

# Heurísticas Gulosas de Roteamento de Veículos Capacitados

Igor Pires dos Santos e Luis Augusto Toscano

## I. INTRODUÇÃO

Nas últimas décadas um grande esforço de pesquisa tem sido empreendido à modelagem e à métodos de otimização no campo do roteamento de veículos. Isto deve-se ao fato do grande custo do transporte dos produtos, muito desse custo se reflete no preço da matéria-prima e consequentemente no produto-final.

Este trabalho trata o problema de roteamento de veículos capacitados, visando encontrar menor caminho possível, cujo objetivo é desenvolver um conjunto de rotas para atender um conjunto de consumidores percorrendo a distância mínima possível entregando produtos. Cada rota representa o caminho que cada veículo deve percorrer partindo e retornando da base. Cada consumidor recebe uma quantidade de carga do veículo e só pode ser atendido apenas uma única vez.

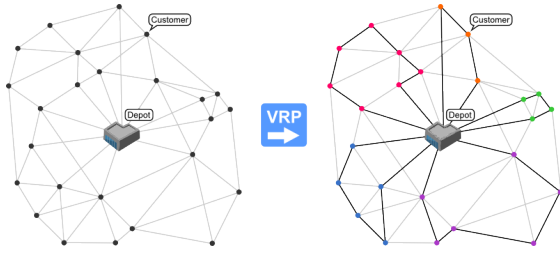


Fig. 1. Exemplo de Roteamento de Veículos com diversas rotas [1]

## II. METODOLOGIA UTILIZADA

A princípio foi modelado o problema em grafos, sendo os nós os clientes a serem atendidos e as arestas os caminhos entre os nós. Sendo os pesos das arestas definidas como distância e os pesos dos nós como tamanho de pedido. As distâncias aqui definidas como distâncias euclidianas, como se os clientes estivessem em um espaço 2D, formando sempre um grafo completo, com todos os nós interligados.

Em seguida a heurística de Inserção do Mais Próximo foi implementada. A heurística de Inserção do Mais Próximo consiste em criar uma rota simples com apenas um nó e a partir desta inserir o cliente mais próximo a rota, ou seja, aquele que pode ser inserido na rota com o menor custo.

E finalmente foram desenvolvidos os teste com instâncias encontradas em [1]

## III. ESTRUTURA UTILIZADA

Foi utilizada a linguagem de programação C++ para modelar o grafo e a heurística.

O grafo é representado como uma lista de adjacência, que utiliza de um vetor para armazenar todos os nós e cada um desses possui o seu peso e uma lista encadeada de adjacências tendo cada adjacência com o seu respectivo peso. Sobre esta lista de adjacências ainda foi implementado uma espécie de InsertSort<sup>1</sup>, ao adicionar cada adjacência é verificado, de acordo com o peso, o local de inserção para que a lista continue ordenada.

A heurística se baseia na criação de "pétalas", isto é, caminhos que tem o formato de pétala, isto se dá porque a heurística monta caminhos que partem da base e visitam os clientes baseados em um setor geográfico[2]. Com isto dito a heurística funciona da seguinte forma:

Sendo  $N$  o conjunto de nós do grafo e  $n$  seu tamanho,

$$vj \in N, \quad \forall j \in \mathbb{N} \text{ e } j < n \quad (1)$$

E  $A$  o conjunto de arestas e  $m$  seu tamanho,

$$aj \in A, \quad \forall j \in \mathbb{N} \text{ e } j < n \quad (2)$$

Sobre estes conjuntos são realizadas as seguintes operações:

- 1º A primeira rota é montada entre a base e o nó mais próximo que ainda não foi visitado, aqui intitulado de  $v1$ , sendo constituída por duas arestas, uma que parte da base e outra que aponta para a base.

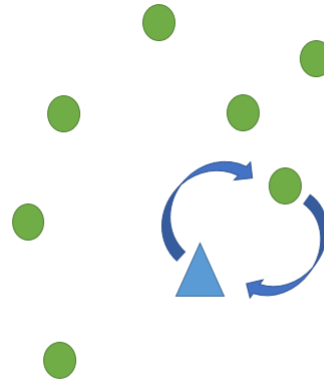


Fig. 2. Representação gráfica do primeiro passo da heurística, arestas em azul representam as arestas adicionadas

<sup>1</sup>InsertSort é um algoritmo de ordenação, eficiente quando aplicado a um pequeno número de elementos

- 2º Para cada aresta dessa rota são analisados todos os nós adjacentes aos dois extremos da aresta que ainda não foram visitados e que ainda há capacidade no veículo é calculado o impacto de retirar esta aresta e adicionar as duas arestas que completariam novamente a rota cobrindo o nó  $v_j$ . Após finalizar todos os cálculos é adicionado o par de arestas que menos aumentar o caminho já obtido.



