ROTEAMENTO DE VEÍCULOS DINÂMICO USANDO ALGORITMOS GENÉTICOS

Glaydston Mattos Ribeiro[1]
Departamento de Ciência da Computação e Informática
UniAracruz – Faculdade de Aracruz
Luiz Antonio Nogueira Lorena[2]
Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais

RESUMO
Devido aos grandes avanços tecnológicos obtidos, principalmente na área de geo-referenciamento, muitas empresas do setor de transportes estão revisando seus modelos de distribuição para acompanhar o dinamismo das operações, procurando alterar suas rotas durante a operação dos veículos por causa de novas requisições dos clientes, de congestionamentos ou ainda, por problemas nos veículos. Neste artigo é analisada a utilização de Algoritmos Genéticos no Problema de Roteamento de Veículos Dinâmico com Janelas de Tempo, com o objetivo de auxiliar o processo de decisão reduzindo assim, os custos logísticos durante a reprogramação das rotas.

ABSTRACT
Considering the technological advances in georeferencing, several transportation companies are reviewing their distribution models to consider the dynamism of operations, changing the routes when the vehicles are in operation and there are new orders, traffic jam or if the vehicles present problems themselves. In the literature, this is known as Dynamic Vehicle Routing Problem with Time Windows. So, this paper presents an application of Genetic Algorithm to solve this kind of problem and shows that it improves the decision process and reduces the logistics costs during the re-scheduling.

1. INTRODUÇÃO

Uma distribuição eficiente de mercadorias ou serviços é uma importante atividade logística pois os custos de transporte hoje em dia representam uma parcela significativa do preço de muitos produtos. Essa eficiência é obtida através da determinação de boas rotas para uma frota de veículos. Existe hoje em dia uma vasta literatura sobre o assunto, principalmente para o caso estático conhecido como o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) (Desrosiers *et al*., 1995). No problema estático, todos os dados relativos aos clientes são conhecidos antes das rotas serem construídas e nenhuma mudança é realizada após a fase de roteamento dos veículos.

Porém, o desenvolvimento da tecnologia da informação vem exigindo das empresas uma ampla revisão de seus modelos de distribuição, para que possam acompanhar as necessidades e as mudanças globais. Os avanços tecnológicos permitem explorar novas informações obtidas durante a operação dos veículos: um veículo que quebrou ou está muito atrasado, uma nova requisição de um cliente chegou e deve ser atendida, entre outras. Isso levou ao crescimento de uma nova classe de problemas não muito explorada (Dror e Powell, 1993; Psaraftis, 1995) conhecida como Problemas de Roteamento de Veículos Dinâmico com Janelas de Tempo (PRVDJT) (Psaraftis, 1995). O PRVDJT tem várias aplicações em diferentes campos como na entrega de produtos de petróleo (Bell *et al*., 1983) e serviços de emergência (Gendreau *et al*., 1997). Recentes trabalhos discutem modelagens e heurísticas de inserção para o PRVDJT (ver Chen *et al*., 2005; e Potvin *et al*., 2005).

Sendo assim, no PRVDJT as rotas podem ser alteradas durante a operação dos veículos mediante o surgimento de novas informações. De modo mais geral, o PRVDJT pode ser visto com uma série de problemas estáticos a serem resolvidos durante a operação dos veículos.
Nesse problema, uma frota de veículos com capacidade limitada deve sofrer uma reprogramação em tempo real, para atender a um novo conjunto de clientes com suas respectivas demandas e intervalos de atendimento (janelas de tempo) de tal modo que o custo seja o menor possível (normalmente distância percorrida ou tempo de viagem). Além disso, algumas requisições dos clientes podem ser conhecidas previamente, ou seja, antes da saída dos veículos do centro de distribuição.

Dado que o PRVDJT pode ser visto como uma série de PRVJTs, este trabalho tem como objetivo solucionar o problema dinâmico utilizando Algoritmos Genéticos nas reprogramações. A estratégia utilizada considera informações de posicionamento dos veículos, espaço disponível nos mesmos (peso e volume), localização e janelas de tempo dos clientes programados e ainda não atendidos, e as novas requisições.

O restante do trabalho está organizado como segue: na próxima seção é apresentada a definição do problema estático, seguida pela definição do problema dinâmico e pela descrição do método de solução proposto. Na Seção 5 são mostrados os resultados computacionais seguida pelas conclusões.

2. O PROBLEMA ESTÁTICO

O problema estático pode ser definido sobre um grafo completo $G=(V,A)$ sendo $V=\{v_0, v_1,...,v_n\}$ um conjunto de vértices e $A=\{(v_i,v_j): v_i, v_j \in V, i \neq j\}$ um conjunto de arestas. O vértice v_0 representa o depósito, e o restante, os clientes a serem atendidos com suas requisições. Cada vértice i possui uma demanda (carga) c_i, um tempo fixo de atendimento tf_i, uma razão de carregamento rc_i relacionada à demanda c_i para calculo do tempo de atendimento, e uma janela de tempo $[e_i, l_i]$ no qual e_i e l_i representam, respectivamente, a abertura e o fechamento da janela de tempo do vértice i (e_0 e l_0 estão associados à abertura e o fechamento do depósito, respectivamente). Por último, têm-se duas matrizes $D=(d_{ij})$ e $T=(t_{ij})$ que representam, respectivamente, a distância e o tempo de deslocamento associado à aresta (i,j).

