

# Computação Natural

## Algoritmos Genéticos

Bruno Pereira  
A75135

João Coelho  
A74859

Luís Fernandes  
A74748

Maria Ana de Brito  
A73580

6 de Maio de 2018

### Resumo

Este documento propõe e descreve uma possível resolução do problema LRP - Location Routing Problem, recorrendo a algoritmos genéticos.

Para a aplicação do algoritmo utiliza-se a biblioteca GA, da linguagem R, sendo a estratégia aplicada em três conjuntos de instâncias da literatura, todos eles bem conhecidos em problemas deste tipo. Além da implementação do algoritmo genético, este documento permite constatar sobre os tempos computacionais para resolver problemas LRP desta forma.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Algoritmo genético</b>	<b>3</b>
2.1	Representação do cromossoma . . . . .	3
2.2	Função de fitness . . . . .	4
2.2.1	Custo da distância . . . . .	4
2.2.2	Custo da circulação de veículos . . . . .	4
2.2.3	Custo de abertura de um depósito . . . . .	5
2.3	Treino do Algoritmo Genético . . . . .	5
<b>3</b>	<b>Resultados</b>	<b>5</b>
<b>4</b>	<b>Conclusão</b>	<b>5</b>

# 1 Introdução

Para introduzir este problema é importante perceber o que caracteriza os **Location Routing Problems (LRP)**. Analisando o artigo que consta nas referências [1], concluiu-se que a principal observação é o facto de lidarem com a combinação de dois tipos de decisão logística: a localização de depósitos e a distribuição de rotas. Deste problema surgiram algumas variantes, de entre as quais os chamados CLRP - Capacitated Location Routing Problems, que se distinguem dos anteriores pela existência de depósitos/veículos com capacidades de armazenamento/transporte limitadas. Pode-se, portanto, definir o caso em estudo como um CLRP.

Relativamente a estes problemas, há um grupo de variáveis que importa conhecer, desde logo o grafo  $G(V,E)$  sobre o qual assenta. Aqui, os vértices ( $V$ ) representam a reunião do conjunto dos depósitos com o conjunto dos clientes, isto é, todas as posições possíveis no "mapa" do problema, entre as quais os veículos se deslocarão durante as suas rotas. Já as arestas ( $E$ ) simbolizam as rotas entre as posições representadas pelos vértices. Sabendo que as rotas efetuadas por um veículo têm como pontos de partida e chegada o mesmo depósito, com paragens em um ou mais clientes, as outras variáveis relevantes são enumeradas de seguida:

- Cada cliente  $i \in I$  possui uma procura  $d_i$ , que é satisfeita por um único depósito  $j \in J$ ;
- Associada a um depósito está uma capacidade máxima  $w_j$ ;
- Não há restrição para o número de veículos, sabendo que estes têm uma capacidade de transporte fixa  $Q$ ;
- A abertura de um depósito, isto é, a atribuição de clientes a um depósito, acarreta um custo  $f_j$ ;
- Ao uso de cada veículo acresce a quantia fixa  $F$ ;
- À deslocação numa rota está associado um custo  $c_{ij}$ , por unidade de distância.

A soma das procuras dos clientes atribuídos a um depósito não pode ultrapassar a sua capacidade máxima. Todas as variáveis anteriores serão definidas em função do *dataset* em análise, à exceção do custo por unidade de distância ( $c_{ij}$ ), que será sempre de 1.

O objetivo final de problemas LRP ou CLRP é encontrar o conjunto de depósitos a abrir e o esquema de rotas a ser coberto pela frota de veículos associada a um depósito, de modo a minimizar os custos da operação, sem esquecer que toda a procura dos clientes tem de ser satisfeita.

No que resta deste documento, far-se-á a explicitação da implementação do algoritmo genético, terminando na apresentação dos resultados e averiguação de conclusões.

## 2 Algoritmo genético

Introduzido o problema, antes de se passar aos detalhes da abordagem escolhida relembram-se os conceitos base por detrás dos algoritmos genéticos. Este tipo de algoritmos, baseados em mecanismos de seleção natural, começam com uma população de soluções, geralmente aleatórias, e vai evoluindo através da aplicação de operações genéticas aos indivíduos, a cada iteração, entre as quais a seleção, mutação e *crossover*.

### 2.1 Representação do cromossoma

Um ponto fundamental quando se utilizam algoritmos genéticos (GA) é a definição da representação de um indivíduo da população, isto é, do cromossoma. Nos GA, um indivíduo da população representa uma solução para o problema. No caso desta abordagem, a representação escolhida para os cromossomas surge na Fig. 1.

