

Adaptação do Algoritmo Genético no Tratamento do Problema de Roteamento de Veículos com Janela de Tempo

HUMBERTO CÉSAR BRANDÃO DE OLIVEIRA¹
MARIANE MOREIRA DE SOUZA¹
GUILHERME BASTOS ALVARENGA²
RICARDO MARTINS DE ABREU SILVA²

UFLA - Universidade Federal de Lavras
DCC - Departamento de Ciência da Computação
Cx Postal 37 - CEP 37200-000 Lavras (MG)

¹(humberto.brandao, mariane.souza)@gmail.com

²(bastos, rmas)@dcc.ufla.br

Resumo. Problemas de Roteamento de Veículos vêm sendo cada vez mais estudados para obter uma maior economia nos gastos com o transporte de pessoas e mercadorias. Este trabalho propõe uma alteração no algoritmo genético básico, que mesmo sem a aplicação de operadores de cruzamento, foi capaz de encontrar bons resultados no tratamento do Problema de Roteamento de Veículos com Janela de Tempo.

Palavras-Chave: roteamento de veículos com janela de tempo, otimização combinatória, pesquisa operacional, algoritmo genético, *PRV*, *PRVJT*.

1 Introdução

Custos com o transporte de mercadorias vêm ganhando uma atenção especial nas últimas décadas, onde a minimização dos gastos é um grande foco para empresas que visam sobreviver e crescer no mercado competitivo atual. Os resultados da literatura vêm contribuindo de maneira significativa no Problema de Roteamento de Veículos (*PRV*). Como este problema no mundo real apresenta uma enorme dinamicidade, foi proposta a subdivisão desta classe de problemas a fim de se estudar de maneira específica algumas destas situações presentes em instâncias reais. Uma delas é a consideração da capacidade de carga do veículo e o tempo em que os consumidores podem ser atendidos (janela de tempo), tendo apenas um depósito central de mercadorias, ficando conhecido como Problema de Roteamento de Veículos com Janela de Tempo (*PRVJT*).

Segundo Alvarenga [Alvarenga (2004)], os custos relacionados ao transporte de pessoas e mercadorias é muito grande, com tendência ao crescimento, motivado pela expansão atual das fronteiras comerciais de todo tipo de negócio. Pesquisas sugerem que de 10% a 15% do valor final das mercadorias comercializadas correspondem ao custo de seu transporte [Fisher et.Al. (1997)]. Segundo Bodin [Bodin et.Al. (1983)], o custo de distribuição nos Estados Unidos em 1980 foi estimado em 400 bilhões de dólares. Uma parcela destes custos po-

deria ser reduzida com o tratamento de diversos problemas de roteamento, onde o *PRVJT* tem uma importante contribuição.

2 Problema de Roteamento de Veículos com Janela de Tempo (*PRVJT*)

Como descrito por Alvarenga [Alvarenga (2004)], o Problema de Roteamento de Veículos com Janela de Tempo pode ser formulado da seguinte maneira: Um conjunto de veículos idênticos, representado pelo conjunto $V = 1, \dots, M$, necessita realizar entregas em uma região. Os N consumidores dentro desta região estão representados pelo conjunto C , que são vértices de um grafo $G = (C, A)$. Adicionalmente, incluem-se dois outros vértices, o vértice 0 representa o depósito central de onde partirão todos os veículos. Para facilitar a representação matemática do problema, se duplica o vértice do depósito central, ou seja, o vértice $N + 1$ também representará o depósito central, significando no problema o vértice de chegada de todos os veículos. As variáveis t_{ij} e c_{ij} representam respectivamente o tempo e a distância necessários para ir do vértice i ao vértice j . Cada consumidor i também está associado a uma demanda, ou seja, uma quantidade de encomenda q_i . Além disso, cada consumidor deverá ser atendido por um único veículo, não sendo permitido a divisão de uma encomenda por dois ou mais veículos. Quanto à janela de tempo,

definida como o intervalo $[a_i, b_i]$, indica que a partir do instante inicial a_i é permitido o início da entrega ou coleta no consumidor i . Caso a chegada do veículo no consumidor i se dê antes do instante a_i , o veículo deverá esperar. O veículo nunca poderá chegar depois do instante b_i , pois viola a restrição de tempo do problema. Este tipo de restrição de tempo é conhecido na literatura como janela de tempo rígida ou *hard time window*. Os veículos são idênticos e possuem uma capacidade máxima de carga Q . A variável de decisão x_{ijv} determina se o veículo v faz o percurso do consumidor i para o consumidor j , recebendo o valor 1, se verdadeiro, e 0 em caso contrário.

