



Heurística de Busca de Vizinhança Variável para Otimização do Problema de Roteamento de Veículos Capacitados

Pedro Paulo M. Silva e Leandro C. Resendo

Sumário

- Introdução
- Proposta do Trabalho
- Busca em Vizinhança Variável
- Vizinhanças
- Instâncias
- Calibração
- Resultados
- Conclusões

Introdução

Custos Logísticos

O transporte é uma das atividades que mais contribui nos custos logísticos e qualquer redução nos custos de transporte poderá acarretar em uma economia significativa para empresas e consumidores.

- Consumidores finais chegam a pagar de 10% a 15% do valor da mercadoria com transporte do mesmo.
- O gasto com transporte de carga corresponde a 12,7% do PIB (Produto Interno Bruto).

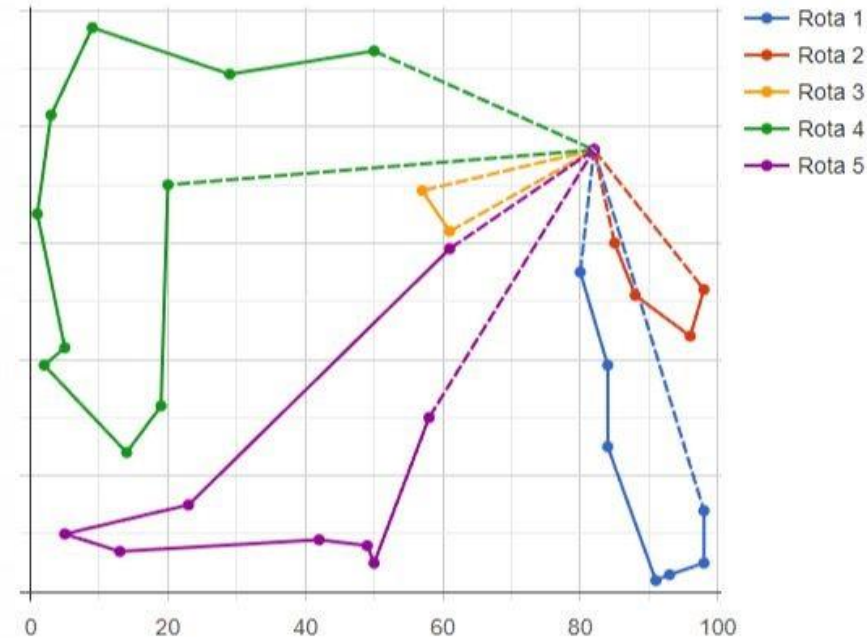
Introdução

Melhorias nos custos

- Existe uma real necessidade de melhorar esses sistemas em termos de custo e eficiência.
- Desenvolvimento de metodologias que auxiliem o planejamento e utilização da frota, fornecendo informações sobre as melhores rotas.
- Minimização de custos por meio de Otimização Combinatória.
- Este problema de minimização dos custos de transporte é modelado pelo Problema do Roteamento de Veículos (PRV).

Descrição do Problema

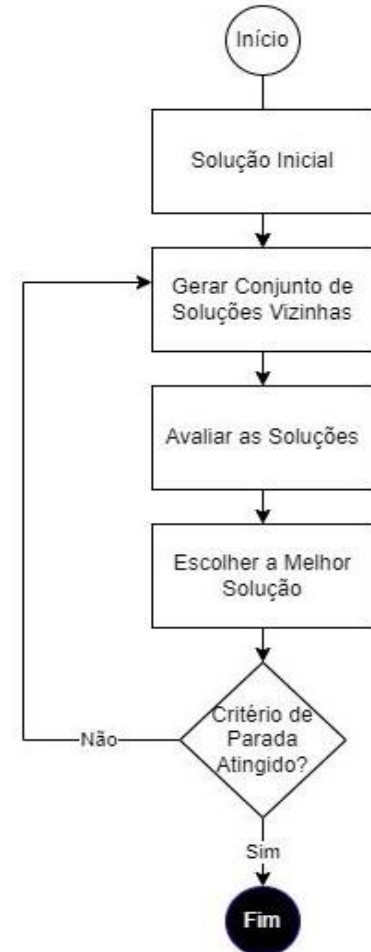
- Em resumo, este trabalho busca desenvolver um método heurístico para encontrar boas soluções para o Problema do Roteamento de Veículos Capacitados.
- Nessa situação, a soma das demandas dos clientes da rota de um veículo, não podem ultrapassar a capacidade total do veículo.
- O número de rotas não deve exceder o número de veículos disponíveis,
- As capacidades de cada veículo não podem ser extrapoladas,
- Toda rota começa e termina no depósito,
- Cada cliente é visitado apenas uma vez e por apenas um veículo.



