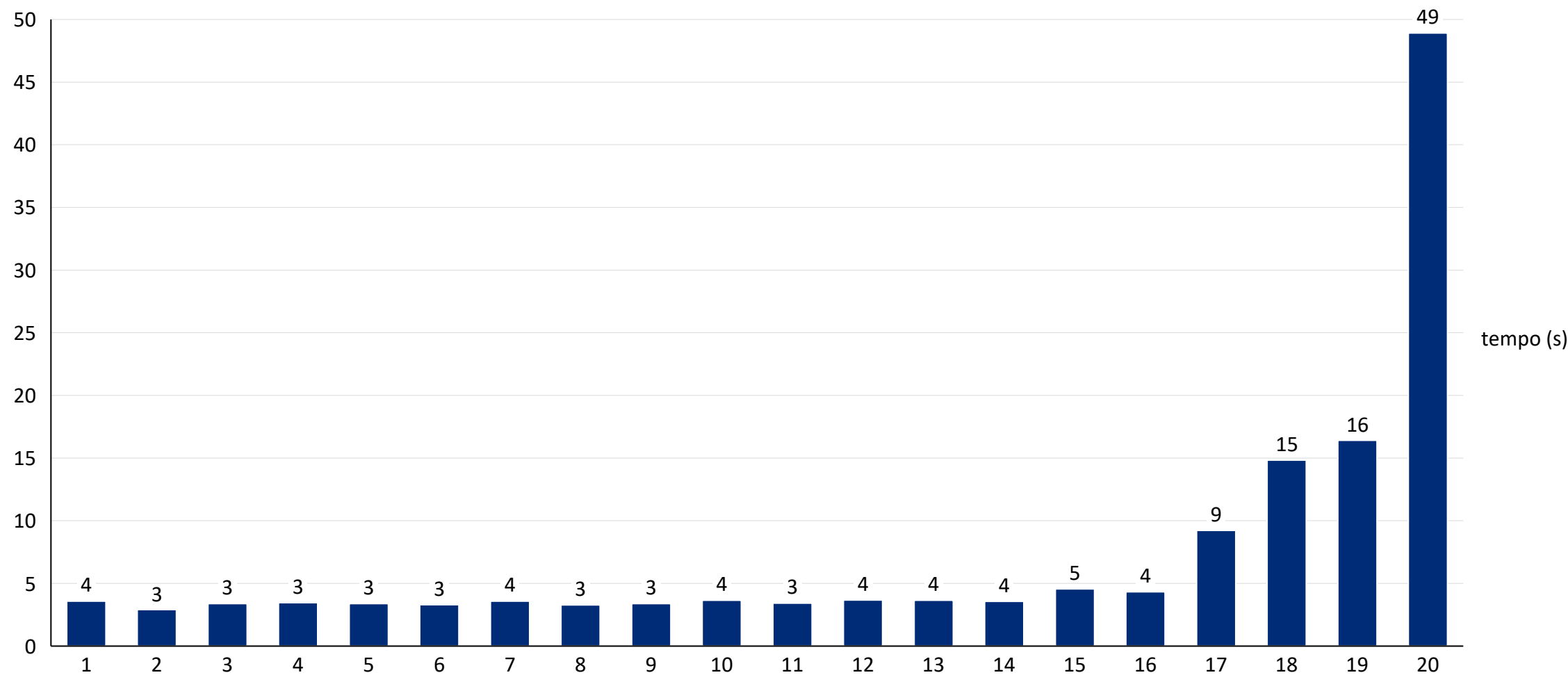
Comparativo de permutações avaliadas utilizando as estratégias

