FORMULAÇÕES DE PROGRAMAÇÃO LINEAR INTEIRA PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS CUMULATIVO

Lorenzo Henrique Zanetti (PET-Informática/UEM), Mauro Henrique Mulati (Orientador). E-mail: mhmulati@uem.br

Universidade Estadual de Maringá, Centro de Tecnologia, Maringá, PR.

Área e subárea do conhecimento: Ciência da Computação / Teoria da Computação.

Palavras-chave: Otimização Combinatória; Branch-and-Bound; Relaxação Linear.

RESUMO

O Problema de Roteamento de Veículos Cumulativo (CmVRP) é uma variante do roteamento de veículos em que o custo de percorrer um arco é o produto da distância pelo peso de transporte acumulado, ao invés da distância clássica, tornando o modelo mais adequado para representar um consumo energético aproximado. Este trabalho avalia computacionalmente duas formulações de Programação Linear Inteira para o CmVRP: a Arco-Carga (AL), que discretiza a carga em níveis por arco, e a Arco-Item (AI), que rastreia individualmente cada coleta. As implementações das formulações foram feitas em linguagem Julia e o *solver* CPLEX e avaliadas sobre 16 instâncias de benchmark da CVRPLIB, com limite de uma hora por execução. A AL atingiu o critério de parada por gap (*gap* ≤ 0,01%) em 5 instâncias e encontrou solução viável em 14 delas, com gap médio de 5,33%. A AI atingiu o mesmo critério em 4 instâncias, mas não encontrou nenhuma solução viável em 9 das 12 instâncias não resolvidas. Já na única instância com capacidade muito acima do padrão das demais, a AI resolveu o problema na otimalidade do solver, enquanto a AL não obteve sequer um limitante válido, devido ao grande número de variáveis de carga. Os resultados sugerem que a AL encontra soluções viáveis com mais facilidade em instâncias de porte usual, enquanto a AI é mais robusta em instâncias com capacidade desproporcional ao número de clientes.

INTRODUÇÃO

Os Problemas de Roteamento de Veículos (VRPs) estão entre os problemas de otimização combinatória mais estudados em pesquisa operacional, com ampla aplicação em logística de transporte e distribuição. Tradicionalmente, a função objetivo desses problemas minimiza a distância total percorrida pela frota, o que

nem sempre reflete adequadamente os custos operacionais reais, sobretudo em cenários nos quais o consumo de combustível ou energia depende diretamente da carga transportada pelo veículo ao longo da rota (TOTH; VIGO, 2014).
O Problema de Roteamento de Veículos Cumulativo (CmVRP) aborda essa limitação ao definir o custo de cada arco como o produto da distância pelo peso acumulado do veículo naquele trecho, soma do peso próprio do veículo com a carga já coletada dos clientes visitados até então (KARA; KARA; YETIŞ, 2008). Esse modelo captura de forma aproximada o consumo energético de operações de coleta, sendo particularmente relevante em cenários com restrições severas de autonomia, como veículos elétricos em rotas de coleta seletiva de resíduos ou drones utilizados na coleta de amostras em campo.
Diferentemente do CVRP clássico, no qual o custo de cada arco depende apenas da distância, no CmVRP focado em coleta o custo cresce progressivamente à medida que o veículo avança e acumula carga. O arco inicial saindo do depósito não transporta nenhuma demanda de clientes (apenas o peso próprio do veículo), enquanto o último arco de retorno carrega a carga máxima coletada na rota. Essa característica faz com que a ordem de visita influencie diretamente o custo total, mesmo quando a distância percorrida é idêntica: visitar primeiro os clientes de menor demanda tende a reduzir o custo acumulado, pois garante que o veículo trafegue o mais leve possível durante a maior parte do trajeto.
O CmVRP é um problema NP-difícil, e a eficiência dos métodos exatos de resolução baseados em Programação Linear Inteira (PLI) depende fortemente da qualidade da formulação matemática utilizada: formulações mais fortes produzem relaxações lineares mais próximas do valor ótimo inteiro, reduzindo o espaço de busca explorado por algoritmos de branch-and-bound. Embora diversas formulações para o CmVRP tenham sido propostas, poucos trabalhos comparam sistematicamente seu desempenho computacional sob as mesmas condições experimentais. Diante disso, este trabalho tem como objetivo implementar e comparar computacionalmente duas formulações de PLI para o CmVRP disponíveis na literatura, avaliando sua qualidade de relaxação linear e desempenho na resolução exata de instâncias de benchmark.

MATERIAIS E MÉTODOS

Foram implementadas duas formulações de PLI para o CmVRP. A formulação Arco-Carga (AL), proposta por Fukasawa, He e Song (2016), associa a cada arco variáveis binárias correspondentes aos possíveis níveis de carga discretos do veículo, indicando se o trajeto é percorrido carregando exatamente aquela quantidade de peso. A Arco-Item (AI) usa variáveis por item de demanda, indicando se foi coletado e está no trecho, além de um item-guia que indica o uso do arco independentemente da carga.
A literatura demonstra que AL e AI não dominam uma à outra: suas relaxações lineares são incomparáveis, cada uma capturando uma estrutura diferente do problema: carga acumulada, no caso de AL, e identidade do item transportado, no caso de AI (MULATI; FUKASAWA; MIYAZAWA, 2022).