Pode-se agora definir matematicamente o problema de otimização como:

$$\text{Minimize } \sum_{v \in V} \sum_{i \in G} \sum_{j \in G} c_{ij} x_{ijv} \text{ sujeito a:} \quad (1)$$

$$\sum_{v \in V} \sum_{j \in G} x_{ijv} = 1; \forall i \in C \quad (2)$$

$$\sum_{i \in C} q_i \sum_{j \in G} x_{ijv} \leq Q; \forall v \in V \quad (3)$$

$$\sum_{j \in G} x_{0jv} = 1; \forall v \in V \quad (4)$$

$$\sum_{j \in G} x_{i(N+1)v} = 1; \forall v \in V \quad (5)$$

$$\sum_{i \in G} x_{ijv} - \sum_{j \in G} x_{h j v} = 0; \forall h \in C, \forall v \in V \quad (6)$$

$$s_{iv} + t s_i + t_{ij} - K(1 - x_{ijv}) \leq s_{jv}; \forall i, j \in G, \forall v \in V \quad (7)$$

$$a_i \leq s_{iv} \leq b_i; \forall i \in G, \forall v \in V \quad (8)$$

$$x_{ijv} \in \{0, 1\}; \forall i, j \in G, \forall v \in V \quad (9)$$

A fórmula 2 garante que somente um veículo v chega a cada consumidor i . Cada veículo v atenderá somente

um conjunto de consumidores cuja demanda total não ultrapasse a sua capacidade Q (fórmula 3). As fórmulas 4 e 5 garantem que cada veículo v parte e retorna ao depósito central, respectivamente. A fórmula descrita por 6 indica a continuidade das rotas, ou seja, se um veículo chega a um consumidor ele deverá sair do mesmo para o consumidor seguinte, perfazendo todos os trechos entre consumidores. A restrição de tempo (com relação ao início da janela de tempo) é garantida pela fórmula 7, onde o instante de chegada de um veículo v a um consumidor j (s_{jv}) não poderá ocorrer antes do tempo de chegada no consumidor anterior i (s_{iv}) mais o tempo de serviço no primeiro ($t s_i$), mais o tempo de percurso no trecho (i, j) que é t_{ij} . É assumido uma velocidade constante tal que o tempo de percurso t_{ij} é igual a distância entre i e j . A constante K sendo suficientemente grande garante que a equação seja somente uma restrição efetiva quando x_{ijv} seja igual a 1, ou seja, quando o veículo v percorra a trecho (i, j) . O respeito ao fim da janela de tempo do consumidor é garantido pela fórmula 8, onde o instante de chegada de um veículo v em um consumidor i está dentro do limite da janela do mesmo. A fórmula 9 garante a integralidade das variáveis do problema.

Observa-se no modelo apresentado, um custo associado a cada arco do grafo, geralmente representado pela distância do consumidor i ao consumidor j , sendo neste caso o objetivo a minimização da distância total percorrida (fórmula 1).

Encontrar solução para o *PRVJT* implica em obter simultaneamente a solução de vários problemas \mathcal{NP} -Difíceis, tais como o Problema do Caixeiro Viajante (*PCV*) e o Problema da Mochila (*Knapsack Problem*), sendo conseqüentemente considerado \mathcal{NP} -Difícil, provado por Johnson em [Johnson et.Al. (1979)].

Como o espaço de busca de problemas \mathcal{NP} -Difíceis não podem totalmente explorados em tempo hábil, heurísticas e meta-heurísticas são utilizadas para buscar soluções satisfatórias para tais problemas. Este trabalho apresenta na seção 3 uma heurística, e na seção 4 uma meta-heurística para solucionar o Problema de Roteamento de Veículos com Janela de Tempo.