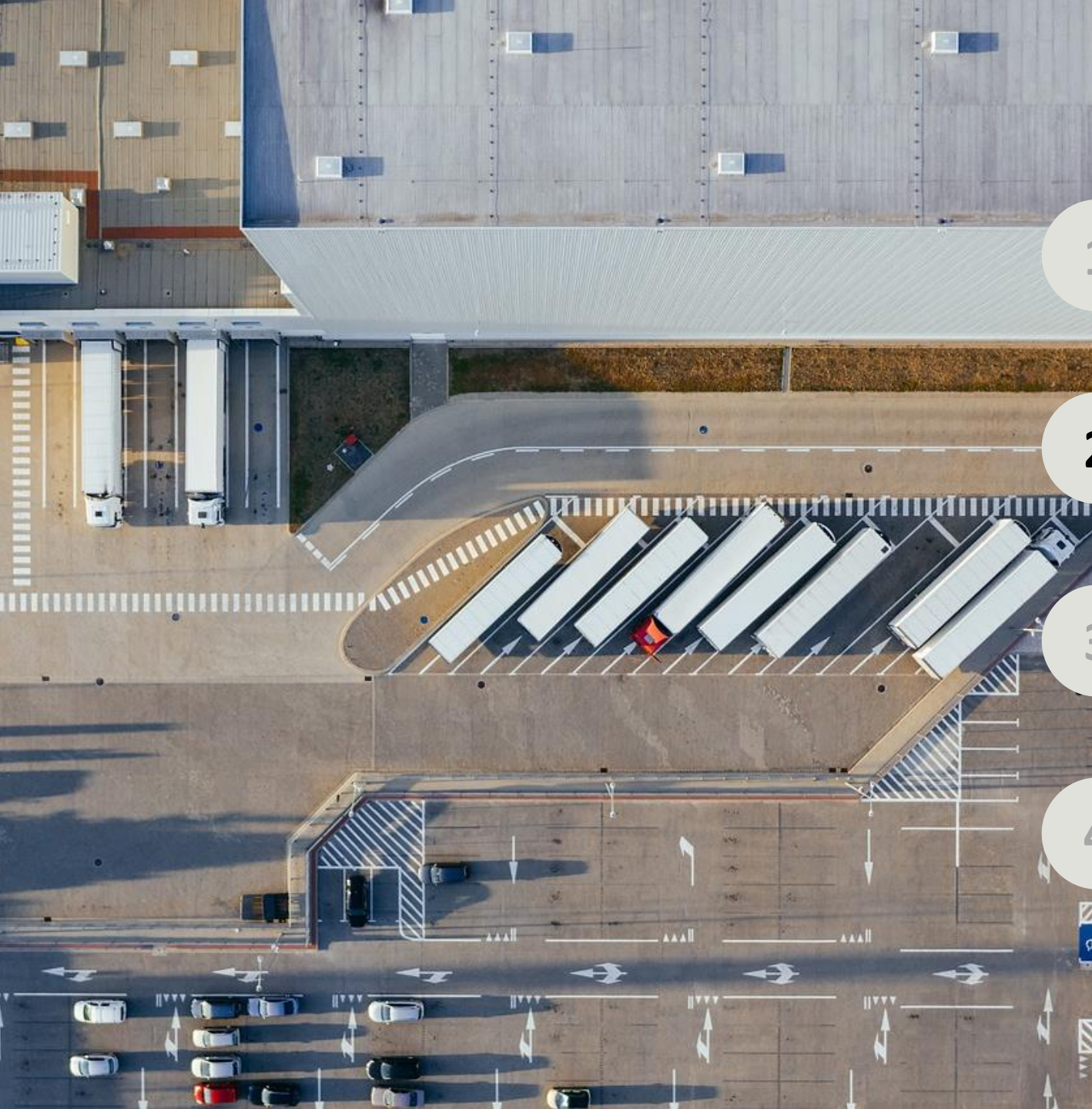
Avaliação do nº de clientes na rota x nº de permutações – (cenário de 20 clientes na base)



Comparativo de tempos de execução

Avaliação do nº de clientes x tempo de execução (s)





