

Problema de roteamento de veículos com inteligência artificial*

Wilton Costa¹[0000–0001–9445–5572]

Universidade Federal de Mato Grosso do Sul, Campo Grande MS, BRAZIL
wilton.costa@ufms.br
Faculdade de Computação (FACOM), Universidade de Mato Grosso do Sul, Campo Grande-MS, Brasil

Abstract. O transporte de objetos, pessoas, comidas, têm se popularizado em aplicativos mobile, mas é um problema enfrentado desde sua introdução com o artigo (2) e veio a tona pelos comerciantes e serviços públicos, neste artigo é demonstrado como resolver o problema de roteamento de veículos capacitado e com janelas de tempo com a ajuda da inteligência artificial, mas precisamente com aprendizagem não supervisionada K-Means e com um algoritmo A*, no intuito de entregar todos os pedidos em um dia normal de trabalho.

Keywords: Vehicle Routing Problem · K-Means · algoritmo A*.

1 O problema de roteamento de veículos (VRP)

O problema de roteamento de veículos foi introduzido pelo (2), e desde então tem se tornado polêmico na literatura, ele consiste na busca de melhores rotas para um frota de veículos, trata-se de um problema NP-difícil, uma vez que sempre será necessário conhecer todas as rotas para verificar qual possui "o menor custo" para os entregadores, este problema se derivou do problema do caixeiro viajante, que procurava o melhor caminho para um único veículo. Entretanto no VRP, o desafio se estende para uma frota de veículos.

Na prática, esse problema é enfrentado por empresas como: Loggi, Uber, iFood, Rappi, entre outros que se preocupam com rotas para os veículos realizarem uma coleta ou entrega de objetos ou pessoas. O VRP se estendeu para outras variantes como: o problema de roteamento de veículos capacitado (CVRP), que se preocupa com a capacidade dos veículos, o problema de roteamento de veículos dinâmicos (DVRP), que permite uma busca na frota de veículos para executar uma nova demanda não esperada e o problema de roteamento de veículos com janela de tempo (VRPTW), que busca entregar todos os pedidos dentro de uma janela de tempo, neste artigo será mostrado como foi realizado a solução para essas variantes com inteligência artificial com intuito de responder as demandas em 1 dia útil, ou seja, enquanto tiver motoristas e veículos para entregar os pedidos.

* Desenvolvido por Wilton.

Foi encontrado outra maneira de responder tal questão com a colônia de formigas no artigo (4), a ideia é definir os veículos existentes e mandá-los executar todos os pedidos sempre salvando a rota da frota anterior, assim a cada rota criada para a frota, gera um rastro que pode ser usado na próxima iteração das novas formigas, ou seja, quanto mais tempo o algoritmo for executado, maior a probabilidade de encontrar os melhores caminhos, a grande questão que se abre é se não for possível computacionalmente realizar este procedimento para demandas mais complexas.

Este artigo foi separado da seguinte maneira, na seção 1 haverá uma explicação melhorada sobre as variantes do VRP e será apresentado uma solução com uma heurística para o algoritmo A*, na seção 2 é demonstrado como foi desenvolvido a solução e como foi a iteração com a parte de inteligência artificial, e por fim quais eram as expectativas. A última seção é mostrado o resultado dos experimentos e uma comparação do que será esperado com a realidade. Vale ressaltar que uma heurística representa uma solução ótima da busca encontrada, entretanto não há garantias que seja a melhor, pois a complexidade do problema pode ser alta, o que se procura dentro de uma heurística é a capacidade de encontrar um melhor objetivo para determinar as melhores distâncias.

Para este projeto foi utilizado a aprendizagem não-supervisionada K-Means e o algoritmo A*, a ideia é a separação dos veículos por zonas do mapa e conforme for aparecendo novas demandas, no mapa é possível verificar se é necessário um novo veículo para executar a demanda ou é possível procurar a zona mais próxima e integrar a demanda para o veículo responsável pela zona. Essa dúvida aparece pois é possível que uma ou outra seja mais eficiente, ou seja, um menor custo para as rotas da frota.

2 Desenvolvimento

2.1 Definições

O problema VRP se estendeu para várias outras variantes como é mostrado em (1). Neste artigo será retomado as variantes CVRP, DVRP e VRPTW que são descritas da seguinte forma:

- **O Problema de roteamento de veículos capacitado(CVRP):** Trata-se de um problema que dado a capacidade dos veículos e o número de itens de cada demanda, é possível separar as rotas de forma que todas as demandas sejam atendidas sem que seja necessário retornar ao depósito e obter novamente os itens para a distribuição, é possível ver uma representação na Figura 1, para mais detalhes também pode ser visto no artigo (5).
- **O Problema de roteamento de veículos dinâmicos (DVRP):** É uma variante do VRP que trata de novas demandas para o problema, como visto na Figura 2 no momento inicial só existem as demandas (A,B,C,D,E) após um tempo t_1 as demandas (X,Y) são acrescentadas a rota e por fim é obtido uma nova rota para o veículo. Esse foi o problema investigado pelo artigo (3).