3 Push-Forward Insertion Heuristic (PFIH)

Para a construção da população inicial do Algoritmo Genético e como complemento de alguns operadores de mutação, este trabalho utilizou a heurística *PFIH*, introduzida por Marius Solomon em [Solomon (1987)]. Como citado por Larsen em [Larsen (1999)], o *PFIH* possui uma estratégia construtiva eficiente para calcular o custo de um novo consumidor em uma rota. Este custo é calculado de acordo com sua posição geográ-

fica, com o fim de sua janela de tempo e o ângulo existente entre ele e o depósito central.

O *PFIH* é descrito da seguinte maneira: Considere uma rota qualquer R_q com m consumidores, $R_q = (C_1, \dots, C_m)$, onde C_1 é o primeiro consumidor atendido e C_m é o último. O depósito central é considerado como C_0 e também C_{m+1} . Considere também, $[a_1, b_1]$ e $[a_m, b_m]$ os intervalos que definem a janela de tempo do primeiro e último consumidor, respectivamente. Lembrando que o intervalo corresponde aos horários inicial e final dentre os quais o veículo deve chegar ao consumidor. A chegada pode acontecer antes, porém será necessário esperar para o início do serviço. A viabilidade de inserir um novo consumidor na rota R_q é testada inserindo-o entre cada par de consumidores, sendo a posição de menor aumento na distância percorrida na rota R_q viável selecionada. Se o consumidor C_j é inserido entre o consumidor C_0 e C_1 , o tempo de chegada t_1 ao consumidor C_1 , irá sofrer um deslocamento para frente, podendo causar o mesmo para todos os consumidores seguintes (*Push Forward*). Como o veículo pode estar aguardando em algum consumidor da rota R_q , é possível que o valor do deslocamento neste consumidor seja zero. A partir deste consumidor, o tempo de chegada do veículo nos consumidores seguintes permanece inalterado.

A heurística *PFIH* inicia uma rota com um único consumidor i , escolhido conforme seu custo de inicialização, dado pela equação 10.

$$custo_i = \left[-\alpha d_{0i} + \beta b_i + \gamma \frac{p_i}{360} d_{0i} \right] \quad (10)$$

Onde:

$$\alpha = 0.7; \beta = 0.1; \gamma = 0.2;$$

d_{0i} = distância do depósito central ao consumidor i ;

b_i = limite superior da janela de tempo de chegada ao consumidor i ;

p_i = ângulo da coordenada polar do consumidor i , referente ao depósito central;

Os valores para os parâmetros α , β e γ foram definidos empiricamente por Solomon em [Solomon (1987)].

A partir do primeiro consumidor escolhido, conforme a equação 10, os demais são testados um a um, entre cada posição possível na rota em construção. A posição e o consumidor que resultarem no menor acréscimo da distância total percorrida, sem violação de capacidade e janela de tempo é escolhido. Após não mais haver

consumidores possíveis para inserção na rota em construção, esta é encerrada e inicia-se novamente o mesmo processo com uma nova rota vazia, sendo o primeiro consumidor aquele de menor custo de acordo com o cálculo da fórmula 10, dentre os que ainda não foram roteados.

4 Algoritmo Genético (AG)

Proposto inicialmente por John Holland [Holland (1975)], com intuito de aplicar a teoria da evolução das espécies elaborada por Darwin [Darwin (1859)], ou seja, utilizar os conceitos da evolução biológica, tais como, genes, cromossomos, cruzamento, mutação e seleção em outros problemas de otimização através de algoritmos computacionais.

A idéia básica do Algoritmo Genético pode ser descrita de maneira sucinta:

Algoritmo Genético Clássico

- 1: $P \leftarrow$ População Inicial;
- 2: **enquanto** condição não satisfeita **faça**
- 3: $P' \leftarrow$ Seleção(P);
- 4: $P \leftarrow$ Cruzamentos(P');
- 5: $P \leftarrow$ Mutações(P);
- 6: **fim enquanto**
- 7: Solução \leftarrow Melhor indivíduo(P);

A execução do algoritmo *PFIH* sobre uma determinada instância gera apenas uma solução. Em contrapartida, a população do Algoritmo Genético deve iniciar-se apresentado uma diversidade entre seus indivíduos. Para alcançar tal diversidade, foram utilizadas algumas alterações sobre o algoritmo original do *PFIH*, como sugerido por Alvarenga em [Alvarenga (2004)].