1

Geração Exaustiva de Rotas & Permutações

2

Algoritmo de Clarke & Wright

3

Heurística Construtiva Probabilística

4

Tempos de Execução

Descrição do algoritmo

Premissa

- Há uma economia em colocar dois clientes próximos em uma mesma rota

1 Calcular economias

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

2 Ordenar arcos por ordem decrescente de economia

3 Selecionar arco com maior economia (não alocado) e iniciar a rota

3 Adicionar clientes adjacentes até esgotar a capacidade da rota iniciada

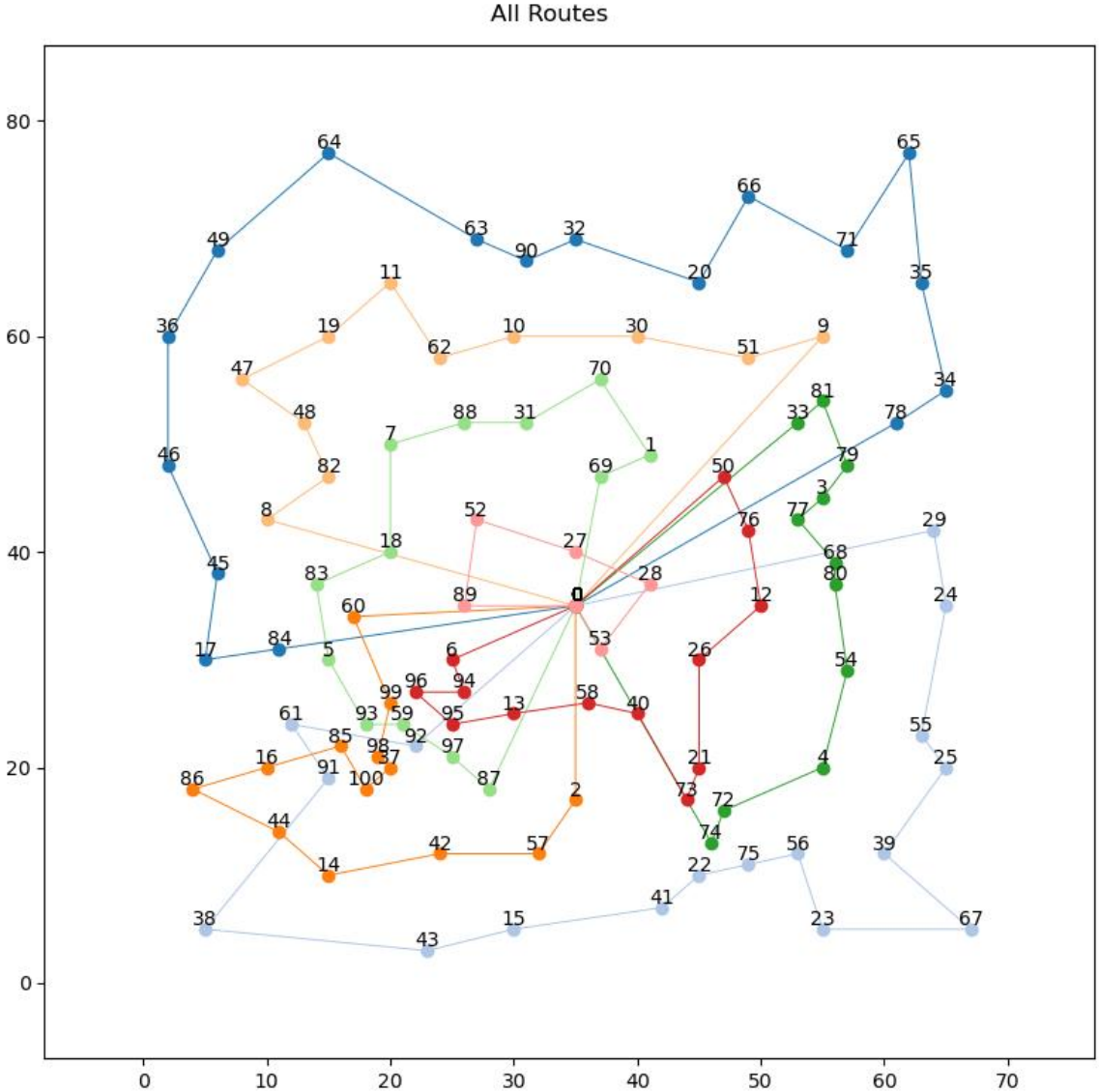
Filtrar a lista de economias e adicionar o cliente com maior economia que respeita as seguintes restrições:

- São adjacentes a rota
- Possuem uma demanda que não estoura a capacidade da rota

4 Retornar ao passo 3 até que todos clientes tenham sido atendidos

Solução gerada

Distância total: 982.5 km



Descrição do algoritmo com múltiplos recomeços

1

Adequação feita (pseudo código abaixo)

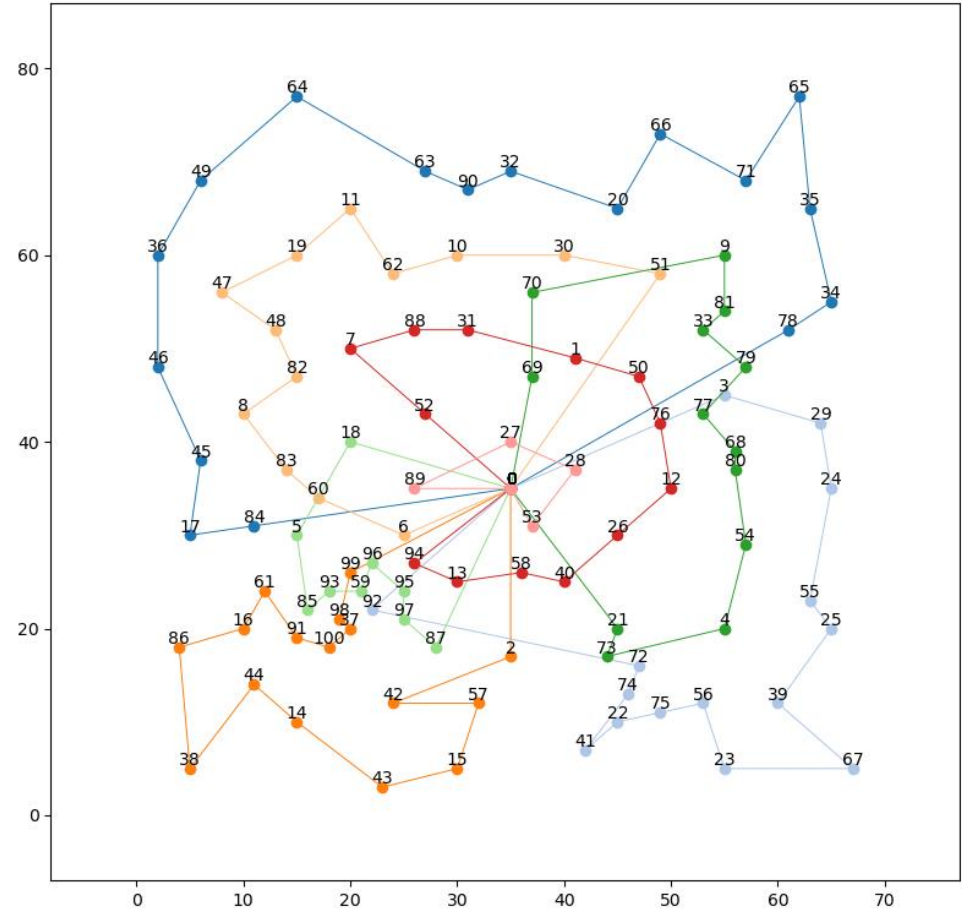
Adicionar um perturbador no calculo das economias, variando entre 0.5 e 1.5 a cada iteração (1000x)

```
1 procedure RandomCWS
2     f_best = infinity
3     while stopping criterion not satisfied Do
4         m = random float between 0.5 and 1.5
5         subprocedure calculateEconomies(m)
6         subprocedure ClarkeWright
7             S = RandomSolution
8         if f(S) < f_best
9             S_best = S
10            f_best = f(S)
11        end if
12    end while
13    return S
14
15 procedure calculateEconomies(m)
16     s_ij = d_i0 + d_j0 - m*d_ij |
```

