

## OPERADORES DE RECOMBINAÇÃO PARA UM ALGORITMO GENÉTICO HÍBRIDO APLICADO AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO

**Thiago Muniz Stehling**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte, Minas Gerais, Brasil  
thiagostehling@gmail.com

**Sérgio Ricardo de Souza**

Departamento de Computação - Centro Federal de Educação Tecnológica de Minas Gerais  
Belo Horizonte, Minas Gerais, Brasil  
sergio@dppg.cefetmg.br

**Thiago Ferreira de Noronha**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais  
Belo Horizonte, Minas Gerais, Brasil  
tfn@dcc.ufmg.br

### RESUMO

Este artigo avalia diferentes operadores genéticos propostos para solucionar o Problema de Roteamento de Veículos com Janela de Tempo (VRPTW, do inglês, *Vehicle Routing Problem with Time Windows*). Os operadores analisados foram testados em um conjunto clássico de instâncias para VRPTW. Os resultados obtidos por cada operador foram analisados estatisticamente e os pontos positivos e negativos de cada operador foram identificados. Por fim, os experimentos computacionais mostram que os melhores resultados foram obtidos pelo operador que realiza cruzamento a partir de troca de rotas.

**PALAVRAS CHAVE.** Problema de Roteamento de Veículos. Algoritmo Genético. Operadores Evolutivos.

**Tópicos(Otimização Combinatória, Algoritmos Evolutivos)**

### ABSTRACT

This paper is concerned with a comparison between different crossover operators proposed for solving the Vehicle Routing Problem with Time Windows. The analyzed operators were tested on a set of well-known benchmark instances for this problem. The obtained results by each operator were statistically analyzed and the strengths and weaknesses of each operator have been identified. Finally, the computational experiments show that the best-known results were obtained by the crossover operator which performs exchange of routes.

**KEYWORDS.** Vehicle Routing Problem. Genetic Algorithm. Crossover Operators.

**Paper topics(Combinatorial Optimization, Evolutionary Algorithms)**

## 1. Introdução

A principal diferença entre VRPTW e o problema clássico de roteamento é a restrição de janela de tempo. Sendo assim, o primeiro é capaz de representar situações nas quais o tempo de atendimento é considerado uma restrição forte. VRPTW é um problema que visa minimizar o custo total das rotas geradas entre o depósito e os demais consumidores [Gendreau e Tarantilis, 2010].

Formalmente, VRPTW pode ser definido através de um grafo completo  $G(V, A)$ . O conjunto de vértices  $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$  corresponde aos consumidores que necessitam de atendimento e o conjunto de arestas  $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$  representa elos entre cada par distinto de consumidores. Os vértices  $v_0$  e  $v_{n+1}$  referem-se ao depósito. Cada vértice do grafo, incluindo o depósito, possui uma demanda  $D_i$  a ser atendida, um tempo de serviço  $TS_i$  referente ao trabalho a ser realizado, a janela de tempo inicial  $JTI_i$  e a janela de tempo final  $JTF_i$ , indicando o intervalo de tempo no qual o atendimento deve ser iniciado.

Cada consumidor deve ser atendido por um único veículo  $k$ , com exceção do depósito. A capacidade de carga  $Q$  de cada veículo é constante. Logo, a frota de veículos é homogênea. O tempo de deslocamento do veículo  $k$  entre os consumidores  $v_i$  e  $v_j$  é considerado custo do percurso. Em outras palavras, cada aresta  $(v_i, v_j)$  possui um custo  $C_{ij}$  associado.

Algoritmo Genético (AG) é um método estocástico de busca populacional inspirado em processos naturais de evolução [Holland, 1975]. Tal evolução é proporcionada pelo operador de cruzamento responsável pela reprodução dos indivíduos da população e pelo operador de mutação que gera mudanças inesperadas nesses indivíduos [Goldberg, 1989]. O objetivo deste artigo é avaliar quatro operadores de cruzamento dentro do escopo de um AG híbrido aplicado a VRPTW. Por pertencer à classe de problemas NP-Difícil, algoritmos exatos podem solucionar VRPTW, no pior caso, em tempo exponencial [Cordeau et al., 2001]. Além disso, esse tempo aumenta consideravelmente de acordo com o tamanho das instâncias do problema. Pela união desses fatos, justifica-se a utilização de AG para resolver esse problema.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos correlatos sobre operadores de cruzamento para VRPTW; a Seção 3 mostra a estrutura do AG híbrido proposto neste trabalho; cada adaptação realizada nos operadores de cruzamento abordados neste artigo está discriminada na Seção 4; a Seção 5 mostra os resultados obtidos pelo algoritmo proposto, separados por operador genético utilizado; a análise estatística sobre esses resultados está demonstrada na Seção 6; e a Seção 7 conclui este artigo.