Fig. 3. Representação gráfica do segundo passo da heurística, arestas em azul representam as arestas adicionadas e as arestas em laranja representam as arestas analisadas para serem trocadas

- 3º Após a inserção dessas suas arestas é realizado novamente todos os cálculos, porém como substituímos apenas uma aresta por vez podemos manter os cálculos realizados com as arestas que não foram retiradas, retirando apenas os cálculos com nós que já foram visitados.

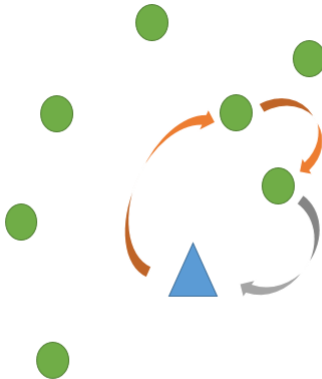


Fig. 4. Representação gráfica do terceiro passo da heurística, arestas em laranjas representam as que serão analisadas e as arestas em cinza representam as que já foram analisadas e o cálculo mantido

Este processo gera uma rota que não necessariamente atende todos os clientes, portanto este processo é repetido até que o faça.

Tendo isso em mente o seguinte algoritmo foi desenvolvido:

```

Data: Grafo  $g$ , int nóBase
Result: int tamanho
1  $visitado[n] \leftarrow \{false\}$ 
2  $solution \leftarrow 0$ 
3 while todos os clientes não forem atendidos do
4    $C_i \leftarrow \min\{cbj \in v_j \mid \forall v_j \in V\}$ 
5    $calculos \leftarrow 0$ 
6    $visitado[v_j] \leftarrow true$ 
7   while houver carga no caminhão do
8      $a1 \leftarrow last\{C_i\}$ 
9      $a2 \leftarrow last\{C_i\}.previous()$ 
10    for  $v_j \in V \mid adjacente(v_j, a1)$  do
11      if  $peso \leq capacidadeRestante$  then
12         $calculos \leftarrow calculos \cup calculos(v_j, a1)$ 
13      end
14    end
15    for  $v_j \in V \mid adjacente(v_j, a2)$  do
16      if  $peso \leq capacidadeRestante$  then
17         $calculos \leftarrow calculos \cup calculos(v_j, a2)$ 
18      end
19    end
20    if  $calculos == \{\emptyset\}$  then
21      Fim da rota
22    end
23     $solution \leftarrow solution \cup \min\{calculos.peso()\}$ 
24     $C_i \leftarrow C_i - arestaRetirada$ 
25     $C_i \leftarrow C_i \cup arestas(\min\{calculos.peso()\})$ 
26    for  $calculo \in calculos$  do
27      Retira os cálculos de nós já visitados
28      Retira os cálculos da aresta retirada
29    end
30  end
31 end

```

Algorithm 1: Heurística Gulosa Implementada

A partir desta heurística gulosa foi implementada a heurística gulosa randomizada e a heurística gulosa randomizada reativa, que diferem na escolha da melhor solução (linhas 23-25). Para estas duas últimas heurísticas, utilizamos a seguinte variável:

$$\alpha \in [0, 1] \quad (3)$$

A heurística gulosa randomizada utiliza um alfa e uma semente randomizada, a partir desses dois identifica a posição da lista de cálculos que será inserida, quanto mais o alfa se aproximar de 0 resultados mais similares a heurística gulosa, ou seja, o alfa representa uma seção da lista da onde será tirada a solução e a semente randomizada cria uma aleatoriedade dentro daquela seção. Obtendo a seguinte equação de posição:

$$pos \leftarrow rand() \% (\alpha * calculos.tamanho()) \quad (4)$$

E por último a heurística gulosa randomizada reativa, intitulada como GRASP<sup>2</sup> Reativa em [3], para esta necessitaremos de:

