

UM ALGORITMO MEMÉTICO HÍBRIDO APLICADO AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO

Thiago Muniz Stehling

Programa de Pós-Graduação em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675 Nova Gameleira 30510-000 Belo Horizonte MG Brasil
thiagostehling@gmail.com

Sérgio Ricardo de Souza

Programa de Pós-Graduação em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675 Nova Gameleira 30510-000 Belo Horizonte MG Brasil
sergio@dpg.cefetmg.br

Moacir Felizardo de França Filho

Programa de Pós-Graduação em Modelagem Matemática e Computacional
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas 7675 Nova Gameleira 30510-000 Belo Horizonte MG Brasil
franca@des.cefetmg.br

Resumo

Este artigo apresenta um Algoritmo Memético Híbrido para a solução do Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). A hibridização está caracterizada na combinação entre PFIH e GRASP, utilizada para gerar a população inicial do algoritmo proposto. O meme inserido no processo evolutivo visa não somente alcançar bons resultados, mas economizar tempo de execução. Para os experimentos computacionais, as 56 instâncias de Solomon foram utilizadas. Os resultados obtidos são comparados com os melhores resultados conhecidos na literatura e com algoritmos similares ao proposto neste trabalho. Uma análise empírica, uma avaliação do tempo de execução e uma análise estatística foram realizadas. Os resultados indicam que o algoritmo alcançou resultados melhores em um tempo de execução razoável, considerando a inserção do meme.

Palavras-Chaves: Problema de Roteamento de Veículos com Janela de Tempo; Algoritmos Meméticos; Algoritmos Híbridos.

Abstract

This article presents a Hybrid Memetic Algorithm for solving the Vehicle Routing Problem with Time Windows (VRPTW). The hybridization is characterized by the combination between the Push Forward Insertion Heuristic (PFIH) and the Greedy Randomized Adaptive Search Procedure (GRASP). This combination is used to generate the initial population of the proposed algorithm. The meme inserted into the evolutionary process aims not only achieve good results, but save runtime. The 56 Solomon instances were used for computational experiments. The obtained results are compared with the best results known in the literature and with similar algorithms to the proposed in this work. An empirical analysis, an evaluation of the run time and a statistical analysis was performed. The results indicate that the algorithm has achieved the best results at a reasonable execution time considering the insertion of the meme.

Keywords: Vehicle Routing Problem with Time Windows; Memetic Algorithm; Hybrid Algorithm.

1. INTRODUÇÃO

O Problema de Roteamento de Veículos com Janela de Tempo (PRVJT) é uma extensão do Problema de Roteamento de Veículos clássico. No PRVJT, o serviço em cada cliente deve iniciar dentro de um intervalo de tempo, caracterizando a janela de tempo do problema. O veículo deve permanecer no local do atendimento enquanto o serviço é executado. Considerando uma frota homogênea de veículos com capacidade especificada, o PRVJT consiste em criar rotas de custo mínimo, de maneira a atender à demanda de todos os clientes. A janela de tempo não pode ser violada, assim como as demais restrições do problema. Veja em [1] e [2] uma análise sobre o PRVJT com maiores detalhes.

De acordo com [1], o PRVJT é um problema que pertence à classe de problemas NP-Difícil. Assim, a utilização de métodos de aproximação, como metaheurísticas, para solucioná-lo é justificada.

Este trabalho apresenta um Algoritmo Memético (AM) híbrido para solucionar o PRVJT. A hibridização proposta consiste em uma combinação entre a metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP), introduzida por [3], e a *Push-Foward Insertion Heuristic* (PFIH), apresentada por [4].

Em [5], foi indicado como trabalho futuro um aprimoramento do Algoritmo Genético (AG) híbrido desenvolvido por esses autores. O aperfeiçoamento sugerido se trata de um investimento em técnicas para reduzir a quantidade de veículos utilizados e melhorar a qualidade das soluções obtidas, não somente na fase de construção do GRASP proposto. Porém, no presente trabalho, além de adicionar uma busca local no processo evolutivo do AG, também é levado em consideração o tempo computacional gasto durante a execução do algoritmo. A ideia é melhorar os resultados utilizando um tempo computacional próximo ao obtido em [5].

A próxima seção deste artigo apresenta conceitos importantes sobre AM, os quais auxiliaram no objetivo de desenvolver um AM eficiente, tanto em qualidade de resultados quanto em tempo de execução. A Seção 3 aborda trabalhos que adotaram AM, e variações, para solucionar o PRVJT. A Seção 4 descreve a metodologia aplicada ao desenvolvimento do algoritmo proposto. A seção 5 apresenta os resultados computacionais obtidos com o algoritmo evolutivo híbrido, incluindo uma análise empírica, uma avaliação do tempo de execução e uma análise estatística dos experimentos. Por fim, a última seção conclui este artigo.