Assim, dado uma frota homogênea de m veículos, o objetivo do PRVJT é encontrar um conjunto de rotas de custo mínimo iniciando e terminando no depósito tal que (Gendreau *et al*., 1999):

- Cada veículo deve atender uma única rota;
- Cada vértice v_i é visitado apenas uma única vez;
- Os veículos devem deixar o depósito em um horário superior a e_0;
- Os veículos devem estar no depósito antes do fechamento da janela de tempo do depósito, ou seja, antes de l_0;
- O início de atendimento em cada vértice v_i é maior ou igual a e_i;
- Se o tempo de chegada tc_i do veículo no vértice v_i for menor que e_i, o mesmo deverá aguardar, caracterizando um tempo espera (te_i) calculado como $te_i=(e_i-tc_i)$.

Com isso, sendo S um conjunto de soluções factíveis, a função objetivo a ser minimizada pode ser definida como (Gendreau *et al*, 1999):

$$v(s) = \sum_{k=1}^{m} d_k + \sum_{i=1}^{n} \alpha_i Max(0, tc_i - l_i) + \beta_i te_i, \qquad s \in S \tag{1}$$

tal que d_k é a distância total percorrida na rota k para todo $k=1,...,m$, α_i é um coeficiente de penalidade associado à chegada do veículo no vértice i após o fechamento da janela de tempo, e β_i é um coeficiente de penalidade associado ao tempo de espera te_i obtido no vértice i. Toda rota factível s deve iniciar e terminar no depósito respeitando a janela de tempo do mesmo.

3. O PROBLEMA DINÂMICO

No PRVDJT algumas requisições surgem durante a operação dos veículos, e com isso, deve-se realizar uma reprogramação dos veículos em operação. Como mencionado, no momento da reprogramação das rotas, esse problema pode ser visto como um problema estático a ser resolvido.

Para que as requisições não programadas possam ser atendidas no mesmo dia da solicitação, faz-se necessário impor um tempo limite de solicitação de tal modo que as demais requisições recebidas após esse tempo limite, sejam colocadas para o próximo dia de operação. No PRVDJT pode não existir conhecimento prévio sobre a localização das novas requisições realizadas quando os veículos já estão em operação, e assumi-se existir uma central de comunicação que informa ao motorista mudanças na sua programação (Montemanni *et al*., 2003).

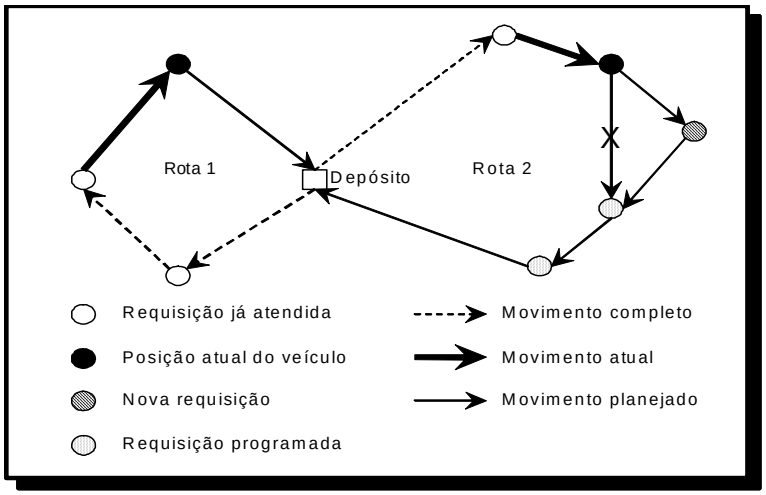

Figura 1: No momento da reprogramação das rotas, o problema pode ser visto como um PRVJT (Adaptado de Gendreau *et al*., 1999).

Para exemplificar a questão da transformação do PRVDJT em vários PRVJTs, considere a Figura 1. Essa figura mostra a situação de duas rotas no momento em que uma nova requisição é realizada, sendo que os círculos negros correspondem às posições atuais de cada veículo. Nesse caso, existe um conjunto de rotas existentes no qual cada uma inicia na posição atual do veículo e termina no depósito. Observa-se que para inserir a nova requisição, assim como no problema estático, o objetivo é minimizar a função (1), dadas as várias soluções factíveis existentes incluindo a nova requisição programada em uma das duas rotas. Considerando ainda que em uma reprogramação a ordem dos clientes não visitados pode ser alterada, este trabalho considera apenas requisições de coleta. Na Figura 1 é mostrada uma possível reprogramação realizada sobre a rota 2.