$$\psi = \{\alpha_1, \dots, \alpha_m\} \quad (5)$$

De acordo com [3], "seja este  $\psi$  um vetor com os valores possíveis de alfa. As probabilidades associadas à escolha de cada valor são inicializadas com  $p_i = 1/m, i = 1, \dots, m$ . Além disto, seja  $s^*$  a melhor solução já encontrada e  $Ai0$  o valor médio de todas as soluções encontradas utilizando-se  $\alpha = \alpha_i, i = 1, \dots, m$ . As probabilidades de seleção são periodicamente reavaliadas como  $p_i = q_i / \sum_{i=1}^m t_i$  onde  $q_i = s^* / Ai, i = 1, \dots, m$ . Maiores valores de  $q_i$  correspondem a valores mais apropriados para o parâmetro  $\alpha$ . As probabilidades associadas aos valores mais apropriados irão aumentar quando forem reavaliadas. O enfoque reativo pode levar a melhorias sensíveis em relação ao procedimento GRASP básico, em torno de uma robustez do algoritmo e a qualidade das soluções, devido à maior diversidade destas últimas e a menos dependência do ajuste de parâmetros."

#### IV. EXPERIMENTOS COMPUTACIONAIS

Com isso foram iniciados os testes utilizando as instâncias encontradas em [1], uma instância pequena ("Instance20.txt")[4], uma média ("Instance50.txt")[5], uma grande ("Instance101-2.txt")[6] e por fim uma gigante ("Instance121.txt")[7] e então os resultados foram comparados com o melhor resultado.

Na tabela está representado a qual instância a linha se refere, o tempo em segundos que demorou para ser obtido, o custo mínimo encontrado, o caminho ótimo e o  $\Delta_{opt}(\%)$ , sendo :

$$\Delta_{opt} = \frac{\frac{100 * Custo}{Otimo} - 100}{100} \quad (6)$$

Os resultado foram obtidos em 2 máquinas diferentes, porém similares, devido ao grande tempo que cada teste demandava.

##### Máquina 1

Processador: Core i7 2<sup>a</sup> Geração, Quadcore 2.8 GHz  
Memória RAM: RAM, 8 GBs DDR3 1333 MHz  
OS: Ubuntu 14.04 64 bits

##### Máquina 2

Processador: Core i7 2<sup>a</sup> Geração, Quadcore 2.8 GHz  
Memória RAM: RAM, 8 GBs DDR3 1333 MHz  
OS: Ubuntu 14.10 64 bits

<sup>2</sup>GRASP: Greedy Randomized Adaptive Search Procedure

Com isso, montou-se a seguinte tabela para a Heurística Gulosa:

TABLE I  
SOLUÇÃO GULOSA

Instância	Tempo(s)	Custo	Ótimo	$\Delta opt(\%)$
Instance20.txt	0,037103	290,463	216	34,47
Instance50.txt	0,155425	753,856	554	36,07
Instance101-2.txt	18,605	1294,61	820	57,87
Instance121.txt	4,27945	1600,02	1034	54,74

Em seguida foram montadas as tabelas para as heurísticas gulosas randomizadas e seus diferentes valores de  $\alpha = \{0.1, 0.2, 0.5, 0.7\}$  executado 1000 vezes cada um.

TABLE II  
SOLUÇÃO GULOSA RANDOMIZADA PARA 20 CLIENTES

$\alpha \leftarrow 0.1$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance20.txt	8,92801	226,949	0,050689815	216
Instance20.txt	8,85152	224,698	0,040268519	216
Instance20.txt	8,84206	219,938	0,018231481	216
Instance20.txt	9,00722	224,132	0,037648148	216
Instance20.txt	8,84492	230,773	0,068393519	216
Instance20.txt	8,84183	222,72	0,031111111	216
Instance20.txt	9,19115	226,922	0,050564815	216
Instance20.txt	8,89199	226,949	0,050689815	216
Instance20.txt	8,84618	226,922	0,050564815	216
Instance20.txt	8,913	227,051	0,051162037	216

