Caixeiro viajante com janelas de tempo aplicado para a minimização de atraso de chegada em N localidades

Breno C. Zukowski[1], Henrique Ribeiro dos Santos[1], Jean Luca dos Santos Silva[1], Paola Paulina D. J. S. Capita[1]

[1]Faculdade de Tecnologia de Ribeirão Preto - (FATEC)
Ribeirão Preto, SP – Brasil

breno.marques@fatec.sp.gov.br, henrique.santos54@fatec.sp.gov.br, jean.silva88@fatec.sp.gov.br,paola.capita@fatec.sp.gov.br

Abstract. *The traveling salesman problem is a classic of mathematical literature, where a salesperson must visit a set of N cities once and return to his point of origin, through a route that minimizes the traveled distance. This work proposes a solution with a mathematical programming model, for a variant of the original problem that has time limits for arrival at each city, in which delays are allowed and the objective is to minimize the overall delay of the route. Using the PyMathProg library we achieved interesting results, offering plausible routes to the points offered.*

Resumo. *O problema do caixeiro viajante é um clássico da literatura matemática, onde um vendedor deve visitar um conjunto de N cidades uma única vez e retornar ao seu ponto de origem, através de uma rota que minimiza a distância percorrida. Este trabalho propõe uma solução via modelo de programação matemática, para uma variante do problema original que conta com tempos limite para chegada a cada cidade, em que atrasos são permitidos e têm-se como objetivo minimizar o atraso geral da rota. Com a utilização da biblioteca PyMathProg atingimos resultados interessantes, oferecendo rotas plausíveis para os pontos oferecidos.*

1. Introdução

Os problemas de roteirização e otimização de tempo são grandes conhecidos do cotidiano, não é incomum que uma rota tenha diversas possibilidades de percurso. Este tipo de problema é explorado há séculos, sendo um dos seus exemplos mais famosos o Problema do Caixeiro Viajante (PCV), problema este trabalhado desde o século XIX por diversos matemáticos. Com o advento da Pesquisa Operacional (PO), este tipo de situação começou a tomar novas proporções através da utilização de modelos matemáticos e programação de computadores para obter soluções eficientes para diversas variações do PCV.

A PO é a área do conhecimento dedicada a estudar e aplicar métodos analíticos para a resolução de problemas nas mais diversas áreas de atuação humana (SOBRAPO 2017). A partir dela somos capazes de encontrar soluções ótimas com limitações de recursos e restrições, modelando matematicamente a realidade para obter resultados factíveis para a resolução de diversas situações (Hillier e Lieberman 2013).

Neste trabalho exploramos Problema do Caixeiro Viajante com tempo limite de chegada utilizando uma abordagem desenvolvida em aula que será descrita em detalhes na próxima sessão.

2. Descrição do problema

O problema do caixeiro viajante é um clássico da otimização combinatória, que propõe que um determinado caixeiro deverá visitar *n* cidades diferentes, iniciando e encerrando sua viagem na primeira cidade. A ordem das cidades visitadas é irrelevante e de cada uma delas pode-se ir diretamente a qualquer outra (Silveira 2000). Trata-se de um problema de complexidade $R(n) = (n-1)!$ e pertence a classe dos NP-completos. Até o presente momento este é um problema aberto da matemática, sem resolução polinomial descoberta.

Tendo em vista essas informações, existem também variações do PCV e a partir de uma destas desenvolvemos o seguinte trabalho. Dado o enunciado temos:

(a) Um veículo que deve partir do ponto inicial, visitará *n* localidades e retornará ao ponto de partida após as visitas.
(b) Cada localidade a ser visitada tem um prazo limite para receber a visita.
(c) Atrasos são permitidos, porém uma multa que aumenta com o tempo de atraso é imposta.
(d) A distância euclidiana (em linha reta) entre as localidades é contabilizada como tempo de percurso.

O modelo deverá minimizar o atraso total das visitas, eliminando sub-rotas e garantindo que as restrições sejam devidamente atendidas.

3. Modelagem Matemática

3.1. Conjunto de Nós

Representativamente podemos definir um conjunto de valores N como visto em (1) para todos os nós n, sendo que 0 e $n+1$ representam o ponto de origem que também deverá ser o último nó a ser visitado ao fim do percurso.

$$N = \{0, ..., n+1\} \tag{1}$$

Concomitantemente temos o produto cartesiano destes nós dado por:

$$A = N * N \tag{2}$$

3.2. Função Objetivo:

A função objetivo abaixo mostra a minimização do somatório do atraso nos nós, representado pela variável w_j:

$$min \sum_{j \in N \setminus \{0, n+1\}} w_j \tag{3}$$

3.3. Restrições:

A restrição (4) mostra o somatório entre todas as variáveis x_{ij} que necessariamente terá que equivaler a 1, definindo que apenas um arco é utilizado nessa roteirização, fixando os pontos de chegada. A expressão (5) tem função similar, por sua vez, restringindo os pontos de saída. Estas restrições garantem que os nós tenham pontos de chegada e partida únicos.