Busca em Vizinhança Variável

Variable Neighborhood Search (VNS)

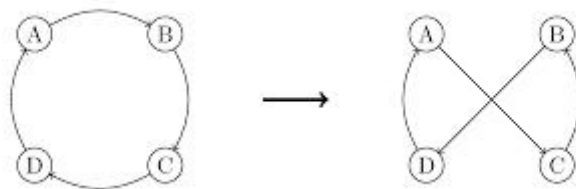
- A meta-heurística Variable Neighborhood Search (VNS) que busca melhorar as rotas incrementalmente procurando melhorias nas vizinhanças.
- Partindo de uma solução inicial, deve-se realizar trocas de estruturas de vizinhanças para explorar um novo conjunto de soluções e analisar seus resultados.
- Neste trabalho foram desenvolvidas 5 vizinhanças para o algoritmo de VNS.
- Vizinhanças: 2 de trocas internas em um veículo, 2 de trocas entre 2 veículos e 1 com um fator aleatório.



Vizinhanças

2-Opt dos Clientes de um Veículo

- Esta vizinhança busca encontrar a melhor rota possível de um veículo apenas fazendo a mudança na ordem de visita dos clientes pelo veículo.
- São realizadas todas as trocas de pares de nós adjacentes.



Exemplo 2-Opt

Troca	Rota	Custo
Rota Inicial	[0, 9, 7, 6, 14, 0]	253.8
1	[0, 7, 9, 6, 14, 0]	231.9
2	[0, 6, 7, 9, 14, 0]	250.7
3	[0, 14, 6, 7, 9, 0]	253.8
4	[0, 9, 6, 7, 14, 0]	229.8
5	[0, 9, 14, 6, 7, 0]	244.1
6	[0, 9, 7, 14, 6, 0]	281.6

Vizinhanças

Circ-Opt dos Clientes de um Veículo

- A troca da ordem de atendimento dos clientes será feita como uma lista circular, o último cliente passará a ser o primeiro e o restante terá a posição do sequenciamento adicionado de uma posição.
- Ex: O veículo V1 possui os seguintes clientes C1, C2, C3, C4, após o algoritmo de Circ-opt os veículos ficam da seguinte maneira: V1[C4, C1, C2, C3].

Troca	Rota	Custo
Rota Inicial	[0, 6, 14, 9, 7, 10, 0]	263.0
1	[0, 10, 6, 14, 9, 7, 0]	225.3
3	[0, 7, 10, 6, 14, 9, 0]	246.
3	[0, 9, 7, 10, 6, 14, 0]	233.6
3	[0, 14, 9, 7, 10, 6, 0]	337.5

Vizinhanças

Troca de Clientes entre Veículos

- Esta vizinhança busca melhorar o custo total trocando os clientes de dois veículos.
- Cada cliente de um veículo é testado em todas as posições do outro veículo.

