

Análise de Vizinhança de Nós na Otimização por Colônia de Formigas para o Problema de Roteamento de Veículos Capacitados

Rodrigo Silva Borges

Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675 - Nova Gameleira, Belo Horizonte - MG, 30510-000
rodrigo.sil.borges@gmail.com

Flávio Vinícius Cruzeiro Martins

Centro Federal de Educação Tecnológica de Minas Gerais
Av. Amazonas, 7675 - Nova Gameleira, Belo Horizonte - MG, 30510-000
prof.flaviocruzeiro@gmail.com

RESUMO

Este trabalho implementa a meta-heurística Otimização por Colônia de Formigas no Problema de Roteamento de Veículos Capacitados, aplicando variações já propostas na literatura e propondo uma nova variação através do estudo de vizinhança de cada nó do problema. Este estudo melhora estatisticamente o resultado encontrado pelo algoritmo original na maioria das instâncias trabalhadas.

PALAVRAS CHAVE. Problema de Roteamento de Veículos Capacitados, Otimização por Colônia de Formigas, Heurística.

Logística e Transportes, Metaheurísticas, Otimização Combinatória

ABSTRACT

This research implements the Ant Colony Optimization heuristic to solve the Capacitated Vehicle Routing Problem, implementing already proposed modifications and proposing a new one analysing the neighbourhood of each node of the problem. This analysis statistically improves results found by the original algorithm in most of the benchmarks used to test it.

KEYWORDS. Capacitated Vehicle Routing Problem. Ant Colony Optimization. Heuristic.

Logistics and Transport, Metaheuristics, Combinatorial Optimization

1. Introdução

Problemas de logística tem enorme aplicação no mercado atual, e a resolução de problemas de roteamento tem se tornado crítica para este meio. Desta forma, diferentes algoritmos, exatos ou inexatos, são propostos para este fim. Um dos algoritmos inexatos heurísticos propostos foi a Otimização por Colônia de Formigas, que trabalha a busca por rotas de forma análoga à busca das formigas pelo caminho entre o ninho e uma fonte de comida.

Abordagens heurísticas clássicas costumam abranger uma enorme gama de problemas, mas não necessariamente os resolvem da melhor maneira. É possível, então, trabalhar com algoritmos híbridos, que misturam técnicas de diferentes heurísticas em uma só, como já fizeram [Ning e Guo, 2012] e [Bin e Baozhen, 2009] ao trabalharem a colônia de formigas com operadores genéticos. [Bell e McMullen, 2004], por sua vez, propuseram variações à colônia de formigas implementando múltiplas colônias que percorriam o espaço de busca independentemente.

Este trabalho propõe e desenvolve, então, uma variação para a abordagem clássica da colônia de formigas, visando melhorar as soluções para o problema de roteamento de veículos ao adicionar uma etapa de inteligência às formigas na decisão do próximo vértice a ser percorrido.

2. Problema de Roteamento de Veículos Capacitados

O Problema de Roteamento de Veículos Capacitados (Capacitated Vehicle Routing Problem - CVRP) consiste em um cenário de distribuição/captação de recursos a partir de um depósito para diferentes clientes. Segundo [Dantzig e Ramser, 1959], a formalização do CVRP pode ser descrita como:

1. Dado um conjunto de n “clientes” $P_i (i = 1, 2, \dots, n)$ para os quais entregas são feitas a partir a partir de um ponto P_0 chamado “depósito”;
2. Uma matriz de distâncias $[D] = [d_{ij}]$ dada que especifica a distância $d_{ij} = d_{ji}$ entre cada par de pontos $(i, j) = 0, 1, 2, \dots, n$;
3. Um vetor de demanda $(Q) = (q_i)$ dado que especifica a quantidade q_i a ser entregue em cada ponto $P_i (i = 1, 2, \dots, n)$;
4. A capacidade do veículo é C , onde $C > \text{argmax}(q_i)$.
5. Se $x_{ij} = x_{ji} = 1$ é interpretado como os pontos P_i e P_j estando ligados na rota criada e se $x_{ij} = x_{ji} = 0$ é interpretado como estes pontos não estando ligados, pode-se tirar a condição

$$\sum_{j=0}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n) \quad (1)$$

já que todo ponto P_i está ligado a P_0 ou a no máximo um outro ponto P_j . Além disso, por definição, $x_{ii} = 0, \forall i = 0, 1, \dots, n$.

6. O problema é encontrar os valores de x_{ij} que fazem a distância total

$$D = \sum_{i,j=0}^n d_{ij} \cdot x_{ij} \quad (2)$$

mínima sob as condições 2 a 5.

