COMPARAÇÃO ENTRE DUAS ABORDAGENS BIO-INSPIRADAS APLICADAS AO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEAS

André Luiz Maravilha Silva
Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)
Rua da Glória, 187 – 39.100-000 – Diamantina – MG – Brasil
msilva.andreluiz@gmail.com

Vinícius Wellington Coelho Morais
Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM)
Rua da Glória, 187 – 39.100-000 – Diamantina – MG – Brasil
vwcmorais@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

Neste artigo é feita uma comparação entre duas abordagens bio-inspiradas para resolução do Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas, uma delas é baseada no algoritmo imunológico e outra no algoritmo colônia de formigas. Este problema é bastante estudado devido uma grande variedade de aplicações práticas e também pela sua complexidade. Os algoritmos bio-inspirados vem se mostrando uma boa estratégia para resolução de problemas complexos. Ambos são descritos, testados e analisados mostrando suas características fundamentais. Nos testes foram utilizadas 14 instâncias encontradas na literatura do problema, das quais o algoritmo imunológico se mostrou superior ao de otimização por colônia de formigas em 9 delas.

PALAVRAS CHAVE. Logística & Transportes, Metaheuristicas, Otimização Combinatória. Área principal. Logística & Transportes.

ABSTRACT

In this paper a comparison is made between two bio-inspired approaches for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery, one of them is based on immune algorithm and another in the ant colony algorithm. This problem is widely studied because of a variety of practical applications and also for its complexity. The bio-inspired algorithms has proved to be a good strategy for solving complex problems. Both are described, tested and analyzed showing its fundamental characteristics. Tests were used in 14 instances found in the literature of the problem, which the immune algorithm was better than the optimization ant colony in 9 of them.

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

1. Introdução

Com o aumento da demanda por produtos industrializados, as organizações procuram melhorar seus processos de produção, buscando mecanismos para aumentar a eficiência desses processos. Estas etapas vão desde a concepção da matéria-prima até a entrega de seus produtos aos consumidores finais.

Durante todos estes processos, o fluxo de materiais é constante e impacta diretamente no custo do produto final. Segundo Alvarenga (2005), no Brasil, devido a enorme malha rodoviária e a extensão territorial, cerca de 10% a 15% do valor final dos produtos pagos pelo consumidor final correspondem ao custo de seu transporte.

O estudo do fluxo destes materiais é um tradicional problema de transporte, o qual está inserido no contexto da logística. Uma especialização da logística é a logística reversa onde além de se planejar todo a atividade de entrega de mercadorias deve-se definir o caminho inverso, onde os insumos do produtos recebidos e processados no cliente devem retornar à central de distribuição.

Um clássico problema de logística reversa é o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas (PRVCES), proposto por Min (1989), que vem sendo tratado por diversas empresas de distribuição, como distribuidoras de bebidas, pneus e de produtos químicos. Neste último, a coleta dos produtos após o uso é de extrema importância, uma vez que os resíduos produzidos devem ser depositados em locais apropriados para que não prejudiquem o meio ambiente, obedecendo as leis ambientais.

O PRVCES é um problema de otimização combinatória classificado como NP-Difícil. Portanto não é conhecido nenhum método que encontre a solução ótima em tempo polinomial. Devido a isso existem diversos estudos com o uso de heurísticas e metaheurísticas para o tratamento deste problema.

O Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas pode ser definido como: dado um conjunto de N consumidores ou clientes, cada um com uma demanda de entrega e uma demanda de coleta, uma frota de K veículos homogêneos de capacidade Q, além de um depósito de onde os veículos deverão partir com os produtos a serem entregues e para onde devem retornar com os resíduos coletados após atenderem os consumidores. As operações de entrega e coleta devem ser realizadas de forma simultânea e cada consumidor deve ser visitado uma única vez por um único veículo, que não pode ter sua capacidade extrapolada em momento algum do trajeto. O objetivo do PRVCES é reduzir os esforços de coleta e entrega, determinando um conjunto de rotas que minimizem o custo total de transporte e o número de veículos.

Formulações matemáticas para o PRVCES são apresentadas nos trabalhos de Dell'Amico et al. (2005) e Subramanian et al. (2008), este último ainda apresenta um algoritmo ILS (*Iterated Local Search*) para resolução do mesmo.