2. ALGORITMO MEMÉTICO

Definido inicialmente em [6], um Algoritmo Memético (AM) é, em linhas gerais, uma combinação de algoritmo evolutivo com busca local. Esse método é baseado em modelos de sistemas naturais, que combinam a adaptação evolutiva populacional com a capacidade de aprendizagem individual dos elementos de uma população.

Segundo [7], o AM é inspirado no conceito de meme. Para um AM, o meme representa um componente de evolução cultural, que pode ser exibido através de refinamentos locais. Em outras palavras, o meme é um aspecto de desenvolvimento individual que é empregado para melhoria do indivíduo de uma população.

Outra maneira de identificar um AM, conforme [8], é como uma união entre busca populacional e busca local. A busca populacional está relacionada ao processo evolutivo já existente, por exemplo, em um algoritmo genético, na forma das estruturas de recombinação e mutação. Veja em [9] uma análise mais detalhada sobre busca populacional proposta para um algoritmo genético. Ainda de acordo com [8], a busca local é um refinamento pontual realizado dentro do processo evolutivo de um algoritmo evolucionário.

Sob a ótica de [10], um AM é um algoritmo adaptativo. A característica adaptativa

marcante dos meméticos está na eleição do método de busca local ou meme a ser utilizado. Essa escolha afeta significativamente o desempenho do AM em problemas complexos. Segundo esses autores, a forma adaptativa de um AM deve promover a cooperação e a concorrência entre vários memes para avaliar as diversas estruturas de vizinhança, que contém soluções de alta qualidade alcançadas com recursos computacionais reduzidos. Em [10], foram propostos diversos tipos e níveis de adaptações. Essa classificação foi realizada de acordo com as diferenças nas estratégias adotadas por cada meme.

3. TRABALHOS CORRELATOS

Inúmeros trabalhos que abordam soluções para o PRVJT estão disponíveis na literatura. Porém, nesta seção, foram priorizadas as contribuições que utilizaram AM, e variações, como solução para esse problema.

Em [11] os autores propuseram um AM para solucionar o PRVJT. A abordagem desse algoritmo envolve a evolução de duas populações em paralelo. A primeira população evolui os indivíduos, visando o objetivo de minimizar a distância total percorrida. Simultaneamente, a segunda população se concentra em reduzir as violações da restrição de janela de tempo, criando soluções viáveis. Os memes utilizados em [11] são: (i) uma heurística de inserção; (ii) uma grande busca na estrutura de vizinhança; e (iii) um método baseado em Colônia de Formigas, para diversificar e intensificar ainda mais o mecanismo de busca populacional.

Em [12] foi proposto um algoritmo genético híbrido para solucionar três variações do PRVJT. A hibridização idealizada por esses autores se caracteriza pela combinação entre um operador de melhoria da busca na estrutura de vizinhança, que promove o aprendizado do indivíduo, e uma fase de reparo da solução. A evolução cultural ocorre através de dois procedimentos de busca local: um que aprimora cada rota separadamente e outro que melhora o padrão de atribuição de escolhas. Uma sequência de execução desses memes é apontada. Segundo [12], o aprendizado e o reparo de uma solução são essenciais para o alcance de soluções de alta qualidade.

Em [13] foi utilizado um AM para resolver o PRVJT. Esses autores propuseram uma heurística de satisfação de restrições, que realiza cortes no espaço de busca. Estas podas guiam a busca na direção de boas soluções do problema. Uma heurística evolutiva também é indicada para o processo de recombinação e mutação das rotas. De acordo com a estratégia adotada por [13], uma parcela da população de indivíduos aprimorados pelo meme de satisfação de restrições é somada a uma parte da população original. A união dessas parcelas cria a próxima geração do algoritmo evolutivo.

No trabalho de [14] foi proposto um AM paralelo como solução do PRVJT. O paralelismo empregado por esses autores se baseia em duas fases: (i) minimizar o número de veículos utilizando um algoritmo heurístico paralelo e (ii) minimizar a distância percorrida através do AM paralelo propriamente dito. Em ambos os algoritmos, existe a permuta periódica de informações, ou seja, as melhores soluções encontradas até o momento da troca são compartilhadas. O meme utilizado por [14] realiza, em um primeiro momento, um movimento de redução de rota e, em seguida, movimentos que visam melhorar a qualidade do indivíduo.

