

# Algoritmos evolucionarios para problemas de *vehicle routing*

André Pimenta, Cedric Pimenta, and Rafael Abreu

Department of Informatics  
University of Minho  
{pg20189,pg19824,pg20978}@alunos.uminho.pt  
www.di.uminho.pt

**Abstract.** Este artigo apresenta a resolução do problema de **CVRP** ( *Capacitated Vehicle Routing Problem* ) utilizando algoritmos evolucionários como metodologia e proposta de resolução.

O problema de **CVRP** ( *Capacitated Vehicle Routing Problem* ) é um problema conhecido da literatura de problemas de optimização, onde se considera um armazém ( centro de distribuição ) que contem produtos para satisfazer necessidades de clientes, entregando os produtos a estes com a utilização de veículos com capacidade limitada.

O objectivo passa por encontrar uma solução que minimize o custo total de transporte, optimizando a utilização dos veículos tendo em conta as respectivas limitações.

Neste estudo iremos usar representações baseadas em permutações, para posteriormente através de algoritmos evolucionarios e operadores inteligentes resolver o problema.

**Keywords:** Capacitated Vehicle Routing Problem, algoritmos evolucionarios, JEColi, operadores, evolução.

## 1 Introdução

*The Capacitated Vehicle Routing Problem (CVRP)* é o um problema de optimização, que pretende criar um conjunto de rotas específicas para um conjunto de veículos de forma a satisfazer as necessidades dos clientes, tendo como objectivo principal minimizar os custo de transporte satisfazendo todas as entregas. Este é um problema muito comum na vida real, e que é uma especificação do *Vehicle Routing Problem (VRP)* , que se encontra normalmente associado a diferentes limitações que vão desde a disponibilidade de motoristas, numero de veículos, entre outros que não serão tido em consideração este estudo. Apenas a capacidade de transporte e o numero de veículos disponíveis serão tidos em conta.

Este é também um problema de complexidade **NP**, e tal como a maior parte dos problemas que fazem parte deste mesmo conjunto de problemas são feitas tipicamente três abordagens clássicas para a resolução do problema: **heurísticas**,

**métodos de aproximação e métodos exactos.** No entanto e devido à natureza do problema não é viável utilizar estas mesmas metodologias para grandes instâncias do problema

A aplicação de algoritmos evolucionários é uma das técnicas usadas para a resolução deste problema, e que será a usada neste estudo, tentando desta forma resolver o problema com ajuda de operadores de melhoria de soluções baseados em algumas heurísticas que serão apresentadas neste artigo.

Neste artigo iremos estudar e apresentar uma solução baseada na utilização de algoritmos evolucionários, mais concretamente algoritmos genéticos para a resolução do problema, sendo o problema representado através de permutações, serão também apresentados resultados obtidos pela utilização desta mesma solução, assim como provenientes dos diferentes operadores utilizados para melhoria de solução.

O artigo segue a seguinte estrutura: secção 2 onde se apresentam as metodologias para a resolução do problema, secção 3 onde são apresentados resultados obtidos e por fim uma breve conclusão.

### 1.1 *Vehicle Routing Problem (VRP)*

*Vehicle Routing Problem (VRP)* é um problema de optimização combinatória que pretende satisfazer as necessidades de um determinado numero de clientes. Proposto por Dantzig and Ramser em 1959 **VRP** é um problema de referencia na área de transportes, de distribuição e logística.

Este problema é caracterizado pela existência de um armazém central que tem como funcionalidade a satisfação das necessidades dos clientes tentando minimizar os custos de transporte, tendo em conta as possíveis restrições, como a do caso de estudo ( **CRVP** ) que é a limitação da capacidade de transporte por parte das suas viaturas.

### 1.2 *Travelling salesman problem (TSP)*

*Travelling salesman problem (TSP)* é a base do problema que pretendemos resolver. É também um problema de optimização que tem suscitado muito interesse por parte de investigadores e que tem levado a muitos estudos em volta desta problemática.

O problema de **TSP** faz parte de um conjunto de problemas de complexidade **NP**, onde se pretende obter a rota mais curta possível que visite um conjunto de cidades apenas uma vez terminado a mesma rota no ponto de partida.

### 1.3 **Algoritmos evolucionários**

Como já referido neste artigo, a utilização de algoritmos evolucionários para a resolução do problema **CVRP** vai ser o principal enfoque de análise.

Os algoritmos evolutivos baseiam-se num conjunto de mecanismos de evolução biológica, que são utilizado por norma para a resolução de problemas de optimização.