3. Otimização por Colônia de Formigas

O algoritmo Otimização por Colônia de Formigas (Ant Colony Optimization - ACO) simula o comportamento de formigas reais na busca por um caminho entre seu ninho e a fonte de comida e na exploração até o encontro do menor caminho entre os dois, proposto por [Dorigo e Colomni, 1996]. Esta evolução da rota é feita através do depósito de feromônio feito pelas formigas que encontram a comida e voltam para o ninho. Outras formigas têm mais tendência a seguir uma trilha de feromônio do que uma trilha seca, encontrando a comida e retornando ao ninho depositando

seu próprio feromônio, o que reforça o caminho. Este feromônio, no entanto, evapora, permitindo a busca de diferentes caminhos por parte das formigas. Como todo o feromônio evapora a uma mesma taxa, conclui-se que caminhos mais curtos tem mais consistência de feromônio, sobressaindo-se às opções mais longas.

Quando implementado computacionalmente, o ACO tenta reproduzir esta interação da formiga com seu espaço de busca para otimização. Ainda que diferentes problemas possam ser adaptados para esta abordagem, a resolução do CVRP faz analogia direta à inspiração real, dado que a busca realizada é pelo menor caminho a ser percorrido.

Este trabalho divide a abordagem do ACO em três módulos:

- Escolha do arco a ser seguido pela formiga;
- Evaporação do feromônio; e
- Depósito de feromônio.

A seguir, são apresentadas as adaptações adotadas por este trabalho e a adaptação desenvolvida por ele, descrita em 3.1.1.

3.1. Escolha do Arco a Ser Seguido

Este módulo do ACO define o processo de tomada de decisão de uma formiga k , que se encontra em um vértice r qualquer, quanto ao próximo passo a ser dado no conjunto de vértices disponíveis. Pelo modelo proposto por [Dorigo e Gambardella, 1997], na iminência de cada passo é sorteado um valor aleatório $q \in (0, 1)$. O próximo vértice visitado pela formiga é definido por:

$$s = \begin{cases} \operatorname{argmax}_{u \in A_k} \{[\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta\} & \text{se } q \leq q_0 \\ S & \text{caso contrário} \end{cases} \quad (3)$$

Onde:

- q_0 : Fator inerente da formiga, tal que $q_0 \in (0, 1)$;
- s : Próximo vértice visitado pela formiga; e
- A_k : Conjunto de vértices os quais a formiga k pode visitar, ou seja, que ela ainda não tenha visitado e cuja demanda não extrapole a capacidade disponível no veículo.

E S é uma decisão tomada probabilisticamente, tal que a probabilidade de um arco (r, s) ser escolhido é dada por:

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha \cdot [\eta(r, s)]^\beta}{\sum_{\forall u \in A_k} [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta} & \text{se } s \in A_k \\ 0 & \text{caso contrário} \end{cases} \quad (4)$$

Onde:

- $p_k(r, s)$: Probabilidade da formiga k percorrer o arco (r, s) ;
- $\tau(r, s)$: Quantidade de feromônio no arco (r, s) ;
- $\eta(r, s)$: O inverso do comprimento do arco (r, s) , ou seja, $1/d_{r,s}$;
- α : Fator de valorização para a quantidade do feromônio no arco; e
- β : Fator de valorização do comprimento do arco.

Caso não haja vértice u que pertença a A_k , a formiga segue para o depósito.

3.1.1. Análise de Vizinhaça

Durante a implementação do algoritmo clássico, a avaliação dos resultados e a comparação com resultados da literatura, observou-se um desenho comum nas soluções: as rotas traçadas pelas formigas assemelham-se a “pétalas”. Supôs-se, então, que influenciar a formação desta geometria nas soluções poderia melhorá-las.

Para tanto, neste trabalho foi elaborado um conceito de vizinhaça para cada vértice do espaço de busca que permite direcionar a formiga para mais longe ou mais perto do depósito a partir do vértice em que se encontra no momento. É realizado um pré-processamento no conjunto de clientes, definindo para cada cliente r uma vizinhaça interna e externa definidas por:

$$\begin{aligned} VI(r) &= \{i \in P \mid d_{P_0,i} < d_{P_0,r}, [\angle(P_0, r) - \theta] < [\angle(P_0, i)] < [\angle(P_0, r) + \theta]\} \\ VE(r) &= \{e \in P \mid d_{P_0,e} > d_{P_0,r}, [\angle(P_0, r) - \theta] < [\angle(P_0, e)] < [\angle(P_0, r) + \theta]\} \end{aligned} \quad (5)$$

Onde:

- $VI(r)$: Conjunto de vértices pertencentes à vizinhaça interna de r ;
- $VE(r)$: Conjunto de vértices pertencentes à vizinhaça externa de r ;
- P_0 : Vértice depósito do problema;
- $\angle(P_0, r)$: Ângulo traçado entre os vértices P_0 e r no plano; e
- θ : Ângulo que define a abertura do cone que limita as vizinhaças.

