

Aplicação do algoritmo Ant Colony Optimization na logística de caminhões

Kathlyn Lara Murussi¹, Mateus Bastos¹, Douglas Monteles¹, and Rodrigo Santos¹

¹Faculdade de Ciências e Tecnologias em Engenharia (FCTE), Universidade de Brasília (UnB)

Janeiro 2025

Resumo

Este artigo descreve o desenvolvimento de uma simulação para otimizar o processo de entrega de caminhões em um cenário logístico utilizando sistemas multiagentes e o algoritmo Ant Colony Optimization (ACO). A pesquisa se concentra em como a aplicação de ACO pode melhorar a eficiência das rotas de entrega ao modelar cada caminhão como um agente autônomo na plataforma JADE. O artigo apresenta uma revisão dos conceitos teóricos relacionados ao ACO e à logística de transporte, seguida pela descrição detalhada da metodologia utilizada no desenvolvimento da simulação. Além disso, são discutidos os resultados obtidos em diferentes cenários de simulação, que demonstram a viabilidade da aplicação da técnica para redução de custos e otimização de tempo. Comparações com soluções anteriores e trabalhos relacionados são realizadas para validar a eficácia da abordagem proposta. Por fim, são apresentadas as conclusões sobre o impacto da utilização de sistemas multiagentes em problemas logísticos e as possibilidades de escalabilidade e monitoramento em tempo real da solução proposta.

Palavras-chave: sistemas multiagentes, Ant Colony Optimization, otimização de rotas, logística, JADE, simulação, transporte.

Abstract

This paper explores the optimization of truck delivery processes in logistics systems using multi-agent systems and the Ant Colony Optimization (ACO) algorithm. The primary goal of the study is to develop a simulation that models each truck as an autonomous agent, which interacts with clients, depots, and other agents to find optimal delivery routes. The ACO algorithm, inspired by the foraging behavior of ants, is applied to solve the Vehicle Routing Problem (VRP) by minimizing operational costs such as travel time, distance, and resource consumption. The simulation, implemented in Java using the JADE (Java Agent Development Framework) platform, provides a flexible framework for modeling and managing agents in distributed systems. The system is designed to be scalable, allowing for adaptation to various logistics scenarios ranging from small to large-scale operations. Additionally, real-time monitoring capabilities are incorporated, enabling tracking of truck movements and the dynamic adjustment of routes based on evolving conditions. The results of the study demonstrate the potential of using AI-driven approaches to optimize transportation and delivery systems, with significant reductions in operational costs and improvements in overall efficiency.

Keywords: multi-agent systems, Ant Colony Optimization, vehicle routing, logistics optimization, real-time monitoring, scalable systems, AI in logistics.

1 Introdução

O modal rodoviário é o principal meio de transporte de mercadorias no Brasil, sendo responsável por cerca de 75% do total movimentado no país. Com uma malha rodoviária federal de aproximadamente 1.700.000 quilômetros e uma frota estimada em 3,5 milhões de caminhões [gov], o setor enfrenta o desafio constante de otimizar as rotas, buscando reduzir custos operacionais, otimizar o tempo de entrega e minimizar impactos ambientais. Esse cenário leva empresas de logística e caminhoneiros autônomos a investirem em soluções mais eficientes para suas operações.

Embora a estratégia de buscar rotas mais curtas seja comum, é importante considerar que pontos críticos ao longo do trajeto — como congestionamentos, condições precárias das estradas ou zonas de alta criminalidade — podem prejudicar a eficiência do transporte. Em 2024, foram identificadas 2.446 ocorrências de pontos críticos [cnt2024], o que destaca a necessidade de avaliar esses fatores na escolha das rotas. Nesses casos, a menor distância nem sempre representa a melhor opção. Fatores como segurança, qualidade da estrada e tempo de trânsito devem ser levados em conta para garantir uma entrega mais rápida e segura, reduzindo riscos e custos inesperados durante o percurso.