Inspirando-se nos conceitos biológicos de **reprodução, mutação, recombinação e selecção**, e aplicando estes mesmo conceitos num conjunto de solução inicial que é vista como uma população vai-se atingido um estado de evolução natural da população, e respectivamente da solução.

**Seleccção:** primeiro, selecciona-se uma parte da população existente para produzir uma nova geração. Esta selecção é feita utilizando uma abordagem baseada na capacidade ( valor de aptidão ) de cada indivíduo, ou seja, os mais aptos têm maior probabilidade de se reproduzir do que os restantes.

**Reprodução:** durante a fase de reprodução é gerada a nova geração, que é criada usando dois métodos, cruzamento e mutação. Para cada novo descendente são seleccionadas características dos progenitores. No processo de cruzamento do génotipo são seleccionadas características de ambos os progenitores para criar o descendente.

A possibilidade de o descendente ser exposto a uma mutação de alguns genes pode ajudar futuramente a expandir o espaço de soluções ou preservar a divergência genética. No entanto a ocorrência de mutações está geralmente associada a baixas probabilidades. Pode-se então afirmar que um equilíbrio entre a diversidade e qualidade genética é o cenário ideal.

**Implementação:** a implementação será feita na linguagem **JAVA** . Esta tem como grande vantagem a aproximação ao paradigma orientado a objectos, assim como permite a utilização da plataforma JEColi\*[16].

## 2 Algoritmos evolucionários para o problema CVRP

### 2.1 Representação genética dos veículos

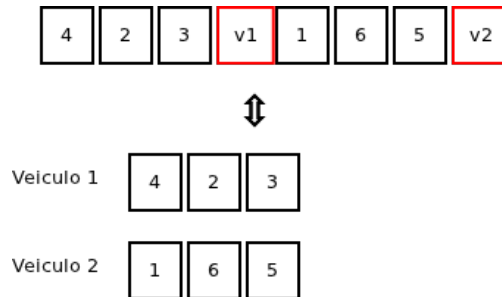
A representação genética dos veículos no problema é o primeiro passo a tomar. Esta representação deverá ser capaz de especificar o numero de veículos necessários assim como as rotas de cada um.

Para a representação do problema iremos usar então permutações que têm como principal característica a representação através de uma lista que não possui elementos repetidos e que permite um múltiplo numero de arranjos possíveis.

Pretendemos através do uso de permutações para a representação do problema conseguir representar um numero fixo de veículos ( **representação directa** ), mas também representar apenas um numero mínimo de veículos ( **representação indirecta** ), pois desta forma será possível resolver dois tipos de problemas, um para o caso de numero fixo de veículos ( **representação directa** ) e outro para o um numero mínimo de veículos ( **representação indirecta** ).

**Representação directa** A representação feita para um determinado numero fixo de veículos. Usando uma representação através de permutações é dada por uma solução de tamanho **numero de veículos + numero de clientes**, onde

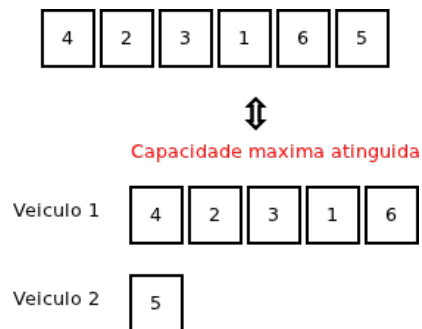
uma rota se encontra definida na solução pela presença do veículo respectivo. Tem como principal objectivo atingir soluções para numero fixo de veículos, o que pode levar consequentemente a soluções invalidas.



**Fig. 1.** Representação directa

**Representação indirecta** A representação indirecta tem como principal objectivo permitir atingir soluções para um numero mínimo de veículos, ou seja , caso se pretenda conhecer um numero mínimo de veículos para resolver determinado problema.

A solução é então definida pelo **numero de clientes**, sendo posteriormente definida a rota retirando os diferentes destinos até atingir a capacidade máxima do veículo. Este tipo de solução origina sempre soluções válidas



**Fig. 2.** Representação indirecta

## 2.2 População inicial e evolução

Como referido cada individuo da população vai ser representado através de permutações, tanto de forma directa como indirecta, que implica uma lista inicial com sem repetidos com o tamanho definido para cada representação. Esta mesma lista irá ser gerada aleatoriamente, criando assim os primeiros genomas que se irão sujeitar à evolução.

Parte-se então de uma solução inicial totalmente aleatória, sendo depois esta exposta aos diferentes processos de *selecção natural* que irão originar as soluções mais aptas ao problema em questão. Este mesmo processo de *selecção natural* é definido em grande parte por uma função avaliação que avalia as capacidades de cada solução, assim como pela influencia de operadores inteligentes que procuram otimizar de forma não aleatória as soluções, ou seja os genes dos diferentes indivíduos de uma população.

