

ABORDAGEM IMUNOLÓGICA AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEAS

Vinícius Wellington Coelho de Moraes

Universidade Federal de Minas Gerais (UFMG)

Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG CEP 31270-901

vwcMorais@dcc.ufmg.br

André L. Maravilha

Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)

Rua da Glória, 187 – 39.100-000 – Diamantina – MG – Brasil

msilva.andreluiz@gmail.com

Luciana Pereira de Assis

Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)

Rua da Glória, 187 – 39.100-000 – Diamantina – MG – Brasil

lupassis@gmail.com

Alessandro Vivas Andrade

Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)

Rua da Glória, 187 – 39.100-000 – Diamantina – MG – Brasil

alessandro.vivas@gmail.com

RESUMO

Esse artigo discute o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas (PRVCES), uma variação do clássico Problema de Roteamento de Veículos, cujo objetivo é minimizar o custo associado às rotas percorridas por veículos que atendem um conjunto de consumidores e suas demandas de coleta e entrega, simultaneamente. O PRVCES é resolvido neste trabalho usando uma abordagem imunológica. Os algoritmos implementados foram o Clonalg, B-Cell, Opt-AiNet e Copt-Ainet. Os resultados mostraram que para três instâncias com até 100 clientes, o presente trabalho obteve melhores resultados comparados aos da literatura para as mesmas instâncias do problema, sugerindo que os algoritmos imunológicos possuem grande potencial de aplicação à variantes dos Problema de Roteamento de Veículos.

PALAVRAS CHAVE. Logística & Transportes, Metaheurísticas, Otimização Combinatória. **Área principal.** Logística & Transportes.

ABSTRACT

This paper addresses the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD), a variation of the classic Vehicle Routing Problem, whose goal is to minimize the cost of routes traveled by vehicles that serve a set of consumers and their demands for pickup and deliveries simultaneously. The VRPSPD is solved in this work using an immune approach. The algorithms implemented were Clonalg, B-Cell, Opt-AiNet and Copt-Ainet. The results showed, that for 3 (three) instances tests with up to 100 customers, the system implemented was better than the best results in the literature for the same instances of the problem, suggesting that immune algorithms have great potential application for variants of the Vehicles Routing Problems.

KEYWORDS. Logistics & Transport, Metaheuristics, Combinatorial Optimization. **Main area.** Logistics & Transport.

1. Introdução

Hoje, com a aplicação de várias tecnologias, a indústria e a agricultura têm superado seus índices de produção anual, mas a forma como os produtos são remetidos ao consumidor final ainda é um problema a ser superado.

Existem diversos fatores que elevam os custos dos transportes de produtos, entre eles destacamos o mau estado da rede viária, a insegurança dos meios de transporte, pedágios, preços dos combustíveis, manutenção da frota de veículos e leis ineficazes. Assim, as empresas procuram adaptar soluções tecnológicas de logística à sua produção, a fim de reduzir seus custos.

Outro fator relevante é a preocupação da sociedade com os problemas ambientais. Atualmente, a fim de agregar valor aos seus produtos, além de produzir com qualidade, menor custo e em grande escala, as empresas também devem se preocupar com o impacto ambiental causado pelas atividades industriais [Caires et al. (2009)].

Assim, além da distribuição de seus produtos, as empresas devem recolher seus resíduos e reciclá-los, para atender às legislações ambientais ou para evitar o desperdício de matéria prima, maximizando seus lucros. Esta prática caracteriza um problema de logística reversa. E, as organizações que implementam soluções tecnológicas para a distribuição de seus produtos podem se destacar no ambiente competitivo [Morais et al. (2009)].

1.1. Definição do Problema

O Problema de Roteamento de Veículos (PRV) proposto por Dantzig e Ramser (1959), é um clássico problema de otimização combinatória. Na literatura há uma série de problemas decorrentes de variações do PRV, sendo um deles o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas (PRVCES). Este pode ser definido como: dado uma frota de k veículos com capacidade Q , que devem atender simultaneamente as demandas de coleta e entrega de um conjunto de N consumidores, retornando para o depósito no final do trajeto com o total coletado. O objetivo do problema é minimizar o custo das rotas percorridas pelos veículos.