Na sua implementação, tem-se, portanto, um *array* de comprimento igual a  $N_{\text{fragmentos}} * N_{\text{genes}} * N_{\text{alelos}}$ , o que, na prática, equivale ao produto entre o número de depósitos e o quadrado do número de clientes. Isto porque, em primeiro lugar, cada fragmento do cromossoma representa um depósito, sendo necessário contabilizar todos eles. Depois, visto que cada depósito tem uma frota ilimitada de veículos, em que cada um, dependendo da procura do cliente, pode visitar um número ilimitado de clientes, é necessário considerar o pior caso em duas situações. Considerando

que é requisitado um veículo por cliente, o pior caso para o número de genes equivale ao número de clientes. O mesmo acontece para o número de alelos, sabendo que o pior caso é um veículo servir todos os clientes.

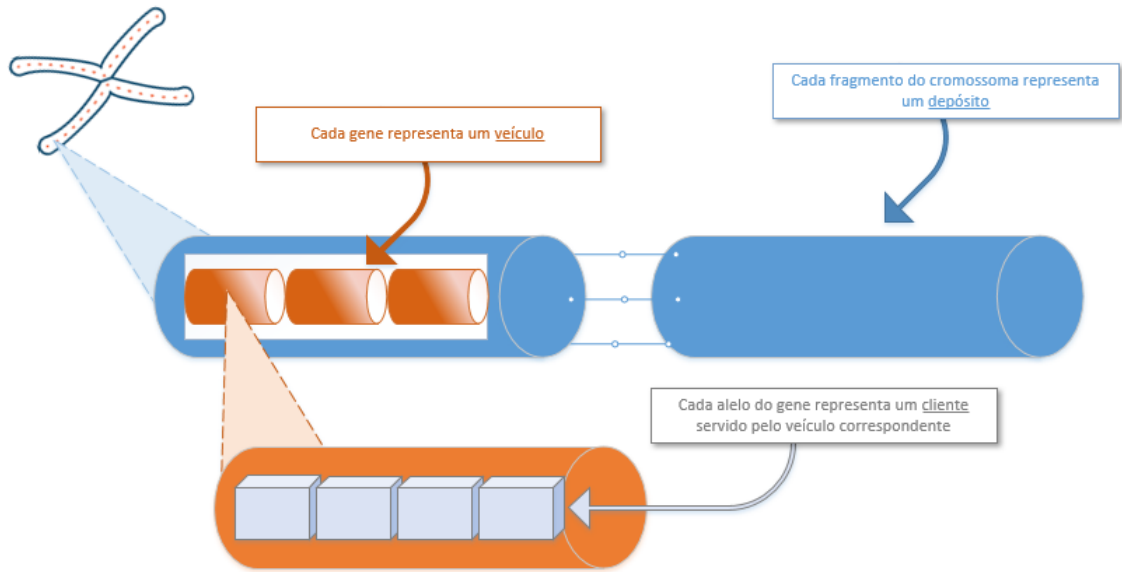


Figura 1: Representação do cromossoma.

## 2.2 Função de fitness

A função fitness, ou função objetivo, é usada para identificar os indivíduos (cromossomas) mais aptos, ou seja, avalia o quão próxima uma dada solução está da solução ótima de um determinado problema.

No problema em questão, o objetivo passa por minimizar o custo, sendo que este custo considera três variáveis: a distância percorrida (cada quilômetro tem o custo de uma unidade), o custo de circulação de um veículo e o custo de abertura de um depósito, ambos diferentes para cada ficheiro. Assim, a função de fitness será representada da seguinte forma:

$$fitness = \frac{1}{distancia + custoVeiculos + custoDeposito}$$

Visto que o objetivo é maximizar a função, é necessário inverter a soma dos custos para que custos maiores (piores soluções) reduzam o valor da função de fitness.

### 2.2.1 Custo da distância

A distância que será percorrida é calculada através dos clientes representados nos alelos. Cada alelo representa um cliente, sendo que, para calcular a distância, basta aceder às duas matrizes que apresentam as distâncias entre clientes e entre clientes e depósitos, somar os respetivos valores e, então, obter a distância percorrida. Inicialmente, os clientes são dispostos segundo uma ordem, cabendo ao algoritmo, fornecendo a função de fitness e as respetivas matrizes, reordenar o cromossoma de modo a maximizar o fitness da solução, maioritariamente minimizando a distância percorrida. Caso seja encontrado um cliente inválido, a distância somada será de 0 unidades. Finalmente, soma-se a distância percorrida ao custo total.