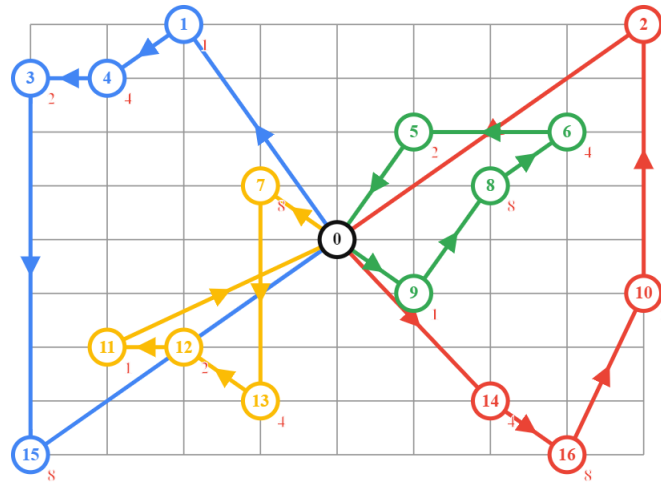


Fig. 1. Capacited Vehicle Routing Problem

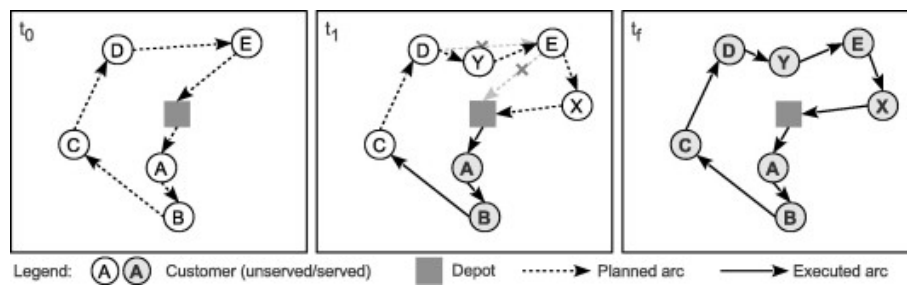


Fig. 2. Dynamic Vehicle Routing Problem

- **O Problema de roteamento de veículos com janelas de tempo (VRPTW):** É uma variante em que cada demanda possui uma janela de tempo para ser atendida. Um bom exemplo para este caso são as ambulâncias, as quais possuem um tempo inicial e por urgência um tempo limite para que o item seja atendido. A ideia é transpor um VRP para VRPTW buscando facilitar o cálculo da rota desejada com a finalidade de atender todos as demandas como solicitado.