## 2. Revisão de Literatura

Prins [2004] utilizou um algoritmo evolutivo para solucionar o Problema de Roteamento de Veículos clássico. O operador adotado por esse autor é o *Ordered Crossover* (OX). Em OX, dois pontos de corte são selecionados aleatoriamente e o gene do pai  $P_1$ , localizado entre os pontos de corte, é copiado para o filho. O restante do filho é preenchido de acordo com o gene remanescente do pai  $P_2$ . A ordem de precedência dos genes de  $P_2$  é mantida. Outro filho é gerado executando as mesmas etapas. Porém, o gene copiado é de  $P_2$  e o restante é preenchido de acordo com as características de  $P_1$ .

Ombuki et al. [2006] propuseram um AG multiobjetivo para resolver VRPTW. O operador adotado na fase de recombinação é o *Best Cost Route Crossover* (BCRC). Esse operador é baseado na escolha da rota de melhor custo. Inicialmente, BCRC seleciona dois cromossomos pais,  $P_1$  e  $P_2$ . Após a identificação dos pontos de corte, aleatoriamente, um gene de cada pai é escolhido. Considerando os filhos gerados,  $O_1$  e  $O_2$ , a parte removida de  $O_2$  é um gene de  $P_1$  e a removida de  $O_1$  é um gene de  $P_2$ . Por fim, os genes removidos são inseridos novamente, de maneira aleatória, nos respectivos filhos.

Zhou et al. [2013] apresentaram um AG aplicado ao Problema de Roteamento de Veículos Baseado em Rotas Balanceadas. O operador utilizado por esses autores é o *One-Point Crossover* (OPC). Em OPC, um único ponto de corte é selecionado aleatoriamente dos pais  $P_1$  e  $P_2$ . Um gene de  $P_1$  é sorteado, com mesma probabilidade de escolha. Esse gene é copiado no filho. O restante do

filho é preenchido pelo  $P_2$ , tentando manter a mesma ordem de precedência do pai. A viabilidade é verificada no momento de inserção dos genes de  $P_2$ . Outro filho é gerado pela inversão dos papéis dos pais no processo descrito.

Chiang e Hsu [2014] adotaram um algoritmo evolutivo como estratégia para solucionar uma versão multiobjetiva de VRPTW. Neste trabalho, o operador proposto por esses autores é identificado como *Exchange Route Crossover* (ERC). A primeira etapa desse operador é a criação de cópias dos pais. Essas cópias representam os filhos. A segunda etapa seleciona a melhor rota para cada filho, ou seja, aquela que possui a menor distância percorrida. Em seguida, essas melhores rotas são trocadas entre os filhos. Consumidores repetidos são retirados das rotas já existentes. Por fim, a pior rota é removida de cada filho. Os consumidores da rota retirada são inseridos novamente de acordo com a Menor Distância Extra (MDE), que é calculada pela seguinte equação:

$$MDE_{kij} = (D_{ik} + D_{kj} - D_{ij}) \quad (1)$$

MDE define que o custo de inserção de um consumidor  $k$  entre os consumidores  $i$  e  $j$  é uma substituição do custo do percurso entre  $i$  e  $j$  pelo custo de deslocamento do veículo iniciado em  $i$ , passando por  $k$  e finalizado no consumidor  $j$ . O objetivo desse operador é garantir que os filhos herdem as boas características dos pais e eliminem as deficiências apresentadas nos mesmos [Chiang e Hsu, 2014].

### 3. Algoritmo Proposto

O AG híbrido proposto neste artigo possui uma estrutura tradicional. A população é composta por 100 indivíduos e estes são submetidos ao processo evolutivo composto pelas seguintes fases: seleção, recombinação e mutação. A taxa de recombinação é de 90% e a taxa de mutação é de 5%. A taxa de cruzamento é alta para que a influência do operador genético seja avaliada. Essas taxas não sofrem alteração durante as gerações do algoritmo.

#### 3.1. População Inicial

A população inicial é formada pela combinação entre *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Push-Forward Insertion Heuristic* (PFIH). Idealizada para VRPTW, PFIH é uma heurística de inserção gulosa que ordena cada um dos consumidores de modo crescente [Solomon, 1987]. Sendo assim, PFIH insere primeiro o consumidor que apresenta menor custo de inserção. Em seguida, insere os demais, até o de maior custo.

GRASP é uma ponderação entre busca gulosa e aleatória [Feo e Resende, 1995]. No âmbito de VRPTW, a característica gulosa é garantida através da seleção dos  $n$  melhores consumidores. Esses  $n$  elementos agrupados formam a Lista Restrita de Candidatos (LRC) [Feo e Resende, 1995]. Neste artigo, PFIH é utilizada para garantir a característica gulosa da fase de construção de GRASP. O Algoritmo 1 mostra como a população inicial é gerada.