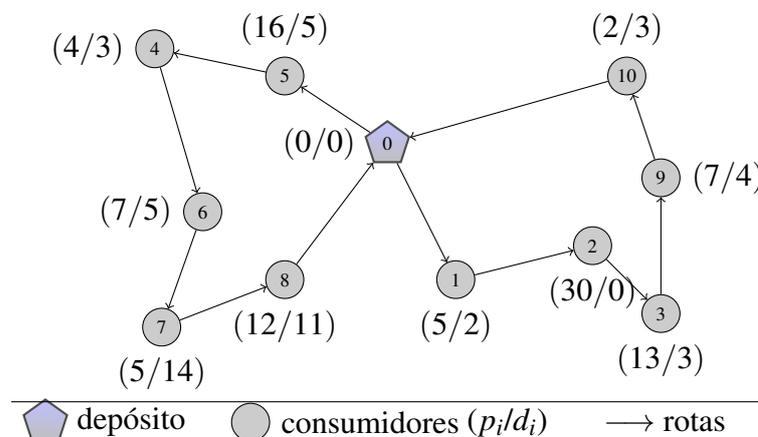


Figura 1: Ilustração do PRVCES

A Figura 1 ilustra o PRVCES, os consumidores estão numerados 1 a N , o depósito é

representado pelo nó 0 (zero) e os valores entre parênteses representam as demandas de coleta (p_i) e entrega (d_i), respectivamente.

Neste exemplo, existem 10 (dez) consumidores que foram divididos em duas rotas a serem percorridas por veículos de capacidade $Q = 70$. Na primeira rota, o veículo parte do depósito com a soma das demandas de entrega igual a 38 ($\sum_{i=0}^D = 38$), atende os consumidores 5, 4, 6, 7, 8, e retorna para o depósito com o total coletado ($\sum_{i=0}^P = 44$). Como o veículo anterior, o segundo atende às demandas dos consumidores 1, 2, 3, 9, 10 e retorna para o depósito ao final do percurso.

Na prática, o PRVCES pode ser aplicado em cervejarias, companhias aéreas, transportadoras em geral, entre outros. Segundo Dethloff (2001) esse problema é NP - Difícil, assim existem muitos estudos na literatura que buscam encontrar boas soluções ao problema em um tempo computacional viável.

Este trabalho propõe uma adaptação de algoritmos imunológicos para resolver o PRVCES. Foram desenvolvidos quatro métodos: o Clonalg, B-Cell, Opt-AiNet e Copt-Ainet. Na próxima subseção será apresentado o histórico do problema, na seção 2 são descritos os algoritmos imunológicos implementados, a seção 3 apresenta os parâmetros e testes, descreve e analisa os resultados, posteriormente, as conclusões são apresentadas na Seção 4.

1.2. Revisão da literatura

Desde que foi proposto por Min (1989), o PRVCES foi pouco explorado, mas a partir dos trabalhos de Dethloff (2001), vários pesquisadores têm contribuído para a divulgação e resolução do problema. Assis (2007), por exemplo, apresenta três heurísticas construtivas baseadas no mecanismo Dividir e Rotear aplicando o conceito de Árvore Geradora Mínima na fase de divisão dos consumidores.

Em 2005, Dell'Amico et al. (2005) usaram um algoritmo *Branch-and-price* para resolver o problema usando duas estratégias: uma abordagem exata e uma relaxação do espaço de estados (*state space relaxation*).

Em 2006, Montané e Galvão (2006) desenvolveram o primeiro trabalho através da aplicação de metaheurísticas para PRVCES, onde um algoritmo de Busca Tabu foi proposto. Nesta linha, diversos autores deram suas contribuições. Chen J-F (2006) e Wassan et al. (2007) também propuseram versões da metaheurísticas Busca Tabu, para o problema.

Subramanian et al. (2008) e Morais et al. (2009) aplicaram algoritmos híbridos baseados nas metaheurísticas *Iterated Local Search* (ILS), *Variable Neighborhood Descent* (VND), *Variable Neighborhood Search* (VNS) e *Greedy Random Adaptive Search Procedure* (GRASP).

Zachariadis et al. (2009), aplicaram um mecanismo híbrido baseado nas metaheurísticas Busca Tabu e *Guided Local Search* (GLS), para resolver o PRVCES.

Quanto aos algoritmos bio-inspirados, Vural (2003) apresentou a primeira implementação de Algoritmo Genético ao problema. Gökçe (2004), Çatay (2006), Gajpal e Abad (2009) e Zhang et al. (2008), trabalharam com o algoritmo de otimização por colônia de formigas para o PRVCES.