Comparação do Resultado entre CWS tradicional e randômico.

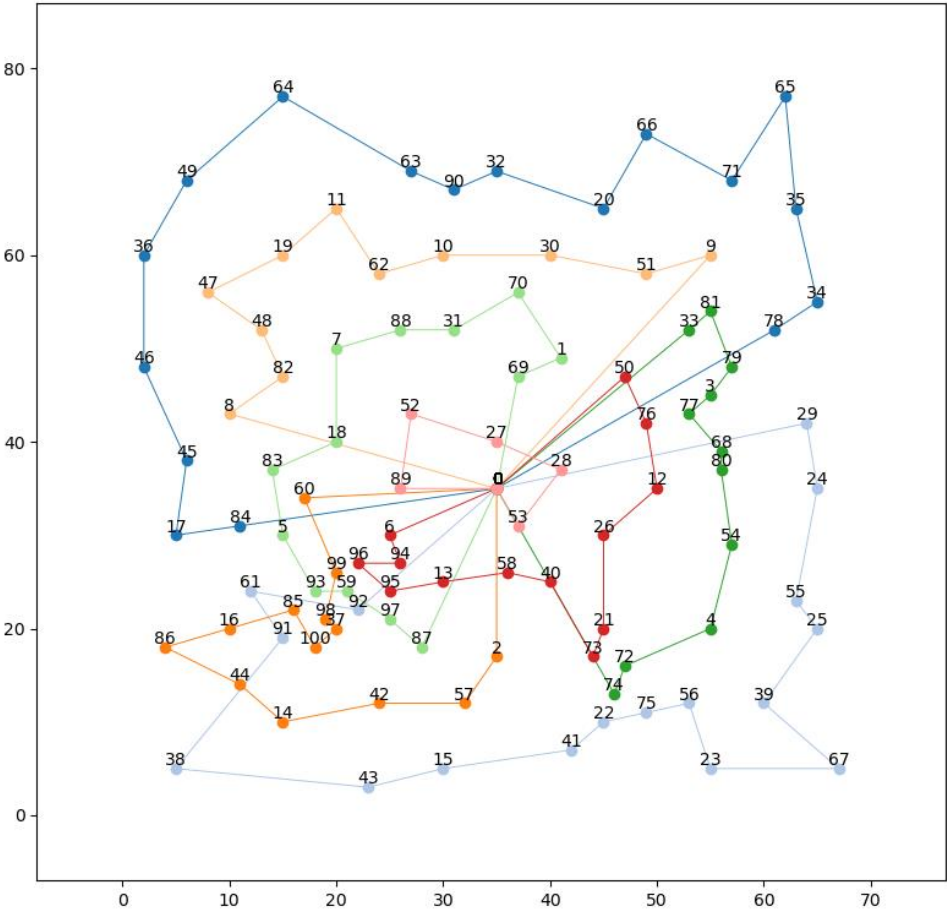
Tradicional: 982.5 km

All Routes



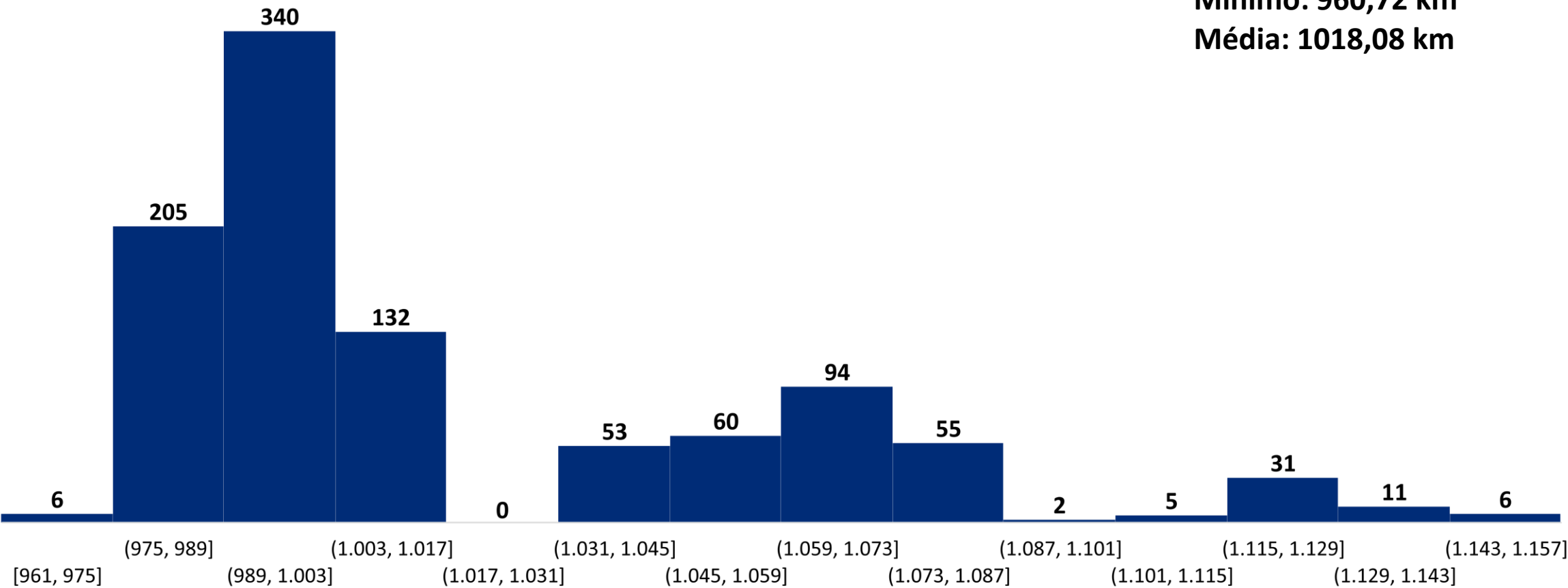
Randômico: 960.7 km

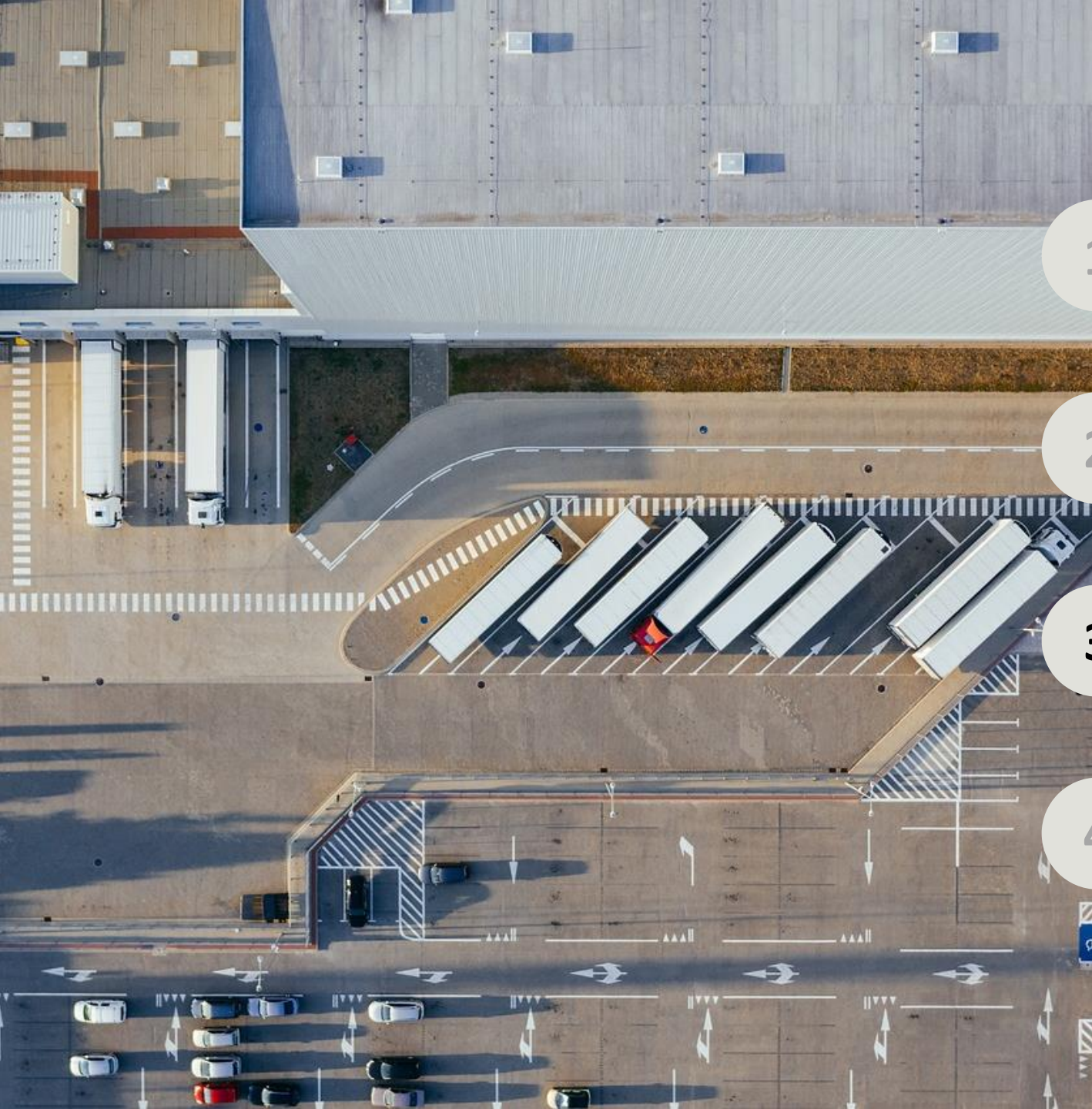
All Routes



Valores da função objetivo

Máximo: 1144,43 km
Mínimo: 960,72 km
Média: 1018,08 km





1

Geração Exaustiva de Rotas & Permutações

2

Algoritmo de Clarke & Wright

3

Heurística Construtiva Probabilística

4

Tempos de Execução