Troca	Rota	Custo
Rota Inicial	[[0, 9, 7, 6, 0], [0, 12, 13, 1, 0]]	366.1
1	[[0, 12, 7, 6, 0], [0, 9, 13, 1, 0]]	356.0
2	[[0, 13, 7, 6, 0], [0, 12, 9, 1, 0]]	399.2
3	[[0, 1, 7, 6, 0], [0, 12, 13, 9, 0]]	357.9
4	[[0, 9, 12, 6, 0], [0, 7, 13, 1, 0]]	391.0
5	[[0, 9, 13, 6, 0], [0, 12, 7, 1, 0]]	345.5
6	[[0, 9, 1, 6, 0], [0, 12, 13, 7, 0]]	384.9
7	[[0, 9, 7, 12, 0], [0, 6, 13, 1, 0]]	353.8
8	[[0, 9, 7, 13, 0], [0, 12, 6, 1, 0]]	385.1
9	[[0, 9, 7, 1, 0], [0, 12, 13, 6, 0]]	355.6

Vizinhanças

Realocação de Clientes entre Veículos

- Esta vizinhança busca melhorar o custo total realocando o cliente de um veículo em todas as posições de sequenciamento do outro veículo, e vice-versa.
- Semelhante ao anterior, porém aqui são testados os clientes dos dois conjuntos.

Realocação	Rota	Custo
Rota Inicial	[[0, 9, 7, 6, 0], [0, 12, 13, 1, 0]]	366.1
1	[[0, 7, 6, 0], [0, 12, 9, 13, 1, 0]]	382.7
2	[[0, 7, 6, 0], [0, 12, 13, 9, 1, 0]]	387.9
3	[[0, 7, 6, 0], [0, 12, 13, 1, 9, 0]]	380.4
4	[[0, 9, 6, 0], [0, 12, 7, 13, 1, 0]]	316.0
5	[[0, 9, 6, 0], [0, 12, 13, 7, 1, 0]]	317.9
6	[[0, 9, 6, 0], [0, 12, 13, 1, 7, 0]]	328.5
7	[[0, 9, 7, 0], [0, 12, 6, 13, 1, 0]]	365.5
8	[[0, 9, 7, 0], [0, 12, 13, 6, 1, 0]]	368.3
9	[[0, 9, 7, 0], [0, 12, 13, 1, 6, 0]]	381.3
10	[[0, 12, 13, 1, 0], [0, 9, 7, 6, 0]]	366.1
11	[[0, 13, 1, 0], [0, 9, 12, 7, 6, 0]]	391.9
12	[[0, 13, 1, 0], [0, 9, 7, 12, 6, 0]]	395.2
13	[[0, 13, 1, 0], [0, 9, 7, 6, 12, 0]]	380.5
14	[[0, 12, 1, 0], [0, 9, 13, 7, 6, 0]]	336.7
15	[[0, 12, 1, 0], [0, 9, 7, 13, 6, 0]]	334.5
16	[[0, 12, 1, 0], [0, 9, 7, 6, 13, 0]]	347.1
17	[[0, 12, 13, 0], [0, 9, 1, 7, 6, 0]]	384.9
18	[[0, 12, 13, 0], [0, 9, 7, 1, 6, 0]]	385.8
19	[[0, 12, 13, 0], [0, 9, 7, 6, 1, 0]]	383.3

Vizinhanças

Recombinação Aleatória de Veículos

- Esta vizinhança tem o intuito de adicionar aleatoriedade ao algoritmo, permitindo que fossem encontradas soluções que não seriam encontradas de modo determinístico.
- Esta vizinhança utiliza o conceito de Recombinação da Computação Evolucionária
- A vizinhança busca melhorar o custo total fazendo a recombinação dos clientes de dois veículos, sorteados aleatoriamente.
- Na prática, são selecionados dois veículos aleatoriamente, após isso um cliente de cada veículo é selecionado aleatoriamente para troca de posição.

Ex:

O veículo V1 possui os clientes: C1, C2, C3, C4.

O veículo V2 possui os clientes C5, C6, C7, C8

Instâncias

Fonte: <http://vrp.galgos.inf.puc-rio.br/index.php/en/>

- Todas as instâncias são homogêneas.
- Capacidade dos veículos e demanda.
- 24 instâncias, divididas em 4 grupos.