Muitas pesquisas estão sendo realizadas considerando o PRVDJT. Gendrau *et al*. (1999) desenvolveram uma heurística paralela de Busca Tabu dividindo o problema dinâmico em problemas estáticos aplicando-a nas instâncias propostas por Solomon (1987). Krumke *et al*. (2001) formalizaram a liberação em tempo real de unidades de serviço e propôs uma técnica de solução baseada em geração de colunas e particionamento. Gunstsh e Middendorf (2002) propuseram um algoritmo heurístico baseado em colônias de formiga para o problema do caixeiro viajante dinâmico. Montemani *et al*. (2002) desenvolveram uma estratégia de solução para o PRVDJT baseada em Sistemas de Colônia de Formigas, que foi testada sobre um conjunto de testes proposto por Kilby *et al*. (1998) que alteraram as instâncias propostas por Taillard (1994), Christofides e Beasley (1984) e Fisher *et al*. (1981).

Maiores informações sobre os diferentes tipos de problemas de roteamento de veículos dinâmico podem ser encontradas em Psarafits (1995), Gendreau e Potvin (1998), Bianchi (2000), Chen *et al*. (2005) e Potvin *et al*. (2005).

4. ALGORITMOS GENÉTICOS PARA O PRVDJT

O Algoritmo Genético (AG) foi desenvolvido por Holland (1975). Mais tarde, Goldberg (1989) disseminou o uso do AG aplicando-o a uma série de problemas de otimização. Os Algoritmos Genéticos empregam um processo adaptativo e paralelo de busca de soluções em problemas complexos, o que o torna uma técnica muito útil em problemas de otimização.

O primeiro passo de um AG é a geração da população inicial de cromossomos. Esta população é formada por um conjunto aleatório de cromossomos que representam possíveis soluções do problema a ser resolvido. Durante o processo evolutivo, esta população é analisada e cada cromossomo recebe uma avaliação (aptidão) que reflete a qualidade da solução que ele representa. Em geral, os cromossomos mais aptos são selecionados e os menos aptos são descartados. Os membros selecionados podem sofrer modificações em suas características fundamentais através dos operadores de cruzamento (*crossover*) e mutação, gerando descendentes para a próxima geração. Este processo é repetido até que uma solução satisfatória seja encontrada. Maiores detalhes sobre o algoritmo e a implementação, podem ser obtidos em Lacerda e Carvalho (1999).

Para o PRVJT existem muitas representações para um cromossomo como pode ser visto nos trabalhos de Potvin e Bengio (1996), Louis *et al*. (1999) e Pereira *et al*. (2002). Além disso, alguns Algoritmos Genéticos consideram uma população dinâmica como o Algoritmo Genético Construtivo proposto por Lorena e Furtado (2001).

4.1 Representação de um cromossomo para o PRVDJT

Um cromossomo para o PRVDJT deve representar as rotas em operação, a programação dos clientes em cada rota e as posições de inserção das novas requisições. Assim, o cromossomo é representado através de blocos contendo as rotas de cada um dos veículos e as posições das novas requisições. A Figura 2 mostra um exemplo de como é representado um cromossomo no PRVDJT. Nesse exemplo, observam-se 3 rotas em execução sendo a rota 1 programada da seguinte maneira: posição atual do veículo 1, cliente 1, 3, 4, 5 e retorno para o depósito. Tanto a localização do veículo no momento da reprogramação das rotas quanto a localização do depósito, não são representadas no cromossomo. Outra questão importante está relacionada às posições das novas requisições que estão associadas ao índice do vetor das rotas. No exemplo,

a primeira requisição deve ser inserida na posição 0, a segunda na posição 2, a terceira na 5 e a quarta na 9. A Figura 3 mostra a solução correspondente ao cromossomo mostrado na Figura 2 após terem sido feitas as inserções das novas requisições.

Rota 1				Rota 2				Rota 3		Novas Requisições				
										11	12	13	14	⇒ Códigos das novas requisições
1	3	4	5	9	8	7	2	6	10	0	2	5	9	⇒ Posições de inserções

Figura 2: Um exemplo de um cromossomo para o PRVDJT.

Rota 1							
Posição atual do veiculo 1	11	1	3	12	4	5	Retorno para depósito

Rota 2						
Posição atual do veiculo 2	9	13	8	7	2	Retorno para depósito

Rota 3				
Posição atual do veiculo 3	6	14	10	Retorno para depósito

Figura 3: Representação real do cromossomo mostrado na Figura 2.

4.2 Operadores genéticos

Dada uma população de tamanho *N*, a cada geração, *N* novos descendentes são obtidos através da aplicação de operadores genéticos de *crossover* e de mutação. Porém, ao se combinar dois cromossomos pais para formar dois filhos alguns cuidados são necessários. Neste trabalho, os clientes já programados devem, após o cruzamento, aparecer nos seus descendentes e nas mesmas rotas, com isso, o cruzamento é realizado entre as respectivas rotas presentes nos cromossomos pai.