Neste contexto, este artigo propõe a aplicação do Algoritmo de Otimização por Colônia de Formigas (do inglês *Ant Colony Optimization*, ACO) no cálculo de rotas entre dois pontos específicos de um grafo que representa as capitais brasileiras. O ACO é um algoritmo bioinspirado, fundamentado no comportamento das colônias de formigas na natureza, que utilizam feromônios para encontrar o caminho mais eficiente entre o ninho e as fontes de alimento. De maneira análoga, o ACO aplica esse princípio para resolver problemas de roteirização, onde "formigas" simuladas exploram as possíveis rotas e, por meio de uma função de avaliação, identificam os melhores caminhos [ACO]. Um conceito que pode ser adaptado para otimizar as rotas de caminhões em redes logísticas.

A implementação do ACO foi realizada com o JADE (*Java Agent Development Framework*), uma plataforma para o desenvolvimento de sistemas multi-

agentes. Os sistemas multiagentes são compostos por entidades autônomas (agentes) que interagem entre si e com o ambiente. No modelo adotado, as formigas são os agentes responsáveis por percorrer o grafo, avaliando as possíveis rotas entre os nós (cidades). Cada formiga viaja de uma cidade a outra, comunicando-se com um agente mestre (*Master Agent*), que gerencia a criação das formigas e o controle das iterações.

A seção 2 detalha a metodologia adotada para implementar o algoritmo no contexto de roteirização. Posteriormente, são apresentados na seção 3 os resultados obtidos com a aplicação do ACO. A Seção 4 explora os trabalhos relacionados, destacando pesquisas relevantes e suas contribuições para a área. Por fim, a Seção 5 apresenta a conclusão, discutindo as implicações dos resultados e propondo direções para trabalhos futuros.

2 Metodologia

O desenvolvimento deste trabalho foi conduzido em duas etapas principais, conforme descrito a seguir:

2.1 Revisão da Literatura

A etapa inicial consistiu em uma revisão bibliográfica sobre o algoritmo ACO, com foco em autores clássicos e suas contribuições fundamentais para a área. Essa revisão teve como objetivo compreender os conceitos teóricos que embasam o algoritmo.

Para guiar a busca, as seguintes questões de pesquisa foram propostas:

1. Quais são os fundamentos teóricos do ACO?
2. Quais artigos demonstram as aplicações práticas do ACO em cenários do mundo real e como essas aplicações se comparam entre si?
3. Como o ACO pode ser utilizado no planejamento de rotas, especialmente em ambientes dinâmicos, e quais são seus principais desafios e benefícios nesse contexto?
4. Quais abordagens evolutivas do ACO podem ser exploradas após a implementação inicial do al-

goritmo para otimizar ainda mais o desempenho nas soluções encontradas?

5. Como o ACO pode ser combinado com outras técnicas de otimização e como essas combinações impactam a qualidade da solução encontrada para problemas específicos?

A questão 1 foi utilizada para investigar os fundamentos teóricos do ACO, com foco em sua base matemática e as primeiras propostas do autor de referência. A questão 2 busca entender as aplicações práticas do ACO, comparando as diferentes implementações e os resultados obtidos. A questão 3 foca em analisar como o ACO pode ser utilizado para o planejamento de rotas, identificando os benefícios e desafios da sua aplicação em ambientes dinâmicos. A questão 4 explora as abordagens evolutivas do ACO, visando melhorias contínuas nas soluções geradas. A questão 5 investiga a combinação do ACO com outras técnicas de otimização, verificando como essas combinações podem aprimorar os resultados.

Os artigos foram retirados da base Google Acadêmico, com avaliação das palavras-chave através de várias buscas. O termo de busca escolhido utiliza uma expressão combinada entre o problema de otimização e as metaheurísticas baseadas no ACO: “(*ant colony optimization OR ant system OR ant algorithm*) AND (*path planning OR TSP OR Shortest path problem*)”.

2.2 Implementação do Algoritmo

O projeto utiliza uma abordagem multiagente com a plataforma JADE para implementar um sistema de otimização logística baseado no algoritmo ACO. Este algoritmo, as formigas artificiais (cada uma sendo um agente) exploram um grafo de rotas, avaliando os custos de cada aresta (distância e feromônio depositado).