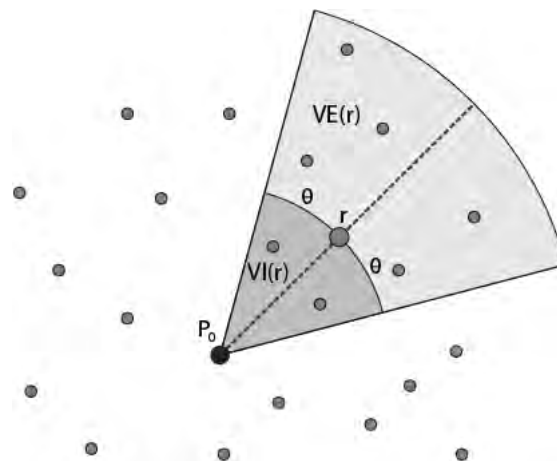


Figura 1: Representação Gráfica de $VI(r)$ e $VE(r)$

O escopo do processo de decisão se mantém, porém agora com fatores influenciadores trazidos por cada uma das vizinhaças. O objetivo é que o veículo, ao sair do depósito, tenda a se afastar do mesmo (sendo incentivada a visita a clientes na vizinhaça externa) e, à medida que elabora sua rota, tenda a voltar ao mesmo (sendo incentivada a visita a clientes da vizinhaça interna).

$$s = \begin{cases} \operatorname{argmax}_{u \in A_k} \{\mu(r, u)\} & \text{se } q \leq q_0 \\ S & \text{caso contrário} \end{cases} \quad (6)$$

Onde S é a decisão tomada probabilisticamente ditada por:

$$p_k(r, s) = \begin{cases} \frac{\mu(r, s)}{\sum_{\forall u \in A_k} \mu(r, u)} & \text{se } s \in A_k \\ 0 & \text{caso contrário} \end{cases} \quad (7)$$

Sendo $\mu(r, s)$ a atratividade do arco (r, s) definida por:

$$\mu(r, s) = [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta \cdot [V_i(r, s)] \cdot [V_e(r, s)] \quad (8)$$

Onde $V_i(r, s)$ e $V_e(r, s)$ são os fatores multiplicadores de vizinhança interna e externa, respectivamente, e são obtidos por:

$$V_i(r, s) = \begin{cases} C_i(c) & \text{se } s \in VI(r) \\ 1 & \text{caso contrário} \end{cases} \quad (9)$$

$$V_e(r, s) = \begin{cases} C_e(c) & \text{se } s \in VE(r) \\ 1 & \text{caso contrário} \end{cases} \quad (10)$$

E C_i e C_e são funções da capacidade disponível no veículo. Na implementação do problema, considerou-se que o veículo sai do depósito com capacidade nula ($c = 0$) e retorna com a capacidade próxima ou igual a 1 ($c \leq 1$). Logo, C_i aumenta à medida que a capacidade aumenta enquanto C_e diminui na mesma relação. Foram escolhidas, para C_i e C_e , respectivamente:

$$C_i(c) = \begin{cases} 1 & \text{se } c \leq 0.5 \\ -2 + 6 \cdot c & \text{se } 0.5 < c \leq 1 \end{cases} \quad (11)$$

$$C_e(c) = \begin{cases} 4 - 6 \cdot c & \text{se } c \leq 0.5 \\ 1 & \text{se } 0.5 < c \leq 1 \end{cases} \quad (12)$$

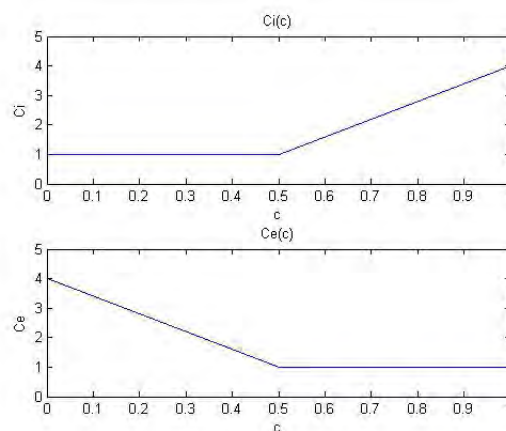


Figura 2: $C_i(c)$ e $C_e(c)$

3.2. Evaporação do Feromônio

Esta etapa da otimização define de que maneira o feromônio no espaço de busca irá evaporar. A evaporação evita o acúmulo excessivo de feromônio em determinados arcos, o que poderia estagnar a busca a poucas rotas.