$$\sum_{i \in N \setminus \{n+1\}} x_{ij} = 1 \qquad j \in N \setminus \{0\} \tag{4}$$

$$\sum_{j \in N \setminus \{0\}} x_{ij} = 1 \qquad i \in N \setminus \{n+1\} \tag{5}$$

Abaixo podemos ver a restrição (6) que garante que não ocorram sub-rotas. A variável φ representa o instante de início do serviço em um nó $i \in N$. Caso o nó x_{ij} não seja utilizado, a constante M_{ij} que é suficientemente grande, garante que toda a expressão assuma um valor negativo, portanto φ_j poderá assumir qualquer valor.

$$\varphi_j \geq \varphi_i + (S_i + t_{ij})x_{ij} - M_{ij}(1 - x_{ij}) \qquad i \in N \setminus \{n+1\} \qquad j \in N \setminus \{0\} \tag{6}$$

A restrição (7) garante que o acúmulo de atraso w_j seja maior que o tempo de chegada de um nó $i \in N$ subtraído da *deadline*. Nesta restrição não estão inclusos os nós 0 e $n+1$ já que estes são considerados depósitos e não possuem *deadline*.

$$w_j \geq \varphi_j - D_j \qquad j \in N \setminus \{0, n+1\} \tag{7}$$

A restrição (8) abaixo garante e não-negatividade de w_j

$$w_j \geq 0 \qquad j \in N \setminus \{0, n+1\} \tag{8}$$

Por fim, a restrição (9) declara o domínio de valores possíveis para a variável x_{ij}.

$$x_{ij} \in \{0, 1\} \qquad i \in N \setminus \{n+1\} \qquad j \in N \setminus \{0\} \tag{9}$$

3.4. Variáveis:

w_j Tempo de atraso para cada nó de chegada j.

$j \in N \setminus \{0, n+1\}$

x_{ij} Variável binária que define se o arco é utilizado na rota.

$i \in N \setminus \{n+1\} \qquad j \in N \setminus \{0, n+1\}$

φ Variável de decisão contínua que representa o instante de inicio do serviço de um nó $i \in N$.

$(i, j) \in A$

D_j *Deadline* de chegada para um nó j.

$j \in N \backslash \{0, n+1\}$

t_{ij} Distância euclidiana entre um nó e outro.

$i \in N \backslash \{n+1\}$ $\quad$ $j \in N \backslash \{0, n+1\}$

S_i Tempo de serviço em um nó.

$i \in N \backslash \{n+1\}$

M Valor suficientemente grande.

4. Resultados Computacionais

O modelo (3)-(9) foi codificado na linguagem de programação Python em conjunto com a biblioteca PyMathProg, que possibilita a elaboração e solução de problemas de programação matemática, e com a biblioteca MatplotLib que permite a criação de visualizações estáticas, animadas e interativas.(https://matplotlib.org/) A biblioteca PyMathProg faz uso do solver GLPK (GNU Linear Programming Kit) que tem a função de resolver problemas de programação linear em larga escala (LP), programação inteira mista (MIP) e outros problemas relacionados. (https://www.gnu.org/software/glpk/).

Para realizar os testes com o modelo matemático elaborado, utilizamos 4 instâncias com dados fictícios, incluindo as coordenadas (x e y), o tempo de serviço e o deadline (tempo máximo de chegada) para cada nó. Cada instância tem, respectivamente, 10, 15, 20 e 25 nós.

Tabela 1. Resultados computacionais.

Instância	Tempo decorrido	Valor ótimo	Gap	Lim. inferior	Lim. superior
inst_10	0	0	0	0	0
inst_15	0	0	0	0	0
inst_20	0	0	0	0	0
inst_25	0	0	0	0	0

5. Considerações Finais

Referências

HILLIER, F.; LIEBERMAN, G. *Introdução a Pesquisa Operacional*. AMGH, 2013. ISBN 9788580551198. Disponível em: <https://books.google.com.br/books?id=-A88a0-KxQ0C>.

SILVEIRA, J. F. Porto da. *Problema do Caixeiro Viajante*. UFRGS, 2000. Acessado em 02 de Jun. de 2022. Disponível em: <http://www.mat.ufrgs.br/ portosil/caixeiro.html>.

SOBRAPO. *O que é pesquisa operacional?* SOBRAPO, 2017. Acessado em 26 de Mai. de 2022. Disponível em: <https://www.sobrapo.org.br/o-que-e-pesquisa-operacional>.