### 2.2.2 Custo da circulação de veículos

Um veículo é representado por um gene, ou seja, para determinar se um veículo entrou em circulação, basta olhar para todos os genes de um cromossoma, mais especificamente para os alelos. Caso esses genes apresentem unicamente clientes inválidos (ou seja, com um número superior ao número de clientes lido do ficheiro), significa que não há qualquer rota a ser satisfeita, logo não

se considera o custo desse veículo. Basta que um gene apresente um cliente válido para a rota ser válida e ser necessário considerar o custo de circulação do veículo. Finalmente, multiplica-se o custo de circulação pelo número de veículos em circulação e soma-se este valor ao custo total.

### 2.2.3 Custo de abertura de um depósito

Um depósito é representado por um fragmento do cromossoma, ou seja, para determinar se um depósito foi aberto basta atentar nos genes do fragmento. Assim, se num depósito existirem veículos nas condições abordadas no ponto anterior, sabe-se que esse depósito foi aberto. Logo, para cada fragmento do cromossoma, verifica-se se existem rotas a serem satisfeitas. Caso tal aconteça, basta somar o custo de abrir o determinado depósito ao custo total.

## 2.3 Treino do Algoritmo Genético

O processo de treino do algoritmo genético depende de vários parâmetros que são definidos na função `ga`. Assim, o primeiro parâmetro explícito é o `type` que toma o valor de `permutation`. Com esta configuração garantiu-se que os clientes são visitados pelos veículos apenas uma vez.

Por outro lado, aos parâmetros `fitness`, `distMatrix` e `distDepot` atribuíram-se, respetivamente, as funções definidas para o cálculo do valor de `fitness`, da matriz de distâncias entre clientes e da matriz de distâncias entre clientes e depósitos.

Além disso, os tamanhos mínimo (`min`) e máximo (`max`) do espaço de procura correspondem a 1 e ao comprimento máximo do cromossoma do problema. O tamanho da população é 30. A probabilidade de mutação de um cromossoma foi fixada em 0.2, enquanto que só podem ocorrer 200 iterações sem que haja uma melhoria do valor de `fitness`. O número de iterações máximo que o algoritmo pode executar foi estabelecido para 1500.

## 3 Resultados

Este algoritmo foi aplicado em três conjuntos de instâncias da literatura, todos eles bem conhecidos em problemas deste tipo. São eles ‘Prins et al.’, ‘Tuzun and Burke’ e ‘Barreto’, sendo que, nos três casos, foram testados vários ficheiros de *input*. Para o primeiro a ser testado, de ‘Prins et al’, o resultado para o ficheiro de *input* inicialmente lido foi o seguinte:

Ficheiro:

`coord20-5-2.dat`

Tempo:

<code>user</code>	<code>system</code>	<code>elapsed</code>
24.63	0.02	25.32

Valores:

```
"SOLUTION 8"
"5 vehicles used!"
"2 depots opened: 2 5"
"The final cost was: 20225.73"
```

Aqui, o tempo foi lido recorrendo à função `proc.time()` do R, importando referir que foi conseguido numa máquina com CPU Intel® Core™ i7 4710HQ Processor, memória de 8 GB, DDR3 MHz SDRAM, incorporada 4 GB e placa gráfica NVIDIA® GeForce® GTX850M 2GB DDR3. Os valores são de uma das soluções de igual valor a que o algoritmo genético chegou, daí a legendagem numerada "SOLUTION 8". O custo final encontra-se arredondado a duas casas decimais. Para todos os ficheiros, este foi o tipo de resultados capturados em ficheiro, um por cada conjunto, os quais se encontram anexados a este relatório.

## 4 Conclusão

Este documento descreveu uma abordagem ao problema de CLRP por meio de um algoritmo genético. Em termos do que foi a implementação do algoritmo, a estratégia utilizada adequou-se

às necessidades na maioria dos ficheiros de *input* utilizados, não deixando de revelar dificuldades nos ficheiros com maior volume de dados. Talvez um aspeto que pudesse conduzir à melhoria deste aspeto seria a inserção de penalizações, adaptadas ao problema, que reduzissem o fitness de soluções aparentemente aptas e encurtassem o leque de indivíduos numa população.

## Referências

- [1] Carlos; Santos Beatriz Sousa Lopes, Rui Borges; Ferreira. *A simple and effective evolutionary algorithm for the capacitated location–routing problem*. Elsevier, 2016.