

# UM ALGORITMO DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS HÍBRIDO COM MEME ACOPLADO PARA SOLUCIONAR O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO

**Thiago Muniz Stehling**

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Avenida Amazonas, 7675 - Nova Gameleira, 30510-000  
thiagostehling@gmail.com

**Sérgio Ricardo de Souza**

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Avenida Amazonas, 7675 - Nova Gameleira, 30510-000  
sergio@dppg.cefetmg.br

**Moacir Felizardo de França Filho**

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Avenida Amazonas, 7675 - Nova Gameleira, 30510-000  
franca@des.cefetmg.br

## RESUMO

Este artigo apresenta um algoritmo de Otimização por Enxame de Partículas híbrido com Meme Acoplado aplicado à solução do Problema de Roteamento de Veículos com Janela de Tempo. A hibridização é caracterizada pela união entre a metaheurística GRASP, a heurística PFIH, um operador de mutação e um método de busca local. Os experimentos computacionais foram realizados utilizando as 56 instâncias de Solomon e os resultados obtidos são comparados com os melhores resultados conhecidos da literatura. Além disso, uma análise empírica, de impacto e uma análise estatística foram realizadas para verificar a eficiência da estratégia adotada. Por fim, foi possível apontar sucesso ao algoritmo desenvolvido.

**PALAVRAS CHAVE.** Problema de Roteamento de Veículos com Janela de Tempo. Otimização por Enxame de Partículas. Busca Local.

**Área Principal:** Metaheurísticas (MH)

## ABSTRACT

This paper presents a hybrid Particle Swarm Optimization algorithm with Coupled Meme applied to the solution of the Vehicle Routing Problem with Time Windows. The hybridization is characterized by the union of GRASP, PFIH, a mutation operator and a local search method. The computational experiments were performed using the 56 Solomon's instances and the results are compared with the literature and with the best known results. In addition, an empirical, impact and statistical analysis were performed to verify the effectiveness of the adopted strategy. Finally, the success of developed algorithm is appointed.

**KEYWORDS.** Vehicle Routing Problem with Time Windows. Particle Swarm Optimization. Local Search.

**Main Area:** Metaheuristics (MH)

## 1. Introdução

Um dos problemas mais estudados em Otimização Combinatória é o Problema de Roteamento de Veículos com Janela de Tempo (PRVJT). Segundo Cordeau et al. (2001), o PRVJT é uma extensão do clássico Problema de Roteamento de Veículos (PRV). Além da restrição de carga, é acrescentada a esta extensão uma característica temporal de atendimento aos clientes. Em linhas gerais, o PRVJT determina rotas de custo mínimo, de modo que uma frota homogênea de veículos inicia o trajeto em um depósito e atende a um conjunto determinado de clientes. Cada veículo possui uma capacidade de carga conhecida. Os clientes apresentam uma demanda a ser atendida e um tempo mínimo e máximo para início do atendimento, caracterizando a janela de tempo do problema. Veja em Gendreau e Tarantilis (2010) uma análise exaustiva sobre o PRVJT.

De acordo com Cordeau et al. (2001), um dos desafios do PRVJT é que este pertence à classe de problemas que não podem ser resolvidos em tempo polinomial. Métodos exatos solucionariam o problema proposto, no pior caso, em tempo exponencial. Portanto, justifica-se o uso de métodos de aproximação para resolver o PRVJT.

Este artigo apresenta um algoritmo de Otimização por Enxame de Partículas (PSO) híbrido como solução para o PRVJT. A hibridização consiste em uma combinação entre a metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP), a heurística *Push-Forward Insertion Heuristic* (PFIH) e um operador de mutação. Além disso, um método de busca local é integrado ao algoritmo populacional.

Stehling et al. (2015) propuseram três versões de PSO para resolver o PRVJT. Estatisticamente, esses autores verificaram que a última versão híbrida do algoritmo proposto por eles é a melhor dentre as três propostas. Essa última versão é utilizada como base para o algoritmo híbrido desenvolvido no atual trabalho. A ideia é validar a significância de melhoria nos resultados obtidos com a inclusão do método de busca local ao PSO.

O restante do trabalho está organizado da seguinte maneira: a Seção 2 apresenta os trabalhos correlatos disponíveis na literatura, abordando em especial os que adotaram as técnicas adotadas neste artigo e as aplicaram ao PRVJT; a Seção 3 inclui todos os detalhes sobre o funcionamento do algoritmo proposto neste trabalho; a Seção 4 apresenta os resultados computacionais obtidos com o algoritmo populacional híbrido desenvolvido. Já a Seção 5 apresenta uma análise empírica dos resultados computacionais, quanto ao impacto do Meme inserido, bem como uma análise estatística. A Seção 6 conclui este artigo.