- Variação sobre os parâmetros α , β e γ , dando-lhes valores aleatórios no intervalo de 0 a 1.
- Eliminação da fórmula 10 na determinação da ordem dos indivíduos as serem roteados, construindo esta fila de maneira totalmente aleatória.

Tais estratégias são eficientes na geração de indivíduos diferentes. A segunda é capaz de gerar uma população com maior diversidade genética, mas em consequência disso, traz consigo o fato de produzir, com mais frequência, indivíduos com um alto valor da função objetivo, se comparados com os resultados da fórmula original. Para um bom funcionamento do Algoritmo Genético, fica proposto uma dosagem dentre tais variações para poder assim construir uma população de

variabilidade genética capaz de não se prender facilmente a mínimos locais. Este trabalho utilizou 10% dos indivíduos seguindo a fórmula original do *PFIH*, 60% utilizando a variação sobre os parâmetros α , β e γ , e 30% utilizando a estratégia da fila totalmente aleatória. O tamanho da população foi de 30 indivíduos.

Alguns autores descrevem alterações sobre o modelo do Algoritmo Genético descrito por Holland em [Holland (1975)], como por exemplo o fato do melhor indivíduo da população sobreviver para a próxima geração (elitismo) [Soares (1997)], a fim de não perder seu bom código genético se de alguma maneira este não for selecionado para o cruzamento, que computacionalmente toma por base valores aleatórios para realizar esta operação. Tal estratégia foi utilizada neste trabalho.

Os indivíduos neste trabalho foram representados por um vetor de rotas, onde cada rota possui um vetor de veículos a serem atendidos.

A estratégia de seleção que melhor se adaptou na aplicação do Algoritmo Genético no *PRVJT* foi a roleta. O torneio demonstrou uma convergência muito rápida para mínimos locais, apresentando uma deficiência na variabilidade da população, tornando os indivíduos muito parecidos no decorrer das gerações.

5 Cruzamento

Uma estratégia de cruzamento proposta por Alvarenga em [Alvarenga (2004)] para a resolução do Problema de Roteamento de Veículo com Janela de Tempo se baseia na escolha aleatória de rotas dos indivíduos sendo cruzados, onde estas são inseridas no indivíduo resultante. Se pelo menos um dos consumidores da rota a ser inserida no novo indivíduo pertencer a alguma rota já inserida na solução, esta rota é descartada. Se ao fim das tentativas de inserções de rotas, ainda existirem consumidores fora da solução (pelo fato das intersecções), estes são enxertados na solução através do algoritmo *PFIH*. Vale ressaltar que este operador tenta equilibrar o número de rotas vindos de ambos os pais.

Suponha que a rota número 2 do indivíduo 2 (*I2R2*), com o conjunto de consumidores $\{2, 3, 6, 7\}$, é a primeira escolhida para compor o resultado do cruzamento. Para qualquer rota escolhida do indivíduo 1, sempre existirá uma intersecção entre das rotas *I1R1* e *I1R2* com a rota *I2R2* já presente no indivíduo resultante.

$$I1R1 \cap I2R2 \neq \emptyset \quad (11)$$

$$I1R2 \cap I2R2 \neq \emptyset \quad (12)$$

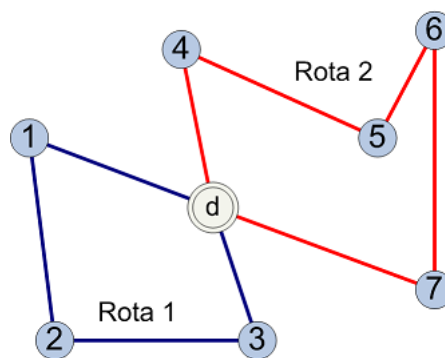


Figura 1: Indivíduo 1

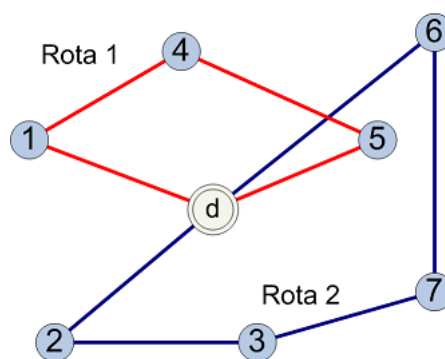


Figura 2: Indivíduo 2

Por este fato, as rotas *I1R1* e *I1R2* são descartadas, e a rota *I2R1* é inserida na solução, completando o indivíduo, sendo exatamente igual a um de seus pais (indivíduo 2).