TABLE III  
SOLUÇÃO GULOSA RANDOMIZADA PARA 20 CLIENTES

$\alpha \leftarrow 0.2$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance20.txt	9,01937	246,782	0,142509259	216
Instance20.txt	8,93468	252,121	0,167226852	216
Instance20.txt	8,88997	240,482	0,113342593	216
Instance20.txt	8,95187	249,817	0,156560185	216
Instance20.txt	9,03934	242,729	0,12374537	216
Instance20.txt	8,97748	249,152	0,153481481	216
Instance20.txt	9,02572	242,414	0,122287037	216
Instance20.txt	9,20375	246,794	0,142564815	216
Instance20.txt	9,10118	244,787	0,133273148	216
Instance20.txt	9,00684	239,287	0,107810185	216

TABLE IV  
SOLUÇÃO GULOSA RANDOMIZADA PARA 20 CLIENTES

$\alpha \leftarrow 0.5$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance20.txt	9,26147	297,943	0,379365741	216
Instance20.txt	9,20447	277,77	0,285972222	216
Instance20.txt	9,35898	301,907	0,397717593	216
Instance20.txt	9,24479	291,887	0,351328704	216
Instance20.txt	9,23668	285,403	0,321310185	216
Instance20.txt	9,32784	281,854	0,30487963	216
Instance20.txt	9,42742	286,504	0,326407407	216
Instance20.txt	9,31575	292,337	0,353412037	216
Instance20.txt	9,34461	295,527	0,368180556	216
Instance20.txt	9,29926	296,197	0,371282407	216

TABLE V  
SOLUÇÃO GULOSA RANDOMIZADA PARA 20 CLIENTES

$\alpha \leftarrow 0.7$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance20.txt	9,78203	330,104	0,528259259	216
Instance20.txt	9,42793	322,392	0,492555556	216
Instance20.txt	9,43179	313,776	0,452666667	216
Instance20.txt	9,46097	324,014	0,500064815	216
Instance20.txt	9,50565	326,881	0,513337963	216
Instance20.txt	10,5062	333,456	0,543777778	216
Instance20.txt	9,54829	332,922	0,541305556	216
Instance20.txt	9,58326	312,96	0,448888889	216
Instance20.txt	9,7799	333,469	0,543837963	216
Instance20.txt	9,74124	296,778	0,373972222	216

TABLE VI  
SOLUÇÃO GULOSA RANDOMIZADA PARA 50 CLIENTES

$\alpha \leftarrow 0.1$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance50.txt	81,9063	809,536	0,461256318	554
Instance50.txt	82,3042	828,017	0,494615523	554
Instance50.txt	81,9324	826,595	0,492048736	554
Instance50.txt	82,1266	829,151	0,496662455	554
Instance50.txt	82,3767	840,977	0,518009025	554
Instance50.txt	81,7367	824,658	0,488552347	554
Instance50.txt	82,0679	823,45	0,486371841	554
Instance50.txt	82,492	826,495	0,491868231	554
Instance50.txt	81,9988	842,692	0,521104693	554
Instance50.txt	81,4494	830,854	0,499736462	554

TABLE VII  
SOLUÇÃO GULOSA RANDOMIZADA PARA 50 CLIENTES

$\alpha \leftarrow 0.2$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance50.txt	82,1032	913,014	0,648039711	554
Instance50.txt	82,6313	911,151	0,644676895	554
Instance50.txt	83,7387	918,808	0,658498195	554
Instance50.txt	82,6068	899,612	0,623848375	554
Instance50.txt	82,7711	922,261	0,664731047	554
Instance50.txt	83,4333	919,703	0,660113718	554
Instance50.txt	82,8172	901,337	0,626962094	554
Instance50.txt	83,0265	891,044	0,608382671	554
Instance50.txt	82,9818	920,764	0,662028881	554
Instance50.txt	82,8336	915,736	0,652953069	554

TABLE VIII  
SOLUÇÃO GULOSA RANDOMIZADA PARA 50 CLIENTES

$\alpha \leftarrow 0.5$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance50.txt	83,6341	1142,43	1,062148014	554
Instance50.txt	83,0461	1160,87	1,095433213	554
Instance50.txt	85,7082	1083,9	0,956498195	554
Instance50.txt	83,2044	1145,39	1,067490975	554
Instance50.txt	83,4283	1099,64	0,984909747	554
Instance50.txt	84,3206	1143,52	1,064115523	554
Instance50.txt	82,2507	1155,06	1,084945848	554
Instance50.txt	83,4182	1151,19	1,077960289	554
Instance50.txt	82,6534	1139,74	1,057292419	554
Instance50.txt	83,6599	1135,25	1,049187726	554