Em Montané e Galvão (2006), Chen J-F (2006), Wassan et al. (2007) e Bianchessi e Righini (2007) são apresentados algoritmos baseados na metaheurística Busca Tabu além de um conjunto de heurísticas para resolução do PRVCES. Assis (2007) propõe três heurísticas construtivas baseadas no método Dividir e Rotear para o problema.

Algoritmos hibridos baseados nas metaheurísticas VNS(*Variable Neighborhood Se-*

arch, O VND (*Variable Neighborhood Descent*), ILS e GRASP (*Greedy Random Adaptive Search Procedure*) para resolução do PRVCES são apresentados em Freitas e Montané (2008) e Morais et al. (2009).

Neste trabalho são apresentadas duas abordagens bio-inspiradas para resolução do PRVCES. A primeira delas é uma versão do algoritmo de Otimização por Colônia de Formigas e a segunda é uma adaptação de um Algoritmo Imunológico.

O trabalho está organizado da seguinte forma: nas Seções 2 e 3 são apresentados os conceitos e as características da Otimização por Colônia de Formigas e dos Sistemas Imunológicos Artificiais, respectivamente. Por fim, são apresentados o resultados obtidos na Seção 4 e conclusão na Seção 5.

2. Otimização por Colônia de Formigas

A Otimização por Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*) se baseia no comportamento das formigas na procura por alimentos. Quando o alimento é encontrado, mesmo que por diferentes caminhos, observa-se que com o passar do tempo, elas tendem a convergir para a menor rota entre o ninho e o alimento.

Este fenômeno é possível pelo fato das formigas se comunicarem, de forma indireta, através de uma substância denominada feromônio, que elas depositam ao longo do trajeto em que percorrem. Como as formigas que trafegaram pelo menor caminho realizam o percurso em menor tempo, a quantidade de feromônio presente neste caminho acaba sendo maior. O feromônio evapora com o passar do tempo, então o caminho mais curto terá uma quantidade desta substância tão alta, que quase todas as formigas seguirão apenas por ele.

Diversos trabalhos sobre esta abordagem para diversas aplicações estão disponíveis na literatura. Dentre eles, Dorigo (1992) propôs o algoritmo, Gökçe (2004), Gajpal e Abad (2009) e Zhang et al. (2008) descrevem e aplicam o algoritmo ao PRVCES, o último com restrição de carga máxima.

O método de Otimização por Colônia de Formigas é capaz de gerar soluções para problemas de otimização onde uma abordagem exata se torna inviável. Ele consiste em gerar soluções de forma construtiva, selecionando os elementos que irão fazer parte da solução de forma probabilística com base na intensidade de feromônio e na atratividade destes.

Uma formiga é um agente computacional simples, que constrói uma solução viável para o problema. Cada formiga parte de uma solução vazia e adiciona componentes até que a solução esteja completa. O Algoritmo 1 apresenta o funcionamento do método Otimização por Colônia de Formigas.

No passo 1 é gerada uma solução através da heurística Rotear e Dividir (Assis (2007)) para que a intensidade das trilhas de feromônio e o número de formigas para a primeira iteração pudessem ser inicializados. A intensidade das trilhas de feromônio são inicializadas com o inverso do custo da solução gerada para todas as arestas e o número de formigas é definido como duas vezes o número de rotas utilizadas por esta solução. Os passos 3–6 são executados até que uma condição de término seja alcançada, que pode ser um número máximo de iterações ou até o algoritmo estabilizar.

No passo 3 cada formiga cria uma solução candidata. O número de formigas utilizadas

Algoritmo 1 Otimização por Colônia de Formigas aplicado ao PRVCES

1: Inicializar a intensidade das trilhas de feromônio e definir o número de formigas para a primeira iteração;
2: **enquanto** condição de término não é alcançada **faça**
3: Gerar uma solução para cada formiga;
4: Melhorar cada solução gerada;
5: Selecionar a melhor solução;
6: Atualizar intensidade de feromônio nas trilhas;
7: **fim enquanto**
8: Retornar a melhor solução;

em todas as iterações seguintes foi igual a duas vezes o número de rotas necessárias na melhor solução encontrada até o momento de execução da iteração. Na construção das soluções, cada formiga é guiada através atratividade dos componentes e a escolha de cada componente é feita de acordo com uma probabilidade P_{ij}. O cálculo desta probabilidade é apresentado em detalhes na Seção 2.1.