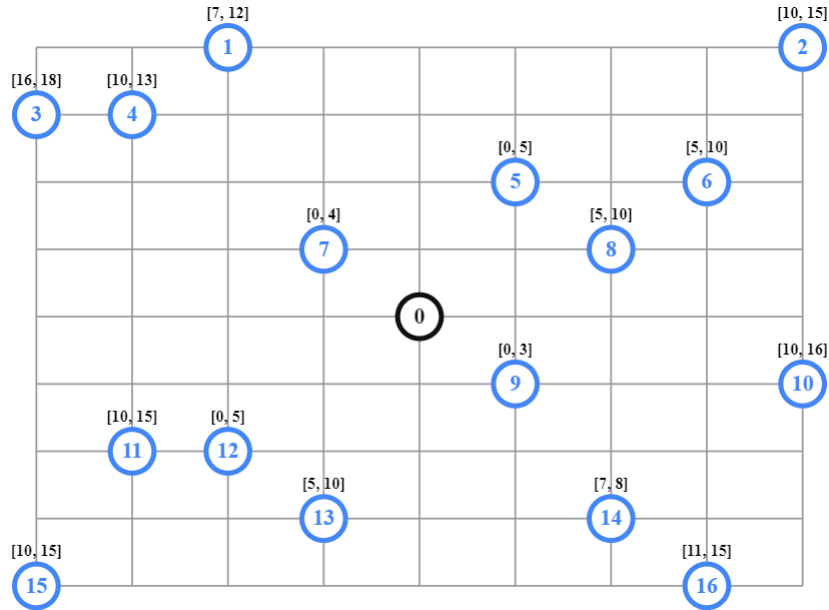


Fig. 3. Vehicle Routing Problem with Time-Window

O objetivo da heurística do algoritmo A* foi descrito no artigo (4), ela tem o objetivo de criar um conjunto de "formigas" sempre procurando as melhores respostas. Entretanto, esse processo dependendo da complexidade do problema pode se tornar muito custoso computacionalmente. A ideia para este projeto é minimizar o tempo de busca, e conseqüentemente diminuir o custo. Inicialmente, foram divididos todos os n-pedidos iniciais para os veículos em zonas, e essas zonas são encontradas com a aprendizagem não-supervisionada k-means, tal aprendizagem particiona os n pedidos em k conjuntos de pedidos, tal que cada conjunto possa ser executado de forma que respeite o CVRP e VRPTW.

A segunda parte do problema é o DVRP, na qual novos pedidos podem ser inseridos no meio do caminho, então uma forma mais eficiente para buscar o melhor veículo para executar a nova demanda é buscar o veículo que está na zona mais próxima do pedido, após encontrar o veículo mais próximo é necessário

verificar se dará tempo para executar todos os outros pedidos acrescentando o pedido novo. Além de respeitar a capacidade do veículo, caso o veículo tenha tempo para executar todas as tarefas respeitando a sua capacidade, pode ser que ele deixe de atender um outro pedido, com isso haverá outro veículo para se preocupar com a demanda que não será atendida, note que podem existir veículos no depósito, que por sua vez estavam com a carga máxima.

Este projeto está dividido nessas duas etapas que serão a parte alpha e a parte beta da solução do problema todo, neste artigo a parte implementada até o momento foi a parte alpha, que foi necessário a integração do A* do artigo 2, e o particionamento do problema em zonas como já descrito anteriormente.

2.2 Inteligência Artificial

K-Means Sua integração na parte alpha foi necessária para entender mais a fundo sobre as variantes do VRP, e para que seja possível tratá-las é preciso considerar que um pedido possui uma quantidade de itens pedidos, uma localização e um tempo de serviço para executar o pedido, a janela de tempo é dividida entre tempo inicial e tempo final. Seu tempo inicial pode ser dado pelo tempo necessário para chegar até a localização do pedido e seu tempo final sendo o tempo inicial somada com o tempo de serviço do pedido, em nosso caso a janela de tempo só estará presente para auxiliar o cálculo do objetivo da melhor rota dentro da zona encontrada.

Ao entender melhor como o pedido está em nosso problema, foi necessário implementar um K-Means que respeitasse tanto a proximidade dos pedidos quanto a quantidade de itens que a rota sugerida pelo K-Means possui, ou seja dentro de um cluster do K-Means se têm :

$$CK_i = \frac{n}{k_i} \quad (1)$$

Onde,

- CK_i é a capacidade total do cluster i
- n é o total de pedidos do problema
- k_i é a capacidade máxima do veículo i

De um aspecto geral é possível imaginar que responder a um cluster inteiro, seja responder a várias demandas, ou seja, caso seja possível calcular a melhor rota dentro do cluster com o A* já proposto e exemplificado na seção seguinte, é possível prever o tempo para responder um cluster, assim é capaz armazenar a informação para uma futura pesquisa, como mostra a Figura 4

Após todo problema ser separado em regiões, cada veículo executa uma heurística A*

Algoritmo A* Para uma melhor compreensão sobre o algoritmo foi exemplificada a Figura 5, através dela é possível notar que após chegar no estado final, ou seja, retornar ao depósito o algoritmo já se encerra. Neste artigo foi utilizado como o melhor estado aquele que possuía a menor distância do depósito até o ponto do estado.