2. Algoritmos Imunológicos

Os Sistemas Imunológicos Artificiais (SIA) surgiram a partir da definição de ferramentas computacionais análogas a procedimentos do sistema imunológico biológico [Dasgupta (1998)]. Similarmente a outros métodos bio-inspirados e metaheurísticas, o objetivo do SIA é resolver problemas complexos que não podem ser resolvidos em tempo hábil por métodos tradicionais. Diversos problemas podem ser modelados usando algoritmos imunológicos, como reconhecimento e classificação de padrões, segurança, análise de dados, otimização, dentre outros problemas do mundo real.

Os algoritmos imunológicos evoluem a partir de um conjunto de soluções, ou população de indivíduos candidatos. Assim é possível obter uma maior diversificação, explorando de forma mais eficiente o espaço de soluções para um determinado problema.

Um clássico algoritmo de redes imunológicas artificiais, o aiNET (*Artificial Immune Network*), originalmente proposto para trabalhar com clusterização e compressão de dados é uma referência aos Algoritmos Imunológicos [Castro e Zuben (2001)].

Numa tentativa de aproveitar a flexibilidade dos algoritmos bio-inspirados, o presente trabalho propõe uma adaptação de algoritmos imunológicos para resolver o PRVCES. Os algoritmos propostos utilizam as seguintes notações:

- um **indivíduo** ou célula é uma solução para o PRVCES. Por exemplo, considere a Figura 1, nesse exemplo 10 consumidores são atendidos por dois veículos, então essa solução seria representada como:

$$s = [0 - 1 - 2 - 3 - 9 - 10 - 0 - 5 - 4 - 6 - 7 - 8 - 0]$$

onde um dos veículos partindo do depósito (0) atenderia aos consumidores 1, 2, 3, 9 e 10, nesta ordem e retornaria ao depósito, já o segundo veículo atenderia aos consumidores 5, 4, 6, 7 e 8, respectivamente.

- um **clone** é uma cópia fiel de um indivíduo.
- o **fitness** é um valor que quantifica o quão bom é um indivíduo frente à função objetivo. No caso específico do PRVCES, é o resultado da função objetivo, ou seja, o somatório dos pesos associados às arestas que constituem a solução.
- o valor da **afinidade** atesta o quão próximo um indivíduo está de outro. Em se tratando de Problemas de Roteamento de Veículos, a afinidade é o número de arestas coincidentes entre duas soluções. Geralmente nos algoritmos implementados neste trabalho, a afinidade é calculada a partir do melhor indivíduo da população corrente.
- **Mutação**, no escopo dos algoritmos imunológicos é um mecanismo de busca, caracterizado por uma perturbação no indivíduo corrente.
- **Maturação de afinidade**, é o processo de mutação parametrizado pela afinidade.

Os operadores de mutação e maturação por afinidade implementados neste trabalho foram adaptados do operador *Insert Mutation* proposto por Oliveira e Vasconcelos (2008). Este

operador remove r vértices de uma rota e os reinsere em outras rotas em posições viáveis. No operador de mutação o valor de r é um parâmetro informado no início da execução do algoritmo. No caso do operador de maturação de afinidade o valor de r representa a taxa de mutação inversamente proporcional ao *fitness*, avaliada em função do melhor indivíduo na população, e equivale a um valor entre 1 e NV (número de vértices da instância). Assim, quanto maior a afinidade do indivíduo, menores serão também as taxas de maturação e mutação aplicadas a este.

2.1. CLONALG

Acompanhando a evolução dos SIA, de Castro e Zuben (2000) propuseram o algoritmo CLONALG para resolver inicialmente problemas de aprendizagem de máquina e reconhecimento de padrões.

CLONALG é um acrônimo de *CLONal selection ALGORITHM*, baseado no processo de seleção clonal do sistema imunológico natural, processo onde somente as células capazes de reconhecer antígenos são selecionadas e melhoradas por meio de mutações por afinidade.

Este algoritmo evolui a partir de uma população inicial, onde alguns indivíduos são selecionados e submetidos a um processo de clonagem. Os clones gerados, por sua vez passam por processos de mutação e maturação da afinidade.

O CLONALG explora algumas características importantes, como a retenção de alguns indivíduos mais adaptados, manutenção da diversidade e taxa de mutação inversamente proporcional à afinidade.

O Algoritmo 1 descreve o funcionamento do CLONALG.