Como uma rota é um problema de permutação entre elementos, se faz necessário utilizar um operador genético de permutação, sendo que neste trabalho foi utilizado o CX (*Cycle Crossover*) (Goldbarg, 1989). Esse operador começa copiando o primeiro elemento de pai_1 para $filho_1$. O correspondente elemento de pai_2 é copiado para o $filho_1$ na posição em que aparece no pai_1. Esse processo é repetido até que um elemento ao ser copiado para o $filho_1$ já esteja presente no $filho_1$. Na etapa final, as posições em branco são obtidas por simples troca de elementos entre pai_1 e pai_2. A Figura 4 mostra o processo descrito acima aplicado a dois pais representando duas rotas de dois cromossomos.

Pai_1	1	2	3	4
Pai_2	3	4	1	2

Copiando o primeiro elemento de pai_1 para $filho_1$

$Filho_1$	1			
$Filho_2$				

Procurando 3 em pai_1 e copiando para $filho_1$

$Filho_1$	1		3	
$Filho_2$				

Como o correspondente elemento de pai_2 já está no $filho_1$, o ciclo termina.

$Filho_1$	1		3	
$Filho_2$				

Trocando os elementos entre pai_1 e pai_2.

$Filho_1$	1	4	3	2
$Filho_2$				

Repetindo o processo para $filho_2$ tem-se o seguinte resultado:

$Filho_1$	1	4	3	2
$Filho_2$	3	2	1	4

Figura 4: Uso do Operador Genético *CX*.

Assim, esse procedimento é aplicado separadamente sobre todas as rotas presentes nos cromossomos pai. Porém, ainda se faz necessário realizar o cruzamento entre as novas requisições, para isso foi utilizado o operador de corte. Cada um dos cromossomos pais tem sua cadeia cortada em uma posição aleatória produzindo duas "cabeças" e duas "caudas". As caudas são trocadas gerando dois novos filhos que são adicionados aos cruzamentos das rotas. A Figura 5 mostra o processo descrito acima.

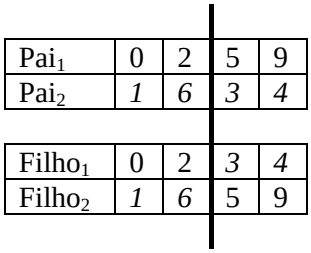

Figura 5: Crossover entre novas requisições.

Após a operação de crossover, o operador de mutação é aplicado, com dada probabilidade, em cada elemento dos dois filhos gerados. A mutação, dentro de uma rota, seleciona 2 elementos e os inverte de posição. A mutação melhora a diversidade dos cromossomos na população, porém destrói informações contidas no cromossomo. A Figura 6 (a) mostra esse processo aplicado aos filhos 1 e 2 da Figura 4.

Antes	$Filho_1$	1	4	3	2
	$Filho_2$	3	2	1	4
Depois	$Filho_1$	**2**	4	3	**1**
	$Filho_2$	**1**	2	**3**	4

Rotas
(a)

Antes	$Filho_1$	0	2	5	9
	$Filho_2$	1	6	3	4
Depois	$Filho_1$	0	**3**	5	**6**
	$Filho_2$	1	**2**	**9**	4

Novas requisições
(b)

Figura 6: Operador de mutação.

O operador de mutação também é aplicado às novas requisições da seguinte maneira: seleciona-se o elemento que sofrerá mutação e em seguida, de forma aleatória, troca-se a posição indicada por esse elemento por uma outra qualquer dentro do cromossomo. Esse processo está exemplificado na Figura 6 (b).

Para evitar convergência prematura, foi utilizado o método de *ranking* linear em que a aptidão é dada por (Baker, 1987):

$$f_i = Min + (Max - Min)\frac{N - i}{N - 1} \qquad (2)$$

tal que i é o índice do cromossomo na população em ordem crescente de valor da função objetivo, $1 \leq Max \leq 2$, $Max + Min = 3$ e N o tamanho da população. Para não perder o melhor cromossomo de uma geração para outra foi adotada a estratégia de elitismo (Dejong, 1975).

5. EXPERIMENTOS COMPUTACIONAIS

Para avaliar o Algoritmo Genético proposto, oito instancias testes foram preparadas. As configurações do AG foram as seguintes: Número de gerações:100; Tamanho da população:

1000; População inicial aleatória; Estratégia de elitismo; Taxa de crossover: 80%; Taxa de mutação: 5%; *Max*=2 e *Min=1* na Equação (2). Para todo conjunto de testes foram realizados 10 experimentos com as mesmas condições iniciais mas com sementes diferentes. Para dar maior liberdade ao algoritmo genético, toda rota encontrada é avaliada mesmo não respeitando a janela de tempo dos clientes, com isso a Equação 1 foi alterada para:

$$v(s) = \sum_{k=1}^{m} d_k + \sum_{i=1}^{n} \alpha_i Max(0, tc_i - l_i) + \beta_i te_i + \varphi_i Max(0, tc_i + tf_i + c_i rc_i - l_i),\ s \in S \qquad (3)$$

tal que φ_i é um coeficiente de penalidade associada à saída do veículo do vértice *i* após o fechamento da janela de tempo l_i. Os valores utilizados para os parâmetros foram: $\alpha_i = 5$, $\beta_i = 1$ e $\varphi_i = 3 \ \ \forall i = 1...n$. O computador utilizado foi um Pentium IV 2,66 GHz com 512 MB de memória RAM.