## 2.3 Função avaliação e valor aptidão

Para seleccionar os indivíduos mais aptos de cada geração é necessário definir uma função avaliação que irá avaliar o *valor aptidão* de cada individuo e consequentemente permitir definir a sua sobrevivência dentro da população ( ou seja medir a sua adaptabilidade ). Os indivíduos com um valor aptidão superior serão os mais indicados para a resolução do problema e para fazer parte da população. Como já apresentado, definimos dois tipos de representação ( directa e indirecta ) que implica avaliações diferentes, pois no caso da representação directa temos o problema de penalizações. Como se parte de um numero fixo de veículos a solução terá de ser primeiramente valida, enquanto numa representação indirecta nunca temos o problema de soluções invalidas sendo apenas julgado o custo total das entregas.

A avaliação de uma representação directa terá como factores principais o excesso de carga total ( penalização) e o custo total, sendo aplicado uma penalização ao excesso de carga. Já a representação directa tem como função avaliação a minimização do custo total.

O calculo do *valor aptidão* de cada individuo vai ajudar a seleccionar os membros da próxima geração.

## 2.4 Operadores genéticos

Assumindo uma população inicial de tamanho  $N$ , em cada geração são escolhidos  $N$  parentes, e  $N$  descendentes são obtidos após a aplicação dos operadores genéticos ao conjunto da população seleccionada.

Serão considerados dois tipos de operadores: *crossover* e *mutation*, sendo que estes devem estar adaptados aos dois tipos de representação utilizadas, permitindo alterar rotas assim como as cargas a entregar pelos diferentes veículos. No caso da representação directa têm sempre em conta gerar soluções válidas, visto ser utilizado um numero fixo de veículos que pode conduzir a essa situação

Os operadores de **crossover** utilizados não utilizam a troca de material genético entre dois progenitores, mas sim através de um progenitor, que poderá ser chamado de **doador**.

Como alternativa ao cruzamento de dois desenvolvemos então operadores "inteligentes" que se baseiam em determinadas heurísticas e que apenas resultam descendentes com melhor aptidão.

Descendentes resultantes da passagem podem ser sujeitos a mutação. Consideramos os seguintes operadores:

- **Inversão** selecciona uma sub-rotas e inverte a ordem visita dos clientes da mesma.
- **Swap** Escolha de dois clientes para trocar, podendo pertencer à mesma rota ou não.
- **Inserção realocação** Escolhe-se uma entrega a remover de
- **Cross move** Troca de parcelas de sub-rotas entre veículos.

Estes operadores apenas resultam em soluções com valores de aptidão superiores, sendo cada um deles aplicada uma probabilidade individual.

### 3 Exemplos de soluções e análises de resultados

Nesta secção iremos apresentar alguns resultados comparativos das diferentes representações e operadores utilizados, tentando dar uma visão do trabalho realizado, mas ao mesmo tempo poder comparar e tirar conclusões dos resultados obtidos.

#### 3.1 Representação directa vs indirecta

Submeteu-se o mesmo problema para resolução usando as duas representações ( directa e indirecta ). Este problema ( **prob1** ) é constituído por 100 clientes, e como numero de veículos para a representação directa de um total de 10.

Apresentam-se de seguida os resultados finais após 2000 mil iterações sobre uma população de 200, onde se apresentam o custo de transporte a penalização de transporte, e as rotas definidas para cada veículo.

##### Representação directa

```
Veiculo:0 0 95 75 85 84 83 82 72 25 16 17 37 58 68.
Veiculo:1 0 89 79 48 46 33 21 11 2 4 5 18 28 38 96.
Veiculo:2 0 23 12 1 31 41 51 42 43 19 20 30 40.
Veiculo:3 0 99 100 90 80 70 69 78 47 57 67.
Veiculo:4 0 92 81 71 62 53 54 64 73 93.
Veiculo:5 0 94 74 44 27 8 29 39 49 59.
```

```

Veiculo:6 0 98 88 77 56 13 3 24 87 97.
Veiculo:7 0 86 76 66 65 55 45 35 26 36.
Veiculo:8 0 34 14 15 6 7 9 10 50 60.
Veiculo:9 0 91 61 52 32 22 63.
CUSTO: 1860.0
PENAL: 0.0