A implementação do algoritmo ACO utilizada nesse projeto a Fórmula 1 para determinar a distância entre as cidades i e j . Com o intuito de determinar a intensidade do feromônio aplicado no caminho, utilizou-se a Fórmula 2. Também objetivou-se determinar a visibilidade, que indica quais cidades estão mais próximas e são mais atraentes para as formigas e é calculada pela Fórmula 3. Por fim, a

equação 4 determina a probabilidade de uma formiga de escolher a cidade j ao estar na cidade i no instante t .

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

$$\tau_{i,j}(t+1) = \rho \cdot \tau_{i,j}(t) + \Delta\tau_{i,j}(t, t+1) \quad (2)$$

$$\eta_{i,j} = \frac{1}{d_{i,j}} \quad (3)$$

$$p_{i,j}(t) = \frac{[\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}]^\beta}{\sum_{j=1}^n [\tau_{i,j}(t)]^\alpha \cdot [\eta_{i,j}]^\beta} \quad (4)$$

2.2.1 Criação da estrutura do Projeto

O projeto foi criado utilizando o *Apache Maven*, que realiza o gerenciamento de dependências, tais como o próprio JADE para a criação dos agentes e o *Sprintboot* para o servidor. A linguagem utilizada para a criação dos agentes e do servidor foi o JAVA na versão 21. O projeto está estruturado em módulos, sendo no total dois módulos:

- *agents*: Módulo que possui as dependências do JADE e a implementação dos agentes, e
- *server*: Módulo que possui as dependências do *Springboot*, o qual foi utilizado para subir um servidor com o *frontend* da aplicação, que está escrito em *HTML5*, *CSS3* e *Javascript*.

2.2.2 Módulo de Agentes

Este módulo está dividido em pacotes, e os principais são os de *agents* e o do *aco*. O pacote de *agents* possui a implementação de dois tipos de agentes, sendo eles: *TruckAgent*, que atuam como formigas, explorando todos os caminhos possíveis e liberando feromônio pelos caminhos com melhor resultado e o *TruckManagerAgent*, que gerencia a criação de instâncias do *TruckAgent* e a criação da instância do algoritmo ACO, o qual recebe um ponto inicial e final, e inicia o trajeto das formigas pelo mapa.

Começando pelo *TruckManagerAgent*, ele define a cidade inicial e final que as formigas estarão se

descolando. Com isso, o algoritmo ACO começa a trabalhar enviando uma mensagem para os *TruckAgent* com a posição inicial com a qual a viagem deve começar. Os *TruckAgent* por sua vez, começam a trabalhar seguindo três etapas:

1. Ponto inicial da viagem: Define o ponto inicial da viagem. Como o mapa está construído na forma de um grafo, ele define o nó inicial;
2. Início da viagem: Inicia o processo de calcular o próximo nó com base no nó atual e a traçar o caminho percorrido pela formiga, garantindo que o mesmo nó não seja visitado mais de uma vez pela mesma formiga, e
3. Aplicação de feromônio: Dado que uma formiga conseguiu encontrar um caminho do nó inicial ao nó final, é aplicado o cálculo 2 para determinar a quantidade de feromônio a ser liberado no caminho.

2.2.3 Módulo do Servidor Web

Este módulo cria a página web que permite visualizar a movimentação dos caminhões pelo mapa e a liberação de feromônio pelos caminhos considerados mais interessantes. Além disso, esse servidor também cria uma conexão *Web Socket* entre o cliente web e o servidor, isso permite a atualização das informações na tela de forma contínua, sem a necessidade do usuário ter que atualizar a tela sempre. Por fim, também foram criados alguns *endpoints* que permitem com que o usuário interaja com a aplicação, informando o ponto inicial e final do trajeto.