Em [5], foi proposto um AG híbrido para solucionar o PRVJT. A hibridização utilizada por esses autores é uma combinação da PFIH com a metaheurística gulosa e aleatória GRASP. Essa união foi utilizada para aprimorar a qualidade dos indivíduos da população inicial do algoritmo evolutivo proposto. O trabalho de [5] foi utilizado como base para o presente artigo, que apresenta o próximo passo da estratégia adotada por esses autores. Esse embasamento será mais bem explicado na seção seguinte.

4. ALGORITMO PROPOSTO

Este trabalho apresenta um Algoritmo Memético Híbrido (AMH), que possui, como base, o algoritmo evolutivo denominado AGH5 apresentado em [5]. Este algoritmo mostrou-se, estatisticamente, com desempenho superior aos demais apresentados nesta referência citada. Por isso a escolha dessa versão para a continuidade do desenvolvimento. No presente trabalho, o algoritmo AGH5 passa a ser identificado, por simplicidade, apenas como AGH.

4.1. FORMULAÇÃO DO PROBLEMA

Matematicamente, o PRVJT pode ser modelado por um grafo $G(V, A)$ sendo $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$ o conjunto de vértices que representa os clientes que necessitam de atendimento e $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ o conjunto de arestas que indicam elos entre cada um desses clientes. Os vértices v_0 e v_{n+1} correspondem ao depósito. Cada aresta (v_i, v_j) , pertencente ao conjunto A , possui um custo C_{ij} associado. Todo vértice do grafo apresenta, como características, o tempo de serviço, a demanda a ser atendida e a janela de tempo, que indica o intervalo de tempo em que o serviço deve ser iniciado. Os veículos possuem uma capacidade de carga especificada e a frota é homogênea. Em [1], é possível verificar, com maiores detalhes, a modelagem matemática adotada neste trabalho.

4.2. REPRESENTAÇÃO DE UMA SOLUÇÃO

Uma solução do PRVJT é representada computacionalmente por um vetor solução que combina n rotas $r = [r_1, r_2, \dots, r_n]$ geradas para atender um conjunto de m clientes $c = [c_1, c_2, \dots, c_m]$. O elemento zero nesse vetor solução é utilizado como limite entre rotas e identifica o depósito. Para cada rota r_i , sendo $i = 1, 2, \dots, n$, existe um único veículo que realiza o trajeto. As demais informações do PRVJT são armazenadas em estruturas de dados.

4.3. PARÂMETROS INICIAIS

Os parâmetros básicos do AMH são idênticos aos do AGH, conforme [5]. O tamanho da população é fixado em 100 indivíduos, sendo que, durante o processo evolutivo, esse número dobra de valor. Porém, apenas a metade desses indivíduos sobrevive para a próxima geração. A taxa de cruzamento e a taxa de mutação não se alteram durante a execução do algoritmo e são, respectivamente, 80% e 5%, tendo sido definidas empiricamente. A população inicial é gerada através da combinação do GRASP [3] com a PFIH [4].

4.4. FUNÇÃO DE APTIDÃO

Dois objetivos são intrínsecos ao PRVJT: minimizar a distância total percorrida e minimizar o número de veículos utilizados. Esses objetivos devem ser convertidos em um único problema de minimização. Em [15], foi proposta uma FA baseada no método da soma ponderada. Essa função é dada por:

$$FA = \gamma \times N + \delta \times D \quad (1)$$

Os parâmetros de peso γ e δ estão associados, respectivamente, ao número de veículos N e à distância total percorrida D . Os valores desses parâmetros foram definidos empiricamente por [15]. Essa FA é utilizada para medir a qualidade de cada indivíduo da população, inclusive os indivíduos que foram aprimorados através da evolução cultural.

4.5. PROCESSO EVOLUTIVO

Os operadores de cruzamento e de mutação utilizados neste trabalho são idênticos aos usados em [5]. Esses autores utilizaram os operadores propostos por [15]: o *Best Coast Route Crossover* (BCRC) e o *Constrained Route Reversal Mutation* (CRRM). As adaptações feitas no BCRC em [5] foram mantidas no AMH. Após a seleção dos pais P_1 e P_2 e identificação dos pontos de corte, o gene a ser cruzado é escolhido com base na FA,

verificando qual dos pais é o melhor. Do melhor pai é selecionada a maior cadeia de gene e do pior, a menor. A ideia é garantir uma maior alteração no filho correspondente ao pior pai e uma menor modificação no correspondente ao melhor pai. Já o CRRM possui as características propostas por [15]. Uma rota mínima é selecionada para inversão, de tamanho contido no intervalo [2, 3]. Um diferencial desses operadores é a garantia de factibilidade em relação à janela de tempo do PRVJT. Além disso, o operador de elitismo e o cruzamento arbitrário, ambos implementados por [5] para o AGH, também se encontram no AMH.

