Uma abordagem para o problema da optimização de rotas de veículos baseada em operadores genéticos

G. Schütz * F. M. Pires †

* Escola Superior de Tecnologia – Universidade do Algarve

† Faculdade de Ciências e Tecnologia – Universidade do Algarve
mpires@ualg.pt

Abstract

In its basic form the vehicle routing problem (VRP) consists of serving a set of clients with known locations and demands by a fixed fleet of vehicles of limited capacity, minimizing the total route length. All vehicles are located at a common depot with known location.

This is a NP-hard problem. The few methods developed to solve the problem exactly are very time consuming and they are only feasible for small dimension problems. So, the most efficient approaches are based on heuristic algorithms. More recently some meta-heuristics, mainly based on Tabu search, have been developed to the VRP.

In this work we present a Genetic algorithm to the basic version of the VRP. We describe the way we have adjusted the genetic operators to the problem. We report and analyse the results of a computational study with a set of known test problems.

Resumo

A versão básica do Problema da Optimização de Rotas de Veículos consiste em determinar um conjunto de rotas a efectuar por uma frota de veículos, estacionada num depósito central, para servir um conjunto fixo de clientes, minimizando a distância total percorrida. O número de veículos e respectivas capacidades, as localizações e procuras dos clientes, bem como a localização do depósito são conhecidas.

Este problema pertence à classe dos problemas *NP*-difíceis. Os poucos métodos exactos que foram desenvolvidos para o resolver são demasiado demorados e apenas exequíveis para a generalidade dos problemas de pequena dimensão (até cerca de 50 clientes). Assim, as abordagens mais comuns e eficientes baseiam-se em métodos heurísticos. Mais recentemente várias meta-heurísticas, principalmente baseadas em Pesquisa Tabu, têm sido desenvolvidas para o Problema da Optimização de Rotas de Veículos.

Neste trabalho expõe-se um Algoritmo Genético para a versão básica do Problema de Optimização de Rotas de Veículos. Apresenta-se a forma como se adaptaram os operadores genéticos a este problema. Descreve-se o estudo computacional realizado, com um conjunto de problemas-teste conhecidos da literatura, e analisam-se os resultados obtidos.

Apdio © 2003 *Associação Portuguesa de Investigação Operacional*

Keywords: Vehicle Routing Problem, Heuristics, Genetic Algorithms

Title: A Genetic Based Approach to the Vehicle Routing Problem

---

1 Introdução

A versão básica do Problema de Optimização de Rotas de Veículos (PORV), genericamente conhecido na literatura de língua inglesa por "Vehicle Routing Problem", consiste em determinar um conjunto de rotas a efectuar por uma frota de veículos, a partir de um depósito central, para servir um conjunto de clientes. O número de veículos e as suas capacidades, as localizações e procuras dos clientes, bem como a localização do depósito são conhecidas. Pretende-se, então, determinar quais os clientes que serão servidos por cada veículo e qual a configuração das rotas a efectuar pelos veículos de forma a minimizar a distância total percorrida.

O PORV é um Problema de Optimização Combinatória pertencente à classe dos problemas *NP*–difíceis [12]. Os algoritmos exactos raramente podem ser usados para resolver, em tempo útil, instâncias com mais de 50 clientes, mesmo para instâncias menores (com menos de 50 clientes) estes métodos são lentos [4], [9]. Nos poucos casos em que instâncias com 100 clientes foram resolvidas exactamente, é referido que ou não foi necessário recorrer à Pesquisa em Árvore, ou o número de vértices da árvore que foi necessário pesquisar era muito pequeno [4].

Assim, a maior parte da investigação sobre o PORV tem-se dirigido a métodos heurísticos. Mais recentemente, várias meta-heurísticas e algoritmos híbridos têm sido desenvolvidos para o PORV. Podem-se citar, entre outros, o algoritmo híbrido [13] que combina Arrefecimento Simulado e Pesquisa Tabu, o algoritmo baseado num Sistema de Colónia de Formigas [1] e os trabalhos baseados em algoritmos de Pesquisa Tabu [6], [11], [15] e [18].

Neste trabalho, descreve-se uma heurística baseada em operadores genéticos e os resultados computacionais obtidos num conjunto de problemas-teste conhecidos da literatura. Apresenta-se, na secção 2, uma formalização matemática para o PORV. Na secção 3, descreve-se o Algoritmo Genético desenvolvido e implementado. Apresenta-se, na secção 4, o estudo computacional realizado, os resultados obtidos e respectiva análise. Finaliza-se com algumas conclusões e comentários, na secção 5.