Algoritmo 1 CLONALG

- 1: gerar população inicial P ;
 - 2: **enquanto** $contador \leq numMaxGerações$ **faça**
 - 3: selecionar os N melhores indivíduos em P ;
 - 4: gerar a população (C) de N_c clones dos N melhores indivíduos de P ;
 - 5: submeter a população C a um processo de mutação;
 - 6: submeter a população C a um processo de maturação da afinidade;
 - 7: selecionar os melhores indivíduos de C para recompor a população P ;
 - 8: substituir D indivíduos da população;
 - 9: **fim enquanto**
-

No passo inicial, uma população P é gerada de forma aleatória, sendo que P é formada por indivíduos que representam uma solução válida para o problema. Para cada geração são executados os passos 3-8. Nos passos 3 e 4, os N melhores indivíduos presentes na população P são selecionados e clonados, gerando uma população C . Esses clones darão origem aos possíveis novos indivíduos da população P , na próxima geração. Cada clone passa por um processo de mutação e maturação da afinidade respectivamente nos passos 5 e 6.

No passo 7, os melhores indivíduos presentes na população C , são selecionados para recompor a população P . No passo 8, os D indivíduos menos adaptados, ou seja, os com maiores valores de *fitness* do conjunto P , são substituídos por novos indivíduos gerados de forma aleatória, a fim de manter a diversidade da população, explorando ainda mais espaço de soluções.

2.2. B-CELL

O algoritmo B-Cell (BCA), proposto por Kelsey e Timmis (2003), é inspirado no processo de seleção clonal. A principal diferença entre o BCA e o CLONALG é o fato de o BCA conter apenas um operador de mutação, definido pelos autores como *Contiguous somatic Hypermutation*.

O algoritmo B-Cell evolui a partir de uma população inicial P , e a cada geração todos os indivíduos são clonados para produzirem uma população de clones C . Cada clone é submetido a um processo de mutação. Para cada indivíduo pertencente a P , são gerados x clones, onde x é a taxa de clonagem, assim o tamanho da população de clones C é x vezes maior que P .

Os passos do B-Cell são descritos no Algoritmo 2. No passo 1, uma população inicial de $|P|$ indivíduos é gerada aleatoriamente. Cada indivíduo representa uma solução válida para o problema.

Algoritmo 2 B-Cell

- 1: criar uma população aleatória P ;
 - 2: **enquanto** $contador \leq numMaxGeracoes$ **faça**
 - 3: avalia a afinidade de todos os indivíduos de P ;
 - 4: para cada indivíduo de P , produzir um conjunto de clones e inserir em C ;
 - 5: para cada clone em C , submetê-lo ao processo de mutação proporcional à afinidade;
 - 6: avaliar a afinidade de cada clone em C , se o clone estiver mais apto que seu pai, realizar a troca, substituindo o pai na população P pelo clone;
 - 7: **fim enquanto**
-

Para cada geração, os passos 2-6 são executados. No passo 3, a afinidade de cada indivíduo em relação ao melhor indivíduo da população, é avaliada. No passo 4 cada indivíduo da população gera N_c clones, formando uma população C de clones. No passo 5 todos os clones são submetidos a um processo de mutação. No passo 6, a afinidade dos clones é avaliada, se sua afinidade for melhor que a do seu pai, então, ele o substitui na população P .

2.3. Opt-aiNET

O Opt-aiNET foi proposto por Castro e Timmis (2002), e, é uma extensão do algoritmo aiNET para problemas de otimização. Esse algoritmo evolui a partir de um conjunto de indivíduos (população inicial), tidos como soluções candidatas para um determinado problema. Esse algoritmo tem processo de expansão clonal, mutação e processo de seleção.

A população nesse algoritmo funciona como memória com as melhores soluções candidatas em um dado instante de tempo, avaliadas pela função objetivo.

As principais características do Opt-AiNET são:

- mantém uma população de tamanho dinamicamente ajustável;
- contém fase de exploração e intensificação do espaço de busca;
- pode manter um subconjunto de soluções ótimas locais;
- existência de mecanismo de seleção elitista;

O Algoritmo 3 descreve as ações do Opt-AiNET. No passo 1 a população inicial é gerada. Essa população consiste de P indivíduos gerados de forma aleatória.

Para cada geração iterada são executados os passos 3-14, onde cada indivíduo da população passa por um processo de expansão clonal, que consiste em gerar NC clones e submetê-los ao processo de mutação por afinidade.