[Dorigo e Gambardella, 1997] propuseram a evaporação de feromônio a cada passagem de uma formiga por um arco. Além disso, o feromônio, apesar de evaporar, tem uma fração de sua quantidade repostada para que arcos não visitados preservem uma competitividade mínima na formulação de rotas. Desta forma, a cada movimento de uma formiga por um arco (r, s) , o feromônio neste arco evapora seguindo:

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \quad (13)$$

Onde:

- ρ : Fator de evaporação de feromônio; e
- τ_0 : Quantidade inicial de feromônio em todos os arcos.

3.3. Depósito de Feromônio

O depósito de feromônio define o quão valioso será cada arco uma vez definidas soluções que o contenham. Quanto melhor a solução construída, maior a quantidade de feromônio que será depositada nos arcos que pertencem a ela. O feromônio depositado por cada formiga é dado por:

$$\tau(f) = \frac{\eta(f)}{\sum_{\forall i \in F} \eta(i)} \cdot \tau_{total} \quad (14)$$

Onde:

- $\tau(f)$: Quantidade de feromônio a ser depositado pela formiga f ;
- $\eta(f)$: O inverso do comprimento da solução proposta por f , ou seja, $1/D(f)$;
- F : Conjunto de todas as formigas do processo de otimização; e
- τ_{total} : Quantidade total de feromônio a ser distribuído entre todas as formigas depositantes.

Em seguida, dentro da solução gerada pela formiga depositante, cada rota individual traçada tem seu caminho reforçado por feromônio através de:

$$\tau(k) = \frac{\eta(k) \cdot \gamma(k)}{\sum_{\forall j \in K_f} \eta(j) \cdot \gamma(j)} \cdot \tau(f) \quad (15)$$

Onde:

- $\tau(k)$: Quantidade de feromônio a ser depositado no decorrer da rota k ;
- $\eta(k)$: O inverso do comprimento da rota k , ou seja, $1/D(k)$;
- $\gamma(k)$: A demanda total de carga atendida pela rota k ; e
- K_f : Conjunto de todas as rotas pertencentes à solução proposta pela formiga f .

O depósito de feromônio em cada arco ao fim da criação de cada conjunto de soluções pelas formigas é feito por:

$$\tau(r, s) \leftarrow \tau(r, s) + \sum_{\forall f \in F} \sum_{\forall k \in K_f} \frac{\tau(k)}{n_k} \quad (16)$$

Sendo n_k o número de arcos contidos na rota k .

4. Resultados

Dois variações da heurística ACO foram comparadas:

- ACO: Implementação clássica; e
- ACO-FA: Semelhante à implementação ACO anterior com adição de 3.1.1.

Ambas foram aplicadas a 8 instâncias clássicas da literatura que podem ser facilmente obtidas em OR-library [Beasley, 1990]. Foram realizadas 40 execuções de cada algoritmo para cada instância. Os parâmetros utilizados foram $\alpha = 1$, $\beta = 2$ e $\rho = 0.05$.

Tabela 1: Resultados Comparativos entre ACO e ACO-FA

Instância	n	c	ACO			ACO-FA		
			Melhor	Médio	Pior	Melhor	Médio	Pior
vrpnc1	50	160	604	628	656	587	602	618
vrpnc2	75	140	1002	1045	1066	974	1009	1039
vrpnc3	100	200	1040	1122	1148	1029	1096	1126
vrpnc4	150	200	1302	1425	1541	1186	1345	1491
vrpnc6	50	160	599	627	650	575	601	619
vrpnc7	75	140	996	1048	1084	975	1014	1041
vrpnc8	100	200	1086	1123	1147	1037	1096	1135
vrpnc9	150	200	1304	1429	1602	1249	1370	1571

As figuras 3 e 4 ilustram os testes estatísticos comparativos de boxplot realizados para cada instância. Em cada comparação, a variação 1 refere-se ao ACO, enquanto a variação 2 refere-se ao ACO-FA.