## 2. Trabalhos Correlatos

Uma gama enorme de trabalhos relacionados ao Problema de Roteamento de veículos com Janela de Tempo (PRVJT) está disponível na literatura. Porém, a revisão bibliográfica deste artigo está concentrada nos artigos acadêmicos que utilizaram PSO e hibridizações para solucionar o PRVJT ou variações desse problema.

O conceito de enxame, citado por Millonas (1993), está relacionado à vida artificial e a conceitos de *Swarm Intelligence*: a proximidade entre as partículas, fatores de qualidade de cada partícula, diversidade, estabilidade e adaptabilidade de uma partícula. A metaheurística populacional PSO, proposta por Kennedy e Eberhart (1995), está fundamentada nesse conceito. Essa técnica é motivada pelo comportamento de organismos como peixes, pássaros e insetos. O diferencial do PSO em relação aos demais algoritmos populacionais é a inspiração nos movimentos físicos dos indivíduos dentro do enxame que, de acordo com Chen et al. (2006), auxilia no equilíbrio da exploração global e local.

De acordo com Krasnogor e Smith (2005), alguns algoritmos são inspirados no conceito de Meme. O Meme pode ser exibido através de refinamentos locais, caracterizando um componente de evolução cultural. Em outras palavras, o Meme é um aspecto de desenvolvimento individual que, em relação ao PSO, pode ser empregado para aprimorar cada partícula do enxame isoladamente.

Na visão de Ong et al. (2006), a eleição do método de busca local, ou Meme a ser utilizado, é uma decisão importante. Essa escolha afeta significativamente o desempenho do algoritmo

em problemas complexos, como o PRVJT. Segundo esses autores, uma escolha adequada deve promover a cooperação e a concorrência entre vários Memes, analisando as diversas estruturas de vizinhança e buscando soluções de alta qualidade, alcançadas com baixos recursos computacionais. No trabalho de Ong et al. (2006) foi realizada uma classificação de acordo com as diferenças nas estratégias adotadas por cada Meme.

Amini et al. (2010) desenvolveram um PSO tradicional e o aplicaram ao PRVJT. O enxame de partículas inicial é gerado de maneira aleatória, assim como a atualização dos parâmetros do algoritmo. Esses autores adotaram como critério de parada a quantidade de iterações. Por fim, Amini et al. (2010) utilizaram o PSO proposto em um estudo de caso com condições reais de transporte.

A priori, Chen et al. (2006) propuseram um PSO com variáveis discretas para solucionar o Problema de Roteamento de Veículos Capacitados. De acordo com esses autores, um *bit* representa um estado de valor 0 ou 1. No âmbito do PRVJT, esse *bit* indica se o veículo em questão atende determinado cliente ou não. Em um segundo momento, Chen et al. (2006) hibridizaram o PSO com a metaheurística de busca local *Simulated Annealing*. O Meme visa refinar as partículas do enxame.

Stehling et al. (2014) utilizaram um Algoritmo Genético híbrido para resolver o PRVJT. A população inicial do algoritmo é gerada pela união entre GRASP e PFIH. Conforme esses autores, a metaheurística GRASP, proposta por Feo e Resende (1995), introduz aleatoriedade aos indivíduos da população inicial e a PFIH, proposta por Solomon (1987), auxilia na característica gulosa da fase construtiva do GRASP.

Masrom et al. (2011) desenvolveram um PSO híbrido, incluindo um operador de mutação ao algoritmo. As restrições do PRVJT foram consideradas durante o processo de mutação das partículas do enxame. Caso a mutação não ocorra, a velocidade e a posição das partículas são calculadas normalmente. A ideia proposta por esses autores é atualizar o enxame através de uma mutação gaussiana.

Como solução para o PRVJT, Ombuki et al. (2006) usaram um Algoritmo Genético multiobjetivo. O operador de mutação proposto por esses autores, denominado *Constrained Route Reversal Mutation* (CRRM), visa escapar de ótimos locais. Baseado em um processo simples de inversão, o CRRM determina um trecho que será disposto ao contrário. Devido à forte restrição associada à janela de tempo incluída no PRVJT, uma alteração mínima deve ser aplicada. Por isso, esses autores definiram um limite máximo de 3 clientes para o tamanho do trecho a ser invertido.

Moghaddam et al. (2012) utilizaram um PSO aplicado ao Problema de Roteamento de Veículos com Demandas Incertas. Esses autores afirmam que a eficiência do algoritmo aumenta com a aplicação de um método de codificação apropriado. Para tanto, foi proposta uma representação bidimensional das partículas do enxame. Além disso, Moghaddam et al. (2012) hibridizaram o PSO com os métodos heurísticos *Variable Neighborhood Search*, *Iterated Greedy* e com outros algoritmos específicos do problema de roteamento.