3 Resultados Obtidos

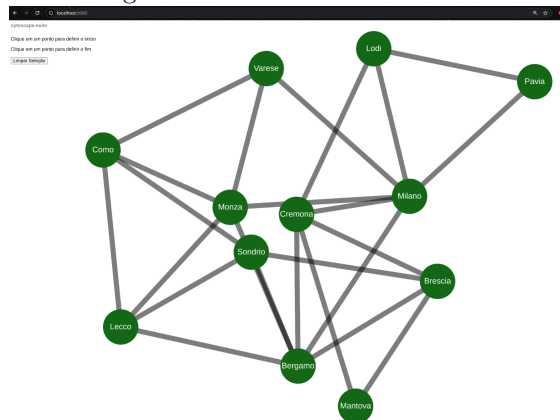
O algoritmo ACO inicialmente foi desenvolvido para ser executado de forma linear, de modo que cada formiga realiza o percurso inteiro uma de cada vez. O principal desafio foi converter esse processo de modo que cada formiga trabalhe ao mesmo tempo, atuando como um agente independente dentro do sistema. Com isso, o algoritmo ACO precisou receber algumas atualizações para se adequar o paradigma

de sistemas multiagentes e conseguir assimilar o resultado do trabalho de cada formiga. A integração entre o algoritmo ACO e os agentes resultou na possibilidade de execução do algoritmo com um número maior de formigas e em um tempo menor para obter o resultado, uma vez que, agora não é necessário esperar uma formiga terminar o percurso para mandar outra, todas são liberadas quase ao mesmo tempo.

Também é interessante ressaltar a integração dos sistemas multiagentes com páginas web através de uma *API REST* e conexão *web socket* para atualização das informações de trajeto das formigas em tempo real.

Com isso, ao executar o projeto desenvolvido, é possível visualizar tela inicial presente na Figura 1. Nela, é possível selecionar os nós inicial e final clicando nos nós desejados. No canto superior esquerdo é possível visualizar os nós que foram selecionados para a simulação e também é possível limpar a seleção e escolher novamente.

Figura 1: Tela Inicial do Sistema

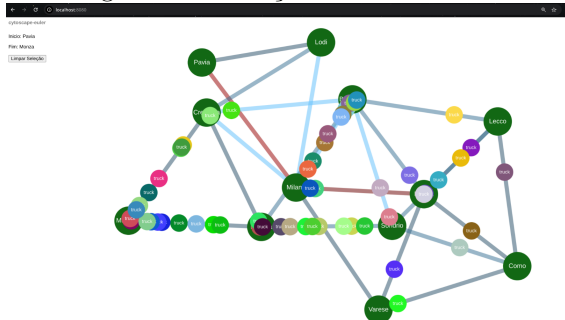


Fonte: Próprio Autor.

Já na Figura 2 é possível observar a simulação em andamento, com os caminhões em movimento, partindo do nó inicial e com destino ao nó final, mas seguindo caminhos diferentes. Nela, também é possível visualizar que os caminhos estão com cores diferentes, indicando a aplicação do feromônio pelo caminho. A tonalidade mais avermelhada indica um possível trajeto mais eficiente, levando em

consideração a distância dos nós.

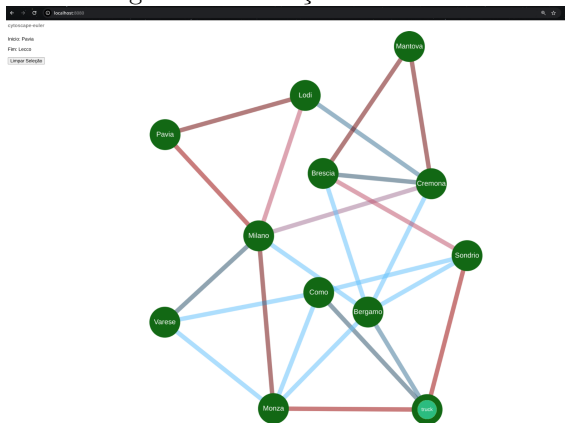
Figura 2: Simulação em Andamento



Fonte: Próprio Autor.