Descrição do algoritmo

1 Definir quantidade de veículos

$m = \text{soma da demanda total} / \text{capacidade do veículo}$

2 Escolher clientes iniciais para cada uma das m rotas

Escolher clientes que necessariamente estariam em rotas diferentes. Para fazer essa escolha, iniciar adicionando o cliente mais distante do depósito a lista de clientes escolhidos (juntamente com o depósito). Em seguida, iterar adicionando os clientes mais distantes aos clientes da lista até que m sejam adicionados.

3 Construir o restante das rotas

1. Selecionar uma rota
2. Elaborar a lista ordenada dos clientes mais próximos (à rota), que ainda estão pendentes de alocação.
3. Sortear um cliente entre os 5 mais próximos
4. Adicionar o cliente na posição com menor acréscimo de custo
5. Seguir para rota seguinte e repetir os passos até que as rotas tenham a capacidade exaurida

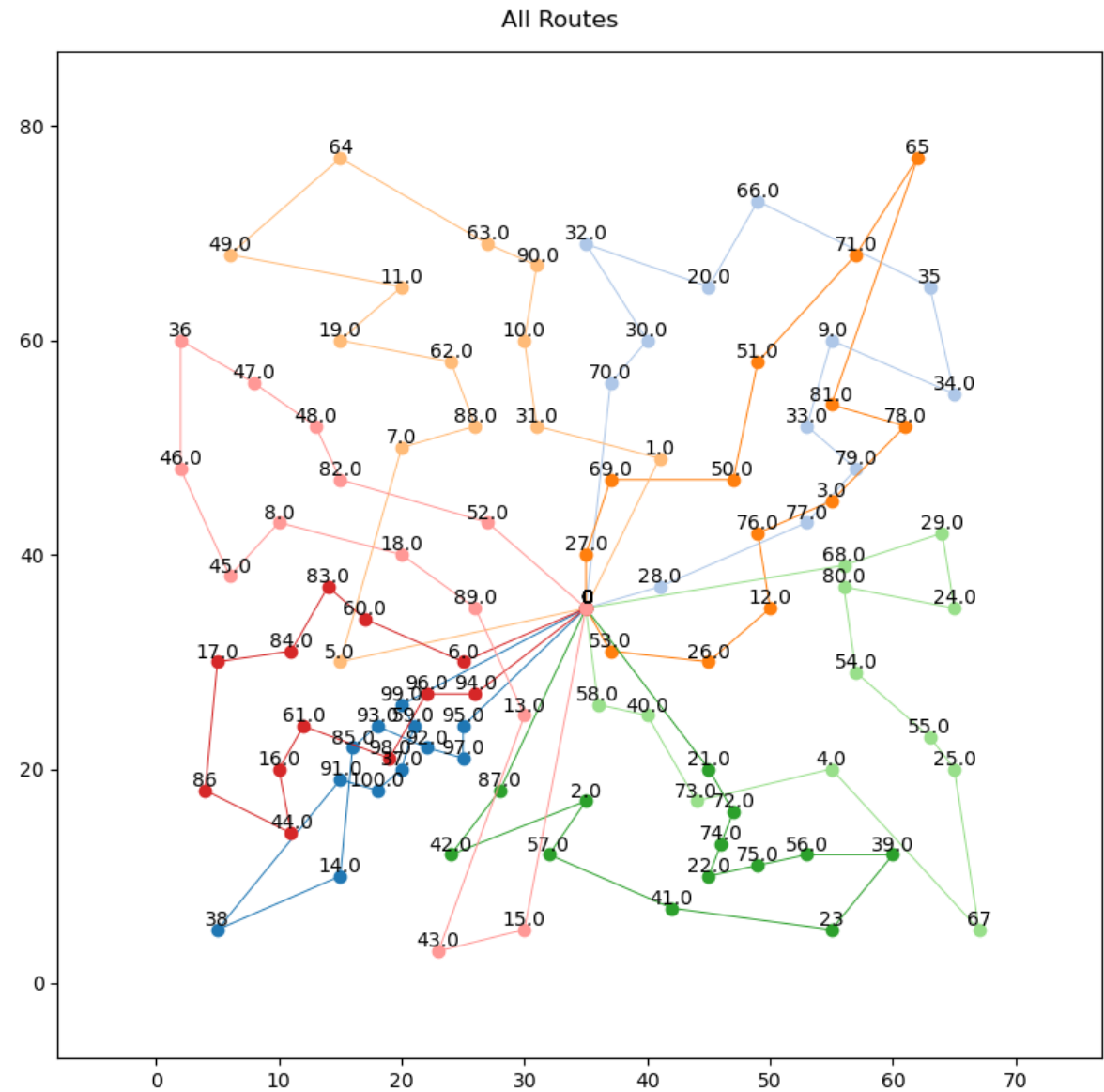
*Caso as rotas fiquem completas e ainda sobrem clientes, reiniciar o algoritmo com $m+1$ veículos