Hu et al. (2013) combinaram um algoritmo de Otimização Caótica ao PSO para reiniciar o enxame de partículas. Além disso, esses autores hibridizaram o PSO com uma estratégia gaussiana, para evitar ótimos locais, e um algoritmo de inserção, para construir soluções para o PRVJT.

Stehling et al. (2015) propuseram um PSO híbrido para solucionar o PRVJT. Três versões do algoritmo foram apresentadas: (i) um PSO tradicional; (ii) um PSO com enxame de partículas inicial gerado por uma variação do GRASP; e (iii) um PSO híbrido com operador de mutação. Esses autores verificaram, estatisticamente, que o enxame de partículas inicial da versão (ii) apresenta soluções melhores que o enxame da versão (i). Por esse fato, a última versão também apresenta essa característica em sua hibridização.

### **3. Algoritmo Proposto**

Este trabalho apresenta um algoritmo PSO híbrido, denominado HPSO-MEME, com Meme acoplado para a solução do PRVJT. Nesta seção, o algoritmo populacional proposto será descrito com maiores detalhes.

### 3.1. Descrição do Problema

O PRVJT pode ser modelado matematicamente através de um grafo  $G(V, A)$ . O conjunto de vértices  $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$  representam os clientes, sendo  $v_0$  e  $v_{n+1}$  vértices correspondentes ao depósito. O conjunto de arestas  $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$  define as conexões entre os clientes. Cada aresta  $(v_i, v_j)$  possui um custo  $C_{ij}$  associado. Cada vértice do grafo apresenta um tempo de serviço, uma demanda a ser atendida e uma janela de tempo. O depósito possui uma frota homogênea de veículos com capacidade especificada. Veja em Cordeau et al. (2001) a modelagem matemática adotada neste trabalho em maiores detalhes.

### 3.2. Codificando e Decodificando uma Partícula

Os dados de entrada do PRVJT são codificados em uma representação computacional das partículas do PSO. Considerando o número de clientes  $M$  e o número de veículos  $K$ , cada partícula possui  $K$  seções e, para cada uma destas,  $M$  posições. Esse cenário gera uma matriz de dimensão  $K \times M$ . Moghaddam et al. (2012) também utilizaram uma representação bidimensional para o PSO. Cada posição da matriz apresenta valores no âmbito de variáveis discretas. Se o valor é 0, então o cliente  $M_i$  não é atendido pelo veículo  $K_i$ . Se a posição possui valor 1, então o veículo atende o cliente em questão.

Esse processo é inspirado na estratégia adotada por Hu et al. (2013). Porém, neste trabalho, a codificação se trata apenas da representação computacional de uma partícula. A construção de uma solução, avaliando as restrições do PRVJT, é realizada através da combinação GRASP e PFIH, que será mais bem explicada na subseção 3.6. Para demonstrar os resultados, uma partícula é decodificada em uma solução para o PRVJT, gerando rotas de acordo com os  $M$  clientes que são atendidos por cada um dos  $K$  veículos. De acordo com a Figura 1, cada linha  $i$  da matriz, correspondente a um veículo  $K_i$ , representa uma rota do problema e, implicitamente, cada rota tem seu início e término no depósito.

Partícula	$M_1$	$M_2$	$M_3$	(...)	$M_M$
$K_1$	0	0	1	(...)	0
$K_2$	0	0	0	(...)	0
$K_3$	1	1	0	(...)	0
(...)	(...)	(...)	(...)	(...)	(...)
$K_K$	0	0	0	(...)	1

Figura 1: Exemplo fictício da representação computacional de uma partícula.

Na Figura 1, nota-se que o veículo  $K_3$  atende, no mínimo, os clientes  $M_1$  e  $M_2$ . Esse veículo pode atender outros clientes, dependendo da quantidade de clientes  $M$  da matriz. Verifica-se também que as restrições do PRVJT foram satisfeitas. Por exemplo, um cliente deve ser atendido por um único veículo.

### 3.3. Parâmetros do PSO

A velocidade e a posição de cada partícula são calculadas de acordo com o modelo proposto por Shi e Eberhart (1999):

$$V_i = WV_i + \alpha C_1 (P_{best} - X_i) + \beta C_2 (G_{best} - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

Nas expressões (1) e (2), tem-se que:

- $V_i$  indica a velocidade da partícula  $i$ ;
- $X_i$  representa a posição da partícula  $i$ ;
- $W$  é o peso de inércia;
- $C_1$  e  $C_2$  são constantes positivas; e
- $\alpha$  e  $\beta$  são números aleatórios contidos no intervalo  $[0, 1]$ .