```

### Representação indirecta

```

Veiculo:0 0 95 85 75 65 66 76 77 87 97.
Veiculo:1 0 96 86 55 45 54 64 74 83 93.
Veiculo:2 0 38 28 18 9 10 20 30 40 50.
Veiculo:3 0 78 68 58 49 60 70 80 90 99.
Veiculo:4 0 44 34 24 14 4 5 15 25 35.
Veiculo:5 0 43 22 12 11 1 2 3 13 23.
Veiculo:6 0 67 57 48 59 69 79 89 88 98.
Veiculo:7 0 82 72 62 52 53 63 73 84 94.
Veiculo:8 0 56 46 36 26 16 6 33 81 91.
Veiculo:9 0 47 37 27 17 7 8 19 29 39 42 92.
Veiculo:10 0 32 21 31 41 51 61 71 .
CUSTO: 1660.0
PENAL: 0.0

```

Como se pode analisar, através da representação indirecta obtemos melhor resultado sobre o custo total de viagem pois utilizou-se mais um veiculo que permitiu uma escolha rotas mais baratas, no entanto este acaba por exceder o numero de veículos iniciais, o que pode levar a ponderar a aquisição de mais um veiculo caso futuramente fosse mais vantajoso.

Pode-se analisar também que conseguiu-se uma solução valida para a representação directa, o que significa que o numero de veiculos (**N=10**) é suficiente para resolver o problema.

## 3.2 Operadores

Para analise dos diferentes operadores submeteram-se os mesmo operadores ao problema 2 (**prob2**) que tem 110 clientes e 10 veículos ( usada a representação directa ) juntamente com o operador **PermutationOnePtCrossover** presente na *JEColi* de forma a obter resultados em condições idênticas para todos os operadores por nós desenvolvidos.

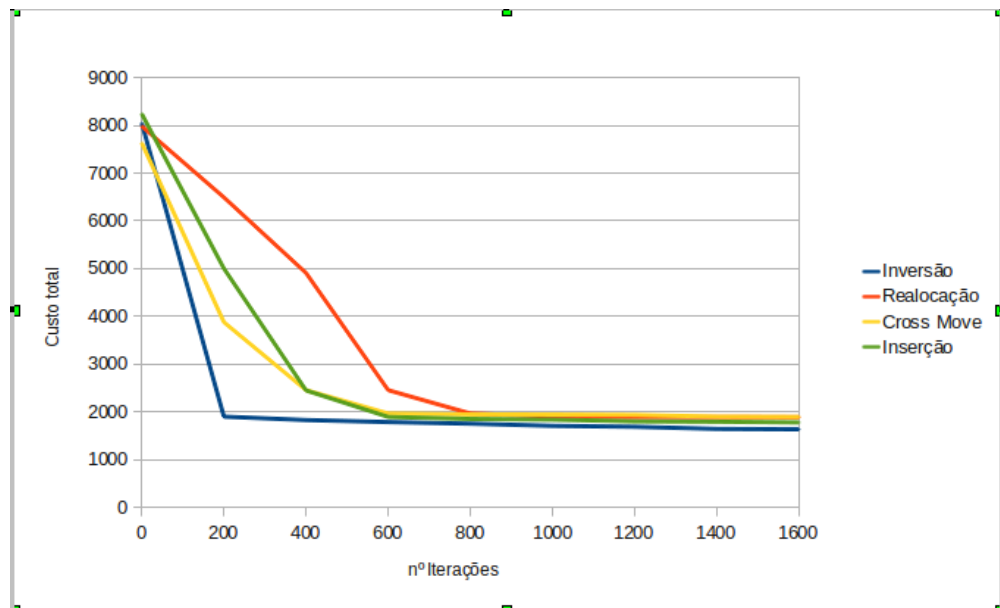
Apresentam-se de seguida resultados obtidos para os diferentes operadores:

Como se pode ver através do gráfico e tabela anterior parte-se de soluções iniciais diferentes, devido ao facto de a solução inicial ser gerada aleatoriamente, no entanto relativamente próximas para obter resultados mais confiáveis.

No entanto podemos constatar que ambos os operadores convergem para soluções próximas, e que não se encontram grandes diferenças abruptas de solução, no

n° Iteracção op	Inversão op	Realocação op	Cross Move op	Inserção
0	8045	7980	7629	8239
200	1902	6492	3879	5002
400	1830	4908	2456	2450
600	1789	2456	1967	1896
800	1755	1967	1950	1854
1000	1709	1904	1946	1840
1200	1689	1894	1938	1804
1400	1645	1894	1902	1795
1600	1637	1890	1897	1780

**Table 1.** Custo total para os diferentes operadores



**Fig. 3.** Grafico da tabela 1.

entanto o operador *inversão* consegue obter resultados mais ligeiramente superiores, assim como atingir o resultado final mais rapidamente com um numero de iterações bem inferior aos restantes. Contrariamente pode-se destacar a realocação que se obtém o pior resultado, assim como demora muito mais a atingir uma solução próxima da final.