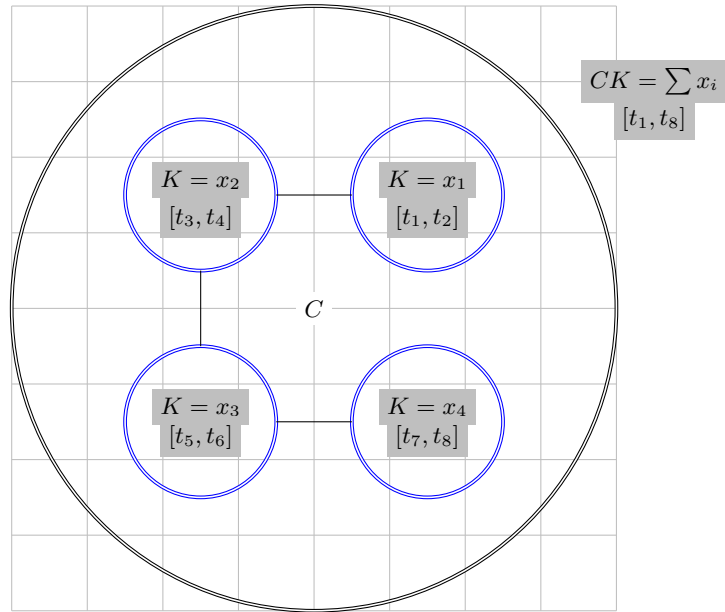


Fig. 4. Centroid responsável por Pedidos

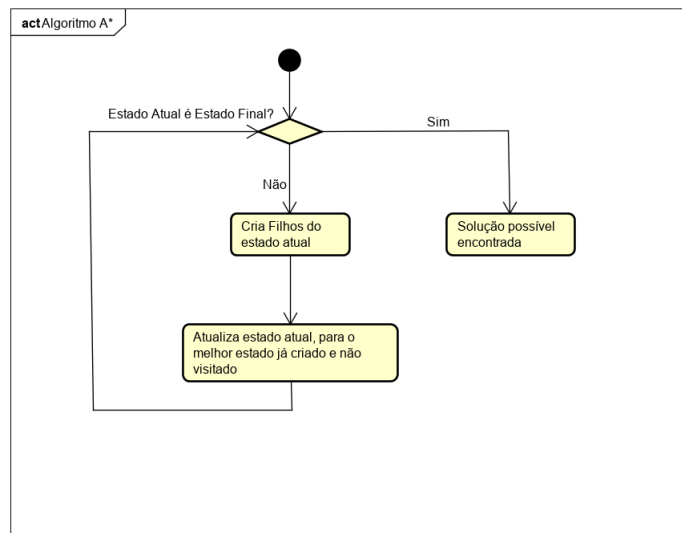


Fig. 5. Diagrama de atividades do algoritmo A*

2.3 Expectativas

Como é preciso tratar o problema em 1 dia útil, ou seja, enquanto os motoristas puderem trabalhar no dia, é esperado que o algoritmo atenda todos os pedidos no dia, inclusive aos pedidos que são chamados depois que toda rota já está toda construída.

A parte alpha poderá ficar melhor com a reformulação da mesma ideia descrita no artigo (6), que ao considerar que os veículos estão separados por regiões é esperado que o cálculo para a nova rota seja mais rápido do que recalcular todos os trajetos já propostos para os outros veículos.

3 Experimentos

Após realizar os testes fica nítido a diferença entre a quantidade de nós visitados, enquanto que no algoritmo da colônia de formigas foi obtido cerca de 26126 nós percorridos em 100 segundos, no A* juntamente com o K-Means cerca de 13655 nós para busca dos melhores clusters e 189 nós para busca do algoritmo A*, o que totaliza 13844 nós percorridos em 4 segundos, porém o A* não retornou a melhor rota uma vez que ele busca chegar no estado final.

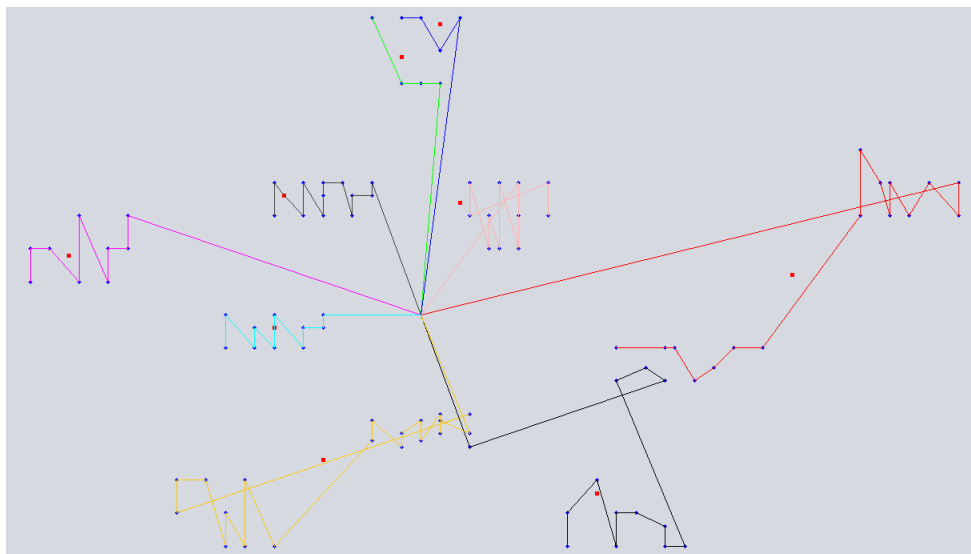


Fig. 6. Resposta final deste Artigo, Centro dos centroides em vermelho

3.1 Experimentos realizados

Para os experimentos foi descrito um problema, seus pedidos estão mais próximos um do outro no mapa inteiro. Para melhor visualização, na prática foi desen-