4 Recomeçar (MultiStarts) os passos acima

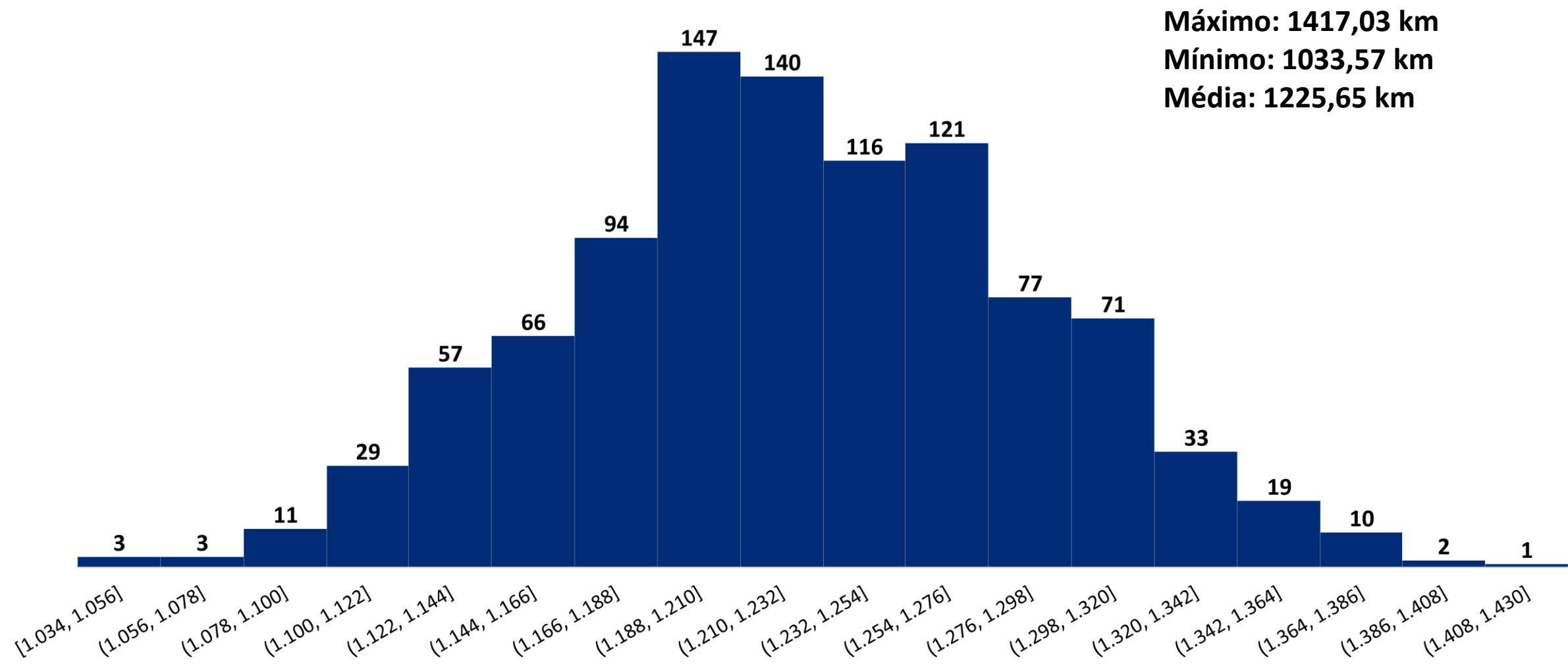
Executar os passos acima 1000x, comparando o valor da função objetivo para identificar qual das iterações gera a melhor solução.