2 Formalização

Apresenta-se uma formalização do PORV como um problema de Programação Não Linear com variáveis binárias [5], por ser a que está subjacente ao algoritmo desenvolvido neste trabalho.

Considerem-se as constantes:

n – o número de clientes

(o depósito corresponde ao índice 0)

m – o número de veículos

c_{ij} – a distância entre i e j , i , j $=$ $0, ..., n$

q_i – a procura do cliente i , i $=$ $1, ..., n$

Q – a capacidade de cada veículo

e as variáveis:

$$x_{ijk} = \begin{cases} 1\,, \text{ se o veículo } k \text{ vai de } i \text{ para } j \text{ directamente} \\ 0, \text{ caso contrário} \end{cases}$$

$$y_{ik} = \begin{cases} 1, \text{ se o cliente } i \text{ é servido pelo veículo } k \\ 0, \text{ caso contrário} \end{cases}$$

Seja $y_k = (y_{0k}, y_{1k}, \dots, y_{nk})$

então tem-se:

$$min \sum_k f\,(y_k) \tag{1}$$

s.a

$$\sum_k y_{ik} = \begin{cases} 1 & , i = 1, ..., n \\ m & , i = 0 \end{cases} \tag{2}$$

$$\sum_i q_i y_{ik} \leq Q \quad , k = 1, ..., m \tag{3}$$

$$y_{ik} \in \{0, 1\} \quad , i = 0, ..., n\,;\; k = 1, ..., m \tag{4}$$

em que para cada k , k $=$ $1, ..., m$

$$f\,(y_k) = min \sum_{i,j} c_{i\,j}\, x_{i\,j\,k} \tag{5}$$

s.a

$$\sum_j x_{ijk} = y_{ik} \quad , i = 0, ..., n \tag{6}$$

$$\sum_i x_{ijk} = y_{jk} \quad , j = 0, ..., n \tag{7}$$

$$\sum_{i,j \in S \mathrm{x} S} x_{ijk} \leq |S| - 1\,, \text{para todo o } S \subseteq \{1, ..., n\} \text{ tal que } 2 \leq |S| \leq n-1 \tag{8}$$

$$x_{ijk} \in \{0, 1\} \quad , i, j = 0, ..., n \tag{9}$$

O objectivo (1) é minimizar uma soma de funções não lineares, $f\,(y_k)$. O valor de $f\,(y_k)$, para cada k, k $=$ $1, ..., m$, é obtido resolvendo um subproblema cuja função objectivo (5) minimiza a distância percorrida por cada veículo, para servir um determinado subconjunto de clientes. As restrições (2) garantem que os clientes estão afectos a um veículo e que o depósito é servido por todos os veículos. As condições (3) não permitem que as capacidades dos veículos sejam excedidas. As restrições (4) definem as variáveis como binárias. Além disso, em cada subproblema as restrições (6) e (7) obrigam a que um veículo que chega a um cliente também parta desse cliente e a que cada cliente seja visitado por exactamente um veículo. As restrições

(8) garantem que não existem subcircuitos que não contenham o depósito. As condições (9) definem as variáveis do subproblema como binárias.

Esta formalização engloba dois Problemas de Optimização Combinatória. Por um lado, as restrições (2), (3) e (4) têm a estrutura de um Problema de Afectação Generalizada, onde é determinada a afectação dos clientes aos veículos. Por outro lado, para as variáveis y_{ik} fixas satisfazendo (2), (3) e (4) e para cada veículo k, cada parcela$f$ (y_k) da função objectivo (1), dada por (5) a (9), define um Problema do Caixeiro Viajante (PCV), onde é estabelecida a configuração de cada rota. Assim, nesta formalização o PORV é definido como um Problema de Afectação Generalizada Não Linear.

3 Uma Implementação de um Algoritmo Genético

Os Algoritmos Genéticos (AGs) foram inicialmente desenvolvidos para resolver problemas da área de Inteligência Artificial [10]. Baseiam-se fortemente em processos aleatórios e em probabilidades. Como é do conhecimento geral, este método utiliza um conjunto de soluções (*população*) que, em cada iteração (*geração*), são combinadas (*reprodução*) originando uma nova população. Assim, algumas cópias dos indíviduos da população actual são escolhidas (*selecção*) para, por aplicação de operadores genéticos (*cruzamento* e *mutação*), produzirem outros indivíduos que vão fazer parte, alguns ou todos, de uma nova população. Estes aspectos envolvidos nos AGs são descritos detalhadamente em [7].