4.6. MEME

A decisão sobre qual meme escolher para o AMH está embasada nos conceitos vistos na Seção 2 e em testes computacionais realizados. Segundo [10], um meme pode ser classificado de acordo com seu tipo: estático, adaptativo ou auto adaptativo. O aspecto cultural também pode ocorrer em diferentes níveis: local, global ou externamente ao processo.

Neste trabalho, a hibridização utilizada para gerar a população inicial ocorre externamente ao processo evolutivo. A recombinação e a mutação promovem uma busca global na população através da diversidade e intensificação. Portanto, foi definido empiricamente que o meme escolhido seria classificado no nível local. O tipo do meme foi identificado como adaptativo. De acordo com [10], o tipo adaptativo pode ser dividido em dois subtipos: adaptação qualitativa e quantitativa. Respectivamente, estas indicam uma estratégia randômica e uma gulosa.

Portanto, o meme utilizado neste trabalho realiza uma busca local para refinar os indivíduos da população através da evolução cultural. A estrutura de vizinhança do meme é identificada através dos seguintes movimentos propostos por [5]:

- eliminar rota;
- trocar dois clientes de posição em uma mesma rota;
- inserir um cliente em uma determinada posição em uma mesma rota;
- trocar dois clientes de posição de rotas diferentes; e
- inserir um cliente em uma determinada posição em outra rota.

O movimento que elimina rotas funciona de forma gulosa para atender à adaptação quantitativa. A estratégia é eliminar as rotas menores em primeiro lugar. Esse movimento é sempre executado e auxilia na redução dos veículos utilizados. Os demais movimentos são escolhidos aleatoriamente, com mesma probabilidade de escolha, para atender à adaptação qualitativa. Após o movimento eliminar rota, um dos quatro movimentos restantes é executado para aprimorar o indivíduo, visando a minimização da distância total percorrida.

4.7. PSEUDOCÓDIGO

Nesta subseção, estão demonstrados todos os detalhes contidos nas subseções anteriores através do pseudocódigo do AMH proposto. O algoritmo utiliza como entrada: o número de veículos, a capacidade dos veículos, a lista de clientes com suas coordenadas geográficas, demandas e janelas de tempo. Para melhor compreensão, o pseudocódigo foi dividido em duas partes. Porém, vale ressaltar que se trata de um único algoritmo híbrido. A numeração das linhas foi mantida para indicar a sequência do pseudocódigo.

A Figura 1 mostra o trecho do pseudocódigo do AMH referente à geração da população inicial, onde PFIH é utilizada na característica gulosa da metaheurística GRASP. A Subseção 4.3 apresenta detalhes quanto a esse trecho, que caracteriza a hibridização do algoritmo proposto.

```

01 PopulacaoInicial =  $\emptyset$ 
02 enquanto Tamanho(PopulacaoInicial) < 100 faça
03      $s = 0$ 
04      $s = \text{ConstrucaoPFIH}()$ 
05      $s' = \text{BuscaLocal}(s)$ 
06     se  $s'$  é melhor que  $s$  então
07         PopulacaoInicial =  $s'$ 
08     fim se
09 fim enquanto

```

Figura 1 – Pseudocódigo da hibridização GRASP-PFIH proposta.

A Figura 2 mostra o restante do pseudocódigo do AMH. A população inicial foi gerada através das linhas de código mostradas pela Figura 1. A FA, detalhada na subseção 4.4 deste artigo, também é utilizada na seleção e sobrevivência dos indivíduos através do método da roleta. A recombinação e a mutação, citadas na subseção 4.5 deste artigo, estão contidas nas linhas 18 e 19 da Figura 2. Das linhas 22 a 24, estão demonstrados o operador de elitismo e o cruzamento arbitrário implementados. Veja mais detalhes sobre esses métodos em [5].

```

10 Evolucao =  $\emptyset$ 
11 Criterio = AvaliaPopulacao (PopulacaoInicial)
12 enquanto Criterio não aceito faça
13     Evolucao = PopulacaoInicial
14     enquanto Tamanho(Evolucao) < 200 faça
15          $p = \text{Selecao}(\text{PopulacaoInicial})$ 
16          $p = \text{OperadorBuscaLocal}(p)$ 
17          $p' = 0$ 
18          $p' = \text{Cruzamento}(p)$ 
19          $p' = \text{Mutacao}(p')$ 
20         InsereIndividuo (Evolucao,  $p'$ )
21     fim enquanto
22      $p^* = \text{OperadorElitismo}(Evolucao)$ 
23     InsereIndividuo (PopulacaoInicial,  $p^*$ )
24     InsereIndividuo (PopulacaoInicial, Cruzamento( $p^*$ ))
25     PopulacaoInicial = PopulacaoSobrevivente (Evolucao)
26     Evolucao =  $\emptyset$ 
27     Criterio = AvaliaPopulacao (PopulacaoInicial)
28 fim enquanto
29 Mostra a melhor solução contida na população