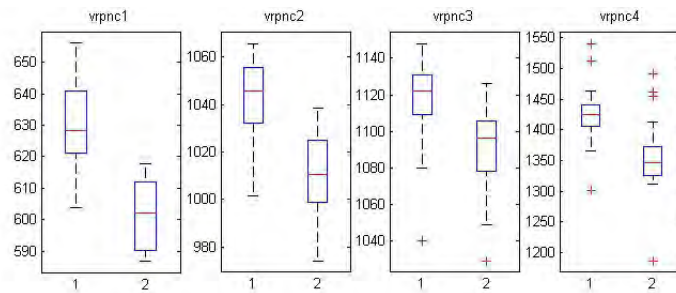


Figura 3: Instâncias *vrpnc1* a *vrpnc4*

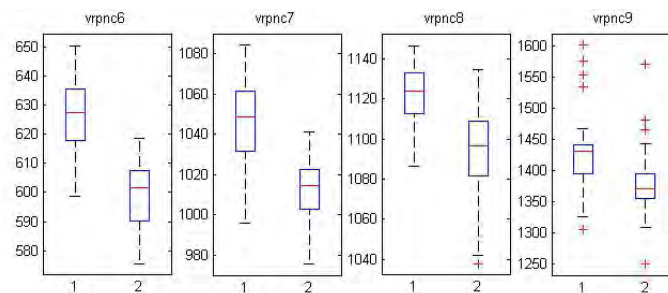


Figura 4: Instâncias *vrpnc6* a *vrpnc9*

A figura 5 mostra uma rota encontrada pelo algoritmo ACO e outra encontrada pelo algoritmo ACO-FA para a instância *vrpnc2*.

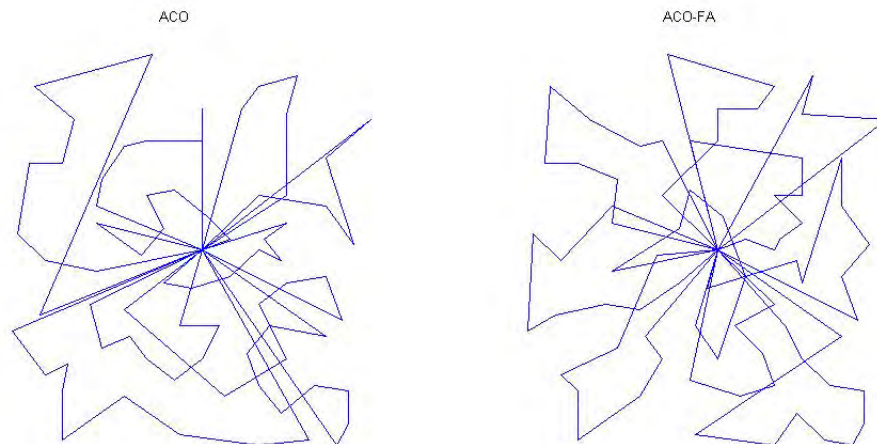


Figura 5: Instância *vrpnc2*

5. Análise dos Resultados

O algoritmo ACO-FA é melhor estatisticamente em seus resultados em 7 dos 8 cenários, pois as caixas do boxplot não estão sobrepostas e estão abaixo das da implementação clássica.

A rota feita pelo ACO na figura 5 mostra o problema que este trabalho tenta tratar, que é diminuir a frequência com que uma rota se afasta e se aproxima do depósito. Na mesma figura, a rota feita pelo ACO-FA demonstra o melhoramento esperado pelo algoritmo, estreitando as pétalas e tornando-as mais suaves na maioria dos casos.

6. Conclusão

A variação proposta pelo trabalho melhorou significativamente os resultados na resolução do PRVC em 7 dos 8 cenários testados. Isto demonstra que a geometria observada nas soluções se relaciona com um menor custo das mesmas.

Estando o projeto ainda em etapa preliminar de desenvolvimento, pretende-se aprimorar a definição dos conjuntos $VI(r)$ e $VE(r)$, assim como as relações para os fatores $C_i(c)$ e $C_e(c)$.

7. Agradecimentos

Este projeto de pesquisa só pode ser realizado graças ao apoio das instituições FAPEMIG, Capes, CNPq e CEFET-MG. Destinam-se a elas estes agradecimentos.

Referências

- Beasley, J. E. (1990). Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072.
- Bell, J. E. e McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18:41–48.
- Bin, Z.-Z. Y., Y. e Baozhen, Y. (2009). An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 196:171–176.
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6: 80–91.
- Dorigo, M. e Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem. *Biosystems*, 43:73–81.

Dorigo, M.-V., M. e Coloni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 26:29–41.

Ning, T. e Guo, C. (2012). Solving vrp using ant colony optimization algorithm. *International Conference on Information and Computing Science*, 5:15–18.