Para verificar a qualidade do Algoritmo Genético proposto, foi implementada a heurística de melhoria *k-opt* proposta por Lin e Kernighan (1973) no qual *k* arcos são removidos de uma rota e substituídos por outros *k* arcos, com a finalidade de reduzir uma função objetivo. Essa heurística foi escolhida por ser indicada a problemas em que se deseja melhorar roteiros existentes. De forma resumida, para *k=2*, o método testa trocas possíveis entre pares de arcos, refazendo as conexões quando houver uma melhoria na rota (ver Figura 8). Porém, para que o método funcione no PRVDJT, se faz necessário incorporar às rotas em operação as novas requisições. Isso foi feito considerando os centróides das rotas programadas. Cada nova requisição é inserida em uma posição aleatória dentro da rota que possui o centróide mais próximo do novo cliente. Para maiores detalhes sobre a heurística *k-opt* ver Lin e Kernighan (1973), Laporte (1992), Novaes (2001) e Cunha *et al*. (2002).

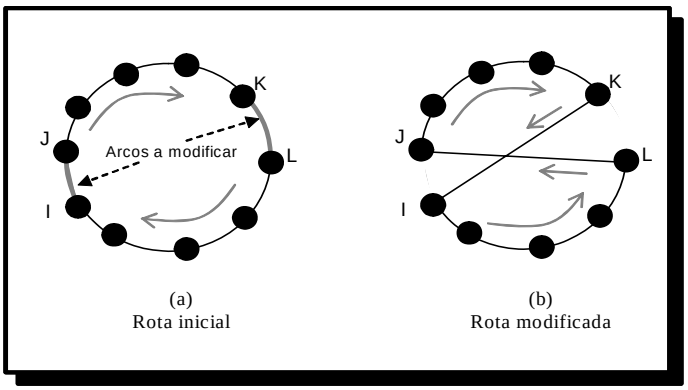

Figura 8: Modificação realizada pela Heurística *2-opt* (Novaes, 2001).

As Tabelas 1, 2 e 3 apresentam os resultados encontrados para todas as instâncias testadas. As descrições das colunas estão listadas a seguir:

- "Instância" - Refere-se ao nome da instância;
- "Número de Pontos" - Número de clientes já programados;
- "Número de Novas Requisições" - Número das novas requisições realizadas;
- "Roteamento Atual" - Valor da função objetivo para o roteamento em operação, com os clientes já programados;
- "Melhor Solução com o AG" - Melhor solução encontrada com o Algoritmo Genético após a reprogramação das rotas;

- “Média das Soluções Encontradas com o AG” - Média obtida para 10 execuções do Algoritmo Genético;
- “Tempo Médio (s)” – Tempo médio obtido para 10 execuções do Algoritmo Genético;
- Heurística *2-opt* – Resultado encontrado pela heurística *2-opt* após a inserção das novas requisições nas rotas mais próximas.

As instâncias mostradas nas tabelas a seguir foram criadas variando-se o grau de dificuldade. Isso foi possível através da variação do número de clientes a serem atendidos e do número de novas requisições, além de atrasos e esperas impostas. Cada veículo presente em cada rota de cada instância, tem capacidade suficiente para atender a todas as novas requisições.

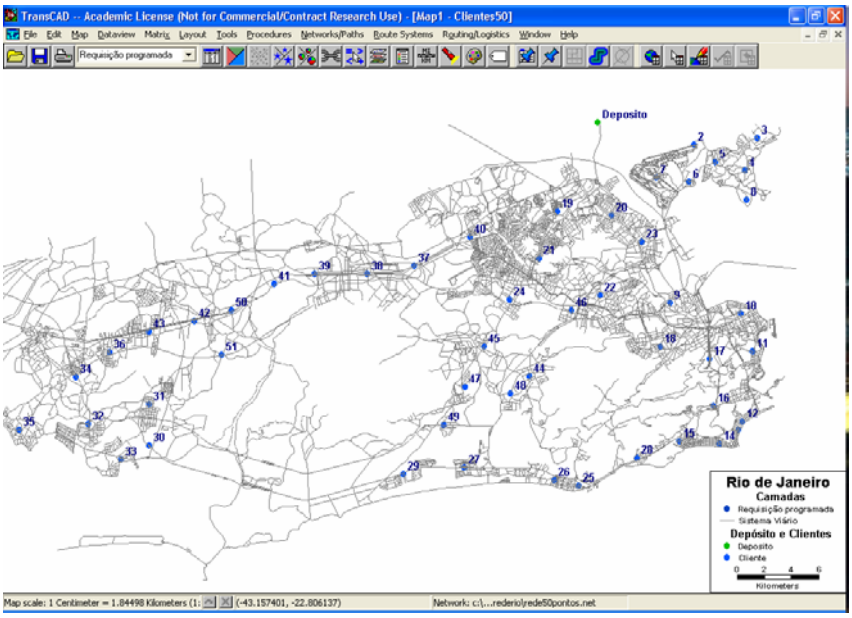

Figura 9: Localização do depósito e dos clientes para a instância 50B4R.