```

Figura 2 – Pseudocódigo do processo evolutivo e memético do AMH.

O meme e todo o processo descrito na subseção 4.6 deste artigo se encontram no operador de busca local desenvolvido. Este é identificado pela linha 16 da Figura 2. Esse operador é o ponto central do AMH. Após a seleção dos pais P_1 e P_2 , estes são submetidos ao processo de evolução cultural. Uma vez refinados, esses pais são submetidos ao processo evolutivo tradicional. Neste trabalho, apenas soluções factíveis são aceitas. Portanto, caso o meme gere soluções infactíveis, os pais que não se desenvolveram individualmente, são submetidos ao processo evolutivo normalmente.

5. RESULTADOS COMPUTACIONAIS

O algoritmo híbrido proposto foi implementado em linguagem C e os experimentos computacionais foram realizados em um notebook Intel Core i7 2,40 GHz, com 8 GB de memória RAM e sistema operacional Windows 8.1. Esses experimentos foram realizados com base nas 56 instâncias de [4] para o PRVJT, com 100 consumidores em cada. De acordo com [1], estas são divididas em seis classes: C1, C2, R1, R2, RC1 e RC2. As instâncias do tipo C possuem clientes distribuídos geograficamente em grupos. As do tipo R apresentam clientes de coordenadas geográficas aleatórias. Já as instâncias do tipo RC representam uma união entre os dois primeiros casos.

Por se tratar de um algoritmo estocástico, o AMH foi executado 50 vezes para cada uma das instâncias. As seis primeiras tabelas a seguir mostram, respectivamente, os resultados obtidos com o AMH nas instâncias C1, C2, R1, R2, RC1 e RC2. Em cada célula das tabelas são destacados o número de veículos e o valor da distância total percorrida. Os resultados obtidos com o AMH são comparados com os alcançados pela última versão desenvolvida por [5], denominada neste trabalho como AGH, e com os melhores resultados conhecidos na literatura (MRL). Os MRL podem ser acessados através do seguinte endereço eletrônico: <http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/>.

Tabela 1 – Instâncias C1

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
C101	10 828,94	10 828,94	10 828,94
C102	10 828,94	10 828,94	10 828,94
C103	10 828,06	10 828,06	10 828,06
C104	10 824,78	10 824,78	10 824,78
C105	10 828,94	10 828,94	10 828,94
C106	10 828,94	10 828,94	10 828,94
C107	10 828,94	10 828,94	10 828,94
C108	10 828,94	10 828,94	10 828,94
C109	10 828,94	10 828,94	10 828,94
TOTAL	90 7455,42	90 7455,42	90 7455,42

Tabela 2 – Instâncias C2

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
C201	3 591,56	3 591,56	3 591,56
C202	3 591,56	3 591,56	3 591,56
C203	3 591,17	3 591,17	3 591,17
C204	3 590,60	3 590,60	3 590,60
C205	3 588,88	3 588,88	3 588,88
C206	3 588,49	3 588,49	3 588,49
C207	3 588,29	3 588,29	3 588,29
C208	3 588,32	3 588,32	3 588,32
TOTAL	24 4718,87	24 4718,87	24 4718,87

Tabela 3 – Instâncias R1

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
R101	19 1645,79	19 1645,79	19 1645,79
R102	17 1486,12	17 1486,12	17 1486,12
R103	13 1292,68	13 1292,68	13 1292,68
R104	9 1007,24	9 1007,24	9 1007,24
R105	14 1377,11	14 1377,11	14 1377,11
R106	12 1251,98	12 1251,98	12 1251,98
R107	10 1104,66	10 1104,66	10 1104,66
R108	9 960,88	10 971,92	9 960,88
R109	11 1194,73	11 1194,73	11 1194,73
R110	10 1118,59	11 1174,89	10 1118,59
R111	10 1096,72	10 1096,72	10 1096,72
R112	9 982,14	10 1001,33	9 982,14
TOTAL	143 14518,64	146 14605,17	143 14518,64

