Roteamento de veículos usando a heurística baseada em Centroide e Simulated Annealing

Juan C. E. O. Valdivia, Eduardo A. Pacheco e Lucas F. da Nóbrega
Instituto de Ciências Matemáticas e de Computação e Escola de Engenharia de São Carlos - Universidade de São Paulo (USP)
juan.valdivia@.usp.br, ea.pacheco@usp.br e lucas.fernandes.nobrega@usp.br

1. Resumo

O Problema de Roteamento de Veículos Capacitados(CVRP) é um famoso problema NP-difícil da literatura. O objetivo é atender todos os clientes gerando uma rota com custo mínimo para cada veículo que parte de um mesmo depósito. Os veículos têm capacidade uniforme e precisam ir e voltar para o depósito. Cada cliente possui uma demanda conhecida e deve receber apenas uma visita de um único veículo. Neste trabalho da disciplina de Inteligência Artificial é implementado o algoritmo heurístico baseado em Centroide para resolver o CVRP em tempo polinomial. O algoritmo é composto por três fases: construção de clusters, ajuste de clusters e estabelecimento da rota. A noção de centro geométrico guia as duas primeiras fases. Após o estabelecimento dos grupos é encontrada a rota subótima de cada cluster. Uma heurística baseada em Simulated Annealing é utilizada para encontrar a melhor rota que cada veículo deverá seguir.

2. CVRP

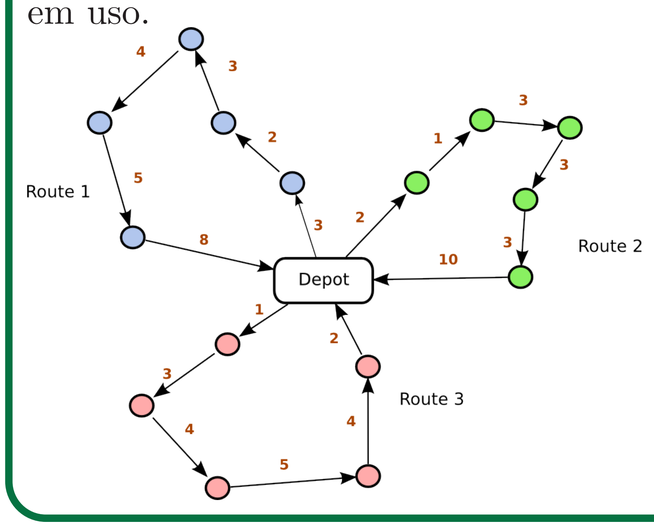
Definição: $G = (V, A)$ é um grafo não direcionado. $V = \{0, 1, ..., n\}$ é o conjunto dos nós e A é o conjunto das arestas. O nó 0 é o depósito e o conjunto $W = V - \{0\}$ de n nós corresponde aos clientes. Cada vértice $\{i, j\}$ de A possui um custo $(c_{i,j} \geq 0)$. Cada cliente i de W precisa de q_i unidades do depósito. Um conjunto de M veículos idênticos com capacidade Q está no depósito.
Solução: Uma solução factível para o problema é composta pela partição $S = \{r_1, r_2, ..., r_m\}$ de V e pela permutação i de r_i que especifica a ordem dos clientes na rota com a restrição de que a demanda total de todos os clientes suportados pela rota r_i não ultrapasse a capacidade Q do veículo. O custo da rota $r_i = \{v_0, v_1, ..., v_{k+1}\}$ é dada por:

$$C(r_i) = \sum_{i=0}^{k} c_{i,i+1} \quad (1)$$

O custo total da solução S é:

$$TC(S) = \sum_{i=0}^{m} C(r_i) \quad (2)$$

A função objetivo de custo mínimo requer que as rotas visitem todos os clientes exatamente uma vez, independentemente do número de veículos em uso.