Além disso,  $P_{best}$  é a melhor posição obtida individualmente pela partícula  $i$  e  $G_{best}$  é a melhor posição obtida dentre todas as partículas do enxame. Os termos  $C_1$  e  $C_2$  correspondem, respectivamente, ao parâmetro cognitivo e ao parâmetro social de uma partícula. Neste trabalho, os valores dos parâmetros adotados são similares aos propostos empiricamente por Chen et al. (2006).

### 3.4. Função de Avaliação

No PRVJT, dois objetivos conflitantes são considerados: (i) minimizar o número total de veículos utilizados e (ii) minimizar a distância total percorrida. Uma solução para o problema precisa ser convertida em um único problema de otimização. Por esse fato, Ombuki et al. (2006) propuseram uma Função de Avaliação (FA) baseada no método da soma ponderada. Essa FA é utilizada neste trabalho para avaliar a qualidade de uma partícula do enxame e é dada por:

$$FA = \gamma \times N + \delta \times D \quad (3)$$

As ponderações  $\gamma$  e  $\delta$  estão associados, respectivamente, ao número de veículos  $N$  e à distância total percorrida  $D$ . Esses valores foram definidos empiricamente em Ombuki et al. (2006) e utilizados neste artigo.

### 3.5. Meme

De acordo com Ong et al. (2006), um Meme pode ser classificado como estático, adaptativo ou auto-adaptativo. O desenvolvimento cultural de uma partícula, no caso do PSO, também pode ocorrer em diversos níveis: local, global ou externamente ao processo.

A estratégia populacional do PSO proporciona uma busca global no enxame. A hibridização GRASP e PFIH ocorre externamente ao processo, na geração do enxame de partículas inicial. Consequentemente, o Meme adotado deve ser classificado no nível local e apresenta uma estrutura de vizinhança identificada através dos movimentos propostos por Stehling et al. (2014):

- (i) eliminar rota;
- (ii) inserir um cliente em uma determinada posição em uma mesma rota;
- (iii) trocar dois clientes de posição de rotas diferentes;
- (iv) inserir um cliente em uma determinada posição em outra rota; e
- (v) trocar dois clientes de posição em uma mesma rota.

Segundo Ong et al. (2006), um Meme adaptativo pode ser subdividido em qualitativo e quantitativo. Isso indica uma estratégia aleatória e uma estratégia gulosa, respectivamente. O movimento para eliminar rotas do PRVJT funciona de maneira gulosa, eliminando primeiramente as rotas menores. Esse movimento é sempre executado e auxilia na redução do número de veículos utilizados. Os demais movimentos são sorteados com mesma probabilidade de escolha. Após o movimento guloso, um dos quatro restantes é executado para melhorar a qualidade da partícula do enxame e minimizar a distância total percorrida.

### 3.6. Pseudocódigo

O pseudocódigo do Algoritmo HPSO-MEME está demonstrado pelo Algoritmo 1. As linhas 1 a 6 são referentes à combinação GRASP e PFIH proposta. Até que o enxame inicial alcance um tamanho de 100 partículas, uma solução é gerada de maneira gulosa pela PFIH na fase de construção GRASP e, em seguida, é refinada pela fase de busca local. Os movimentos dessa fase são idênticos aos propostos por Stehling et al. (2014). Ao atingir o limite de partículas, o critério de parada é avaliado e o processo do PSO é iniciado.

---

#### Algorithm 1: HPSO-MEME

---

**Entrada:** Número de Veículos, Capacidade dos Veículos, Lista de Clientes com suas Coordenadas Geográficas, Demandas e Janelas de Tempo

**Saída:** Melhor indivíduo da população

```

1  $EnxameParticulas \leftarrow \emptyset$ 
2 enquanto  $Tamanho(EnxameParticulas) < 100$  faça
3    $s \leftarrow Construc\tilde{a}oPFIH()$ 
4    $s' \leftarrow BuscaLocal(s)$ 
5   se  $s' < s$  então
6      $\lfloor Inse\tilde{r}eIndiv\tilde{u}do(s', EnxameParticulas)$ 
7 repita
8    $Aptidao \leftarrow CalculaAptidao(EnxameParticulas)$ 
9    $P_{best} \leftarrow AtualizaMelhorLocal(EnxameParticulas, Aptidao)$ 
10   $G_{best} \leftarrow AtualizaMelhorGlobal(EnxameParticulas, Aptidao)$ 
11   $ProximoEnxame \leftarrow$ 
     $AtualizaParticulas(EnxameParticulas, P_{best}, G_{best})$ 
12   $EnxameParticulas \leftarrow ProximoEnxame$ 
13  para Cada part\tilde{u}cula do enxame faça
14     $p' \leftarrow Mutacao(p)$ 
15     $p' \leftarrow OperadorBuscaLocal(p')$ 
16    if  $p' < p$  then
17       $\lfloor EnxameParticulas \leftarrow AtualizaParticula(p, p')$ 
18 at\tilde{e} Criterio de parada aceito
  