Ao término da construção de uma solução candidata, ela é melhorada com o uso de buscas locais (passo 4). No passo 5, a melhor solução encontrada na iteração atual é comparada com a melhor solução até o momento. Se a melhor solução desta iteração for melhor ela é armazenada como a melhor solução corrente.

Após todas as formigas criarem suas soluções, ocorre a atualização das trilhas de feromônio, explicada em maior detalhe na Seção 2.2. No último passo, é retornada a melhor solução gerada.

2.1. Determinação da Probabilidade de Escolha dos Componentes da Solução

A atratividade ξ_{ij} da aresta (v_i, v_j) é calculada através do produto entre dois valores: a intensidade de feromônio (τ_{ij}) presente na aresta e uma informação específica do problema (η_{ij}). No caso do PRVCES, o valor de η_{ij} é dada pela Equação 1 que indica o ganho em se visitar o consumidor v_j à partir do consumidor v_i ao invés de criar uma nova rota onde o consumidor v_j é o primeiro a ser visitado. Aqui, $c(0,i)$, $c(0,j)$ e $c(i,j)$ são o custo das arestas (v_0, v_i), (v_0, v_j) e (v_i, v_j), respectivamente.

$$\eta_{ij} = c(0,i) + c(0,j) - c(i,j) \tag{1}$$

A probabilidade P_{ij} da aresta (v_i, v_j) ser escolhida é apresentada na Equação 2, onde Ω_m é o conjunto de vértices ainda não visitados que podem ser alcançados à partir do vértice v_i.

$$P_{ij} = \begin{cases} \frac{\xi_{ij}}{\sum_{l \in \Omega_m} \xi_{il}} & \text{se } j \in \Omega_m, \\ 0 & \text{caso contrário.} \end{cases} \tag{2}$$

2.2. Atualização das Trilhas de Feromônio

A atualização das trilhas de feromônio ocorre após as formigas terminarem de criar suas soluções. Esta atualização é feita com base nas γ melhores soluções geradas na iteração

atual. A nova intensidade de feromônio na aresta (v_i, v_j) é dada pela Equação 3 apresentada a seguir:

$$\tau_{ij}^{novo} = \rho \times \tau_{ij}^{antigo} + \sum_{\mu=1}^{\gamma} \Delta\tau_{ij}^{\mu}, \qquad i = 1,\ldots,n,\ j = 1,\ldots,n,\ i \neq j. \tag{3}$$

Aqui, a variável ρ define o nível de persistência do feromônio presente na iteração anterior, onde seu valor varia no intervalo $[0,1]$. O próximo termo presente na equação é quem determina a quantidade de feromônio que será depositado na aresta, e é apresentado na Equação 4.

$$\Delta\tau_{ij}^{\mu} = \begin{cases} \frac{1}{L^{\mu}} & \text{se a aresta } (v_i, v_j) \in \mu\text{-ésima solução,} \\ 0 & \text{caso constrário.} \end{cases} \tag{4}$$

Onde L^{μ} representa o custo total da μ-ésima melhor solução da iteração.

3. Sistemas Imunológicos Artificiais

O sistema imunológico biológico protege o organismo contra ataques de microorganismos externos, chamados patógenos. Este sistema é composto por dois sub-sistemas, o sistema imunológico inato e o sistema imunológico adaptativo. O primeiro defende o organismo de forma semelhante para diferentes patógenos, enquanto o adaptativo combate cada patógeno de forma específica, evoluindo com o tempo, garantindo a imunidade do organismo de forma duradoura.

A principal característica do sistema imunológico adaptativo é a presença das células B. São elas quem identificam e combatem os patógenos do organismo. O reconhecimento é feito através dos antígenos, que são moléculas presentes na superfície do patógeno.

Os Sistemas Imunológicos Artificiais (SIA) (Dasgupta (1998)) são mecanismos computacionais inspirados no sistema imunológico biológico, principalmente nas características do sistema imunológico adaptativo, que são maturação da afinidade, seleção clonal e teoria da rede imunológica. Na literatura existem diversos modelos de SIA dentre os quais podemos citar o Opt-AiNet de Castro e Timmis (2002) e o Copt-AiNet de Souza et al. (2004).