Codificação dos Indivíduos

Apesar de os AGs terem surgido baseados numa codificação binária dos indivíduos [7], [10], tem-se mostrado necessária a utilização de outro tipo de codificação, nomeadamente em alguns Problemas de Optimização Combinatória [14].

Para o PORV a codificação binária não parece adequada. Uma alternativa natural seria usar abordagens análogas às que têm sido utilizadas para o PCV [14]. No entanto, no PCV tem-se apenas uma rota e no PORV tratam-se $m$rotas, com número de clientes por rota e subconjunto de clientes em cada rota diferentes de uma solução para outra. Assim, considerou-se que as representações já utilizadas para o PCV, também não seriam muito adequadas para o PORV.

Então, optou-se por codificar cada indivíduo x^i , i = 1,...,Pop, por um vector com n variáveis,x^i_j, j = 1,...,n, onde n é o número de clientes do problema, podendo cada variávelx^i_j tomar um valor de 1 a m, onde m é o número de veículos, previamente definido.

Neste caso, em vez de um indivíduo representar a configuração das rotas, representa as afectações dos clientes aos veículos. De acordo com a formalização dada anteriormente tem-se:

$$y_{0k} = 1 \text{ e } y_{jk} = \begin{cases} 1 & \text{se } x^i_j = k \\ 0 & \text{caso contrário} \end{cases} , \; j = 1, ..., n; \; k = 1, ..., m \tag{10}$$

Por exemplo, o indivíduo [1 1 3 2 2 2 2 3 1 2]

indica que:

os clientes 1, 2, 9 são servidos pelo veículo 1,

os clientes 4, 5, 6, 7, 10 são servidos pelo veículo 2,

os clientes 3, 8 são servidos pelo veículo 3.

e, de acordo com (10), tem-se:

$$\begin{array}{l} y_{11} = 1\,,\ y_{21} = 1\,,\ y_{91} = 1\,, \\ y_{42} = 1\,,\ y_{52} = 1\,,\ y_{62} = 1\,,\ y_{72} = 1\,,\ y_{10\,2} = 1\,, \\ y_{33} = 1\,,\ y_{83} = 1\,, \end{array}$$

sendo as restantes variáveis y_{ij} iguais a zero. Esta afectação dos clientes aos veículos é uma solução admissível do Problema de Afectação Generalizada se verificar as restrições (2), (3) e (4).

Represente-se por $f\ (x^i)$ o valor da função objectivo (1), para cada indivíduo x^i. Este valor é igual à distância total percorrida pelos m veículos para servirem os clientes de acordo com a afectação definida por esse indivíduo e verificando as restrições (6) a (9). Para cada veículo k, $k = 1, ..., m$, e respectiva afectação $y_k = (y_{0k}, ..., y_{nk})$ obtida segundo (10), a rota a efectuar (valores de x_{ijk}) e a distância a percorrer (valor de $f\ (y_k)$) são determinadas através da resolução de um PCV. Assim vem:

$$f\left(x^i\right) = \sum_{k=1}^{m} f\ (y_k)\ ,\ i = 1, ..., Pop \tag{11}$$

Nesta implementação, os PCV foram resolvidos heuristicamente, devido ao facto do PCV ser também um problema pertencente à classe dos problemas *NP*–difíceis. Utilizou-se o Método da Inserção Mais Afastada, por ser computacionalmente muito rápido e se encontrarem bons resultados referidos na literatura [8], [17].

População Inicial

Em geral, nos AGs a população inicial é gerada aleatoriamente, o que, neste caso, produziria soluções não admissíveis. Assim, optou-se por gerar a população inicial heuristicamente, através de um algoritmo do tipo construtivo de rota descrito em [16]. As m rotas são formadas paralelamente e os clientes são inseridos nas rotas segundo uma abordagem ávida. O algoritmo é inicializado escolhendo m clientes e formando m rotas do tipo depósito–cliente–depósito. Seguidamente, em cada iteração, insere-se em cada rota o cliente que não viola as restrições e minimiza o aumento da distância percorrida pelo veículo para o inserir entre os clientes que já estão nessa rota.