Para verificar a eficiência do algoritmo em situações em que se deseja alterar a seqüência de atendimento dos clientes já programados, foram criadas instâncias que não possuem novas requisições. Por exemplo, a instância “19A0R” possui 19 clientes já programados e nenhuma nova requisição. A letra “A” presente no nome da instância indica que as rotas apresentam poucos atrasos, por outro lado à letra “B” indica atrasos consideráveis. As rotas iniciais foram criadas utilizando o software TransCAD (Caliper, 2000) muito utilizado na área de transportes, considerando distâncias Euclidianas e tempos proporcionais a problemas reais (velocidade média de 50 Km/h).

A Figura 9 apresenta a disposição dos clientes assim como a localização do depósito para a instância 50B4R. A Figura 10 apresenta a localização dos veículos (estes ainda no depósito), as rotas programadas e as novas requisições.

Como pode ser visto na Tabela 1, em todos os casos o AG reduziu o valor da função objetivo após a reprogramação e procurou, quando possível, reduzir todos os tempos envolvidos (atrasos e esperas). As médias também foram muito próximas da melhor solução encontrada, a maior diferença foi encontrada na instância 50B4R no qual a melhor solução apresentou uma variação de 3,70% da média. A Figura 11 mostra o desempenho do melhor cromossomo

e a média da população em cada geração para a instância 50B4R. É possível notar que com aproximadamente 25 gerações, a população já está bem evoluída.

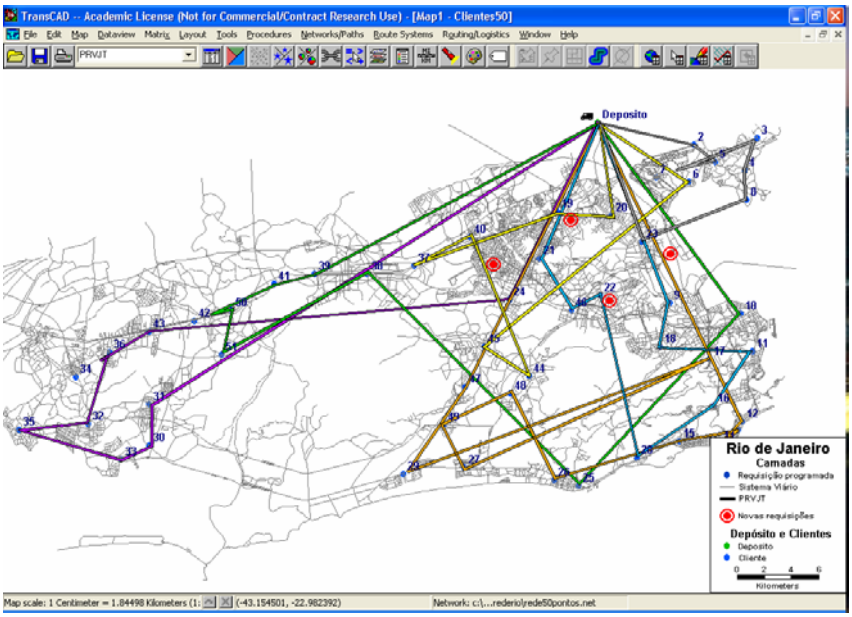

Figura 10: Rotas, localização dos veículos e das novas requisições para a instância 50B4R

Tabela 1: Resultados encontrados com o AG

Instância	Número de Pontos	Número de Novas Requisições	Roteamento Atual	Melhor Solução com o AG	Médias das Soluções Encontradas com o AG.	Tempo Médio (s)
19A0R	19	0	254,11	**171,77**	**171,77**	2,71
19A1R	19	1	254,11	**172,60**	**172,60**	2,82
50A0R	50	0	1068,00	**993,92**	**993,92**	3,28
50A4R	50	4	1068,00	**882,80**	**883,28**	3,54
50B0R	50	0	1741,81	**1496,17**	**1496,56**	5,50
50B4R	50	4	1741,81	**1299,69**	**1350,08**	6,20
100B0R	100	0	1231,03	**1174,00**	**1180,06**	10,80
100B10R	100	10	1231,03	**1105,10**	**1134,30**	13,20