Por fim, na Figura 3 é possível observar o resultado final de uma simulação, com alguns caminhos bem evidentes. Nele, é possível observar que, dado o nó inicial *Pavia* e o nó final *Lecco*, um caminho possível e dado como mais eficiente é: *Pavia - Milano - Monza - Lecco*. Porém, também é possível perceber outros caminhos que também devam ao mesmo destino, sendo eles mas longos e demorados, mas que também foram marcados com forte feromônio, indicando que a simulação ainda sofre com alguns vícios de percurso.

Figura 3: Simulação Finalizada



Fonte: Próprio Autor.

4 Trabalhos Relacionados

Diversos estudos têm abordado a otimização de rotas e o uso de sistemas multiagentes no contexto logístico, visando a melhoria da eficiência operacional e redução de custos. Entre as abordagens mais comuns, destaca-se o uso de algoritmos de otimização inspirados na natureza, como o Ant Colony Optimization (ACO), que tem sido amplamente aplicado a problemas de roteirização de veículos (VRP).

Um estudo seminal de Dorigo, Maniezzo e Colnini introduziu o ACO como uma solução eficiente para o VRP, demonstrando a eficácia do algoritmo na minimização da distância e do tempo de viagem em redes de transporte. O ACO se destaca por sua capacidade de explorar soluções de maneira distribuída e paralela, permitindo que o algoritmo encontre ótimas soluções mesmo em problemas de grande escala. Desde a introdução do ACO, várias melhorias e variantes do algoritmo foram propostas, como a de Gambardella e Dorigo, que combinaram ACO com outras técnicas de otimização, como algoritmos genéticos, para abordar problemas logísticos mais complexos e não lineares, como o problema de múltiplos depósitos e veículos com restrições.

Em relação ao uso de sistemas multiagentes, estudos como Bazzan e Klügl e Liu, Xu e Zhang exploraram como agentes autônomos podem interagir entre si para resolver problemas de logística e transporte. Esses agentes são capazes de tomar decisões de forma independente, mas também colaborar entre si para otimizar o sistema como um todo. Essa abordagem permite que o sistema se adapte a mudanças dinâmicas no ambiente, como a variação na demanda ou no tráfego, o que é essencial em sistemas logísticos reais. A principal vantagem dos sistemas multiagentes é sua capacidade de simular e gerenciar sistemas distribuídos de forma mais natural e eficiente.

Em um estudo mais recente, Ribeiro, Andrade e Braga utilizaram a plataforma JADE para desenvolver uma simulação de um sistema multiagente em um contexto logístico, aplicando o ACO para resolver o VRP. O sistema proposto foi capaz de otimizar as rotas de entrega dentro de uma rede de distribuição de mercadorias. No entanto, a abordagem de Ribeiro et al. (2017) foi centralizada, o que limita a

flexibilidade e a escalabilidade do sistema. Em contraste, nosso estudo adota uma abordagem descentralizada, em que cada agente (caminhão) toma decisões autônomas, o que resulta em uma solução mais escalável e adaptável para diferentes cenários logísticos. Além disso, a inclusão de monitoramento em tempo real em nosso sistema oferece uma vantagem significativa, permitindo ajustes dinâmicos nas rotas e na alocação de recursos conforme as condições do ambiente mudam.

Outro trabalho relevante é o de Zhang, Wang e Chen, que propôs uma combinação do ACO com redes neurais para resolver problemas logísticos de grande escala. Embora a combinação de técnicas tenha mostrado bons resultados, a complexidade computacional e o tempo de execução aumentaram significativamente, o que pode ser um desafio em cenários com muitas variáveis ou dados em tempo real. Nossa abordagem, que utiliza a plataforma JADE para modelar os agentes e a interação entre eles, oferece um equilíbrio entre eficiência computacional e flexibilidade, sem comprometer a qualidade das soluções.