Algoritmo 3 Opt-AiNET

```
1: gera população inicial;
2: enquanto  $contador \leq numMaxGeracoes$  faça
3:   enquanto  $critParada \neq false$  faça
4:     gera  $NC$  clones para cada indivíduo da população;
5:     para cada clone gerado executar a mutação;
6:     se  $Clone(i)$  melhor que seu pai então
7:       substitui o pai na população, pelo  $clone(i)$ ;
8:     fim se
9:     se  $(preAvgFitness - postAvgFitness) < errorThres$  então
10:       $critParada \leftarrow false$ ;
11:    fim se
12:  fim enquanto
13:  executa supressão;
14:  introduz novas células na população;
15: fim enquanto
```

O *fitness* do clone mutado e do seu pai são avaliados, o mais adaptado é mantido na população. Esse processo se repete até que o *fitness* médio da população se estabilize (linhas 9-10), isso ocorre quando a diferença entre o *fitness* médio da população na geração anterior (*preAvgFitness*) e na geração corrente (*postAvgFitness*) for menor que o erro de supressão (*errorThres*), que é um parâmetro passado no início da execução do programa.

Assim que a população se estabiliza são executados os passos 13 e 14. Primeiro é executada a supressão por meio de um processo elitista, onde os p indivíduos menos adaptados são removidos da população. Depois de realizada a supressão, novos indivíduos são inseridos na população. Este processo aumenta a diversidade e determina a cardinalidade da população corrente.

2.4. Copt-aiNET

Proposto por Souza et al. (2004), o Copt-aiNET é uma extensão do AiNET para resolver problemas de otimização combinatória. Esse algoritmo, assim como o Opt-aiNET, evolui a partir de uma população inicial, porém, suas principais diferenças são as fases de maturação fraca da afinidade, processo de intensificação, além possuir um tamanho mínimo para a população. O Copt-aiNET avalia a qualidade de cada indivíduo seguindo dois parâmetros: a afinidade e o *fitness*.

O Algoritmo 4 descreve o procedimento Copt-AiNET. No passo 1, a população inicial é gerada de forma aleatória. Para cada geração iterada, são executados os passos 3 a 16. Na linha 3, cada indivíduo da população passa pelo processo de expansão clonal. Na linha 4, cada clone gerado executa o processo de mutação.

O *fitness* do clone mutado e do seu pai são avaliados, o mais adaptado, ou seja, o de menor *fitness* é mantido na população. Depois da clonagem e mutação, é verificado se o *fitness*

Algoritmo 4 Copt-AiNET

```
1: gera população inicial;
2: enquanto critérioParada  $\neq$  false faça
3:   executa a clonagem;
4:   para cada clone gerado fazer a mutação;
5:   se o fitness médio da População não melhorar então
6:     executa a supressão;
7:   fim se
8:   se o tamanho da população for menor que M então
9:     insere novos indivíduos na população;
10:  fim se
11:  se não houver melhoria nos K melhores indivíduos nas últimas MX iterações então
12:    submeta todos os indivíduos da população ao processo de maturação;
13:  fim se
14:  se o fitness dos k melhores indivíduos não tiver sido alterado nas últimas P iterações então
15:    critérioParada  $\leftarrow$  false ;
16:  fim se
17: fim enquanto
18: os k passam por processo de intensificação;
```

médio da população melhorou com relação à última iteração, caso não tenha ocorrido evolução, realiza-se o processo de supressão (linhas 5 e 6).

Nesse algoritmo, o processo de supressão mantém o melhor indivíduo e elimina de forma aleatória os demais até satisfazer a taxa de supressão tx . Depois dessa etapa, se a população atingir um número mínimo de indivíduos, então novos indivíduos são inseridos na população (linha 8 a 10).

Nos próximos passos (11 a 13), os k melhores indivíduos da população são avaliados, depois de MX iterações, se não houver melhorias nestes indivíduos toda a população passará por processo de maturação.

Assim que a população estabilizar (linhas 14-15) o algoritmo é finalizado e os k melhores indivíduos são selecionados para passarem por um processo de intensificação. Esse processo é dado pela metaheurística VND, que é um método de refinamento que explora por meio de trocas sistêmicas o conjunto de soluções vizinhas. Os operadores de vizinhança codificados neste trabalho e que implementam o processo de intensificação são: Realocação, 2-Opt e Eliminação de Rotas [Morais et al. (2009)].

Decorrido todos os passos descritos nos Algoritmos 1, 2, 3 e 4, aquele indivíduo que se mostrar melhor adaptado frente a função objetivo é considerado a solução para o problema.

3. Resultados Computacionais

Os algoritmos imunológicos foram implementados em linguagem Java (jdk 1.6) e executados em uma máquina com processador intel Quad Core Q8200, 2.33 GHz, com 3 GB de RAM e Sistema Operacional Linux - Distribuição Ubuntu 9.04. Embora esse equipamento tenha múltiplos núcleos, esse recurso não foi utilizado.