---

#### Algorithm 1: População inicial do algoritmo proposto

---

**Input:** Informações da instância de VRPTW

**Output:** População Inicial

```

1 PopulacaoInicial ← ∅;
2 while Tamanho(PopulacaoInicial) < Limite do
3   | s ← ConstrucPFIH();
4   | s' ← BuscaLocal(s);
5   | if s' é melhor que s then
6     |   Insere(s', PopulacaoInicial);
7   | end
8 end
9 return PopulacaoInicial;
```

---

No Algoritmo 1, a linha 3 indica a fase de construção de GRASP. A solução é gerada de maneira gulosa por PFIH. Ou seja, os 5 consumidores com menor custo de inserção serão inseridos primeiro em LRC. Porém, para satisfazer a estocasticidade do método, um sorteio é executado dentre os consumidores contidos nessa lista. O tamanho de LRC foi definido empiricamente e cada consumidor possui a mesma probabilidade de ser sorteado.

Na linha 4 do Algoritmo 1, a solução gerada pela fase anterior de GRASP é refinada pela fase de busca local. A estrutura de vizinhança dessa fase é identificada através dos seguintes movimentos: (i) Intra-Troca - trocar dois clientes de posição em uma mesma rota; (ii) Intra-Inserção - inserir um cliente em uma determinada posição em uma mesma rota; (iii) Inter-Troca - trocar dois clientes de posição de rotas diferentes; (iv) Inter-Inserção - inserir um cliente em uma determinada posição em outra rota; e (v) Exclusão-Rota - eliminar rota.

No movimento Intra-Troca, todos os possíveis pares de consumidores dentro de uma mesma rota são trocados de posição. No movimento Intra-Inserção, cada consumidor dentro de uma mesma rota é inserido em uma posição diferente da qual originalmente estava localizado. O movimento Inter-Troca funciona de maneira similar ao Intra-Troca. Porém, existe uma diferença: todos os possíveis pares de consumidores contidos em rotas diferentes são trocados de posição. O movimento Inter-Inserção funciona de maneira similar ao Intra-Inserção. No entanto, cada consumidor é inserido em uma posição localizada em uma rota diferente da qual o mesmo é originado.

O movimento Exclusão-Rota visa eliminar uma rota da solução corrente  $s$ . O objetivo é esvaziar uma determinada rota inserindo seus consumidores em rotas diferentes. Porém, excluir diversas rotas de uma solução pode gerar uma grande alteração, caracterizando uma perturbação e não um movimento. Para evitar esse problema, o Exclusão-Rota inicia sua execução primeiro nas menores rotas. Em ordem crescente de tamanho, cada rota é submetida ao movimento. No entanto, após a primeira rota eliminada, o Exclusão-Rota é encerrado.

Este último movimento é sempre executado a cada iteração de GRASP. Dentre os demais movimentos, apenas um é executado por iteração. Um sorteio é realizado para saber qual movimento será executado, sendo que cada um apresenta a mesma probabilidade de escolha. O Algoritmo 1 é executado até que a população inicial atinja seu tamanho previamente definido. Esta será a população contida na primeira geração do algoritmo proposto.

### 3.2. Processo Evolutivo

A seleção dos cromossomos é realizada através do método da roleta. No âmbito de VRPTW, a diversidade dos indivíduos é aprimorada com a utilização desse método. A seleção pela roleta distribui uma probabilidade de escolha para cada indivíduo da população de acordo com a sua aptidão. Esse fato indica que os indivíduos mais aptos têm mais chance de serem escolhidos para a recombinação. Porém, não exclui a alternativa de indivíduos menos aptos serem selecionados. Ombuki et al. [2006] propuseram uma Função de Avaliação (FA) baseada no método da soma ponderada. FA é utilizada neste artigo para avaliar a qualidade das soluções geradas pelo algoritmo proposto e é dada por:

$$FA = \lambda \times N + \delta \times DT \quad (2)$$

Os parâmetros de peso  $\lambda$  e  $\delta$  estão associados, respectivamente, ao número de veículos  $N$  e à distância total percorrida  $D$ . Esses pesos foram definidos empiricamente por Ombuki et al. [2006] e apresentam os seguintes valores:  $\lambda = 100$  e  $\delta = 0,001$ . Neste artigo, tais parâmetros não se alteram em tempo de execução.

O Algoritmo 2 mostra o restante do pseudocódigo do algoritmo proposto neste trabalho. É importante ressaltar que, apesar de ter sido dividido em duas partes, o AG híbrido é um único algoritmo. Ou seja, após a linha 8 do Algoritmo 1, imediatamente, é executada a linha 1 do Algoritmo 2.

---

**Algorithm 2:** Processo evolutivo do algoritmo proposto

---

**Input:** População Inicial  
**Output:** Melhor indivíduo da população

```

1 Evolucao ← ∅;
2 Criterio ← Avalia(PopulacaoInicial);
3 while Criterio não satisfeito do
4   Evolucao ← PopulacaoInicial;
5   while Tamanho(Evolucao) < (2 × Limite) do
6     p ← Selecao(PopulacaoInicial);
7     p' ← Recombinacao(p);
8     p' ← Mutacao(p');
9     if p' ≠ p then
10      | Inserere(p', Evolucao);
11    end
12  end
13  PopulacaoInicial ← ∅;
14  p* ← OperadorElitismo(Evolucao);
15  Inserere(p*, PopulacaoInicial);
16  Inserere(Recombinacao(p*), PopulacaoInicial);
17  PopulacaoInicial ← Sobrevivencia(Evolucao);
18  Criterio ← Avalia(PopulacaoInicial);
19  Evolucao ← ∅;
20 end
21 return Melhor indivíduo da PopulacaoInicial;

```

---

Na linha 8 do Algoritmo 2, está localizado o operador de mutação. O *Constrained Route Reversal Mutation* (CRRM) foi proposto por Ombuki et al. [2006]. Em CRRM, o gene a ser invertido é aleatoriamente determinado. Devido às restrições de janela de tempo, esse gene apresenta um limite de tamanho, contido no intervalo [2, 3].