Depois executam-se, para cada indivíduo, procedimentos melhorativos, descritos em [16], que consistem na mudança de clientes de veículos com a sua capacidade quase ou completamente preenchida (mais de 95%) para outros com mais capacidade livre (menos de 90%), e na troca de 1 ou 2 clientes entre rotas, desde que a distância total percorrida seja menor.

Em geral, ao mudar os clientes iniciais de cada rota, as afectações subsequentes são diferentes levando a outra solução. Desta forma, variando a escolha dos m primeiros clientes a serem inseridos, pode-se gerar os *Pop* indivíduos para formar a população inicial. Para 3 dos indivíduos escolheram-se os clientes iniciais com base na localização e/ou procura dos clientes. Para os restantes *Pop* - 3 indivíduos, geraram-se aleatoriamente os clientes iniciais.

Tabela 1: Aptidão, roleta, ponteiro e indivíduos seleccionados para uma população de 10 indivíduos (selecção SUS)

Indiv.	1	2	3	4	5	6	7	8	9	10
f	6	4	2	5	4	5	8	12	7	10
Apt	0,89	1,78	2,00	1,33	1,56	1,11	0,44	0,00	0,67	0,22
roleta	0,89	2,67	4,67	6,00	7,56	8,67	9,11	9,11	9,78	10,00

pont.	0,7	1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7

selec.	1	2	3	3	4	4	5	6	7	9

Função de Aptidão e Selecção

As formas de seleccionar a população baseiam-se na aptidão dos indivíduos (valor da função objectivo, classificação dos indivíduos, etc.). Utilizou-se a selecção SUS ("Stochastic Universal Sampling"), da seguinte forma:

Ordenaram-se os indivíduos pelo valor da função objectivo (dada em (11)). Atribuiu-se uma aptidão 0 ao pior indivíduo e uma aptidão 2 ao melhor. Aos outros indivíduos foram atribuídas aptidões entre 0 e 2 de acordo com a sua classificação e igualmente espaçadas.

Assim, sendo

perm (i) o vector que indica a classificação (de 0 a Pop-1) do indivíduo i na população;

e const $= 2/Pop - 1$.

a função de aptidão é dada por:

$$Apt\left(x^i\right) = 2 - perm\,(i) \times const, i = 1, ..., Pop \tag{12}$$

A partir dos valores da aptidão construiu-se uma roleta como se exemplifica na tabela 1. Seguidamente, gerou-se um número aleatório uniformemente distribuído no intervalo [0, 1] e, adicionando sucessivamente uma unidade a esse número, formou-se um ponteiro múltiplo igualmente espaçado que permite identificar os indivíduos a seleccionar, como exemplificado na tabela 1.

Cruzamento

Agruparam-se, aleatoriamente, os indivíduos seleccionados em pares (*pais*), de modo que não haja pares com elementos iguais (se, em algum caso, a população convergir e deixarem de existir indivíduos diferentes o algoritmo termina). Obtiveram-se dois novos indivíduos (*filhos*) trocando, no máximo, uma certa percentagem dos valores das variáveis que ocupam as mesmas posições nos vectores pais, correspondentes aos números aleatórios gerados uniformemente no intervalo $[1, n]$. Esta troca só se efectuou quando a solução se mantinha admissível. Por exemplo, sejam:

Pai 1	[	**1**	**3**	**1**	**4**	**3**	**1**	**2**	**4**	**4**	**2**	**4**	**2**]
Pai 2	[	*2*	*2*	*3*	*3*	*2*	*3*	*3*	*4*	*4*	*1*	*4*	*1*]

e o número máximo de variáveis a trocar igual a 4. Então, geram-se 4 números aleatórios diferentes no intervalo [1, 12], (por exemplo 5, 7, 1, 8). Se a capacidade dos veículos permitir as duas primeiras trocas e não permitir a terceira, obtêm-se:

Filho 1	[	**1**	**3**	**1**	**4**	*2*	**1**	*3*	**4**	**4**	**2**	**4**	**2**]
Filho 2	[	*2*	*2*	*3*	*3*	**3**	*3*	**2**	*4*	*4*	*1*	*4*	*1*]

Com este tipo de actuação do operador cruzamento obtêm-se afectações, em geral, com agrupamentos de clientes pouco diferentes dos existentes nas afectações que lhes deram origem. No entanto, esta troca aleatória de clientes entre agrupamentos possibilita uma pesquisa diversificada pelo espaço das afectações admissíveis. Como, durante o algoritmo, em cada iteração são executados vários cruzamentos entre afectações distintas é esperado que alguns deles possam conduzir às melhores afectações dos clientes aos veículos.