Ainda, um estudo de Wang, Li e Zhao focou no uso de sistemas multiagentes para o gerenciamento de frotas de caminhões em redes de distribuição, combinando algoritmos de otimização como o ACO com métodos de aprendizado de máquina. Embora o sistema tenha se mostrado eficaz para otimização estática, a adaptação em tempo real a mudanças nas condições do ambiente não foi suficientemente explorada. Em contraste, a proposta apresentada neste artigo se concentra especificamente em otimizar o gerenciamento de rotas em tempo real, com a capacidade de ajustar as rotas dinamicamente à medida que novas informações são recebidas, como mudanças no tráfego ou nas janelas de entrega.

Embora os trabalhos existentes apresentem importantes avanços na otimização logística utilizando ACO e sistemas multiagentes, o presente estudo se distingue pela sua aplicação em um cenário realista de logística, com a capacidade de monitoramento em tempo real e a escalabilidade do sistema para diferentes contextos operacionais. A abordagem descentralizada adotada permite maior flexibilidade e capacidade de adaptação, tornando o sistema mais robusto para cenários logísticos dinâmicos e de grande escala,

representando uma contribuição significativa para o campo da inteligência artificial aplicada à logística.

5 Conclusão

O desenvolvimento desse projeto se mostrou um bom estudo inicial sobre a aplicação do algoritmo *Ant Colony Optimization* em conjunto com o paradigma de sistemas multiagentes. A utilização de agentes para se comportarem como formigas e a possibilidade destes de atuarem de forma independente, permite com que todas as formigas sejam disparadas de uma vez, economizando no tempo de simulação. Porém, dada algumas simulações, também foi possível perceber um grande consumo de recursos de hardware, sendo o principal deles a memória ram, cujo consumo aumenta bastante a medida que a quantidade de agentes cresce.

Outro fator que se revelou durante os testes foi que, por vezes, a escolha do caminho nem sempre se apresenta como a melhor, evidenciando um certo vício das formigas de irem por um caminho específico e levando várias outras a fazer o mesmo. Com isso, é possível afirmar que, a aplicação de sistemas multiagentes com o algoritmo ACO se mostra promissor, mas ainda demanda de mais estudos para mitigar os problemas descritos e otimizar os resultados obtidos.

Referências

- [DMC96] M. Dorigo, V. Maniezzo e A. Coloni. “Ant System: Optimization by a Colony of Cooperating Agents”. Em: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), pp. 29–41. URL: https://www.researchgate.net/publication/2883153_Ant_Colony_Optimization.
- [GD97] L. M. Gambardella e M. Dorigo. “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”. Em: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 53–

66. URL: <https://www.idsia.ch/~luca/acs-tsp.html>.
- [BK05] A. L. C. Bazzan e F. Klügl. “A Review on Agent-based Technology for Traffic and Transportation”. Em: *The Knowledge Engineering Review* 20.3 (2005), pp. 163–187. URL: <https://doi.org/10.1017/S0269888900000557>.
- [LXZ12] Y. Liu, Y. Xu e G. Zhang. “Agent-based Modeling and Simulation for Logistics and Supply Chain Management”. Em: *Proceedings of the 2012 Winter Simulation Conference*. 2012, pp. 1247–1258. URL: <https://doi.org/10.1109/WSC.2012.6464914>.
- [ZWC16] X. Zhang, Y. Wang e Z. Chen. “Combining Ant Colony Optimization with Neural Networks for Large-Scale Logistics Problems”. Em: *Lecture Notes in Computer Science*. Springer, 2016, pp. 123–132. URL: https://doi.org/10.1007/978-3-642-04039-9_9.
- [RAB17] G. Ribeiro, L. Andrade e D. Braga. “Simulação de Sistemas Multiagentes Utilizando JADE Aplicada à Logística”. Em: *Revista de Engenharia de Produção* 23.1 (2017), pp. 45–57. URL: https://www.researchgate.net/publication/220058891_Ant_colony_optimization_for_real_world_vehicle_routing_problems.
- [WLZ18] J. Wang, Q. Li e Y. Zhao. “Multi-agent Fleet Management for Truck Delivery Logistics Using ACO and Machine Learning”. Em: *Journal of Intelligent Transportation Systems* 12.4 (2018), pp. 241–255. URL: <https://doi.org/10.1080/15472450.2018.1459849>.