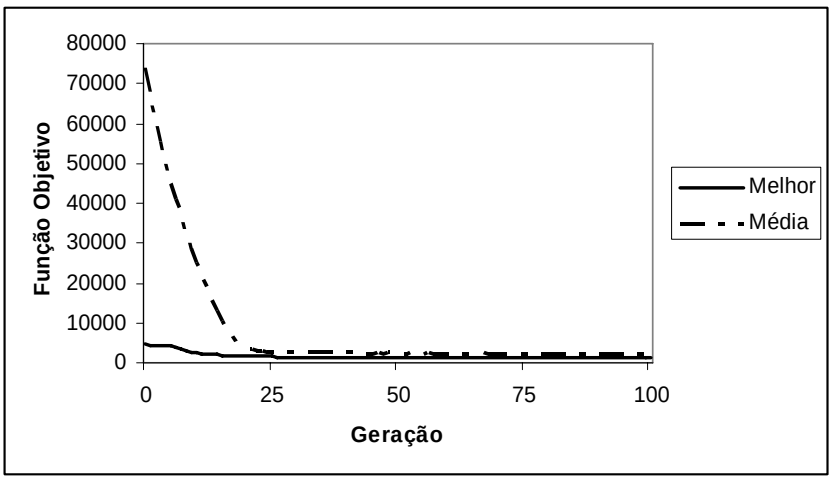

Figura 11: O melhor indivíduo e o valor médio da população.

Comparando-se agora a melhor solução do AG com a Heurística *2-opt* (Tabela 2), para as instâncias sem reprogramação verifica-se que o AG apresentou resultados melhores do que a Heurística *2-opt*. Os dois algoritmos forneceram resultados iguais para duas instâncias (19A0R e 50A0R), com o AG sendo melhor para as demais instâncias. Para as instâncias com novas requisições, o AG apresentou resultados superiores aos da Heurística *2-opt* em todas os testes.

Tabela 2: Comparação entre os melhores resultados AG e a Heurística *2-opt.*

Instância	Melhor Solução com o AG	Heurística *2-opt*
19A0R	171,77	171,77
19A1R	**172,60**	289,46
50A0R	993,92	993,92
50A4R	**882,80**	985,28
50B0R	**1496,17**	1496,24
50B4R	**1299,69**	1364,35
100B0R	**1174,00**	1177,13
100B10R	**1105,10**	1166,47

Ao se analisar os resultados médios (Tabela 3), novamente os resultados encontrados com AG foram interessantes quando comparados à Heurística *2-opt*, tendo 2 resultados iguais (instâncias 19A0R e 50A0R), 4 melhores (instâncias 19A1R, 50A4R, 50B4R e 100B10R) e 2 inferiores (instâncias 50B0R e 100B0R) sendo a maior diferença de 0,24%.

Tabela 3: Comparação entre os resultados médios do AG e a Heurística *2-opt.*

Instância	Médias das Soluções Encontradas com o AG.	Heurística *2-opt*
19A0R	171,77	171,77
19A1R	**172,60**	289,46
50A0R	993,92	993,92
50A4R	**883,28**	985,28
50B0R	1496,56	**1496,24**
50B4R	**1350,08**	1364,35
100B0R	1180,06	**1177,13**
100B10R	**1134,30**	1166,47

6. CONCLUSÕES

Este trabalho procurou analisar e resolver o Problema de Roteamento de Veículos Dinâmico com Janelas de Tempo utilizando Algoritmos Genéticos. Acredita-se que esse problema venha a estar cada vez mais presente no dia-a-dia das empresas de transporte de carga, influenciando significativamente nos custos logísticos. Com isso, se faz necessário desenvolver algoritmos eficientes para resolver tal problema.

Foram realizados vários experimentos e os resultados encontrados com o AG, melhoraram as rotas existentes além de se igualarem ou serem superiores aos da Heurística *2-opt*, muito conhecida e indicada para problemas de roteamento quando o objetivo em questão é o de melhorar as rotas. Em muitos casos os resultados médios encontrados para 10 execuções do Algoritmo Genético foram melhores do que a Heurística 2-*opt*.

Por último, os resultados encontrados sugerem a investigação da utilização da Heurística *2-opt* dentro do AG. Considerando, por exemplo, que a mutação seja realizada utilizando essa heurística, o AG pode convergir mais rapidamente e conseqüentemente, o processo pode ser finalizado antes por algum critério que envolva a similaridade da população.

Agradecimentos
Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq pelo apoio financeiro dado ao desenvolvimento deste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

Bell, W.; G. G. Dalberto, e D. Ronen (1983) Improving the Distribution of Industrial Gases with On-Line Computerized Routing and Scheduling optimizer. *Interfaces*, v. 13, p. 4-23.

Bianchi L. (2000) *Notes on Dynamic Vehicle Routing. The State of the Art*. Technical Report IDSIA-05-01, IDSIA, Switzerland. Disponível em http://www.idsia.ch/idsiareport/IDSIA-05-01.ps.gz.

Caliper (2000), TransCAD 3.61 *Users Guide*.

Chen, H. K.; C. F. Hsueh e M. S. Chang (2005) The Real-Time Time Dependent Vehicle Routing Problem. *Transportation Research Part E*. To Appear.

Christofides, N. e J. Beasley (1984). The Period Routing Problem. *Networks*, v. 14, p. 237-256.

Cunha, C. B., U. O. Bonasser e F. T. M. Abrahão (2002) Experimentos Computacionais com Heurísticas de Melhorias para o Problema do Caixeiro Viajante. *Anais do XVI Congresso de Pesquisa e Ensino em Transportes*, ANPET, Natal-RN, v. 2, p. 105-117.