Esta estratégia de cruzamento pode produzir indivíduos com códigos genéticos exclusivos de ambos os pais, ou ainda, parte do código genético dos pais, e uma parte código genético mutante (enxerto executado pelo algoritmo *PFIH*), ou também ser exatamente igual a um dos pais, como no exemplo desta seção.

6 Mutação

Duas classes de operadores testadas neste trabalho se adaptaram bem ao Problema de Roteamento de Veículos com Janela de Tempo:

- Operadores de troca, transferindo consumidores, um a um;
- Operadores com pontos de corte, transferindo conjuntos de consumidores;

Um operador de troca implementado é determinado pela escolha de uma rota qualquer do indivíduo, onde

nesta, todos os consumidores serão testados em todas as posições das outras rotas, encaixando-o na melhor posição possível caso não viole nenhuma restrição. Este operador pode ser implementado para efetuar a troca somente se encontrar uma melhor solução (intensificação), ou trocar mesmo se isso signifique uma piora na função objetivo (diversificação). Um exemplo de mutação efetuando a troca de consumidores pode ser observado nas figuras 3 e 4.

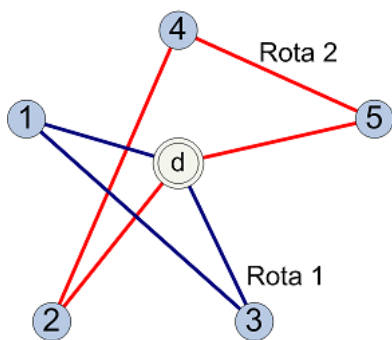


Figura 3: Indivíduo antes da mutação de troca

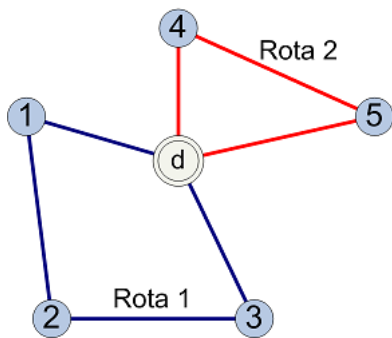


Figura 4: Indivíduo depois da mutação de troca

Suponha que a rota escolhida aleatoriamente seja a rota número 2 da figura 3, tentando encaixar seus consumidores $\{2, 4, 5\}$ na rota 1. Supondo que o algoritmo detecte que os consumidores 4 e 5 não podem migrar para a rota número 1 por violar restrições de tempo ou carga, e o consumidor número 2 pode migrar e ainda oferece um ganho sobre a função objetivo (neste caso, diminuindo a distância). Sendo assim, o consumidor 2 seria encaixado entre os consumidores 1 e 3 da rota número 1, como mostra a figura 4.

Os operadores de mutação do tipo corte tem a capacidade de migrar todo um sub-conjunto de rota. Observe na figura 5 que a rota número 2 contém 2 consumidores $\{3, 4\}$ que podem migrar para a rota número

1, ocasionando uma melhora na função objetivo. Neste caso, este operador efetuará o corte entre os consumidores 4 e 6 da rota número 2, tentando migrar a ramificação $\{3, 4\}$ para as demais rotas, como mostra a figura 6. A ramificação pode não ser transportada com a mesma sequência para a rota número 1. Normalmente é utilizada uma adaptação da heurística *PFIH* para decidir onde os consumidores se encaixarão.