Após esta operação, dos quatro indivíduos envolvidos (pais e filhos) em cada cruzamento, passam a fazer parte da nova população os dois mais aptos.

Mutação

De acordo com a forma como se definiu o operador cruzamento é esperado que a sua actuação produza uma pesquisa diversificada no espaço das afectações admissíveis dos clientes aos veículos e que a intensifique em torno das melhores afectações. Mas, se por um lado a aleatoriedade da pesquisa deverá, em termos probabilísticos, encontrar boas afectações, por outro lado o facto de não ter em conta a configuração das rotas, pode trazer algumas desvantagens.

Assim, substituiu-se o operador mutação usual por outro, que tem em conta a configuração das rotas e a distância nelas percorrida, baseado em procedimentos de k-trocas de clientes entre rotas. Utilizaram-se, para isso, os procedimentos melhorativos, referidos anteriormente, incluídos na heurística construtiva de rota utilizada para gerar a população inicial. Aplicaram-se, uma vez em cada geração, estes procedimentos aos indivíduos resultantes de cruzamentos admissíveis e também, no final, à melhor solução obtida durante as várias gerações. Quando a aplicação do operador mutação, assim definido, tornava o indivíduo mais apto, este substituía o que lhe deu origem.

O operador cruzamento, neste AG, actua especialmente a nível das afectações de clientes aos veículos, provocando, em cada iteração, pequenas alterações nas soluções. O operador mutação (procedimentos melhorativos) também provoca algumas alterações nas afectações mas actua, principalmente, a nível das configurações das rotas a partir das trocas de clientes efectuadas entre as rotas correspondentes às afectações de cada indivíduo.

4 Experiência Computacional

Testou-se o AG descrito num conjunto de problemas-teste conhecidos da literatura (tabela 2), e disponíveis na Internet, no endereço http://branchandcut.org/VRP/data.

Tabela 2: Problemas-teste da literatura obtidos na Internet.

Nº dos problemas	Fonte dos problemas	Tipo dos dados
1 a 5, 8, 11	[3] (Eilon et al (1971))	Gerados uniformemente.
6, 7, 9		Iguais ao 8, com outro número e capacidade dos veículos.
10,12		Iguais ao 11, com outro número e capacidade dos veículos.
13 a 15	[4] (Fisher (1994))	Obtidos a partir de aplicações a casos reais.
16 e 17	[2] (Christofides et al (1979))	Estruturados - os clientes formam grupos (clusters).
18		Junção dos clientes dos problemas 5 e 11.
19		Junção dos clientes do problema 18 com os primeiros 49 clientes do problema 8.

As suas características estão resumidas na tabela 3. Nesta tabela os valores do número de veículos e da melhor solução (ou da solução óptima) são os que constavam dos ficheiros com os dados obtidos da Internet. Os valores sublinhados correspondem à solução óptima.

Todos estes ficheiros de dados contêm as coordenadas cartesianas do depósito e dos clientes. Calcularam-se as distâncias utilizando reais e arredondaram-se os resultados à unidade, trabalhando com uma matriz de distâncias inteiras.

O algoritmo foi implementado em Turbo C e executado num PC com um processador Pentium MMX 200 MHz e com 16 MB de RAM.

Após algumas experiências, com diferentes valores para os parâmetros utilizados, optou-se por fixá-los como está indicado na tabela 4.

Além de se terem comparado as soluções obtidas através do AG, com as melhores conhecidas, utilizaram-se também, para comparar principalmente os tempos de execução computacional, outros dois algoritmos. Escolheram-se: uma heurística de duas fases [16], computacionalmente muito rápida e a heurística construtiva de rota, que serviu para gerar a população inicial, mas recorrendo só aos três critérios de escolha do cliente inicial já mencionados. Na primeira fase da heurística de duas fases, os clientes são afectos aos veículos de uma forma semelhante à do algoritmo descrito em [2], e na segunda fase utiliza-se o Método da Inserção Mais Afastada para resolver heuristicamente os PCV envolvidos. No fim, à solução obtida são aplicados os procedimentos melhorativos utilizados no AG. Com esta implementação obtêm-se duas soluções. Numa os clientes são afectos aos veículos sequencialmente e na outra paralelamente. Com a heurística construtiva de rota obtêm-se três soluções, pois, como já se referiu, utilizaram-se três critérios diferentes para a escolha dos clientes iniciais. Comparou-se a melhor das soluções obtidas com cada um destes métodos com a obtida pelo AG. Repare-se que, qualquer um destes métodos pode, devido ao número de veículos estar previamente definido, não obter solução admissível se ficarem clientes por servir.