Durante o processo evolutivo do algoritmo, a população de indivíduos dobra de tamanho. Em outras palavras, 100 indivíduos pais são somados com os 100 indivíduos filhos, formando uma única população de 200 cromossomos. Porém, apenas a metade destes prossegue para a próxima geração. A sobrevivência é executada na linha 17 do Algoritmo 2 baseada no método da roleta.

Como pode ser observado nas linhas 14 e 15 do Algoritmo 2, uma segunda estratégia aplicada é a inserção arbitrária do melhor indivíduo da população, denominado  $p^*$ , na próxima geração do algoritmo. Esse fato permite ao mesmo ser considerado elitista sem aumentar a pressão seletiva. Essa estratégia se torna interessante uma vez que a seleção realizada através do método da roleta não garante que o melhor indivíduo prossiga no processo evolutivo.

Outra estratégia é uma recombinação entre o  $p^*$  e um segundo pai selecionado de maneira aleatória. O sorteio é realizado entre todos os indivíduos da população, ou seja, o próprio  $p^*$  pode ser selecionado. Cada cromossomo apresenta a mesma probabilidade de escolha nessa etapa descrita pela linha 16 do Algoritmo 2. Os dois filhos gerados por essa recombinação pontual também são inseridos arbitrariamente na próxima geração.

Um fato importante é que, devido à inserção do  $p^*$  e dos dois filhos gerados pela recombinação anteriormente descrita, apenas 97 dentre os 200 indivíduos da população sobrevivem para a próxima geração do algoritmo. O processo evolutivo é executado até que o critério de parada previamente definido seja satisfeito. Neste artigo, esse critério é uma união entre qualidade dos cromossomos e número de gerações sem melhora.

#### 4. Operadores Genéticos

Como dito anteriormente, o objetivo deste artigo é avaliar quatro operadores de cruzamento dentro do escopo de um AG híbrido aplicado a VRPTW. Nesta seção, cada um desses operadores genéticos serão discriminados. Além disso, adaptações realizadas nos mesmos serão explicadas.

##### 4.1. Cruzamento Ordenado Máximo e Mínimo (COMM)

COMM é inspirado no operador de Prins [2004], que foi proposto originalmente por Davis [1985]. A adaptação feita em COMM está no gene a ser cruzado. A Figura 1 mostra o funcionamento desse operador através de uma exemplo fictício de VRPTW.

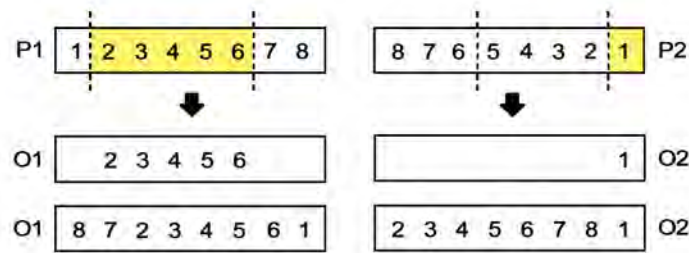


Figura 1: Funcionamento de COMM

Na Figura 1, os pontos de corte em ambos os pais são selecionados aleatoriamente. No âmbito de VRPTW, o gene é um trecho da rota. Em OX, o trecho escolhido era o gene contido entre os pontos de corte. Em COMM, de  $P_1$  é selecionado o maior gene e de  $P_2$  o menor gene. Esses genes são copiados para os respectivos filhos.

Os consumidores restantes de  $O_1$  são preenchidos pelos consumidores de  $P_2$  e os de  $O_2$  pelos de  $P_1$ . A ordem de precedência é mantida. Porém, se ocorrer infactibilidade devido à janela de tempo, por exemplo, esse consumidor é realocado dentro da rota. Na Figura 1, isso pode ser observado pelo consumidor 7 no filho  $O_1$ . A ideia dessa adaptação é gerar um filho com o máximo de características de um pai e outro com o mínimo destas.

##### 4.2. Cruzamento da Rota de Maior Aptidão (CRMA)

CRMA é uma variação do operador proposto inicialmente por Ombuki et al. [2006]. A adaptação realizada nesse operador consiste na avaliação dos pais. O gene a ser cruzado é escolhido com base em FA. Do melhor pai é selecionado o maior gene e do pior é selecionado o menor. A Figura 2 exemplifica o funcionamento de CRMA.