3. Heurística baseada em Centroide

O algoritmo proposto em [1] consiste em três fases:
Construção de clusters: O grupo(cluster) de clientes é construído usando o cliente mais distante do depósito como a semente do grupo. Após isso, os clientes mais próximos do centro geométrico(centroide) do cluster são selecionados, e o centroide é atualizado quando um cliente é adicionado. O processo continua enquanto a capacidade do veículo não for ultrapassada. A construção do grupo é finalizada quando isso ocorre. Após isso uma nova semente é selecionada e o processo de construção de novos grupos continua até que cada cliente esteja agrupado em um cluster.
Ajuste de clusters: Os grupos formados na fase anterior são ajustados. Verifica-se para cada cliente se ele está mais próximo do centroide de um cluster vizinho do que o centroide de seu cluster atual. Caso isso ocorra e a capacidade do grupo vizinho não seja ultrapassada, esse cliente vai para o cluster vizinho e os centroides de ambos grupos são recalculados.
Estabelecimento da rota: A rota subótima é encontrada visitando cada cliente apenas uma única vez dentro do cluster. Isso significa aplicar um algoritmo para resolver o *problema do caixeiro viajante (TSP)* usando a meta-heurística do Recozimento Simulado (Simulated Annealing) para cada grupo encontrado. Em cada cluster i, o cliente v_j faz parte do conjunto $l_i = \{v_0, v_1, ..., v_k\}$ de clientes que estão no cluster i. O centro geométrico (GC) do cluster l_i é definido por: $GC(l_i) = (\sum_{j=0}^{k} \frac{x_j}{k}, \sum_{j=0}^{k} \frac{y_j}{k})$, em que x_j e y_j são as coordenadas de v_j.

4. Simulated Annealing para o TSP

O algoritmo heurístico Simulated Annealing foi usado para resolver o *TSP*. Nesta heurística, a melhor solução é procurada na vizinhança da solução atual. Soluções vizinhas são geradas por meio da troca de dois clientes aleatórios escolhidos de maneira aleatória na rota. Quando uma solução vizinha gerada for melhor, ela será a nova solução atual. Por outro lado, se ela for pior, ela só será aceita de acordo com o valor de uma probabilidade P cuja equação é:

$$P = e^{\frac{\Delta E}{Temperatura}}, \Delta E = custo - novoCusto \quad (3)$$

Essa solução vizinha pior é aceita se $P > random(0, 1)$.
Em cada iteração do algoritmo a temperatura é diminuída gradualmente usando um fator de esfriamento. Já que a temperatura é reduzida, a seleção de uma solução vizinha pior se torna cada vez mais difícil.

5. Resultados

O algoritmo foi testado no dataset de Augerat A, B e P. As instâncias do tipo A a localização dos clientes e suas demandas são geradas aleatoriamente; as instâncias da classe B são criadas por meio de clusters; e as instância do tipo P são versões modificadas de instâncias da literatura. O nome do arquivo A-n32-k5 indica que é um problema do tipo A, com 32 nós (31 clientes e 1 depósito) e com no mínimo 5 veículos necessários. Os experimentos foram realizados no sistema operacional Windows 10 no processador Intel Core i7 8550U 1.8 GHz e 16 GB de RAM. O algoritmo foi implementado na linguagem Java no Eclipse.

Instância	Melhor custo conhecido	Custo	Número Rotas	Tempo de execução (ms)
A-n32-k5	784	882.21	5	21
A-n33-k5	661	729.87	5	22
A-n33-k6	742	787.54	7	22
A-n45-k7	1146	1291.77	7	32
B-n31-k5	672	704.25	5	47
B-n34-k5	788	854.19	6	51
B-n35-k5	955	973.68	5	31
B-n38-k6	805	836.82	6	35
B-n44-k7	909	967.40	7	22
B-n66-k9	1374	1460.70	10	74
P-n16-k8	435	498.51	9	7
P-n19-k2	212	257.7	3	24
P-n20-k2	220	240.93	2	17
P-n23-k8	554	705.07	12	9
P-n40-k5	458	512.61	5	36
P-n50-k10	696	756.21	11	28

6. Conclusões

Foi implementado o algoritmo baseado em Centroide proposto em [1] e para o passo de estabelecimento de rota foi usado o algoritmo Simulated Annealing.Os experimentos mostram que o algortimo consegue resolver problemas de até 66 clientes em um tempo menor que 74 milisegundos.

7.Referências

[1] Kwangcheol Shin and Sangyong Han. A centroid-based heuristic algorithm for the capacitated vehicle routing problem. *Computing and Informatics*, 30:721–732, 2011.