Estas duas heurísticas também foram implementadas em Turbo C e executadas no mesmo PC.

Tabela 3: Características dos problemas-teste

Nº do problema	Nº de Clientes	Nº de Veículos	(1)	Melhor solução
1	21	4	0.938	375
2	22	3	0.755	569
3	29	3	0.944	534
4	32	4	0.918	835
5	50	5	0.971	521
6	75	7	0.886	683
7	75	8	0.947	735
8	75	10	0.974	832
9	75	14	0.974	1032
10	100	4	0.911	681
11	100	8	0.911	817
12	100	14	0.930	1077
13	44	4	0.898	724
14	71	4	0.957	237
15	134	7	0.945	1162
16	100	10	0.905	820
17	120	7	0.982	1065
18	150	12	0.931	1053
19	199	17	0.937	1351

$$(1) = \frac{\text{Total da procura dos clientes}}{\text{Capacidade total dos veículos}}$$

Tabela 4: Valores dos parâmetros utilizados nos testes computacionais.

	Nº clientes ≤ 50		Nº clientes > 50	
Percentagem máxima de variáveis a trocar em cada cruzamento	Dimensão da População	Nº máximo de gerações	Dimensão da População	Nº máximo de gerações
90	50	35	80	60

Tabela 5: Comparação entre a melhor solução conhecida, e as soluções obtidas com o algoritmo de duas fases, o algoritmo construtivo de rota e o AG.

Probl. Nº	*sol**	Heurística de 2 fases			Heurística Construtiva			Algoritmo Genético		
		sol	*dp*	tempo	*sol*	*dp*	tempo	*sol*	*dp*	tempo
1	375	390	4,00	0,000	381	1,60	0,000	375	0,00	2,088
2	569	669	17,57	0,000	570	0,18	0,000	569	0,00	3,956
3	534	(a)		0,000	559	4,68	0,055	534	0,00	5,385
4	835	921	10,30	0,055	882	5,63	0,055	837	0,24	5,989
5	521	601	15,36	0,055	569	9,21	0,110	537	3,07	7,857
6	683	752	10,10	0,165	754	10,40	0,440	725	6,15	213,571
7	735	820	11,56	0,110	840	14,29	0,440	784	6,67	149,341
8	832	987	18,63	0,110	1009	21,27	0,220	878	5,53	44,396
9	1032	1282	24,22	0,055	(a)		0,164	1077	4,36	86,813
10	681	777	14,10	0,440	717	5,29	2,033	705	3,52	177,418
11	817	984	20,44	0,275	902	10,40	1,209	878	7,47	473,462
12	1077	1261	17,08	0,165	1232	14,39	0,769	1153	7,06	440,550
Média			**14,85**	**0,119**		**8,85**	**0,458**		**3,67**	**134,236**
13	724	879	21,41	0,055	778	7,46	0,110	727	0,41	19,725
14	237	260	9,70	0,110	260	9,70	0,495	242	2,11	225,385
15	1162	1483	27,62	0,495	1385	19,19	3,626	1175	1,12	1477,967
Média			**19,58**	**0,220**		**12,12**	**1,410**		**1,21**	**574,359**
16	820	1037	26,46	0,220	1028	25,37	0,769	831	1,34	390,550
17	1065	1334	25,26	0,330	1325	24,41	1,813	1049	-1,50	300,165
18	1053	1278	21,37	0,714	1174	11,49	4,066	1135	7,79	455,824
19	1351	1645	21,76	1,923	1538	13,84	4,835	1434	6,14	1502,582
Média			**23,71**	**0,797**		**18,78**	**2,871**		**3,44**	**662,28**
Média			**17,61**	**0,28**		**11,60**	**1,12**		**3,24**	**314,90**