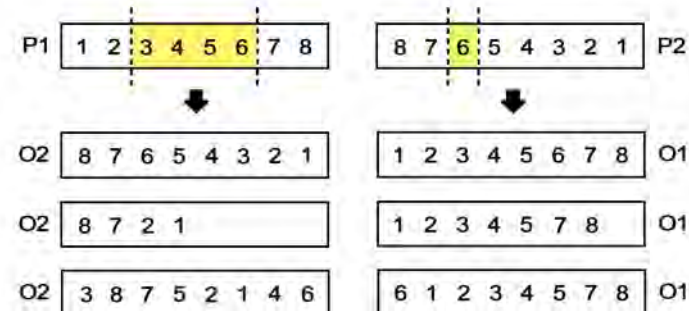


Figura 2: Funcionamento de CRMA

Pela Figura 2, o melhor pai é  $P_1$ . O filho  $O_2$ , inicialmente, é uma cópia de  $P_2$ . O gene selecionado de  $P_1$  será recombinado em  $O_2$ . Cada consumidor contido entre os pontos de corte de

$P_1$  é retirado de  $O_2$  e inserido novamente na melhor posição possível. Nesse momento, as restrições de VRPTW são levadas em consideração e a melhor solução viável gerada é escolhida e atribuída ao filho  $O_2$ . O mesmo ocorre em  $O_1$ , porém, com o gene derivado de  $P_2$ . A ideia dessa adaptação é proporcionar maior alteração no filho correspondente ao pior pai e uma menor modificação no correspondente ao melhor pai.

### 4.3. Cruzamento Duplo de Um-Ponto (CDUP)

CDUP é uma variação do operador proposto por Langdon e Poli [1997], que, por sua vez, é uma adaptação do operador de Davis [1985]. A modificação feita em CDUP consiste na geração de dois filhos com os mesmos pais selecionados. A Figura 3 mostra o funcionamento desse operador através de um exemplo fictício de VRPTW.

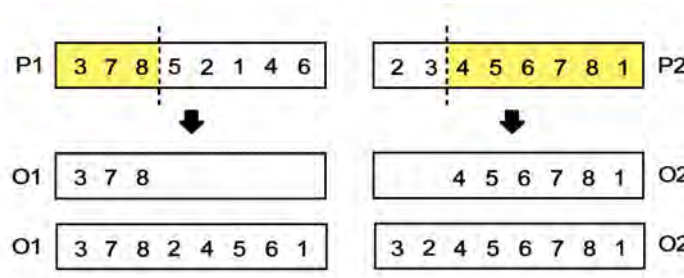


Figura 3: Funcionamento de CDUP

Na Figura 3, após a identificação do ponto de corte, um gene de cada pai é escolhido aleatoriamente. Esse gene é copiado para seu respectivo filho. O restante do filho é derivado do outro pai. Ao invés de repetir o processo, selecionando novos pais e gerando uma nova prole, CDUP utiliza os mesmos passos. Porém, o filho  $O_2$  recebe o gene selecionado de  $P_2$  e o restante do gene de  $P_1$ . A ideia dessa adaptação é gerar o dobro de filhos que OPC produz sem a seleção de novos pais.

### 4.4. Cruzamento de Troca de Rotas (CTR)

CTR é inspirado no operador aprimorado proposto por Chiang e Hsu [2014], originalmente apresentado por Tan et al. [2006]. A versão aprimorada realiza trocas entre as melhores e piores rotas dos pais para gerar filhos mais aptos. A adaptação feita em CTR consiste na maneira de avaliar essas rotas. Nesse operador, a melhor rota apresenta o melhor valor de FA. Logo, a pior rota apresenta o pior valor dessa função. A Figura 4 mostra o funcionamento de CTR.

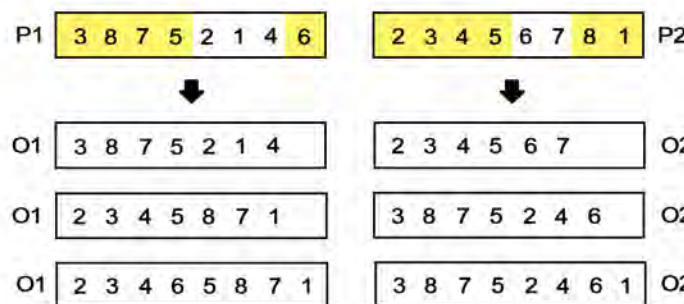


Figura 4: Funcionamento de CTR

Na Figura 4, a melhor rota de  $P_1$  é  $\{3, 8, 7, 5\}$  e a pior rota é  $\{6\}$ . Estas rotas foram classificadas de acordo com FA. O próximo passo de CTR é retirar a pior rota de  $O_1$ . Em seguida,

a melhor rota de  $P_2$  é inserida em  $O_1$ . Consumidores repetidos são retirados e os demais são ajustados. O último passo é reinserir os consumidores que foram retirados da pior rota. Essa inserção é realizada através de MDE. Os mesmos passos são executados para o filho  $O_2$ . Porém, a melhor rota é derivada de  $P_1$ . A ideia dessa adaptação é aprimorar a maneira como as rotas são classificadas.