TABLE IX  
SOLUÇÃO GULOSA RANDOMIZADA PARA 50 CLIENTES

$\alpha \leftarrow 0.7$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance50.txt	87,1514	1284,41	1,318429603	554
Instance50.txt	83,9277	1257,39	1,26965704	554
Instance50.txt	83,9005	1261,91	1,277815884	554
Instance50.txt	84,6898	1207,83	1,180198556	554
Instance50.txt	83,6664	1266,66	1,286389892	554
Instance50.txt	84,2271	1263	1,279783394	554
Instance50.txt	83,3608	1248,27	1,253194946	554
Instance50.txt	82,8037	1258,55	1,271750903	554
Instance50.txt	83,1868	1262,75	1,27933213	554
Instance50.txt	83,2068	1264,6	1,28267148	554

TABLE X  
SOLUÇÃO GULOSA RANDOMIZADA PARA 100 CLIENTES

$\alpha \leftarrow 0.1$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance101-2.txt	1478,04	1366,18	0,666073171	820
Instance101-2.txt	1515,27	1397,48	0,704243902	820
Instance101-2.txt	1601,34	1424,03	0,736621951	820
Instance101-2.txt	1620,11	1424,24	0,736878049	820
Instance101-2.txt	1601,51	1466,15	0,787987805	820
Instance101-2.txt	1612,37	1380,81	0,683914634	820
Instance101-2.txt	1557,51	1489,43	0,816378049	820
Instance101-2.txt	1529,5	1437,95	0,753597561	820
Instance101-2.txt	1604,87	1416,64	0,727609756	820
Instance101-2.txt	1625,83	1388,87	0,693743902	820

TABLE XI  
SOLUÇÃO GULOSA RANDOMIZADA PARA 100 CLIENTES

$\alpha \leftarrow 0.2$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance101-2.txt	1471,39	1761,19	1,147792683	820
Instance101-2.txt	1515,67	1853,68	1,260585366	820
Instance101-2.txt	1602,22	1905,64	1,32395122	820
Instance101-2.txt	1616,18	1795,18	1,189243902	820
Instance101-2.txt	1602,42	1844,65	1,249573171	820
Instance101-2.txt	1613,84	1851,3	1,257682927	820
Instance101-2.txt	1559	1801,93	1,19747561	820
Instance101-2.txt	1526,94	1858,13	1,266012195	820
Instance101-2.txt	1602,94	1857,22	1,264902439	820
Instance101-2.txt	1621,6	1801,43	1,196865854	820

TABLE XII  
SOLUÇÃO GULOSA RANDOMIZADA PARA 100 CLIENTES

$\alpha \leftarrow 0.5$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance101-2.txt	1442,06	2689,88	2,280341463	820
Instance101-2.txt	1487,64	2562,43	2,124914634	820
Instance101-2.txt	1572,09	2696,15	2,287987805	820
Instance101-2.txt	1590,79	2675,18	2,262414634	820
Instance101-2.txt	1577,93	2679,79	2,268036585	820
Instance101-2.txt	1585,32	2688,25	2,278353659	820
Instance101-2.txt	1527,76	2654,16	2,236780488	820
Instance101-2.txt	1501,02	2729,83	2,329060976	820
Instance101-2.txt	1569,46	2705,28	2,299121951	820
Instance101-2.txt	1597,55	2661,03	2,245158537	820

TABLE XIII  
SOLUÇÃO GULOSA RANDOMIZADA PARA 100 CLIENTES

$\alpha \leftarrow 0.7$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance101-2.txt	1411,55	3044,5	2,712804878	820
Instance101-2.txt	1461,37	3115,9	2,799878049	820
Instance101-2.txt	1549,21	3071,96	2,746292683	820
Instance101-2.txt	1556,78	2999,04	2,657365854	820
Instance101-2.txt	1552,3	3080,91	2,757207317	820
Instance101-2.txt	1556,03	3061,31	2,733304878	820
Instance101-2.txt	1492,83	3072,5	2,74695122	820
Instance101-2.txt	1474,91	3067,93	2,741378049	820
Instance101-2.txt	1543,22	2995,66	2,653243902	820
Instance101-2.txt	1569,19	2999,56	2,658	820