Apresenta-se de seguida uma análise ao processamento:

Através da tabela anterior podemos analisar o numero da iteração em que se obteve uma solução já próxima da final, assim como o tempo total necessário de processamento, como o resultado final obtido.

Estes resultados foram obtidos do **prob2** utilizado nos resultados anteriores.



Operador	Nº Iteração	tempo (mls)	Custo Toal
<b>Inversão</b>	200	12877	1720
<b>Realocação</b>	780	520321	1870
<b>Cross Move</b>	600	59500	1896
<b>Inserção</b>	500	12877	1872

**Table 2.** Custo total para os diferentes operadores

Mais uma vez podemos ver que o operador realocação é muito menos eficaz e lento que os restantes.

## 4 Conclusão

Neste artigo foi apresentada uma solução para o problema **CVRP** através do uso de algoritmos evolucionários, mas em concreto através do uso de algoritmos genéticos.

Sendo um problema com restrições de capacidade de cada veículo, o numero de veículos é um factor importante no problema e foram então desenvolvidas duas representações para um numero fixo e outra para um numero mínimo, e tal como apresentado na secção de resultados pode-se concluir que a utilização de um numero mínimo de veículos origina soluções mais baratas, e sem qualquer tipo de penalização, no entanto o acréscimo do numero de veículos não está a ser contabilizado.

Focou-se também um pouco a atenção deste artigo nos operadores desenvolvidos através de heurísticas de melhoria, onde podemos constatar primeiramente que a aproximação à solução final é muito mais rápido nos operadores "inteligentes" desenvolvidos, uma vez que estes geram operações de melhoria apenas, ou seja individuos com valor de aptidão superiores ao seu progenitor. Estas heurísticas permitem então obter soluções mais rápidas, o que se pode afirmar que são muito mais vantajosas para este problema.

Fazendo uma referencia à **JEColi** utilizada no desenvolvimento de todo o sistema, esta pode ser considerada muito útil, pois permite um desenvolvimento rápido e facilitado de qualquer tipo de algoritmo evolucionário, e neste caso algoritmos genéticos, sendo que dispõe se de diversos tipos de representações já implementados.

Como trabalho futuro pretende-se fazer uma analise mais detalhada aos operadores utilizados, assim com a importância e o seu papel específico no ceio de todo o processo de um algoritmo evolutivo.

## References

1. Christian Prins.: *A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem*.  
MIC'2001 - 4th Metaheuristics International Conference.

2. Miura C., Cunha C.: *MODELAGEM HEURÍSTICA NO PROBLEMA DE DISTRIBUIÇÃO DE CARGAS FRACIONADAS DE CIMENTO.*
3. Alba E., Dorronsoro B.: *Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms.*
4. R. AYADI, Y. BENADADA.: *A Genetic Algorithm for a heterogeneous fleet Multi-trip Vehicle Routing Problem.*
5. Irhamah ,Ismail Z.: *A Breeder Genetic Algorithm For Vehicle Routing Problem with Stochastic Demands.*  
Journal of Applied Sciences Research, 5(11): 1998-2005, 2009
6. Cai Z.,Feng Y.,Gan Y.,Zhang B.,Liu S.: *Research on Logistics Vehicle Routing Problem Based on Genetic Algorithm.*
7. khoshtakht M.,Sedighpour M.: *An Optimization Algorithm for the Capacitated Vehicle Routing Problem Based on AntColony System.*  
Australian Journal of Basic and Applied Sciences, 5(12): 2729-2737, 2011
8. Lin Cheng.: *A GENETIC ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS.*
9. Delbem A, Gabriel P.: *Fundamentos de Algoritmos Evolutivos.*
10. OMBUKI B.,HANSHAR B.: *Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows.*  
2006 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
11. OMBUKI B.,HANSHAR B.: *Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows.*  
2006 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
12. Masum A.,Shahjalal M, Faruque M, Sarker M.: *Solving the Vehicle Routing Problem using Genetic Algorithm.*  
(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 7, 2011
13. Tavares G., Macahdo P., Pereira F. Costa E.: *On the Influence of GVR in Vehicle Routing.*
14. Rocha M.: *Optimização de Modelos Conexionalistas de aprendizagem via computação genética e evolucionária.*
15. Rocha M.: *Optimização de Modelos Conexionalistas de aprendizagem via computação genética e evolucionária.*
16. Evangelista P., Maia P., Rocha M.: *Implementing metaheuristic optimization algorithms with JECOLi.*  
2009 Ninth International Conference on Intelligent Systems Design and Applications