```

---

As linhas 8 a 12 do algoritmo 1 indicam a estratégia tradicional do PSO: (i) a aptidão de cada partícula é calculada; (ii) cada  $P_{best}$  é alterado; (iii) o  $G_{best}$  é atualizado avaliando a melhor partícula dentre todas do enxame; (iv) a posição e a velocidade de cada partícula são atualizadas; e (v) o próximo enxame é gerado.

Em seguida, para cada uma das partículas do enxame, o operador de mutação é executado. A adaptação do CRRM está no número máximo de clientes a serem invertidos na rota. Esse número é inversamente proporcional à qualidade da partícula, ou seja, quanto melhor a solução, menor o trecho a ser invertido. Tal número de clientes pertence ao intervalo  $[2, 5]$  e a qualidade das partículas é medida através da FA definida pela expressão (3).

Após a mutação, cada partícula é refinada através do Meme acoplado ao algoritmo, indicado pela linha 15. Os movimentos de busca local estão descritos na subseção 3.5 deste artigo. Vale ressaltar que, se não houve melhora após a mutação e o refinamento, a partícula inicial é mantida no enxame.

### 4. Resultados Computacionais

O algoritmo proposto foi desenvolvido em linguagem de programação C e os experimentos computacionais foram executados em um notebook Intel Core i7 2, 40Ghz com 8GB de memória

RAM e sistema operacional Windows 8.1. Os testes foram realizados utilizando as 56 instâncias de Solomon (1987) de 100 consumidores para o PRVJT.

Devido à estocasticidade do método, o HPSO-MEME foi executado 40 vezes para cada instância. Os resultados por classe, tanto em número de veículos quanto em distância total percorrida, estão demonstrados nas Tabelas 1 a 6. Esses valores são comparados aos obtidos com a versão de Stehling et al. (2015), denominada HPSO, e com os melhores resultados obtidos na literatura, apresentados na coluna MRL. Os dados referentes aos melhores resultados resultados da literatura foram retirados de <http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/>.

Tabela 1: Instância C1

Instância	MRL	HPSO	HPSO-MEME
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 — 828,94	10 — 828,06
C104	10 — 824,78	10 — 828,94	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
<b>TOTAL</b>	<b>90 — 7455,42</b>	<b>90 — 7455,42</b>	<b>90 — 7455,42</b>

Tabela 2: Instância C2

Instância	MRL	HPSO	HPSO-MEME
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
<b>TOTAL</b>	<b>24 — 4718,87</b>	<b>24 — 4718,87</b>	<b>24 — 4718,87</b>

## 5. Análise dos Resultados

Esta seção apresenta, na seção 5.1, uma análise dos resultados alcançados pelo algoritmo HPSO-MEME para a solução das instâncias do PRVJT introduzidas por Solomon (1987). Em seguida, na seção 5.2, mostra-se uma análise do impacto da introdução do Meme no algoritmo. Por fim, na seção 5.3, apresenta-se uma análise estatística dos resultados, de modo a caracterizar de forma efetiva o significado desses resultados.

### 5.1. Análise Empírica

Segundo Cordeau et al. (2001), as instâncias de Solomon (1987) são divididas em 6 classes: C1, C2, R1, R2, RC1 e RC2. As instâncias do tipo C apresentam clientes agrupados. Nas do tipo R, os clientes estão aleatoriamente espalhados no espaço geográfico. As do tipo RC, por fim, são uma combinação das duas anteriores.

Cada uma dessas instâncias é subdividida em dois grupos. A classe C, por exemplo, possui dois subgrupos: C1 e C2. No primeiro, os clientes dispõem de uma janela de tempo curta

Tabela 3: Instância R1

<b>Instância</b>	<b>MRL</b>	<b>HPSO</b>	<b>HPSO-MEME</b>
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	9 — 960,88	9 — 960,88
R109	11 — 1194,73	11 — 1194,73	11 — 1194,73
R110	10 — 1118,59	10 — 1118,59	10 — 1118,59
R111	10 — 1096,72	10 — 1096,72	10 — 1096,72
R112	9 — 982,14	9 — 982,14	9 — 982,14
<b>TOTAL</b>	<b>143 — 14518,64</b>	<b>143 — 14518,64</b>	<b>143 — 14518,64</b>