Tabela 4 – Instâncias R2

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
R201	4 1252,37	4 1252,37	4 1252,37
R202	3 1191,70	3 1191,70	3 1191,70
R203	3 939,50	3 939,50	3 939,50
R204	2 825,52	2 825,52	2 825,52
R205	3 994,42	3 994,42	3 994,42
R206	3 906,14	3 906,14	3 906,14
R207	2 890,61	2 890,61	2 890,61
R208	2 726,75	3 789,01	2 726,75
R209	3 909,16	3 909,16	3 909,16
R210	3 939,34	3 939,34	3 939,34
R211	2 885,71	3 903,87	2 885,71
TOTAL	30 10461,22	32 10541,68	30 10461,22

Tabela 5 – Instâncias RC1

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
RC101	14 1696,94	14 1696,94	14 1696,94
RC102	12 1554,75	12 1554,75	12 1554,75
RC103	11 1261,67	11 1261,67	11 1261,67
RC104	10 1135,48	11 1179,81	11 1179,81
RC105	13 1629,44	13 1629,44	13 1629,44
RC106	11 1424,73	12 1478,93	12 1478,93
RC107	11 1230,48	11 1230,48	11 1230,48
RC108	10 1139,82	11 1177,39	11 1177,39
TOTAL	92 11073,31	95 11209,41	95 11209,41

Tabela 6 – Instâncias RC2

<i>Instâncias</i>	<i>MRL</i>	<i>AGH</i>	<i>AMH</i>
RC201	4 1406,91	4 1406,91	4 1406,91
RC202	3 1365,65	3 1365,65	3 1365,65
RC203	3 1049,62	3 1049,62	3 1049,62
RC204	3 798,41	4 811,73	4 811,73
RC205	4 1297,19	4 1297,19	4 1297,19
RC206	3 1146,32	3 1146,32	3 1146,32
RC207	3 1061,14	3 1061,14	3 1061,14
RC208	3 828,14	4 835,98	4 835,98
TOTAL	26 8953,38	28 8975,98	28 8975,98

5.1. ANÁLISE EMPÍRICA

Cada uma das seis classes das instâncias de [4] é subdividida em dois grupos. Por exemplo, a classe C apresenta o subgrupo C1, cujos clientes possuem janela de tempo curta, e o subgrupo C2, onde cada cliente apresenta uma janela de tempo larga.

A janela de tempo ser considerada curta implica que um número de veículos maior será necessário para atender aos clientes. Uma restrição forte será que o veículo já utilizado não consegue atender todos os clientes, considerando o tempo de deslocamento entre um cliente e outro e a janela de tempo inicial do próximo cliente. Por esse fato, mais veículos serão utilizados. A janela de tempo larga é o inverso. Requer um número menor de veículos para atender a todos os clientes.

A classe R também apresenta os subgrupos R1 e R2 e a classe RC os subgrupos RC1 e RC2. Cada uma mantém sua característica de grande classe e as mesmas propriedades dos subgrupos C1 e C2, no quesito tamanho da janela de tempo.

Pelas seis tabelas anteriores, nota-se que a dificuldade apresentada pelo AGH, principalmente nas classes R1 e RC1, onde os clientes possuem janela de tempo curta, indica uma deficiência na capacidade de intensificação do algoritmo, ou seja, na busca por soluções próximas às obtidas.

O AMH, por ser derivado do AGH e apresentar a inclusão de uma busca local no processo evolutivo, melhorou esse problema. Passou de duas para quatro classes que obtiveram os MRL. Mas ainda apresentou deficiência nas classes com características mescladas: RC1 e RC2.

A Tabela 7 mostra os resultados médios por classe e totais obtidos, comparando o MRL com os algoritmos AGH e AMH. Este proposto neste trabalho e aquele proposto em [5]. O AMH conseguiu melhorar o resultado, utilizando cinco veículos a menos que o AGH. Além disso, a coluna OMB indica os resultados obtidos com o algoritmo de [15], que também são comparados com os resultados médios por classe e totais alcançados pelo AMH. Este utiliza sete veículos a menos que o OMB.