Durante os testes, os algoritmos foram executados para populações de 10, 30, 50 e 100 indivíduos, quanto ao número de iterações ou gerações, testes foram feitos para 100, 200, 300, 500 e 1000, o que determinou o erro médio de supressão do algoritmo Opt-aiNet.

Para ajustar a taxa de clonagem os valores utilizados foram 3, 4, 5, 10 e 20. Nos algoritmos Opt-aiNet e Copt-aiNet, a taxa percentual de supressão foi de 30%, o que indica o número de indivíduos descartados da população corrente. O valor percentual de novos indivíduos introduzidos é avaliado sob a cardinalidade da população depois de realizada a supressão.

Os parâmetros utilizados para se gerar os resultados descritos a seguir são apresentados na Tabela 1.

Tabela 1: Parâmetros utilizados na execução dos Algoritmos Imunológicos

Parâmetros	Valor
tamanho da população	50
número de iterações ou gerações	300
erro médio de supressão	0.001
taxa de clonagem	10
taxa percentual de supressão	30%
percentual de indivíduos introduzidos	43%

Tabela 2: Melhores resultados da literatura para instâncias de Salhi e Nagy (1999)

Instâncias	Nome	N	Montané & Galvão		Wassan et al.		Zachariadis et al.		Subramanian et al.		Chen & Wu	
			Veículos	Custo	Veículos	Custo	Veículos	Custo	Veículos	Custo	Veículos	Custo
CMT01X	50	3	472	3	468,3	3	469,8	4	466,77*	3	478,59	
CMT01Y	50	3	470	3	458,96*	3	469,8	4	466,77	3	480,78	
CMT02X	75	7	695	6	668,77*	6	684,21	6	684,21	6	688,51	
CMT02Y	75	7	700	6	663,25*	6	684,21	6	684,21	6	679,44	
CMT03X	100	5	721*	4	729,63	5	721,27	5	721,4	5	744,77	
CMT03Y	100	5	719*	4	745,46	5	721,27	5	721,4	5	723,88	
CMT12X	100	6	675	5	644,7*	5	662,22	5	662,22	6	678,46	
CMT12Y	100	6	689	6	659,52*	5	662,22	5	662,22	6	676,23	
CMT11X	120	4	900	4	861,97	4	838,66	4	839,39	4	858,57*	
CMT11Y	120	5	910	4	830,39*	4	837,08	4	841,88	5	859,77	
CMT04X	150	7	880	7	876,5	7	852,46*	7	852,83	7	887	
CMT04Y	150	7	878	7	870,44	7	852,46	7	852,46	7	852,35*	
CMT05X	199	11	1098	9	1044,51	10	1030,55*	10	1030,55	10	1089,22	
CMT05Y	199	10	1083	9	1054,46	10	1030,55*	10	1031,17	10	1084,27	
Average	-	6,14	675	5,5	475,09	5,71	543,09	5,86	717,91	5,93	647,92	

* Melhores resultados da literatura.

A Tabela 2 apresenta os melhores resultados encontrados na literatura para as 14 instâncias propostas por Salhi e Nagy (1999), que serão base para as comparações e análises a seguir. As colunas Nome e N são, respectivamente, o nome da instância e seu número de consumidores. Para cada resultado dos referidos autores, são apresentados o número de veículos e o custo da melhor solução.

Dentre as 14 instâncias de Salhi e Nagy (1999), os melhores resultados encontrados na literatura até então pertencem a: Montané e Galvão (2006) para 2 instâncias de 100 consumidores, Wassan et al. (2007) com 6 instâncias, Zachariadis et al. (2009) com 3 instâncias, Subramanian et al. (2008) para a instância CMT01X e Chen J-F (2006) para as instâncias CMT11X e CMT04Y. Analisando os melhores resultados da literatura, podemos observar que para instâncias de 50 a 120 consumidores, a metaheurística Busca Tabu reativa de Wassan et al. (2007)

Tabela 3: Comparação entre algoritmos imunológicos e abordagens da literatura para as instâncias de Salhi e Nagy (1999)