Tabela 4: Instância R2

<b>Instância</b>	<b>MRL</b>	<b>HPSO</b>	<b>HPSO-MEME</b>
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	3 — 809,36	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	3 — 912,44	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
<b>TOTAL</b>	<b>30 — 10461,22</b>	<b>34 — 10547,35</b>	<b>30 — 10461,22</b>

Tabela 5: Instância RC1

<b>Instância</b>	<b>MRL</b>	<b>HPSO</b>	<b>HPSO-MEME</b>
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	10 — 1135,48	10 — 1135,48
RC105	13 — 1629,44	13 — 1629,44	13 — 1629,44
RC106	11 — 1424,73	11 — 1424,73	11 — 1424,73
RC107	11 — 1230,48	11 — 1230,48	11 — 1230,48
RC108	10 — 1139,82	10 — 1139,82	10 — 1139,82
<b>TOTAL</b>	<b>92 — 11073,31</b>	<b>92 — 11073,31</b>	<b>92 — 11073,31</b>

que, conseqüentemente, implica na utilização de um número maior de veículos. O tempo de deslocamento de um cliente para outro é o maior obstáculo para a formação das rotas. No segundo subgrupo, os clientes apresentam uma janela de tempo larga. Isso significa que um veículo consegue atender vários clientes em uma mesma rota, reduzindo a quantidade de veículos utilizados em uma solução.

Tabela 6: Instância RC2

Instância	MRL	HPSO	HPSO-MEME
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 — 831,98	3 — 798,41
RC205	4 — 1297,19	5 — 1317,92	4 — 1297,19
RC206	3 — 1146,32	4 — 1087,99	3 — 1146,32
RC207	3 — 1061,14	4 — 1071,03	3 — 1061,14
RC208	3 — 828,14	3 — 828,14	3 — 828,14
<b>TOTAL</b>	<b>26 — 8125,24</b>	<b>30 — 8104,95</b>	<b>26 — 8125,24</b>

Conforme as Tabelas 1 a 6, nota-se que o HPSO enfrentou dificuldades apenas em duas classes de instâncias: R2 e RC2. Isso indica que o maior desafio desse algoritmo está nos clientes distribuídos de maneira aleatória e que apresentam janela de tempo larga. A técnica populacional PSO, em geral, utiliza uma forte exploração do espaço de busca, verificando todas as partículas do enxame para avaliar qual direção seguir. Por isso, a versão HPSO-MEME integra um método de busca local ao processo que, ainda de acordo com essas tabelas, eliminou a deficiência do HPSO.

A Tabela 7 mostra os resultados médios por classe e totais obtidos, comparando os melhores resultados da literatura com os seguintes algoritmos:

- OMB: algoritmo genético multiobjetivo proposto por Ombuki et al. (2006);
- AGH: algoritmo genético híbrido desenvolvido por Stehling et al. (2014);
- HU: PSO híbrido utilizado por Hu et al. (2013), executado em 500 iterações; e
- HPSO e HPSO-MEME foram apresentados na seção 4 deste artigo.

Tabela 7: Resultados médios por classe e totais obtidos

Instância	MRL	OMB	AGH	HPSO	HU	HPSO-MEME
C1	10 - 828,38	10 - 828,48	10 - 828,38	10 - 828,38	10 - 828,38	10 - 828,38
C2	3 - 589,86	3 - 590,60	3 - 589,86	3 - 589,86	3 - 589,86	3 - 589,86
R1	11,92 - 1209,89	12,07 - 1204,48	12,16 - 1217,1	11,92 - 1209,89	11,92 - 1215,79	11,92 - 1209,89
R2	2,73 - 951,02	3,1 - 893,03	2,91 - 958,33	3,1 - 958,85	2,73 - 953,53	2,73 - 951,02
RC1	11,5 - 1384,16	12,05 - 1370,79	11,86 - 1401,18	11,5 - 1384,16	11,5 - 1414,24	11,5 - 1384,16
RC2	3,25 - 1160,75	3,5 - 1025,31	3,5 - 1122,0	3,75 - 1157,35	3,25 - 1135,99	3,25 - 1160,75
<b>TOTAL</b>	<b>405 - 56352,70</b>	<b>417 - 55627,01</b>	<b>415 - 57506,53</b>	<b>413 - 56418,54</b>	<b>405 - 57648,42</b>	<b>405 - 56352,70</b>

Analisando a Tabela 7, nota-se que o HPSO-MEME alcançou os melhores resultados da literatura apresentados na coluna MRL. Mesmo obtendo a mesma quantidade de veículos utilizados pelo HPSO-MEME, o HU perdeu em distância total percorrida para todos os algoritmos.