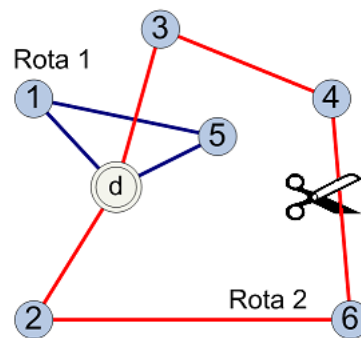


Figura 5: Indivíduo antes da mutação de corte

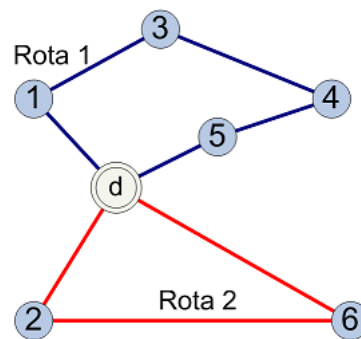


Figura 6: Indivíduo depois da mutação de corte

Note que estes operadores também podem ser aplicados a qualquer classe dos problemas de roteamento de veículos. Como o *PRVJT* possui um alto número de restrições, estes simples operadores não são suficientes para proporcionar uma evolução satisfatória para toda a heterogeneidade das instâncias. Por este motivo, foram desenvolvidas variações para alcançar resultados satisfatórios. Uma dessas variações é o operador com duplo ponto de corte, produzindo 3 fragmentos de rota. Os fragmentos das extremidades são retirados da rota original para o melhor roteamento dos consumidores.

Suponha que uma rota R já atende C consumidores e estes já ocupam toda a capacidade de carga do veículo. Para qualquer outro consumidor roteado nesta rota R , por qualquer operador de mutação, não haverá

intervalo para tal operação sem que haja uma restrição de carga violada. Para tal dificuldade, este trabalho propõe um operador de mutação que retira K consumidores da rota R , com K variando de 1 até o número de consumidores de R . Após esta retirada, pode-se direcionar este conjunto de rotas (solução parcial do problema, pois não contém todos os consumidores) para qualquer outro operador de mutação. Após sua ação, os consumidores previamente retirados da solução devem ser inseridos na mesma, utilizando qualquer heurística de roteamento, por exemplo, o *PFIH*. Esta estratégia na fuga de mínimos locais também é eficiente para restrições temporais dos consumidores (janela de tempo), não se limitando apenas a capacidade de carga dos veículos.

7 Resultados

Para a avaliação dos resultados encontrados neste trabalho, o mesmo foi comparado a publicações utilizando a distância total como primeiro objetivo na minimização do Problema de Roteamento de Veículos com Janela de Tempo ([Taillard et. Al. (1997)], [Rousseau (1999)], [Tan et. Al. (2001)] e [Alvarenga (2004)]). Todos eles foram aplicados sobre as 56 instâncias de Solomon, onde cada um possui 100 consumidores.

As instâncias de Solomon são divididas em seis classes: R1, R2, C1, C2, RC1 e RC2. As instâncias dos tipos R1 e R2 apresentam consumidores com as coordenadas euclidianas totalmente aleatórias. Já as instâncias dos tipos C1 e C2, apresentam os consumidores de maneira aglomerada (em *clusters*). As instâncias dos tipos RC1 e RC2 apresentam um misto das duas primeiras características (esparcos e aglomerados). Uma característica entre os tipos R1, C1 e RC1 é que suas instâncias possibilitam que poucos consumidores sejam atendidos por um veículo, necessitando de um número maior de veículos para atender todo a demanda. Já os tipos R2, C2 e RC2 apresentam na solução poucos veículos, assim atendendo uma grande quantidade de consumidores em cada rota. As instâncias de Solomon podem ser encontradas no site [Instâncias de Solomon].

O algoritmo implementado neste trabalho foi nomeado *AAGI* (Adaptação do Algoritmo Genético 1). Foram colhidas 11 amostras da execução do *AAGI*. As médias dos resultados, agrupadas pelos tipos das instâncias são comparadas com os outros algoritmos que também consideram o minimização de distância, como principal objetivo. Os resultados apresentam o número médio de veículos, distância média percorrida e o percentual a que cada resultado está dos melhores resultados publicados na literatura (com relação a média da distância total).

Tabela 1: Resultados da categoria R1

Trabalho	Veículos	Distância	% Distância
Melhores publicados	12.25	1195.63	-
Taillard (1997)	12.17	1209.35	+1.1475%
Rousseau (1999)	12.83	1201.10	+0.4575%
Tan (2001)	13.83	1260.71	+5.4432%
Alvarenga (2004)	13.33	1200.05	+0.3697%
AAG1 (2004)	13.42	1195.34	-0.0243%

Tabela 2: Resultados da categoria C1

Trabalho	Veículos	Distância	% Distância
Melhores publicados	10.00	828.38	-
Taillard (1997)	10.00	828.38	0.0000%
Rousseau (1999)	10.00	828.38	0.0000%
Tan (2001)	10.11	858.81	+3.6734%
Alvarenga (2004)	10.00	828.38	0.0000%
AAG1 (2004)	10.00	828.38	0.0000%