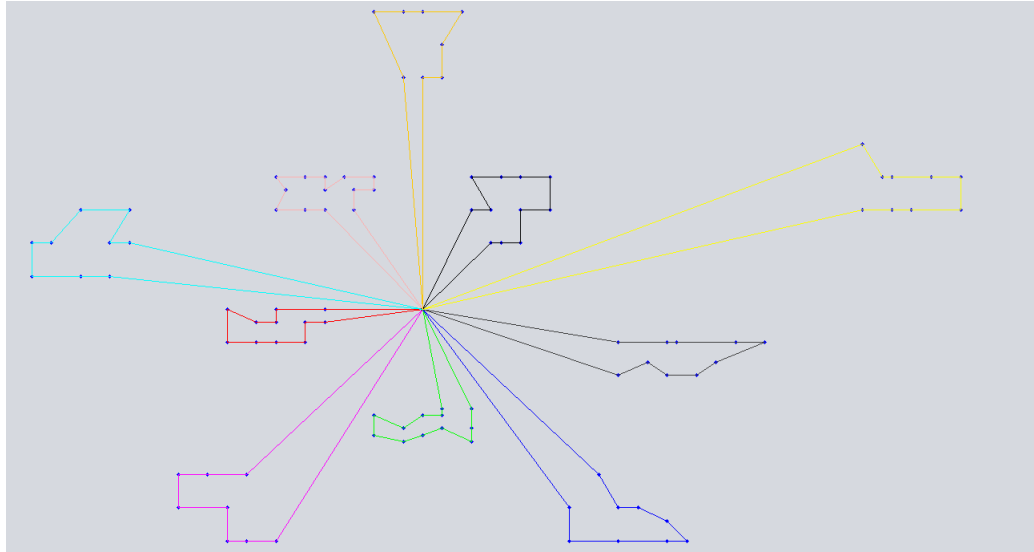


Fig. 7. Resposta final do Artio (4)

volvido um programa em Java que recebesse esse arquivos de entrada para demonstrar o andamento do código realizado pelo código responsável pelo artigo (4), para mais detalhes do código acessar ao github:

<https://github.com/Wiltoon/ProjectDVRPTW>.

Ao colocar em comparação a Figura 1, responsável pela execução final do algoritmo do artigo (4), com a Figura 2 responsável pela execução final deste artigo, é possível ver a diferença entre buscar uma ótima solução com colônia de formigas, que pode ser interpretado por um algoritmo genético com a busca de uma única solução para otimizar a rota de toda frota.

4 Conclusão

4.1 Conclusões

A parte alpha do projeto foi finalizada com êxito, como demonstrado na seção 3, e uma resposta mais eficiente pode ser demonstrada na parte beta do problema, uma vez que a parte alpha trata as demandas como estáticas, posteriormente será necessário se preocupar com a adição de novos pedidos no projeto, para cobrir a ideia de responder a todos os pedidos no dia de trabalho.

4.2 Novas perspectivas

Com os testes realizados, uma nova porta se abre, e num futuro próximo é possível pensar em realizar a junção da colônia de formigas com o K-Means,

no intuito de devolver uma possível solução mais otimizada, assim como já foi realizado em (6), onde é usado uma colônia de formiga com a aprendizagem k-means. A principal ideia será otimizar o tempo de busca inicial, que por sua vez terá uma solução mais rápida e eficiente em um curto espaço de tempo na parte dinâmica do problema

Bibliography

- [1] Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering* **99**, 300–313 (2016)
- [2] Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management science* **6**(1), 80–91 (1959)
- [3] Ghiani, G., Manni, E., Quaranta, A., Triki, C.: Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E: Logistics and Transportation Review* **45**(1), 96–106 (2009)
- [4] Necula, R., Breaban, M., Raschip, M.: Tackling dynamic vehicle routing problem with time windows by means of ant colony system. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 2480–2487. IEEE (2017)
- [5] Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E.: On the capacitated vehicle routing problem. *Mathematical programming* **94**(2-3), 343–359 (2003)
- [6] Reed, M., Yiannakou, A., Evering, R.: An ant colony algorithm for the multi-compartment vehicle routing problem. *Applied Soft Computing* **15**, 169–176 (2014)