## 5. Resultados Computacionais

O AG híbrido proposto neste artigo foi implementado em linguagem C, no ambiente Dev C++ e compilado com o GCC 4.7.2. Os experimentos computacionais foram realizados em um único processador Intel Core i7, 2,40 Ghz com 8 GB de memória RAM. O sistema operacional utilizado foi o Windows 10.

Quatro versões do algoritmo proposto foram analisadas. Elas se diferem apenas pelo operador de recombinação utilizado. Portanto, no Algoritmo 2, as únicas alterações realizadas se encontram nas linhas 7 e 16. Cada versão é denominada de acordo com o operador de cruzamento utilizado: COMM, CRMA, CDUP e CTR. Devido à estocasticidade do método, cada versão foi executada 50 vezes, variando-se a semente do gerador de números aleatórios utilizado.

Para avaliar as quatro versões, foram utilizadas as 56 instâncias de Solomon [1987] de 100 consumidores. Esse conjunto de instâncias foi criado especificamente para VRPTW e é distribuído em 3 grandes classes: C, R e RC. A primeira apresenta consumidores que possuem coordenadas geográficas distribuídas no espaço de maneira agrupada, ou seja, em *clusters*. A segunda grande classe apresenta consumidores distribuídos no espaço geográfico de maneira aleatória. Finalmente, a terceira apresenta uma mescla de características das classes anteriores. Além disso, cada classe é subdividida em 2 grupos: C1 e C2, R1 e R2 e RC1 e RC2. Por exemplo, no grupo C1, os consumidores possuem janela de tempo curta e no grupo C2 apresentam janela de tempo larga. As mesmas características valem para os respectivos grupos.

Na Tabela 1 são apresentados os resultados obtidos por cada versão do algoritmo proposto. Em cada coluna, há uma subdivisão na qual o número de veículos utilizados fica à esquerda e a distância total percorrida fica à direita. As 56 instâncias estão distribuídas em cada um dos grupos anteriormente explicados. Portanto, os resultados demonstrados são os resultados médios obtidos. Na última linha da tabela, diferentemente das demais linhas, é mostrado o resultado total obtido, ou seja, o somatório do número de veículos utilizados e da distância total percorrida por todas as instâncias. Além disso, a coluna identificada por *Best-known* apresenta os melhores resultados conhecidos para essas instâncias. Esses resultados não são obtidos por um único trabalho. Quando algum resultado é melhorado, a veracidade é avaliada e este é atualizado no seguinte endereço eletrônico: <http://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/>.

Tabela 1: Resultados médios por grupo e totais obtidos

	<i>Best-known</i>		CDUP		COMM		CRMA		CTR	
C1	10	828,38	10	828,38	10	828,38	10	828,38	10	828,38
C2	3	589,86	3	589,86	3	589,86	3	589,86	3	589,86
R1	11,92	1210,34	12,17	1217,53	12,17	1217,53	12,17	1217,53	11,92	1210,34
R2	2,73	951,03	2,91	958,33	2,91	958,33	2,91	958,33	2,73	951,03
RC1	11,5	1384,16	12	1402,69	12	1402,69	11,88	1401,18	11,75	1396,48
RC2	3,25	1119,24	3,75	1131,14	3,63	1129,25	3,5	1124,41	3,5	1126,02
<b>Total</b>	405	56352,70	418	57596,93	417	57581,77	415	57530,98	409	57339,64

De acordo com a Tabela 1, as versões CDUP, COMM e CRMA ficaram distantes dos melhores conhecidos. A primeira utilizou 13 veículos a mais, a segunda ficou com uma diferença de 12 veículos de ótimo e a terceira versão utilizou 10 veículos a mais que os melhores alcançados. Além disso, essas versões tiveram dificuldades em 4 grupos de instâncias: R1, R2, RC1 e RC2. Esse fato indica uma deficiência dessas versões ao lidar com consumidores aleatoriamente distribuídos. Já a



versão CTR atingiu bons resultados, utilizando apenas 4 veículos a mais que o melhor conhecido. A maior dificuldade dessa versão está nos grupos onde os consumidores possuem características mescladas.

## 6. Análise Estatística

As primeiras estatísticas descritivas analisadas foram as medidas de tendência central e de variabilidade dos dados gerados. O objetivo dessa análise é comparar o impacto dos operadores genéticos nas soluções finais. Essas soluções são mostradas em relação à qualidade das mesmas, que é estimada através da FA definida pela Equação (2). Quanto mais próximo de 1 for o valor de FA, melhor é a solução. Esse fato indica que a FA foi convertida em um problema de maximização. Cada *boxplot* da Figura 5 auxilia na interpretação visual dos dados gerados.