Tabela 3: Resultados da categoria RC1

Trabalho	Veículos	Distância	% Distância
Melhores publicados	13.00	1344.89	-
Taillard (1997)	11.50	1389.22	+3.2962%
Rousseau (1999)	12.50	1370.26	+1.8864%
Tan (2001)	13.63	1447.06	+7.5969%
Alvarenga (2004)	13.00	1344.89	0.0000%
AAG1 (2004)	13.62	1366.90	+1.6366%

Tabela 4: Resultados da categoria R2

Trabalho	Veículos	Distância	% Distância
Melhores publicados	4.64	910.17	-
Taillard (1997)	2.82	980.27	+7.7019%
Rousseau (1999)	3.18	966.94	+6.2373%
Tan (2001)	3.82	1058.52	+16.2992%
Alvarenga (2004)	4.64	910.17	0.0000%
AAG1 (2004)	4.64	880.53	-3.2565%

Tabela 5: Resultados da categoria C2

Trabalho	Veículos	Distância	% Distância
Melhores publicados	3.00	589.86	-
Taillard (1997)	3.00	589.86	0.0000%
Rousseau (1999)	3.00	594.01	+0.7036%
Tan (2001)	3.25	617.10	+4.6180%
Alvarenga (2004)	3.00	589.86	0.0000%
AAG1 (2004)	3.00	589.86	0.0000%

Este trabalho superou ou igualou os melhores algoritmos da literatura nas classes de problemas R1, C1, R2, C2 e RC2, sendo inferior somente na classe RC1.

Um dos testes efetuados durante o projeto foi a execução do algoritmo genético sem o operador de cruza-

Tabela 6: Resultados da categoria RC2

Trabalho	Veículos	Distância	% Distância
Melhores publicados	6.00	1032.88	-
Taillard (1997)	3.38	1117.44	+8.1868%
Rousseau (1999)	3.75	1113.29	+7.7850%
Tan (2001)	7.00	1169.41	+13.2184%
Alvarenga (2004)	6.00	1032.88	0.0000%
AAG1 (2004)	5	1018.02	-1.1437%

mento proposto por Alvarenga em [Alvarenga (2004)] (explicado na seção 5 deste trabalho). Este teste merece destaque pois demonstra que tal operador não surte efeitos benéficos perante os resultados, ou seja, não traz vantagens sobre o objetivo do problema (minimização da distância total). Note que esta afirmação não condena tal operador como sendo inútil dentro de qualquer algoritmo genético, mas sim que dentro do contexto deste sistema, ele não destacou-se perante os resultados finais. Por este motivo, este trabalho foi nomeado de "Adaptação do Algoritmo Genético 1", pois é executado somente com mutações. A seguir temos a tabela com a média dos resultados, categorizados pelas classes de problemas, representando a execução do algoritmo com o cruzamento (AGI) e a execução do algoritmo sem o cruzamento (AAGI).

Tabela 7: Comparativo entre AGI e AAGI

Classe	AGI		AAGI	
	Veículos	Distância	Veículos	Distância
R1	13.33	1201.69	13.42	1195.34
C1	10.00	828.38	10.00	828.38
RC1	13.25	1385.51	13.62	1366.90
R2	3.82	897.92	4.64	880.53
C2	3.00	589.86	3.00	589.86
RC2	4.25	1035.07	5	1018.02

Ambos os algoritmos foram limitados a executar durante 30 minutos para cada instância. Nota-se uma melhor média de veículos na execução do AGI se comparado ao AAGI, mas em todos os casos foi igual ou inferior na distância total percorrida.

Para a implementação dos algoritmos AGI e AAGI utilizou-se a linguagem de programação Java, mais especificamente a plataforma J2SE [J2SE] na versão 5.0 sobre o sistema operacional Slackware 10.0 [Slackware] com o Kernel Linux na versão 2.4.26.

8 Conclusão

Este trabalho apresenta uma importante colaboração para o Tratamento do Problema de Roteamento de Veículos

com Janela de Tempo. Os resultados sobre as instâncias de Solomon alcançados pelo algoritmo AAGI podem ser considerados expressivos, já que superam ou igualam os melhores resultados da literatura em cinco das seis classes do problema.