## 5.2. Análise de Impacto do Meme Acoplado

Para avaliar a melhoria nos resultados após a inclusão do Meme, não somente nos resultados finais, mas também durante o processo do PSO, foi realizado um teste para analisar a qualidade das partículas em cada iteração do HPSO-MEME. Tal qualidade foi medida com base na FA proposta neste trabalho. A Figura 2 mostra o comparativo entre o enxame de partículas refinadas e o de partículas não refinadas.

O algoritmo foi executado 40 vezes. Em cada iteração está apresentado o valor médio obtido dentre todas as execuções. O critério de parada do algoritmo não é o número de iterações. Porém, o teste foi limitado às primeiras 50 iterações do HPSO-MEME. Uma bateria de teste foi

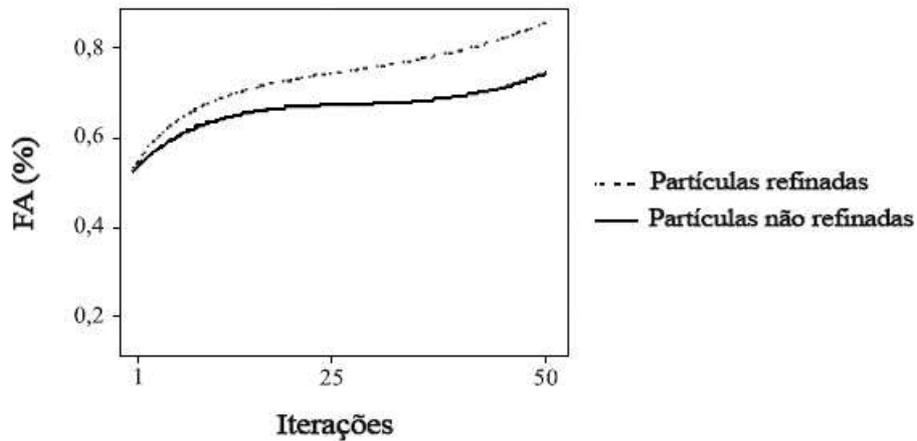


Figura 2: Comparativo entre as partículas refinadas e as não refinadas.

executada com o Meme refinando as partículas e outra foi executada sem o trecho de código referente ao Meme acoplado. Nota-se que, desde o início da execução do algoritmo, as partículas refinadas apresentam maior qualidade que as partículas não refinadas.

### 5.3. Análise Estatística

Inicialmente, as medidas de tendência central e de variabilidade dos dados foram avaliadas através de uma análise estatística descritiva. A Figura 3 refere-se a esses dados. O *boxplot* apresenta os resultados obtidos nas versões HPSO e HPSO-MEME de acordo com a qualidade das soluções dos mesmos. Essa qualidade é calculada através da FA das partículas do enxame.

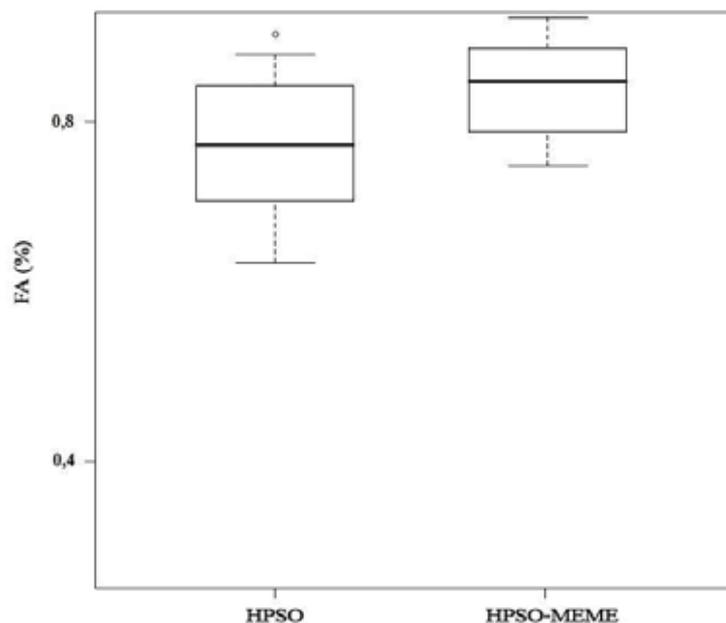


Figura 3: *Boxplot* dos enxames de partículas finais.

Visualmente, nota-se que a versão HPSO-MEME obteve melhores resultados, tanto pela mediana ter melhorado, quanto pela dispersão dos dados ter diminuído. Porém, como houve sobreposição dos dados, nada pode ser afirmado. Portanto, a próxima etapa foi realizar um teste de múltiplas comparações.