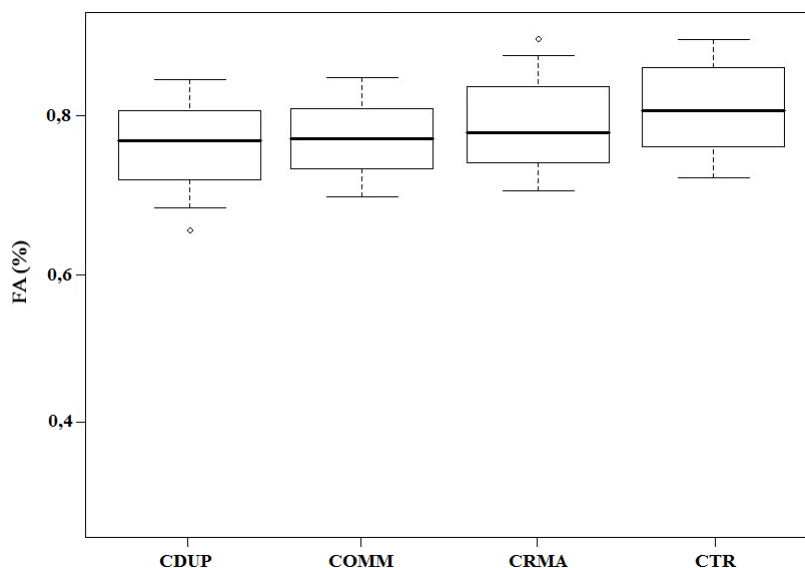


Figura 5: *Boxplot* referente à qualidade das soluções finais obtidas em cada versão

Pode-se observar na Figura 5 que a mediana foi aumentando em cada versão. No cenário proposto, a mediana aumentar é um fator positivo, pois FA refere-se a um problema de maximização. Outra característica positiva é a concentração da maior parte dos dados no quartil superior, ou seja, acima da mediana, principalmente em CRMA e em CTR. Nota-se também a melhora gradativa do *whisker* superior, isso indica que as soluções estão cada vez melhores.

Visualmente, várias informações podem ser retiradas de cada *boxplot* da Figura 5. Porém, como existe sobreposição entre os dados dos mesmos, nada pode ser concluído sobre qual versão obtém os melhores resultados. Portanto, faz-se necessário a realização de algum teste estatístico para avaliar se há diferença entre os tratamentos.

O teste estatístico ANOVA permite estimar médias de populações [Montgomery e Runger, 2009]. Essa análise verifica se existem diferenças significativas entre tais médias e avalia se determinados fatores influenciam em alguma variável do problema. Antes de aplicar o ANOVA nos dados gerados, as premissas necessárias para realização do teste foram avaliadas: (i) a normalidade dos resíduos, ou seja, verificar se as populações apresentam uma distribuição normal; (ii) a homocedasticidade dos erros, em outras palavras, analisar se a variância dos erros é constante; e (iii) a independência e a aleatoriedade da amostra.

Considerando a hipótese nula ( $H_0$ ), que define a ausência de diferenças significativas entre os resultados das quatro versões apresentadas, a hipótese alternativa ( $H_1$ ) indicando que existe diferença entre, pelo menos, uma destas e  $\alpha = 0,05$ , criou-se um teste ANOVA resultando em um *p-valor* de 0,041. Com base nesse valor, foi possível refutar  $H_0$  com 95% de confiança estatística.

Portanto, conclui-se que existem diferenças significativas entre os resultados obtidos sob o efeito de cada operador genético proposto.

Porém, o teste ANOVA não identifica a fonte dessa diferença estatística. O Teste de Tukey é baseado na Diferença Mínima Significativa e permite que qualquer contraste entre duas médias de tratamentos seja testado [Hinton, 2004]. Por esse fato, o próximo passo foi realizar esse teste em todas as combinações possíveis dessas versões. As hipóteses  $H_0$  e  $H_1$  são idênticas às utilizadas no teste ANOVA anteriormente citado. A significância estatística foi fixada em 0,05, ou seja,  $\alpha$  se manteve constante em todos os testes. A Figura 6 mostra o resultado do Teste de Tukey realizado em cada comparação.

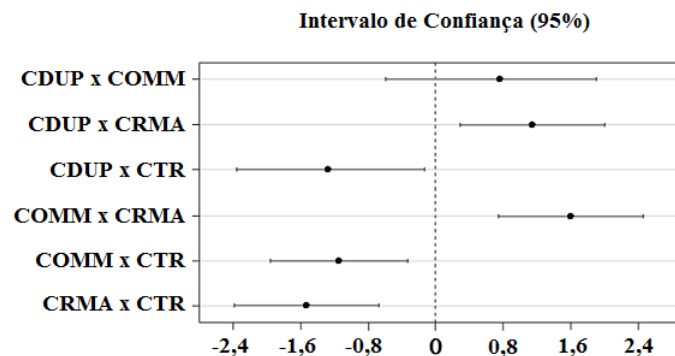


Figura 6: Diferenças entre as médias dos níveis dos fatores

Nota-se na Figura 6 que cada comparação gerou um limite inferior e um superior, os quais definem o intervalo de confiança. Analisando o intervalo de cada teste, devido à ausência do valor zero no intervalo de confiança, foi possível sustentar a hipótese alternativa, que indica a existência de diferenças entre as médias das populações testadas, exceto entre CDUP e COMM. Nesse último caso, com 95% de confiança,  $H_0$  foi corroborada, indicando que não há diferença entre as médias dos tratamentos.