Dejong, K. (1975) *The Analysis and Behaviour of a Class of Genetic Adaptive Systems*. PhD Tesis, University of Michigan.

Desrosiers J.; Y. Dumas, M. M. Solomon e E. Soumis (1995) Time Constrained Routing and Scheduling. *In*: Ball , M. O.; T. L. Magnanti; C. L. Monma e G. L. Nemhauser (*eds.*) *Network Routing, Handbooks in Operations Research and Management Science*. Amsterdam, Netherland, p. 35-139.

Dror, M. e W. B. Powell (1993) Special Issue on Stochastic and Dynamic Models in Transportation. *Operations Research*, v. 41.

Fisher, M., R. Jaikumar e V. Wassenhove (1981) A Generalized Assignment Heuristic for Vehicle Routing. *Networks*, v. 11, p. 109-124.

Gendreau M.; G. Laporte; e F. Semet (1997) *Solving an Ambulance Location Model by Tabu Search*. Techinal Report CRT-97-18, Center de recherche sur les transports. Université de Montréal.

Gendreau M., e J-Y Potvin (1998) Dynamic Vehicle Routing and Dispatching. *In:* Crainic, T. G. e G. Laporte (*eds*) *Fleet Management and Logistics*, p. 115-226. Kluwer.

Gendreau, M.; F. Guertin; J-Y. Potvin e E. Taillard (1999) Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, v. 33, p. 381-390.

Goldbarg, D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., New York.

Guntsch M., e M. Middendorf (2002) Applying Population Based ACO to Dynamic Optmization Problems. *Anais do ANTS 2002*, p. 149-162.

Holland, J. (1995) *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.

Kilby P.; P. Prosser e P. Shaw (1998) *Dynamic VRPs: a Study of Scenarios*. Technical Report APES-06-1998, University of Strathclyde.

Krumke S. O.; J. Rambau e L. M. Torres (2001) *Real-time Dispatching of Guided and Unguided Automobile Service Units with Soft Time Windows*. Technical Report 01-22, Konrad-Zuse-Zentrum für Information Stechnik Berlin.

Lacerda, E. G. M. e A. C. P. L. F. Carvalho (1999) Introdução a Algoritmos Genéticos. *Anais do SBC 99*, v. 2, p. 51-126.

Laporte, G. (1992) The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *European Journal of Operations Research*, v. 59, p. 345-358.

Lin, S. e B. W. Kernighan (1973) An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research*, v. 21, p. 498-516.

Lorena, L. A. N. e J. C. Furtado (2001) Constructive Genetic Algorithm for Clustering Problems. *Evolutionary Computation*, v. 9, n. 3, p. 309-327.

Louis S. J.; X. Yin e Z. Y. Yuan (1999) Multiple Vehicle Routing with Time Windows Using Genetic Algorithms. *Anais do Proceedings of the Congress on Evolutionary Computation*.

Montemanni R.; L. M. Gambardella; A. E. Rizzoli e A. V. Donati (2003) A New Algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System. *Anais do Proceedings of ODYSSEUS 2003*. Palermo, Italy, p. 26-31. Disponível em http://www.idsia.ch/~luca/.

Novaes, A. G. (2001) *Logística e Gerenciamento da Cadeia de Distribuição*. Ed Campus, Rio de Janeiro.

Pereira, F. B.; J. Tavares; P. Machado e E. Costa (2002) GVR: A New Genetic Representation for the Vehicle Routing Problem. *Anais do Proceedings of AICS 2002 - 13th Irish Conference on Artificial Intelligence and Cognitive Science*, Limerick, Ireland.

Potvin, J-Y. e S. Bengio (1996) The Vehicle Routing Problem with Time Windows - Part II: Genetic Search. *INFORMS Journal on Computing*, v. 8, p. 165-172.

Potvin, J-Y.; Y. Xu e I. Benyahia (2005) Vehicle Routing and Scheduling with Dynamic Travel Times. *Computers and Operations Research*. To Appear.

Psaraftis, H. N. (1995) Dynamic Vehicle Routing: Status and Prospects. *Operations Research*, v. 61, p. 143-164.

Solomon M. M. (1987) Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research*, v. 35, p. 254-265.

Taillard, E. (1994) Parallel Iterative Search Methods for Vehicle-Routing Problems. *Networks*, v. 23, p. 661-673.

UniAracruz - Faculdade de Aracruz [1]
Departamento de Ciência da Computação e Informática
R. Prof. Berilo Basílio dos Santos, 180, Bairro Vila Rica,
Aracruz – ES. CEP 29194-910.

Fone: (027) 3256-1102
Fax: (027) 3256-1102
Email: glaydston@fsjb.edu.br

Instituto Nacional de Pesquisas Espaciais[1,2]
Laboratório Associado de Computação e Matemática Aplicada
Av. dos Astronautas, 1758, Jardim da Granja,
São José dos Campos – SP. CEP 12.227-010

Fone: (012) 3945-6555
Fax: (012) 3945-6357
Email: glaydston@lac.inpe.br
lorena@lac.inpe.br