As formulações foram implementadas na linguagem Julia, com o framework JuMP e o solver CPLEX (versão 22.2). Os experimentos foram conduzidos sobre 16 instâncias das Classes A, B, E e P da CVRPLIB (20 a 76 clientes). O peso próprio do veículo foi definido como o piso de 15% da sua capacidade, e o número de veículos disponíveis foi fixado conforme especificado em cada instância. Cada execução foi limitada a uma hora, mantendo-se a tolerância de *gap* de otimalidade padrão do solver em 0,01%. Para cada formulação e instância, foram registrados a relaxação linear, o valor incumbente, o gap de otimalidade final e o tempo de resolução.

RESULTADOS E DISCUSSÃO

A Tabela 1 apresenta os resultados obtidos para as formulações AL e AI em todas as 16 instâncias planejadas. A AL encerrou a busca pelo critério de *gap* otimalidade (*gap* ≤ 0,01%) em 5 instâncias e manteve *gaps* abaixo de 12% em 6 das 9 instâncias restantes com solução viável encontrada, não obtendo nenhum limitante válido apenas em E-n23-k3 e E-n76-k8. A AI atingiu a tolerância de otimalidade em 4 instâncias, incluindo a E-n31-k7 e a E-n23-k3 (na qual a AL falhou), mas não encontrou nenhuma solução viável dentro do limite de tempo em 9 das 12 instâncias restantes.

O caso de E-n23-k3 ilustra essa diferença: com capacidade Q=4500 (muito acima do padrão Q ≈ 100-200), a AL cria uma variável por unidade de capacidade, expandindo para mais de 1,7 milhão e impedindo até o cálculo de um limitante. Já a AI, cujo número de variáveis escala com o número de clientes e não com a capacidade, resolveu essa mesma instância à otimalidade com apenas 11 mil variáveis. Nas instâncias de porte usual, porém, a AI encontrou soluções viáveis com bem menos frequência que a AL, sugerindo que sua estrutura é mais difícil de explorar heurísticamente pelo *solver*.

Uma análise das médias consolida a vantagem da AL nos casos de porte usual. Considerando todas as instâncias em que cada formulação encontrou ao menos uma solução viável, a AL registrou um gap médio de 5,33% (em 14 instâncias) e a AI, de 7,06% (em 7 instâncias). No entanto, uma comparação mais interessante ocorre no subconjunto de 6 instâncias solucionadas com sucesso por ambas as abordagens simultaneamente. Nesse cenário, a AL demonstrou uma melhor performance, obtendo um gap médio de apenas 2,53% em 1495 segundos de processamento, enquanto a AI resultou em um gap médio de 8,23% exigindo, em média, 2102 segundos.

Tabela 1: Resultados obtidos para as formulações AL e AI

Classe	Instância	Gap AL (%)	Tempo AL (s)	Gap AI (%)	Tempo AI (s)
A	A-n33-k6	0,00 (ótimo)	247	0,01	755
A	A-n38-k5	7,10	3600	sem solução	3600
A	A-n45-k7	0,74	3600	sem solução	3600
A	A-n55-k9	5,28	3600	12,86	3600
A	A-n63-k10	11,50	3600	sem solução	3600
B	B-n35-k5	9,86	3600	9,94	3602
B	B-n44-k7	3,83	3601	sem solução	3600
B	B-n51-k7	20,62	3600	sem solução	3600
B	B-n63-k10	15,42	3600	sem solução	3605
E	E-n23-k3	sem limitante	3602	0,01	215
E	E-n31-k7	0,00 (ótimo)	31	0,01	85
E	E-n76-k8	sem limitante	3606	sem solução	3611
P	P-n20-k2	0,01	387	0,01	967
P	P-n40-k5	0,00 (ótimo)	1104	26,55	3601
P	P-n51-k10	0,01	1990	sem solução	3602
P	P-n60-k10	0,22	3600	sem solução	3600
Média geral	-	5,33	2583	7,06	1832
Média Comparativa	-	2,53	1495	8,23	2102

Gaps de 0,01% indicam que o solver atingiu sua tolerância padrão, encerrando a busca antes da prova do ótimo absoluto. "Média geral" considera os resultados apenas das instâncias onde cada formulação encontrou solução (14 para AL, 7 para AI). "Média comparativa" considera apenas as 6 instâncias solucionadas simultaneamente por ambas as formulações).

CONCLUSÕES

Os resultados indicam que AL e AI têm comportamentos complementares: a AL encontra soluções viáveis com mais facilidade na maioria das instâncias, mas falha quando a capacidade é desproporcional ao número de clientes, justamente onde a AI se destaca. Próximos passos incluem estender os testes para o restante das instâncias de referência a fim de aprofundar a comparação de desempenho.

AGRADECIMENTOS

Ao Programa de Educação Tutorial em Informática (PET-Informática/UEM), pela bolsa concedida ao primeiro autor, que viabilizou a dedicação a este trabalho.

REFERÊNCIAS

FUKASAWA, R.; HE, Q.; SONG, Y. A Branch-Cut-and-Price Algorithm for the Energy Minimization Vehicle Routing Problem. **Transportation Science**, v. 50, n. 1, p. 23-34, 2016.

KARA, İ.; KARA, B. Y.; YETIŞ, M. K. Cumulative Vehicle Routing Problems. In: **Vehicle Routing Problem**. Rijeka: IntechOpen, 2008. cap. 6, p. 85-98.

MULATI, M. H.; FUKASAWA, R.; MIYAZAWA, F. K. The arc-item-load and related formulations for the cumulative vehicle routing problem. **Discrete Optimization**, v. 45, p. 100710, 2022.

TOTH, P.; VIGO, D. (org.). **Vehicle Routing**: Problems, Methods, and Applications. 2. ed. Philadelphia: SIAM, 2014.