## 7. Conclusão

Este artigo apresentou uma avaliação de quatro operadores de cruzamento dentro do escopo de um AG híbrido aplicado a VRPTW. Cada operador, inserido em uma estrutura padrão de algoritmo, gerou uma versão distinta do mesmo. Cada versão foi definida como: COMM, CRMA, CDUP e CTR. Essas versões foram avaliadas utilizando as 56 instâncias de Solomon [1987] de 100 consumidores. O desempenho de cada versão proposta foi comparada a partir de análises estatísticas. Além disso, os resultados obtidos foram confrontados com os melhores resultados conhecidos da literatura.

A partir dos resultados obtidos, foi possível observar os pontos fortes e fracos de cada versão. CDUP, COMM e CRMA ficaram distantes dos melhores resultados conhecidos. Cada versão, respectivamente, utilizou 13, 12 e 10 veículos a mais que o ótimo. Além disso, essas versões apresentaram dificuldades em 4 grupos de instâncias: R1, R2, RC1 e RC2. Esse fato indica uma deficiência na capacidade de lidar com consumidores aleatoriamente distribuídos. Porém, a versão CTR atingiu bons resultados, utilizando apenas 4 veículos a mais que o melhor conhecido.

A estatística descritiva possibilitou uma observação visual das medidas de tendência central e de variabilidade dos dados gerados por cada versão proposta neste artigo. Analisando as medidas de dispersão, como a amplitude interquartil e os *whiskers*, por exemplo, percebe-se que a robustez das versões foi sendo aprimorada.

A análise de variância comparou os resultados finais obtidos por cada versão. Com base no *p-valor* gerado pelo teste ANOVA, foi possível refutar  $H_0$  com 95% de confiança estatística.

Esse fato indicou que há diferenças entre os resultados das quatro versões propostas neste trabalho. No intuito de identificar a localização dessa diferença, foi realizado um Teste de Tukey que, com base nos intervalos de confiança gerados, com 95% de confiança, foi possível sustentar a hipótese alternativa, que indica a existência de diferenças entre as médias das populações testadas em todas as versões, exceto entre CDUP e COMM.

Se há diferença entre as médias dos tratamentos e as versões alcançam resultados melhores, acredita-se que, para o cenário proposto, a melhor versão é CTR. Porém, a análise de variância não identificou diferenças entre as médias dos tratamentos de CDUP e de COMM. Por esse fato, acredita-se que essas versões tenham desempenho similar quanto à qualidade dos resultados obtidos.

Como perspectivas para trabalhos futuros, destaca-se uma ideia relevante: avaliar outros operadores de cruzamento para VRPTW que utilizem uma estratégia similar à adotada em CTR. Este operador foi aplicado na versão que gerou os melhores resultados no cenário proposto neste trabalho. Além disso, foi notado que realizar trocas entre rotas é uma boa estratégia para obter soluções melhores e com baixo índice de infactibilidade.

## 8. Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Fundação de Apoio à Pesquisa do Estado de Minas Gerais (FAPEMIG) e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

## Referências

- Chiang, T. C. e Hsu, W. H. (2014). A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Computers and Operations Research*, 45: 25–37.
- Cordeau, J. F., Desaulniers, G., Desrosiers, J., e Solomon, M. M. (2001). The VRP with time windows. In Toth, P. e Vigo, D., editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, p. 157–194. SIAM, Philadelphia, EUA.
- Davis, L. (1985). Job shop scheduling with genetic algorithms. In *Proceedings of an international conference on genetic algorithms and their applications*, volume 140. Carnegie-Mellon University Pittsburgh, PA.
- Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search. *Journal of Global Optimization*, 3:109–133.
- Gendreau, M. e Tarantilis, C. D. (2010). Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Technical Report CIRRELT-2010-04, CIRRELT.
- Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addison Wesley, Reading*.
- Hinton, P. R. (2004). *Statistics explained: a guide for science students*. Psychology Press.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, MI.
- Langdon, W. e Poli, R. (1997). An analysis of the max problem in genetic programming. *Genetic Programming*, 1(1997):222–230.
- Montgomery, D. e Runger, G. (2009). *Applied statistics and probability for engineers*. Wiley, 4th edition.

- Ombuki, B., Ross, B. J., e Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 1(24):17–30.
- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31(12):1985–2002.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.
- Tan, K. C., Chew, Y. H., e Lee, L. H. (2006). A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Computational Optimization and Applications*, 34(1):115–151.
- Zhou, W., Song, T., He, F., e Liu, X. (2013). Multiobjective vehicle routing problem with route balance based on genetic algorithm. *Discrete Dynamics in Nature and Society*, 2013.