(a) - Não obteve solução admissível
tempo - tempo de execução em segundos

Apresenta-se um quadro comparativo (tabela 5) entre o AG, estas duas heurísticas e a melhor solução inteira conhecida. Nas tabelas desta secção tem-se:

sol - solução obtida

*sol** - melhor solução conhecida, (que é a óptima se o valor estiver sublinhado)

dp - diferença percentual, sendo

$$dp = \frac{sol - sol*}{sol*} \text{ x } 100$$

Verifica-se que a solução obtida com o AG atinge em 3 instâncias o óptimo, em 1 (problema 17) obtém um valor melhor do que a melhor solução inteira conhecida, em 4 está a menos de 2% da melhor solução conhecida. No total, em 12 dos 19 problemas a solução obtida com o AG está a menos de 5% da melhor solução conhecida, o que já não ocorre com as outras duas heurísticas. Enquanto as outras duas heurísticas obtêm as melhores soluções para os dados de [3], o AG obtém, também, muito boas soluções para os problemas de menor dimensão deste conjunto, mas já não obtém bons resultados para os de dimensão 75 e 100.

Nas instâncias para as quais este algoritmo obteve as melhores soluções, a procura é, em geral, inferior a 95% da capacidade total dos veículos. No entanto, este facto não explica o bom comportamento do algoritmo, visto noutras instâncias, em que a procura total dos clientes é igual ou menor que 95% da capacidade total dos veículos, não se obterem bons resultados, enquanto que numa instância em que a procura total é cerca de 98% da capacidade total (problema 17) obtém-se uma solução muito boa. Parece, pois, não ser muito significativa a relação entre o comportamento do algoritmo e a percentagem de ocupação dos veículos. Mas, nas instâncias em que a procura total dos clientes esteja muito próxima da capacidade total dos veículos, este AG não deve ser muito eficiente, visto ser mais provável que a maioria das trocas de valores entre os indivíduos pais conduzam a soluções não admissíveis Assim, o AG pode terminar sem que tenha causado grandes alterações à população inicial. Isto ocorreu nos problemas 8 e 9.

É de salientar que os melhores resultados obtidos com o AG correspondem aos problemas de [4], extraídos de aplicações reais e aos problemas 16 e 17, que são os únicos que têm os clientes em grupos [2]. Como, normalmente nas aplicações reais os clientes estão agrupados é de esperar que o algoritmo tenha um bom funcionamento na resolução de problemas reais.

Em relação aos tempos de execução computacional nota-se que os correspondentes ao AG são bastante superiores aos dos outros dois algoritmos (ambos muito rápidos, principalmente o de duas fases). No entanto, o tempo de execução do AG foi em geral inferior a 5 minutos, mesmo trabalhando com uma população de 80 indivíduos, o que é ainda razoável. Em contrapartida os resultados obtidos com o AG foram sempre melhores (na média a *dp* foi de 3% contra 12% e 18% dos outros algoritmos), o que justifica o acréscimo de tempo computacional.

Constata-se, assim, que este AG é mais eficiente do que algumas heurísticas (construtiva e de duas fases). Em relação às melhores soluções conhecidas obtidas com métodos muito mais elaborados, já não é tão eficiente.

5 Conclusões e Trabalhos Futuros

Adaptar um Algoritmo Genético ao PORV não foi tarefa fácil. As maiores deficiências da abordagem realizada devem-se a:

(i) Não se ter permitido que na geração da população inicial fossem produzidos indivíduos (soluções) que violassem as restrições, o que obrigou a utilizar uma heurística para gerar a população inicial;

(ii) Na aplicação dos operadores genéticos também não foram permitidas violações das restrições;

(iii) Cada indivíduo representar uma afectação dos clientes aos veículos e não uma solução (as rotas) do PORV.

Em relação aos aspectos (i) e (ii) também se fizeram experiências gerando aleatoriamente a população inicial e permitindo violações das restrições, mas obtiveram-se fracos resultados. Nestas experiências penalizaram-se os indivíduos não admissíveis atribuindo-lhes uma aptidão

menor. Assim, não eram escolhidos para se reproduzirem tantas vezes como os indivíduos admissíveis, o que pode ter contribuído para os fracos resultados desta abordagem. A utilização de indivíduos não admissíveis de uma outra forma que lhes permita serem seleccionados mais vezes para reprodução, combinando-os com indivíduos admissíveis poderá diversificar a pesquisa.