Eles têm como objetivo resolver problemas do mundo real de alta complexidade computacional. Este tipo de abordagem possui aplicação em diversas áreas como otimização, reconhecimento de padrões, agrupamento de dados, robótica, mineração de dados e segurança computacional. As características mais importantes são a manutenção da diversidade do conjunto de soluções candidatas, a grande exploração do espaço de soluções e parâmetros adaptativos.

3.1. Algoritmo CLONALG

O CLONALG (*Clonal Selection Algorithm*) proposto por Castro e Zuben (2000) é baseado no sistema imunológico natural. Este método considera que apenas as células mais adaptadas para o reconhecimento do antígeno são selecionadas para proliferar, gerando clones. Os clones são submetidos aos processos de mutação e maturação da afinidade.

Ele foi proposto inicialmente para resolução de problemas de reconhecimento de padrões e depois foi adaptado para resolução de problemas de otimização. No trabalho original, o CLONALG se mostrou uma técnica capaz de resolver tarefas complexas como aprendizagem de máquina, reconhecimento de padrões e otimização multimodal.

Este algoritmo apresenta algumas características importantes para a geração de novas soluções como:

- retenção dos indivíduos mais adaptados;
- manutenção da diversidade;
- taxa de mutação é inversamente proporcional à qualidade do clone com relação ao indivíduo mais adaptado presente na população.

O Algoritmo 2 descreve o funcionamento do algoritmo CLONALG para o PRVCES, para isso é assumido a seguinte notação:

- *População:* uma população é um conjunto de indivíduos.
- *Célula:* é indivíduo que representa um anticorpo e corresponde a uma solução candidata válida ao problema. Neste problema, uma solução é representada por um vetor de números naturais, onde cada número presente no vetor representa um consumidor. Os consumidores são numerados de 1 a n e são divididos em r segmentos, sendo r o número de rotas e/ou veículos utilizados pela solução. O elemento utilizado para separar em segmentos é o 0 (zero), simbolizando o depósito. Para um melhor entendimento, considere o seguinte indivíduo representando uma solução para uma instância de 18 clientes: $[0-1-14-17-9-2-0-13-12-4-19-5-15-0-7-16-11-6-8-3-18-10-0]$. Este é dividido em três segmentos, cada um representando uma rota que deverá ser atendida por um único veículo. Os segmentos são $[0-1-14-17-9-2-0]$, $[0-13-12-4-19-5-15-0]$ e $[0-7-16-11-6-8-3-18-10-0]$.
- *Clone:* um clone é uma cópia idêntica de um indivíduo da população.
- *Antígeno:* seria a solução ótima para o problema, mas como não se tem o conhecimento desta solução, consideramos como antígeno a melhor solução presente na população.
- *Fitness:* é o valor da avaliação de um indivíduo a partir da função objetivo, que no PRVCES, esta função é o somatório dos pesos associados às arestas que compõem a solução.
- *Afinidade:* representa a semelhança entre dois indivíduos da população, verificada pelo número de arestas coincidentes entre estes dois indivíduos. Neste trabalho, a afinidade foi obtida comparando cada indivíduo com o melhor presente na população.

No passo inicial, uma população P é gerada de forma totalmente aleatória, sendo que P é formada pela união do conjunto M de células de memória (melhores soluções encontradas) com o conjunto P_r (demais soluções) de indivíduos restantes, onde a cardinalidade do conjunto M é bem menor que a cardinalidade do conjunto P_r. Cada indivíduo presente na população P representa uma solução válida para o problema.

Para cada geração são executados os passos 3 ao 9. No passo 3 os N melhores indivíduos presentes na população P são selecionados para que, a partir deles, um conjunto composto por clones, denominado população C, seja formado no passo 4. Esses clones darão origem aos possíveis novos indivíduos da população P. Cada clone passa por um processo de mutação no passo 5, onde são removidos n vértices de uma rota e estes são reinseridos em outras rotas.