Segundo Montgomery e Runger (2009), o Teste de Tukey é baseado na Diferença Mínima Significativa e permite a avaliação do contraste entre duas médias de tratamentos. No Teste de Tukey realizado, com  $\alpha$  fixado em 0,05, foi gerado um intervalo de confiança com limite inferior de 0,2042 e limite superior de 0,5314. 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 o algoritmo HPSO-MEME é melhor que o HPSO.

## 6. Conclusão

O presente artigo apresenta um PSO híbrido com Meme acoplado para solucionar o PRVJT. O algoritmo proposto foi comparado com os melhores resultados da literatura e com trabalhos apresentados na revisão bibliográfica deste artigo. Esta comparação foi realizada utilizando as 56 instâncias de Solomon (1987) para o PRVJT, tendo 100 consumidores cada.

A análise empírica mostrou que o HPSO-MEME eliminou as deficiências encontradas na versão HPSO. Além disso, destacou que o HPSO-MEME atingiu os melhores resultados da literatura e obteve o melhor resultado dentre os algoritmos comparados.

A partir da análise de impacto da inclusão do Meme, nota-se que as partículas do enxame tiveram a qualidade aprimorada não somente no resultado final, auxiliando, assim, no processo de busca por melhores soluções para o PRVJT.

Por fim, a análise estatística dos experimentos permite afirmar que a robustez do algoritmo melhorou e que, com uma significância estatística de 95%, o HPSO-MEME é melhor que o HPSO. Portanto, conclui-se que o algoritmo populacional proposto atendeu ao objetivo de solucionar o PRVJT.

## Agradecimentos

Os autores agradecem ao CEFET-MG, à CAPES e à FAPEMIG pelo apoio à realização deste trabalho.

## Referências

- Amini, S., Javanshir, H. e Tavakkoli-Moghaddam, R.** (2010), A PSO approach for solving VRPTW with real case study, *Int. J. Res. Rev. Appl. Sci.*, 118-126.
- Chen, A. L., Yang G. K., e Wu, Z. M.** (2006), Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, *Journal of Zhejiang University Science A*, 607-614.
- Cordeau, J. F., Desaulniers, G., Desrosiers, J. e Solomon, M. M.**, The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Toth, P. e Vigo, D. (Eds.), *The VRP with Time Windows*, SIAM, Philadelphia, EUA, 157-194, 2001.
- Feo, T. A. e Resende, M. G. C.** (1995), Greedy randomized adaptive search procedures, *Journal of Global Optimization*, 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.
- Hu, W., Liang, H., Peng, C., Du, B. e Hu, Q.** (2013), A hybrid chaos-particle swarm optimization algorithm for the vehicle routing problem with time window, *Entropy*, 1247-1270.
- Kennedy, J. e Eberhart, R.** (1995), Particle swarm optimization, *In Proceedings of the 1995 IEEE International Conference on Neural Network*.
- Krasnogor, N. e Smith, J.** (2005), A tutorial for competent memetic algorithms: Model, taxonomy, and design issues, *IEEE Trans. Evol. Comput.*, 474-488.
- Masrom, S., Abidin, S. Z. Z., Nasir, A. M. e Rahman, A. S. A.** (2011), Hybrid particle swarm optimization for vehicle routing problem with time windows, *In Proceedings of the 2011 International Conference on Recent Researches in Computational Techniques, Non-Linear Systems and Control*, 142-147.
- Millonas, M. M.**, Artificial Life III, Addison-Wesley (Eds.), *Swarms, phase transitions, and collective intelligence*, Reading, MA, USA, 417-445, 1993.

**Moghaddam, B. F., Ruiz, R., Sadjadi, S. J.** (2012), Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm, *Computers and Industrial Engineering*, 306-317.

**Montgomery, D. e Runger, G.**, *Applied statistics and probability for engineers*, Wiley, 4nd edition, 2009.

**Ombuki, B., Ross, B. J. e Hanshar, F.** (2006), Multi-objective genetic algorithms for vehicle routing problem with time windows, *Applied Intelligence*, 17-30.

**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*, 141-152.

**Shi, Y. e Eberhart, R. C.** (1999), Empirical study of particle swarm optimization, *In Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, 1945-1950.

**Solomon, M. M.** (1987), Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, 35, 254-265.

**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)*.

**Stehling, T. M., Souza, S. R. e França Filho, M. F.** (2015), A hybrid particle swarm optimization for solving vehicle routing problem with time windows, *Poster Section of Genetic and Evolutionary Computation Conference (GECCO 2015)*, no prelo.