O aspecto (iii) pode levar a que, eventualmente, se encontre a melhor afectação possível dos clientes aos veículos, e nesse sentido o Algoritmo Genético seja muito eficiente, mas as rotas dos veículos, para servirem os clientes que lhes estão afectos, obtidas através da resolução heurística dos PCV não sejam as melhores. Assim, a solução do PORV pode não ser a óptima mesmo tendo sido obtida a afectação óptima dos clientes aos veículos. A determinação das rotas óptimas, para uma afectação obtida pelo Algoritmo Genético, envolvia a resolução exacta de um PCV para cada veículo, o que é proibitivo em termos de tempo de execução computacional.

Este algoritmo poderá eventualmente ficar mais eficiente mediante algumas alterações que tenham em conta os aspectos (i), (ii) e (iii) focados atrás. As alterações, que podem ser testadas, separadamente ou em conjunto em trabalhos futuros, são:

- Utilizar heurísticas mais eficientes para a resolução dos PCV envolvidos;
- Permitir afectações não admissíveis, testando outros métodos de as penalizar, por exemplo, recorrendo a parâmetros penalizadores incluídos no cálculo da função objectivo;
- Alterar a actuação dos operadores genéticos tendo em conta que as soluções podem ser avaliadas segundo dois critérios. Um relativo à violação das restrições de capacidade e o outro às distâncias percorridas;
- Gerar a maior parte da população inicial aleatoriamente e somente alguns (poucos) indivíduos heurísticamente.

6 Bibliografia

[1] BULLNHEIMER, B.; R. F. HARTL; C. STRAUSS, *Applying the ant system to the vehicle routing problem*, in Metaheuristics: Advances and Trends in Local Search for Optimization, editado por S. Voss, S. Martello, I. Osman, C. Roucairol, Kluwer Academic Publishers (1999).

[2] CHRISTOFIDES, N.; A. MINGOZZI; P. TOTH, *The vehicle routing problem*, in Combinatorial Optimization, editado por N. Christofides, A. Mingozzi, P. Toth e C. Sandi, John Wiley (1979).

[3] EILON, S.; C. WATSON-GRANDY; N. CHRISTOFIDES, *Distribution Management: Mathematical Modelling and Practical Analysis*, Griffin London (1971).

[4] FISHER, M., *Optimal solution of vehicle routing problems using minimum k-trees*, Ops. Res. 42 (1994) 626-642.

[5] FISHER, M. e R. JAIKUMAR, *A decomposition algorithm for large-scale vehicle routing*, Work. Pap. 78-11-05, Dep. of Decision Sciences, Univ. Pennsylvania, Philadelphia (1978).

[6] GENDREAU, M.; A. HERTZ; G. LAPORTE (1994), *A tabu search heuristic for the vehicle routing problem*, Manag. Sci. 40 (1994) 1276-1290.

[7] GOLDBERG, D., *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company (1989).

[8] GOLDEN, B.; L. BODIN; T. DOYLE; W. STEWART, *Aproximate travelling salesman algorithms*, Ops. Res. 28 (1980) 694-711.

[9] HADJICONSTANTINOU E.; N. CHRISTOFIDES; A. MINGOZZI, *A new exact algorithm for the vehicle routing problem based on q-path and k-shortest path relaxations*, Ann. Ops. Res. 61 (1995) 21-44.

[10] HOLLAND, J., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (1975).

[11] KELLY, J. e J. XU, *A set partitioning-based heuristic for the vehicle routing problem*, INFORMS J. Computing 11 (1999) 161-172.

[12] LENSTRA, J. e A. RINNOOY KAN, *Complexity of vehicle routing and scheduling problems*, Networks 11 (1981) 221-227.

[13] OSMAN, I., *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Ann. Ops. Res. 41 (1993) 421-451.

[14] POTVIN, J.-Y., *Genetic algorithms for the traveling salesman problem*, Ann. Ops. Res. 63 (1996) 339-370.

[15] ROCHAT, Y. e E. TAILLARD, *Probabilistic diversification and intensification in local search for vehicle routing*, J. of Heuristics 1 (1995) 147-167.

[16] SCHÜTZ, G., *Resolução de Problemas de Optimização Combinatória em Máquinas Paralelas*, Tese de Doutoramento, Universidade do Algarve (1999).

[17] SYSLO, M.; N. DEO; J. KOWALIK, *Discrete Optimization Algorithms with PASCAL Programs*, Prentice-Hall (1983).

[18] TAILLARD, E., *Parallel iterative search methods for vehicle routing problems*, Networks 23 (1993) 661-673.