Algoritmo 2 CLONALG aplicado ao PRVCES

1: Gerar População Inicial P;
2: **enquanto** $contador \leq numMaxGeracoes$ **faça**
3: Selecionar os N melhores indivíduos em P;
4: Gerar N_c clones dos N melhores indivíduos gerando uma população temporária C de clones;
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 recomporem as células de memória M;
8: Selecionar alguns indivíduos restantes em C para recomporem P_r;
9: Substituir D indivíduos com menor afinidade em P_r por novos indivíduos gerados de forma aleatória;
10: **fim enquanto**

No passo 6 é realizado a maturação da afinidade, que realiza a evolução de um indivíduo resultando em uma melhora da afinidade entre o anticorpo e o antígeno. Esta etapa foi feita com o uso de buscas locais sobre o indivíduo corrente, explorando ao máximo a vizinhança onde se encontra, a fim de encontrar um ótimo local para este sub-espaço. Para este processo foi utilizado a metaheurística *Variable Neighborhood Descent (VND)* proposta por Mladenovic e Hansen (1997) implementado com os operadores de vizinhança Realocação, Eliminação de Rotas e 2-Opt (Assis (2007)).

No passo 7, os melhores indivíduos presentes na população C, após as etapas 5 e 6, são selecionados para recomporem o conjunto M de células de memória. Esta recomposição do conjunto M considera também as células atualmente presentes nele, assim, sempre preservando os melhores indivíduos. Alguns indivíduos que não foram selecionados para recomporem as células de memória podem ser selecionados para substituírem indivíduos pouco adaptados presentes no conjunto P_r no passo 8.

No passo 9, os D indivíduos menos adaptados do conjunto P_r são substituídos por novos indivíduos gerados de forma aleatória a fim de manter a diversidade da população, explorando ainda mais espaço de soluções.

4. Testes Computacionais

Antes de realizar os testes com as instâncias, foram necessários alguns experimentos para que os parâmetros pudessem ser ajustados. Os parâmetros utilizados nos algoritmos ACO e CLONALG são apresentados nas Tabelas 1 e 2, respectivamente.

Parâmetro	Valor
γ	10
ρ	0.95
Número máximo de iterações	100

Tabela 1: Parâmetros para o ACO

Parâmetro	Valor
Tamanho da população	50
Número de gerações	300
Indivíduos selecionados para clonagem	30
Clones por indivíduo	10
Taxa de supressão	30%
Tamanho mínimo da população	12
Número dos K melhores indivíduos	4

Tabela 2: Parâmetros para o CLONALG

Os algoritmos foram implementados na linguagem Java (jdk 1.6), no ambiente de desenvolvimento Eclipse. Os testes foram realizados em uma máquina com processador Intel Core 2 Quad Q8200, com 3 GB de memória RAM e sistema operacional GNU/Linux distribuição Ubuntu 9.04 de 64 bits. Apesar desta máquina contar com 4 núcleos de processamento, não foi utilizado o recurso de programação paralela.

Para testar a eficiência dos algoritmos foram utilizadas 14 instâncias propostas por Salhi e Nagy (1999) com o número de clientes variando entre 50 a 199. Os resultados obtidos estão presentes na Tabela 3.

Instância	ACO		CLONALG		Subramanian et al. (2008)	
	Veículos	Custo	Veículos	Custo	Veículos	Custo
CMT01X	3	470	3	468	4	**466,77**
CMT01Y	4	**464**	3	472	4	466,77
CMT02X	7	707	6	691	6	**684,21**
CMT02Y	7	707	6	691	6	**684,21**
CMT03X	5	728	5	730	5	**721,4**
CMT03Y	5	729	5	722	5	**721,4**
CMT12X	6	687	6	697	5	**662,22**
CMT12Y	6	683	6	703	5	**662,22**
CMT11X	4	900	4	868	4	**839,39**
CMT11Y	4	896	4	967	4	**841,88**
CMT04X	8	900	7	873	7	**852,83**
CMT04Y	8	891	7	863	7	**852,46**
CMT05X	11	1108	10	1043	10	**1030,55**
CMT05Y	10	1100	10	1053	10	**1031,17**
Média	6	783,57	6	774,36	6	**751,25**

Tabela 3: Comparação entre o ACO e CLONALG

Ao se analisar as soluções obtidas, o algoritmo CLONALG se mostrou superior ao Otimização por Colônia de Formigas em 9 das 14 instâncias testadas, em relação ao custo. O CLONALG obteve uma média de custo de 774,36 contra 783,57 do ACO. Com relação ao número de veículos utilizados, novamente o CLONALG supera o Colônia de Formigas, sendo que em 6 das 14 instâncias o CLONALG utilizou um veículo a menos.