TABLE XIV  
SOLUÇÃO GULOSA RANDOMIZADA PARA 120 CLIENTES

$\alpha \leftarrow 0.1$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance121.txt	2403,63	1777,9	0,719439072	1034
Instance121.txt	2386,84	1741,03	0,683781431	1034
Instance121.txt	2423,39	1701,17	0,645232108	1034
Instance121.txt	2392,99	1737,1	0,679980658	1034
Instance121.txt	2370,45	1790,6	0,73172147	1034
Instance121.txt	2572,99	1703,53	0,647514507	1034
Instance121.txt	2638,8	1798,63	0,739487427	1034
Instance121.txt	3970,55	1781,09	0,722524178	1034
Instance121.txt	2458,12	1720,18	0,663617021	1034
Instance121.txt	2354,12	1746,34	0,688916828	1034

TABLE XV  
SOLUÇÃO GULOSA RANDOMIZADA PARA 120 CLIENTES

$\alpha \leftarrow 0.2$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance121.txt	2361,27	2116,67	1,047069632	1034
Instance121.txt	2363,32	2142,96	1,072495164	1034
Instance121.txt	2427,68	2141,71	1,071286267	1034
Instance121.txt	2414,67	2143,49	1,073007737	1034
Instance121.txt	2356,69	2152,11	1,081344294	1034
Instance121.txt	2714,33	2100	1,030947776	1034
Instance121.txt	2631,24	2050,44	0,983017408	1034
Instance121.txt	3975,42	2214,35	1,141537718	1034
Instance121.txt	2411,29	2106,4	1,037137331	1034
Instance121.txt	2345,4	2212,63	1,139874275	1034

TABLE XVI  
SOLUÇÃO GULOSA RANDOMIZADA PARA 120 CLIENTES

$\alpha \leftarrow 0.5$				
Instância	Tempo(s)	Custo	$\Delta opt(\%)$	Ótimo
Instance121.txt	2446,57	3570,92	2,453500967	1034
Instance121.txt	2369,98	3609,14	2,490464217	1034
Instance121.txt	2324,3	3657,39	2,53712766	1034
Instance121.txt	2411,08	3611,86	2,493094778	1034
Instance121.txt	2398,04	3688,42	2,567137331	1034
Instance121.txt	2463,7	3549,78	2,433056093	1034
Instance121.txt	2494,77	3666,35	2,545793037	1034
Instance121.txt	3910,09	3627,41	2,508133462	1034
Instance121.txt	2426,57	3696,55	2,575	1034
Instance121.txt	2378,66	3674,31	2,553491296	1034

TABLE XVII  
SOLUÇÃO GULOSA RANDOMIZADA PARA 120 CLIENTES

$\alpha \leftarrow 0.7$				
Instância	Tempo(s)	Custo	$\Delta_{opt}(\%)$	Ótimo
Instance121.txt	2388,12	4590,43	3,439487427	1034
Instance121.txt	2380,66	4370,13	3,226431335	1034
Instance121.txt	2396,6	4472,86	3,325783366	1034
Instance121.txt	2392,11	4557,86	3,407988395	1034
Instance121.txt	2354,89	4492,8	3,345067698	1034
Instance121.txt	2593,91	4549,58	3,399980658	1034
Instance121.txt	2512,24	4544,11	3,394690522	1034
Instance121.txt	3946,86	4454,14	3,307678917	1034
Instance121.txt	2430,69	4441,48	3,295435203	1034
Instance121.txt	2491,19	4483,67	3,336237911	1034

E por último foi testada a heurística gulosa randomizada reativa, porém não foi possível testá-la em todas as instâncias devido a grande lentidão do algoritmo. Para a instância de 20 nós foram 10 execuções de 2500 iterações e para a instância de 50 nós foram 8 execuções de 500 iterações.