Um fato a se destacar, é a adaptação do Algoritmo Genético, que se comportou bem perante o problema, mesmo não efetuando nenhum tipo de cruzamento durante a execução do algoritmo, sendo representado pela simplificação:

Algoritmo Genético Simplificado

- 1: $P \leftarrow$ População Inicial;
- 2: **enquanto** condição não satisfeita **faça**
- 3: $P' \leftarrow$ Seleção(P);
- 4: $P \leftarrow$ Mutações(P');
- 5: **fim enquanto**
- 6: Solução \leftarrow Melhor indivíduo(P);

Esta simplificação não propõe a não utilização da operação de cruzamento para o PRVJT, nem mesmo afirma que o operador aqui utilizado é inútil, mas sim declara que no contexto deste trabalho, o operador implementado não resultou em melhoras sobre os resultados finais.

Referências

- [Alvarenga (2004)] G.B. Alvanrega, G. R. Mateus. *A Two-Phase Genetic and Set Partitioning Approach for the Vehicle Routing Problem with Time Windows*, Fourth International Conference on Hybrid Intelligent Systems (HIS04), IEEE Computer Society Press, (2004).
- [Bodin et.Al. (1983)] Bodin, L., Golden, B., Assad, A., Ball, M. *Routing and scheduling of vehicles and crews - the state of the art*, Comput. Oper. Res. 10, 63-211, (1983).
- [Darwin (1859)] Charles Darwin; *On The Origin of Species*, 1st edition, Harward University Press, MA, (1859).
- [Fisher et.Al. (1997)] Marshall L. Fisher; Kurt O. J'ornsteen; Oli B. G. Madsen *Vehicle routing with time windows: Two optimization algorithms*, Operations Research, 45(3):488 - 492, (1979).
- [Holland (1975)] J. H. Holland; *Adaptation in natural and artificial systems*, (1975).
- [Instâncias de Solomon] Solomon's 100 customers Problems Instances (1987),

- URL: [http://neo.lcc.uma.es/radi-
aeb/WebVRP/data/instances/solomon/
solomon_100.zip](http://neo.lcc.uma.es/radi-
aeb/WebVRP/data/instances/solomon/
solomon_100.zip),
27 de Outubro de 2004.
- [J2SE] Sun microsystems - Java Virtual Machine 5.0,
URL: <http://java.sun.com>,
19 de Novembro de 2004.
- [Johnson et.Al. (1979)] R. Michael Garey; David
Johnson; *Computers and intractably: A guide to
the theory of \mathcal{NP} -completeness*, (1979).
- [Larsen (1999)] J. Larsen; *Parallelization of the vehi-
cle routing problem with time windows*, Phd The-
sis, Department of Mathematical Modelling, Te-
chnical University of Denmark, (1999);
- [Rousseau (1999)] L. M. Rousseau, M. Gendreau and
G. Pesant; *Using Constraint-based Operators
with Variable Neighborhood Search to Solve the
Vehicle Routing Problem with Time Windows*, Pre-
sented at the CP-AI-OR'99 Workshop, February
25.-26., University of Ferrara, Italy, (1999).
- [Slackware] Slackware 10.0, kernel linux 2.4.26, URL:
<http://www.slackware.com>,
19 de Novembro de 2004.
- [Soares (1997)] Soares, Gustavo Luís. *Algoritmo Ge-
néticos: Estudo, Novas Técnicas e Aplicações*,
Dissertação de Mestrado - CPDEE - UFMG,
(1997).
- [Solomon (1987)] Marius M. Solomon; *Algorithms for
the Vehicle Routing Problem and Scheduling Pro-
blem with Time Window Constraints*, (1987).
- [Taillard et.Al. (1997)] E. Taillard, P. Badeau, M. Gen-
dreau, F. Geurtin, and J.Y. Potvin; *A Tabu Search
Heuristic for the Vehicle Routing Problem with
Time Windows*, Transportation Science, 31, 170-
186, (1997).
- [Tan et.Al. (2001)] K.C. Tan, L.H. Lee, Q.L. Zhu and
K. Ou *Heuristic methods for vehicle routing pro-
blem with time windows*, Artificial Intelligence in
Engineering, 15, 281-295, (2001).