Instâncias		Melhor da literatura		Opt-aiNet		Clonalg		Copt-aiNet		B-Cell		*Gap
Nome	N	Veículos	Custo	Veículos	Custo	Veículos	Custo	Veículos	Custo	Veículos	Custo	-
CMT01X	50	4	466,77	3	464	3	468	3	467	3	468	-0,59
CMT01Y	50	3	458,96	3	<u>464</u>	3	472	3	471	3	471	1,1
CMT02X	75	6	668,77	7	<u>682</u>	6	691	7	692	7	706	1,98
CMT02Y	75	6	663,25	7	<u>689</u>	6	691	7	720	7	700	3,88
CMT03X	100	5	721	5	714	5	730	5	730	5	740	-0,97
CMT03Y	100	5	719	5	<u>717</u>	5	722	5	723	5	744	-0,28
CMT12X	100	5	644,7	6	<u>669</u>	6	697	6	675	6	696	3,77
CMT12Y	100	6	659,52	6	<u>669</u>	6	703	6	693	6	696	1,44
CMT11X	120	4	858,57	5	868	4	<u>868</u>	4	876	4	924	1,1
CMT11Y	120	4	830,39	4	<u>869</u>	4	967	5	965	4	972	4,65
CMT04X	150	7	852,46	8	<u>856</u>	7	873	7	893	7	920	0,42
CMT04Y	150	7	852,35	7	<u>858</u>	7	863	8	882	7	922	0,66
CMT05X	199	10	1030,55	11	1058	10	<u>1043</u>	10	1147	11	1149	1,21
CMT05Y	199	10	1030,55	10	1063	10	<u>1053</u>	10	1084	11	1131	2,18
Média	-	5,86	746,92	6,21	760	5,86	774,36	6,14	787	6,14	802,79	-

Negrito: Melhores resultados da literatura.

Sublinhado: Melhores resultados entre os algoritmos imunológicos

apresenta os melhores resultados para uma maior número de instâncias.

A Tabela 3 compara os resultados para todos os algoritmos imunológicos. Usando a Equação 1, foi calculado o desvio percentual entre o melhor trabalho da literatura e o custo do melhor resultados obtido neste estudo com os algoritmos imunológicos, e apresentamos na coluna (*Gap).

$$*Gap = 100x \frac{costBestImuno - costBestLiterature}{costBestLiterature}; \quad (1)$$

Visualizando os resultados, nota-se que para instâncias de até 100 consumidores, o algoritmo Opt-AiNET obteve os melhores resultados para as instâncias CMT01X, CMT03X e CMT03Y, com um gap de -0,59%, -0,97% e -0,28%, respectivamente, frente aos melhores resultados da literatura, apontando o método implementado como uma excelente alternativa ao PRVCES para esse número de consumidores. Porém, ao comparar instâncias com um maior número de consumidores o Opt-AiNET não superou os melhores resultados.

Entre os algoritmos imunológicos, outro que obteve bons resultados, próximos aos da literatura, foi o CLONALG, que igualou o número médio de veículos com os melhores resultados já publicados para este conjunto de instâncias. Os algoritmos Copt-aiNET e B-Cell, encontraram rotas com o mesmo número de veículos, porém, o primeiro foi superior em relação ao custo.

4. Conclusão

Devido à grande complexidade computacional do PRVCES, o desenvolvimento e análise de novos procedimentos para encontrar soluções de boa qualidade é extremamente importante. Assim, foi apresentada neste trabalho a adaptação de um conjunto de algoritmos imunológicos para resolver o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas.

Os resultados obtidos mostram os mecanismos bio-inspirados como boas alternativas ao PRVCEs. Uma vez que os algoritmos baseados em população possibilitam uma maior diversificação do espaço de busca, já que são constituídos por mais de uma solução de partida.

Dentre os algoritmos imunológicos, ao se avaliar o número de veículos utilizados, o CLONALG mostrou os melhores resultados, porém quanto ao custo, os melhores resultados foram obtidos pelo algoritmo Opt-AiNet, superando em três instâncias de Salhi e Nagy (1999) o melhor da literatura para o problema.

A qualidade dos resultados obtidos pelos algoritmos imunológicos é altamente dependente dos métodos utilizados nos processos de mutação. Pretende-se, como trabalhos futuros, implementar novas estruturas de vizinhança, otimizar as estruturas de dados usadas e utilizar os algoritmos imunológicos descritos neste artigo para resolver outras variantes do Problema de Roteamento de Veículos. Além disso, será preciso combinar os métodos implementados aqui com estratégias que limitam o espaço de busca dos problemas abordados.