Analisando os resultados obtidos pelos algoritmos propostos neste trabalho, com relação aos encontrados em Subramanian et al. (2008), pode-se notar que em apenas uma das instâncias testadas, se conseguiu um resultado melhor (instância CMT01Y com o algoritmo ACO), mas os outros resultados, mesmo não superando os encotrados na literatura, se mostraram próximos a eles.

O CLONALG mantém um grande conjunto de soluções durante toda sua execução, enquanto o ACO mantém apenas uma pequena quantidade de soluções armazenadas. Esta pequena quantidade de soluções mantidas pelo ACO é necessária para o cálculo das novas taxas de feromônio. Outro aspecto importante é que, a cada iteração, o CLONALG realiza os processos de mutação e maturação da afinidade para cada clone dos indivíduos selecionados para evolução.

No algoritmo ACO o número de soluções geradas a cada iteração é definido dinamicamente, durante a execução do algoritmo. Quanto menor o número de veículos necessários para atenderem os consumidores, de acordo com a melhor solução, menor será o número de soluções criadas. Então, à medida em que o algoritmo está sendo executado, as iterações restantes serão executadas de forma mais rápida. Ao contrário do CLONALG, que gera um número fixo de soluções desde a primeira até a última iteração.

5. Conclusão

Algoritmos baseados em processos naturais, por evoluírem geralmente a partir de um conjunto de soluções, ao contrário de heurísticas mais comuns, possibilitam uma maior exploração do espaço de soluções, porém, com um maior custo computacional. Apesar da necessidade de um maior tempo de processamento, ele não é tão alto ao ponto de tornar proibitiva sua utilização.

Os algoritmos bio-inspirados são probabilísticos, possuindo algumas características de aleatoriedade, fazendo que com a cada execução sobre uma mesma instância possamos obter resultados diferentes, mas de mesma qualidade.

O Algoritmo CLONALG apresentou resultados melhores ao de Otimização por Colônia de Formigas, se mostrando uma técnica promissora para resolução de problemas complexos, como problemas de otimização combinatória, que é o caso do PRVCES.

Com a popularização de computadores com múltiplos núcleos de processamento, estudos futuros podem ser realizados para verificação da eficiência destas abordagens com o uso da programação paralela. Isto tornaria possível que parâmetros referentes à quantidade de soluções geradas a cada iteração e o número de iterações fossem configurados com valores maiores, sem um aumento considerável no tempo de execução, ou ainda, apenas reduzir o tempo atualmente necessário para a execução destes algoritmos.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo do Estado de Minas Gerais (FAPEMIG) pelo apoio financeiro que possibilitou a elaboração deste trabalho.

Referências

Alvarenga, G. B., *Um Algoritmo Hibrido para o Problemas de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo*, PhD thesis, Universidade Federal de Minas Gerais, 2005.

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.

Bianchessi, N. e Righini, G., Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, *Computers and Operations Research*, 34(2):578–594, 2007.

Castro, L. N. e Timmis, J., An artificial immune network for multimodal function optimisation, *Proc. Of IEEE World Congress on Evolutionary Computation*, 2002.

Castro, L. N. e Zuben, F. J. V., An evolutionary immune network for data clustering, *Brazilian Symposium on Artificial Neural Network*, 2000.

Chen J-F, W. T.-H., Vehicle routing problem with simultaneous deliveries and pickups, *J Oper Res Soc*, 57:579–587, 2006.

Dasgupta, D., Artificial immune systems and their applications, *Springer-Verlag*, 1998.

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.

Dorigo, M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, 1992.

Freitas, L. M. B. e Montané, F. A. T., Meta-heurísticas vns-vnd e grasp-vnd para problemas de roteamento de veículos com coleta e entrega simultânea, *XI Simpósio de Pesquisa Operacional e Logística da Marinha, Rio de Janeiro*, 2008.

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.

Min, H., The multiple vehicle routing problem with simultaneous delivery and pickup points, *Transportation Research-A*, 23A(5):377–386, 1989.

Mladenovic, N. e Hansen, P., Variable neighborhood search, *Computers and Operations Research*, 24:1097–1100, 1997.

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 simultâneas, *Simpósio de Pesquisa Operacional e Logística da Marinha*, 2009.

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.

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.

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.