Tabela 7 – Comparativo dos resultados médios por classe e totais

<i>Instâncias</i>	<i>MRL</i>	<i>OMB</i>	<i>AGH</i>	<i>AMH</i>
C1	10 828,38	10 828,48	10 828,38	10 828,38
C2	3 589,86	3 590,60	3 589,86	3 589,86
R1	11,92 1209,89	12,07 1204,48	12,16 1217,1	11,92 1209,89
R2	2,73 951,02	3,1 893,03	2,91 958,33	2,73 951,02
RC1	11,5 1384,16	12,05 1370,79	11,86 1401,18	11,86 1401,18
RC2	3,25 1160,75	3,5 1025,31	3,5 1122,0	3,5 1122,0
TOTAL	405 56352,70	417 55627,01	415 57506,53	410 57339,54

5.2. ANÁLISE DO TEMPO DE EXECUÇÃO

Tanto o AGH quanto o AMH foram executados 50 vezes para cada uma das 56 instâncias. Os tempos, em segundos, indicados por [5] e os obtidos com o algoritmo proposto no presente trabalho se encontram discriminados na Tabela 8. A última linha da tabela mostra o tempo total, em horas, gasto na bateria de testes. Vale ressaltar que, devido à estocasticidade dos algoritmos, os tempos estimados podem variar de uma execução para outra.

Tabela 8 – Comparativo dos tempos de execução

<i>Tempo (s)</i>	<i>AGH</i>	<i>AMH</i>
Mínimo	6,39	9,78
Médio	8,41	15,09
Máximo	19,7	27,33
HORAS	6,5411	11,7366

De acordo com a Tabela 8, a quantidade de horas totais gastas nos experimentos do AMH é bem maior que as consumidas pelo AGH. Porém, essa bateria de testes somente é necessária para comprovar a convergência e a robustez do algoritmo estocástico. O foco principal está no tempo médio gasto pelo AMH, que alcançou uma diferença de menos de 7 segundos do tempo gasto pelo AGH. Importante ressaltar que foi inserido um método de busca local no processo evolutivo. Portanto, tal diferença de tempo é considerada razoável.

5.3. ANÁLISE ESTATÍSTICA

Em um primeiro momento, uma estatística descritiva para analisar as medidas de tendência central e de variabilidade dos dados gerados foi realizada. A Figura 3 mostra o boxplot referente a esses dados. Este foi gerado através da análise de qualidade das soluções obtidas na versão desenvolvida por [5] e pelo AMH proposto. Tal qualidade das soluções é estimada através da FA dos indivíduos, mencionada na subseção 4.4 deste artigo. Quanto mais próximo de 1, melhor é a solução, pois a FA foi convertida em um problema de maximização.

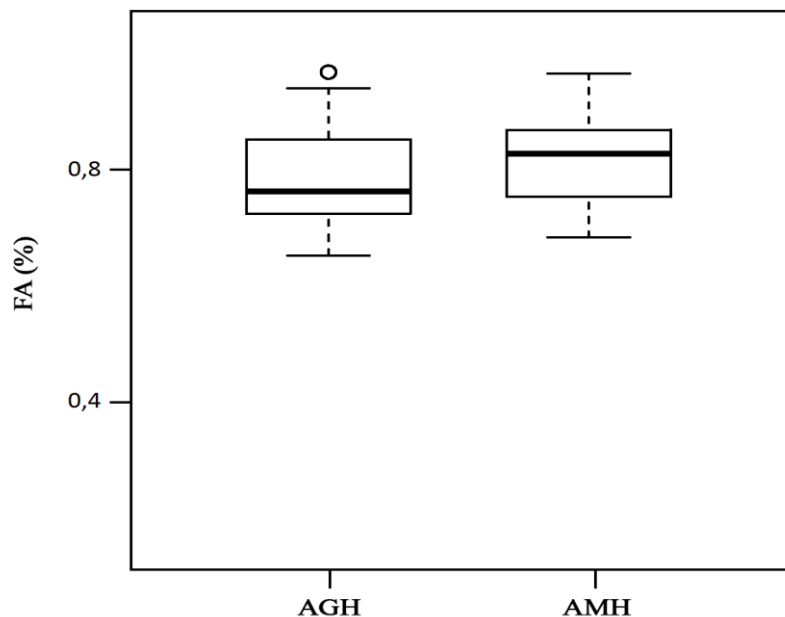


Figura 3 – Boxplot referente aos dados obtidos após 50 execuções dos algoritmos.

Pela Figura 3 observam-se duas informações importantes: a mediana melhorou e a dispersão dos dados diminuiu. Isso indica que a robustez do AMH melhorou em comparação ao AGH. Porém, como existe sobreposição entre os dados dessas duas versões, nada pode ser concluído pela análise visual do boxplot. Portanto, faz-se necessário a realização de algum teste estatístico para avaliar se há diferença entre os tratamentos.