Referências

- Assis, L. P.**, Algoritmos para o problema de roteamento de veículos com coleta e entrega simultâneas, Master's thesis, Universidade Federal de Minas Gerais - UFMG, 2007.
- Caires, L. F. V., Assis, L., de Moraes, V. W. C., e Vivas, A.**, Roteamento de veículos com coleta e entrega simultâneas: Heurística divisão com Árvore geradora mínima de grau restrito, *XLI Simpósio Brasileiro de Pesquisa Operacional*, 2009.
- Castro, L. N. e Zuben, F. J. V.**, ainet: An artificial immune network for data analysis, *Data Mining: A Heuristic Approach*. Abbas, H, Sarker, R and Newton, C (Eds). Idea Group Publishing, 2001.
- Castro, L. N. e Timmis, J.**, An artificial immune network for multimodal function optimisation, *Proc. Of IEEE World Congress on Evolutionary Computation*, 2002.
- Chen J-F, W. T.-H.**, Vehicle routing problem with simultaneous deliveries and pickups, *J Oper Res Soc*, 57:579–587, 2006.
- Dantzig, G. B. e Ramser, J. H.**, The truck dispatching problem, *Management Science*, 1959.
- Dasgupta, D.**, Artificial immune systems and their applications, *Springer-Verlag*, 1998.
- de Castro, L. N. e Zuben, F. J. V.**, The clonal selection algorithm with engineering applications, *Workshop on Artificial Immune Systems and Their Applications*, páginas 36–37, 2000.
- Dell'Amico, M., Righini, G., e Salani, M.**, A branch-and-price algorithm for the vehicle routing problem with simultaneous delivery and collection, *Transportation Science*, 40:235–247, 2005.
- Dethloff, J.**, Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up, *OR Spectrum*, 23(1):79–96, 2001.
- Gajpal, Y. e Abad, P.**, An ant colony system (acs) for vehicle routing problem with simultaneous delivery and pickup, *Computers and Operations Research*, 2009.

- Gökçe, E. I.**, A revised ant colony system approach to vehicle routing problems, Master's thesis, Graduate School of Engineering and Natural Sciences, Sabanci University., 2004.
- Kelsey, J. e Timmis, J.**, Immune inspired somatic contiguous hypermutation for function optimisation, *GECCO*, páginas 207–218, 2003.
- Min, H.**, The multiple vehicle routing problem with simultaneous delivery and pickup points, *Transportation Research-A*, 23A(5):377–386, 1989.
- Montané, F. A. T. e Galvão, R. D.**, A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service, *Computers and Operations Research*, 33(3): 595–619, 2006.
- Morais, V. W. C., Assis, L., Caires, L. F. V., e Vivas, A.**, Algoritmo híbrido grasp/vnd/ils aplicado ao problema de roteamento de veículos com coleta e entrega simultaneas, *Simpósio de Pesquisa Operacional e Logística da Marinha*, 2009.
- Oliveira, H. C. B. e Vasconcelos, G. C.**, A hybrid search method for the vehicle routing problem with timewindows, *Springer Science Business Media, LLC*, 2008.
- Salhi, S. e Nagy, G.**, A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling., *Journal of the Operational Research Society*, 50:1034–42., 1999.
- Souza, J. S., de C. T. Gomes, L., Bezerra, G. B., de Castro Silva, L. N., e Zuben, F. J. V.**, An immune-evolutionary algorithm for multiple rearrangements of gene expression data, *Genetic Programming and Evolvable Machines*, 5:157–179, 2004, ISSN 1389-2576, doi: <http://dx.doi.org/10.1023/B:GENP.0000023686.59617.57>.
- Subramanian, A., Ochi, L. S., e dos Anjos Formiga Cabral, L.**, An efficient ils heuristic for the vehicle routing problem with simultaneous pickup and delivery, *Technical Report – RT 07/08*, 2008, ISSN optional.
- Vural, A. V.**, A ga based meta-heuristic for capacitated vehicle routing problem with simultaneous pick-up and delivery, Master's thesis, Faculty of Management, Istanbul Technical University, 2003.
- Wassan, N. A., Wassan, A. H., e Nagy, G.**, A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries, *Journal of Combinatorial Optimization*, 15(4):368–386, 2007.
- Zachariadis, E. E., Tarantilis, C. D., e Kiranoudis, C. T.**, A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service, *Expert Systems with Applications*, 36(2):1070–1081, 2009.
- Zhang, T., xin Tian, W., jie Zhang, Y., e xin Liu, S.**, Improved ant colony system for vrpspd with maximum distance constraint, *Systems Engineering - Theory and Practice*, 28(1):132–140, 2008.
- Çatay, B.**, An ant based algorithm for the vehicle routing problem with simultaneous delivery and pick-up, *INFORMS Meeting*, 2006.