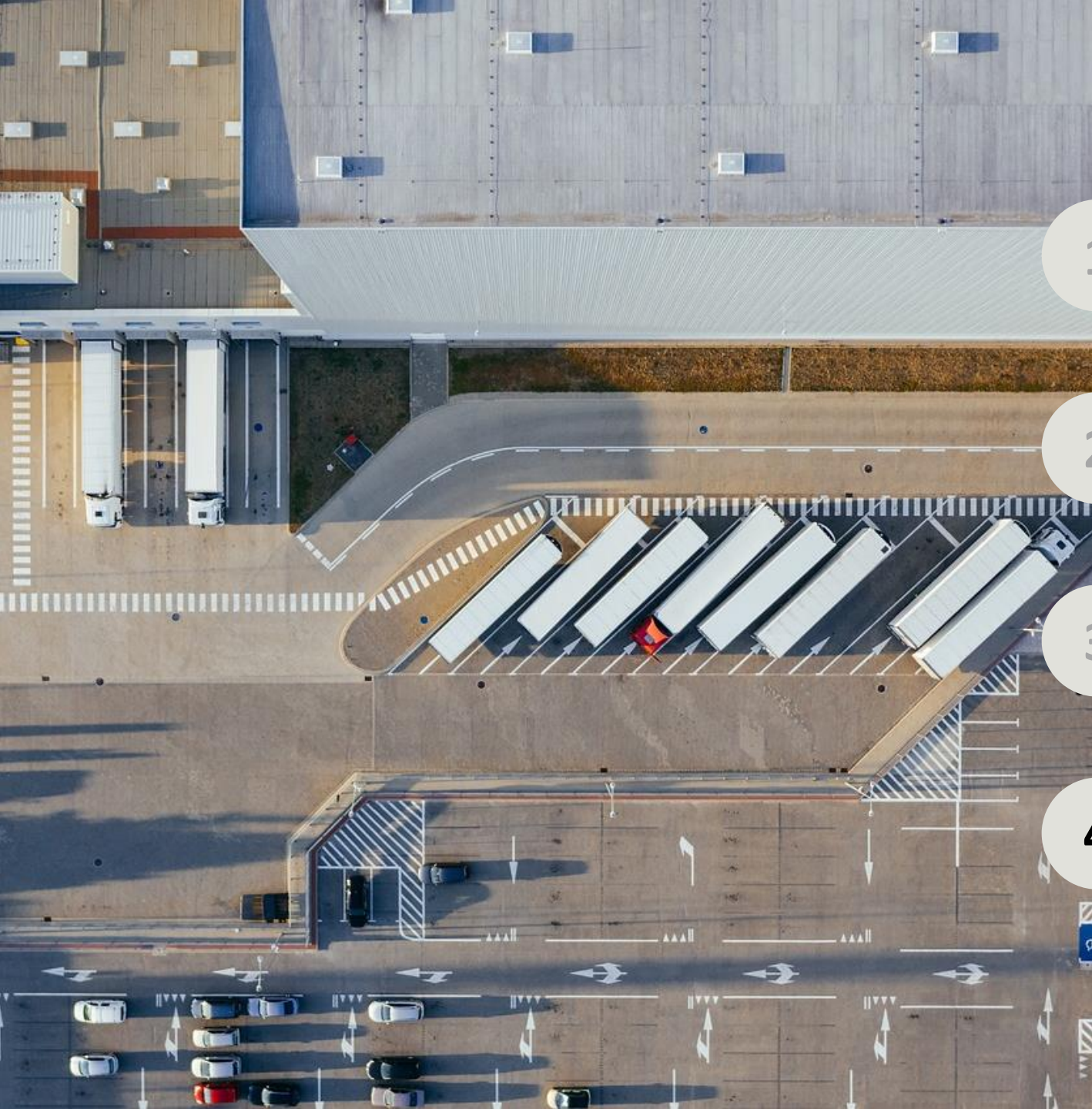
Solução gerada

Distância total: 1033.57 km



Valores da função objetivo





1

Geração Exaustiva de Rotas & Permutações

2

Algoritmo de Clarke & Wright

3

Heurística Construtiva Probabilística

4

Tempos de Execução

Tempos de execução e resultados obtidos

O maior tempo de execução das abordagens probabilísticas não mostrou grande capacidade em reduzir a solução do Clarke & Wright original