Então, o próximo passo foi realizar um teste de múltiplas comparações. De acordo com [16], o Teste de Tukey é baseado na Diferença Mínima Significativa e permite que qualquer contraste entre duas médias de tratamentos seja testado. No Teste de Tukey realizado, α foi fixado em 0,05. O teste gerou um intervalo de confiança com limite inferior de 0,0298 e limite superior de 0,0566. Como o valor zero não pertence ao intervalo, conclui-se que existem diferenças entre as médias dos tratamentos. Portanto, com 95% de confiança, pode-se afirmar que AMH é melhor que AGH.

6. CONCLUSÃO

Este artigo apresentou um AM híbrido para solucionar o PRVJT. A versão desenvolvida foi comparada com o algoritmo proposto por [5], tanto em qualidade dos resultados quanto em tempo de execução. Essa comparação foi realizada com base nas 56 instâncias de [4] para o PRVJT, de 100 consumidores cada.

A análise empírica mostrou que o AMH não somente atingiu melhores resultados nas instâncias R1 e R2, mas também identificou que o AMH utilizou 5 veículos a menos que o AGH e ficou a uma diferença de 5 veículos do ótimo conhecido.

A análise do tempo de execução indicou que, mesmo com a inclusão do meme para realizar a evolução cultural dos indivíduos da população, o AMH conseguiu atingir um tempo médio de execução de 15,09 segundos, ou 6,68 segundos a mais que o AGH. Um tempo razoável.

Além disso, a análise estatística dos experimentos mostra que a robustez do algoritmo melhorou e que, com uma significância estatística de 95%, o AMH é melhor que o AGH. Portanto, conclui-se que o algoritmo evolutivo AMH atendeu ao objetivo de solucionar o PRVJT.

7. AGRADECIMENTOS

Os autores agradecem ao apoio fornecido pelo CEFET-MG, pela FAPEMIG e pela CAPES ao desenvolvimento deste trabalho.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Cordeau, J. F.; Desaulniers, G.; Desrosiers, J. e Solomon, M. M. (2001). The vrp with time windows. Toth, P. e Vigo, D., editors, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, p. 157–194. SIAM, Philadelphia, EUA.
- [2] 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.
- [3] Feo, T. A. e Resende, M. G. C. (1995). Greedy randomized adaptive search. *Journal of Global Optimization*, v. 3, p. 109–133.
- [4] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265.

- [5] Stehling, T. M.; Souza, S. R.; França Filho, M. F. e Silva, M. A. L. (2014). Um algoritmo genético híbrido para a solução do problema de roteamento de veículos com janela de tempo. *Proceedings of the XXXV Iberian Latin-American Congress on Computational Methods in Engineering (CILAMCE 2014)*, Nov 2014.
- [6] Moscato, P. (1989). *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms*, California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.
- [7] Krasnogor, N. e Smith, J. (2005). A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–488, Oct. 2005.
- [8] Moscato, P. e Cotta, C. (2003). A gentle introduction to memetic algorithms, in: F. Glover and G. A. Kochenberger (eds.), *Handbook of Metaheuristics*, Kluwer, Boston, pp. 105–144.
- [9] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.
- [10] Ong, Y. S.; Lim, M. H.; Zhu, N. e Wong, K. W. (2006). Classification of adaptive memetic algorithms: A comparative study. *IEEE Trans. Syst., Man, Cybern. Part B*, vol. 36, no. 1, pp. 141–152, 2006.
- [11] Berger, J. e Barkaoui, M. (2002). A memetic algorithm for the vehicle routing problem with time windows. In *Proceedings of the 7th International Command and Control Research and Technology Symposium*, 2002.
- [12] Vidal, T.; Crainic, T.; Gendreau, M. e Prins, C. (2011). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. Tech. Rep. 2011-61, CIRRELT 2011.
- [13] Cruz-Chávez, M. A.; Díaz-Parra, O.; Juárez-Romero, D. e Martínez-Rangel, M. G. (2008). Memetic Algorithm base on a Constraint Satisfaction Technique for VRPTW. L. Rutkowski et al. (Eds.). *Artificial Intelligence and Soft Computing – ICAISC 2008*.
- [14] Nalepa, J. e Czech, Z. J. (2014). A Parallel Memetic Algorithm to Solve the Vehicle Routing Problem with Time Windows. *arXiv preprint arXiv:1402.6942*. 27 Feb 2014.
- [15] Ombuki, B.; Ross, B. J. e Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, v. 1, n. 24, p. 17–30.
- [16] Montgomery, D. e Runger, G. (2009). *Applied statistics and probability for engineers*. Wiley, 4nd edição.