TABLE XVIII  
SOLUÇÃO GULOSA RANDOMIZADA REATIVA PARA 20 CLIENTES

Instância	Tempo(s)	$\alpha$	Custo	$\Delta_{opt}(\%)$	Ótimo
Instance20.txt	1197,12	0.1	294,938	0,365453704	216
Instance20.txt	1250,27	0.1	294,567	0,363736111	216
Instance20.txt	1197,01	0.1	294,725	0,364467593	216
Instance20.txt	1197,4	0.1	294,748	0,364574074	216
Instance20.txt	1197,47	0.1	294,457	0,363226852	216
Instance20.txt	1197,83	0.1	294,811	0,364865741	216
Instance20.txt	18680,1	0.1	294,577	0,363782407	216
Instance20.txt	1208,55	0.1	294,549	0,363652778	216
Instance20.txt	1211,09	0.1	294,798	0,364805556	216
Instance20.txt	1199,83	0.1	294,839	0,36499537	216

Como exemplo, abaixo está a representação de um vetor de probabilidade obtido no caso anterior:

TABLE XIX  
VETOR DE PROBABILIDADE DA SOLUÇÃO GULOSA RANDOMIZADA REATIVA PARA 20 CLIENTES

$p_0$	0.129767
$p_1$	0.120638
$p_2$	0.1127
$p_3$	0.106038
$p_4$	0.0998544
$p_5$	0.0950296
$p_6$	0.0905791
$p_7$	0.0860078
$p_8$	0.0820323
$p_9$	0.077353

TABLE XX  
SOLUÇÃO GULOSA RANDOMIZADA REATIVA PARA 50 CLIENTES

Instância	Tempo(s)	$\alpha$	Custo	$\Delta_{opt}(\%)$	Ótimo
Instance50.txt	2047,61	0.1	965,12	0,742093863	554
Instance50.txt	2052,76	0.1	965,67	0,743086643	554
Instance50.txt	2075,41	0.1	965,595	0,742951264	554
Instance50.txt	2064,2	0.1	964,994	0,741866426	554
Instance50.txt	2068,7	0.1	964,947	0,741781588	554
Instance50.txt	2083,43	0.1	965,477	0,742738267	554
Instance50.txt	2045,11	0.1	965,134	0,742119134	554
Instance50.txt	2044,64	0.1	964,757	0,741438628	554

Como exemplo, abaixo está a representação de um vetor de probabilidade obtido no caso anterior:

TABLE XXI  
VETOR DE PROBABILIDADE DA SOLUÇÃO GULOSA RANDOMIZADA REATIVA PARA 50 CLIENTES

$p_0$	0.133859
$p_1$	0.123781
$p_2$	0.120373
$p_3$	0.103967
$p_4$	0.0973015
$p_5$	0.0882108
$p_6$	0.084135
$p_7$	0.0896335
$p_8$	0.0829123
$p_9$	0.0758267

## V. CONCLUSÕES

Observando os testes notou-se que para este problema quanto mais gulosa a heurística, ou menor o alfa, melhor o resultado, pois trata-se de distância e quanto maior o alfa, maior o aumento da rota ao se adicionar um novo nó.

Observou-se também que em instâncias pequenas a heurística se aproxima muito do resultado ótimo chegando a  $\Delta_{opt} < 2\%$ , enquanto em instâncias maiores atinge  $\Delta_{opt} \approx 65\%$ .

Sobre o tempo de execução pode se dizer que a complexidade do algoritmo é fatorial,  $O(n!)$ , pois os cálculos são executados  $n$  vezes e em seguida  $n-1$  vezes e assim por diante até atender todos os clientes, por este motivo os tempos de execução em instância de tamanhos diferentes são tão diferentes. Fora isso, este trabalho teve problemas com vazamentos de memória devido ao esquecimento de desalocação de ponteiros, causando diversas perdas de informações que foram resolvidas utilizando o *shared\_ptr*<sup>3</sup>, porém também devido a uma implementação incorreta, este diminuiu a velocidade da heurística.

## REFERENCES

- [1] U. de Málaga — Spain. (2014) Neo - networking and emerging optimization. [Online]. Available: <http://neo.lcc.uma.es/vrp/>
- [2] C. H. M. Ryan and F. Glover, "Extensions of the petal method for vehicle routing," 1993.
- [3] A. C. M. MACIEL, "Heurísticas para o problema do caixeiro viajante branco e preto," 2005.
- [4] A. et al, "P-n20-k2."
- [5] —, "P-n50-k7."
- [6] C. et al, "M-n101-k10."
- [7] —, "M-n121-k7."

<sup>3</sup>*shared\_ptr* é um dos tipos de SmartPointer que são ponteiros inteligentes que possuem mecanismos de coleta de lixo automática, desalocando os objetos que não possuem mais nenhuma referência, utilizado neste caso para que não houvesse "vazamento de memória" devido ao esquecimento da desalocação de ponteiros