Instâncias

- O G1 busca validar a eficiência do algoritmo para casos com uma pequena diferença na quantidade de clientes e o mesmo número de veículos.
- O G2 valida a eficiência do algoritmo para casos com uma diferença maior na quantidade de clientes e com o mesmo número de veículos
- O G3, diferentemente do G1 e G2, busca avaliar casos com o mesmo número de clientes, mas com diferentes frotas de veículos.

Grupo	Número de Clientes	Número de Veículos
G1	[32, 33, 34, 36, 37, 38, 39, 45, 51]	5
G2	[16, 23, 50, 62, 76, 101]	8
G3	101	[4, 8, 10, 14]

- O G4 procura validar a escalabilidade do método com instâncias com muitos clientes e veículos.

Número de Clientes	Número de Veículos
101	10
121	7
151	12
200	16
200	17

Calibração

- Importante para encontrar os melhores parâmetros de uma heurística.
- Número de Iterações: 10, 20, 30, 50, 100, 500, 1000.
- Também foi avaliado o sequenciamento de uso das vizinhanças. Porém, não foi observado variação dos resultados.

Resultados: G1 - G3

- G1 - G2 - Diferença de aproximadamente 7% dos resultados ótimos.
- G3 – Para a instância P-n101-k4, com 101 clientes e 4 veículos, a heurística levou aproximadamente 30 minutos para retornar o resultado encontrado. Porém, obteve um resultado aproximadamente 3% do ótimo.
- G3 - Diferença media de aproximadamente 6% dos resultados ótimos.

Grupo	Instância	Capacidade	Custo Ótimo	Custo Encontrado	Tempo (segundos)
G1	A-n36-k5	100	799	875,17	10,48
	E-n51-k5	160	521	577,3	37,25
	B-n45-k5	100	751	842,66	22,71
	A-n32-k5	100	784	836,06	8,36
	A-n37-k5	100	669	734,44	15,55
	A-n34-k5	100	778	796,16	7,46
	A-n39-k5	100	822	865,64	13,3
	A-n38-k5	100	730	763,95	11,32
	A-n33-k5	100	661	676,1	7,08
G2	A-n62-k8	100	1288	1410,94	23,85
	P-n16-k8	35	450	452,94	0,78
	P-n23-k8	40	529	537,55	1,71
	E-n76-k8	180	735	748,32	40,46
	E-n101-k8	200	817	870,79	133,08
	P-n50-k8	120	631	778,65	9,17
G3	P-n101-k4	400	681	700,82	1933,38
	E-n101-k14	112	1071	1166,02	35,01
	M-n101-k10	200	820	883,24	64,71
	E-n101-k8	200	817	863,95	112,37

Resultados: G4

- Diferença entre o custo encontrado e o custo ótimo de 13%.
- Esse resultado foi um pouco maior que os outros grupos, porém devido ao tamanho das instâncias ainda pode ser considerado relativamente baixo e com o tempo de execução razoável para uso em sistemas reais.

Grupo	Instância	Capacidade	Custo Ótimo	Custo Encontrado	Tempo (segundos)
G4	M-n151-k12	200	1053	1152,71	148,32
	M-n200-k17	200	1373	1425,96	180,07
	M-n121-k7	200	1034	1168,67	264,79
	M-n101-k10	200	820	926,4	61,17
	M-n200-k16	200	1274	1623,36	175,09

Conclusões

- Os resultados mostraram que as 5 vizinhanças propostas foram suficientes para encontrar boas soluções, com os custos apenas 8% maiores que os custos ótimos das instâncias na média.
- O tempo de execução da heurística também se mostrou bastante razoável para o tamanho das instâncias.
- Uma limitação encontrada foi a de instâncias com muitos clientes e poucos veículos. Para esses casos o tempo de execução foi significativamente mais alto que a média, entretanto os custos para esses casos se manteve bem próximo do custo ótimo.
- No geral os resultados mostram que a heurística obteve resultado bastante satisfatório e sua implementação poderia ser aplicada para cenários reais.

Perguntas?

Obrigado!