2. REFERENCIAL TEÓRICO

2.1 PROBLEMA DO CAIXEIRO VIAJANTE (*TSP – TAVELLING SALESMAN PROBLEM*)

A origem do problema do Caixeiro Viajante consiste em um modelo de otimização focado no melhor trajeto possível para um vendedor atender n diferentes cidades. A proposta é que o vendedor saia de uma cidade ou base específica e passe exatamente uma vez por cada cidade a ser atendida, ao fim retornando para a base original completando o ciclo.

O problema pode ser descrito através de grafos, orientados ou não orientados, em que cada cidade é um nó e cada trajeto possível é uma aresta. A solução ótima será encontrada a partir da seleção das arestas que completem todo o ciclo de atendimento e seus pesos somados sejam o menor possível. Dessa forma, categoriza-se como um problema clássico de roteamento de nós.

Em se tratando de um problema de seleção de arestas, a modelagem do problema é descrita através de um problema de programação inteira, definindo de maneira principal as variáveis . A variável além de inteira é binária, assumindo os valores 0 ou 1, determinando os seguintes casos:

1. : O trajeto entre as cidades *i* e *j* é selecionado para a solução que minimiza o percurso total;
2. : O trajeto entre as cidades *i* e *j* é descartado e não fará parte do percurso do caixeiro viajante.

Realizada a definição das variáveis de decisão, na próxima seção serão definidas as variáveis intervalares responsáveis por diferenciar cada um dos trajetos de maneira escalar.

2.2 MATRIZ DE DISTÂNCIAS

Para o caso de solução do problema para um cenário de *n* cidades, será necessário definir uma matriz *n*x*n*, fixando uma distância entre as cidades *i* e *j*. Vale salientar que os atributos da matriz não são necessariamente distâncias, podem ser por exemplo avaliados em tempo, desde que mantenham seu papel fundamental de comparar cada trajetória de maneira escalar e determinística.

A determinação da orientação ou não dos grafos, será de responsabilidade dos pesos dos próprios grafos em suas razões práticas. No estudo serão apresentadas diferentes formas de atribuição de valores aos trajetos, em que cada modo será agente implicador de condição para a orientação. Sumariamente, os casos simétricos, nos quais as distâncias são exatamente iguais as distâncias , são os casos em que não se há a necessidade de orientar o grafo. Abaixo estão apresentados alguns exemplos de avaliação:

1. Distâncias Euclidianas: Como sendo a menor distância entre dois pontos, simplifica-se o problema a uma análise simétrica, já que é sempre equivalente a . A questão das distâncias euclidianas é sua constante inviabilidade prática, já que provavelmente existirão barreiras físicas intransponíveis entre o trajeto de um ponto ao outro.
2. Percurso: As distâncias de percurso podem tanto ser simétricas como não, a depender, por exemplo, da maneira de deslocamento. Ao avaliar uma trajetória de um automóvel, por conta da direção de condução definida para a mão das ruas, os caminhos de ida e volta são distintos. Como isso, a solução ótima deverá levar em consideração a orientação do itinerário.
3. Tempo de trajeto: Como dito anteriormente, é possível considerar a duração de deslocamento entre os pontos, podendo ser medida de diferentes maneiras. A depender dos casos o tempo poderá ser avaliado também por uma locomoção através de veículos ou em caminhada, o que impactará também a orientação ou não.

Entre os diferentes tipos de dimensionamento de escala, a quilometragem ou o tempo de percurso têm maior complexidade de aplicação. Conhecidas as coordenadas geográficas dos nós, as distâncias euclidianas são facilmente calculadas, enquanto as propriedades de percurso precisam ser de alguma maneira medidas, principalmente em se tratando de tempo.

Para simplificar a avaliação, os problemas serão tratados sempre como orientados, já que resolverão também os problemas não orientados. Para o último caso, a direção da solução será desconsiderada, já que não traz qualquer ganho de desempenho.

2.3 MODELAGEM INICIAL

As seções 2.1 e 2.2 definem as principais variáveis e restrições do problema, sendo possível iniciar a modelagem e encontrar as primeiras soluções para o *TSP*. A função objetivo será definida pela soma da multiplicação das variáveis e . De forma direta, cada trajeto escolhido com terá o peso de seu grafo somado ao percurso total; Com sentido de minimização a solução ótima será encontrada.

Em caráter ilustrativo, serão utilizadas as localizações dos centros das cidades do estado do Rio de Janeiro como nós do problema, em que o caixeiro deverá passar por cada uma delas. Como o objetivo é apenas de explanação, serão utilizadas as distâncias euclidianas para uma matriz de distâncias mais simplificada.

.

Figura - Coordenadas RJ

Mapa

Descrição gerada automaticamente

Fonte: Autor

Modelo de solução:

As restrições (1) e (2) são respectivamente responsáveis por garantir que se chegará somente uma vez em cada nó, e do mesmo nó só se sairá também uma única vez. A imposição de que *i* deva ser diferente de *j* implicará na proibição de se deslocar para o mesmo nó, o que não tem sentido prático.

A modelagem resulta na solução apresentada a seguir:

Figura - Solução Modelo Inicial

Mapa

Descrição gerada automaticamente

Fonte: Autor

A solução encontrada tem problemas evidentes de viabilidade, revelando a incapacidade da modelagem inicial de restringir casos inverossímeis.

2.4 Subciclos

A inviabilidade apresentada na solução do modelo inicial se dá através da formação de subciclos, onde são formados “*cluster’s*” de um subconjunto de nós interligados. Pelas definições até aqui dissertadas, os objetivos e restrições estão sendo cumpridos, isso porque todos os nós estão sendo atendidos por pelo menos um trajeto e não mais que um. A questão é que a não há garantia que todos os pontos estejam sendo atendidos pelo mesmo roteiro.

Na prática, a solução inicial permite que o viajante se “teletransporte” entre zonas distantes, saindo de uma cidade a outra sem consumir qualquer recurso de distância. Por esse motivo, as soluções com presença de subciclos serão sempre excelentes candidatas ao maior prêmio, sendo necessário um refinamento do modelo em busca de sua proibição.

Na literatura, estão dispostos dois famosos métodos para a resolução dos problemas de subciclo, como a formulação *MTZ* e a formulação *DFJ*, que serão apresentadas nas próximas subseções.

2.4.1 FORMULAÇÃO MILLER-TUCKER-ZEMLIN (MTZ)

Miller, Tucker e Zemlin (1960), propõe a inserção de uma variável artificial *u* para garantir que não sejam formados subciclos. A ideia principal, é garantir um trajeto que se inicie e termine na cidade original fixada, e que o ciclo não passe por menos de *N* cidades, avaliando a ordem de seleção de cada nó. O modelo parte da inequação quando , o que resulta na imposição de que para trajetos escolhidos, a cidade *j* será atendida depois da cidade *i*.

Na prática o modelo MTZ guarda a ordem de seleção das cidades, de forma que não se possa voltar em uma cidade já atendida, impedindo a formação de um subciclo até que se passe por todos os pontos. Para que todo o trajeto seja definido, a primeira cidade é deixada de fora das restrições, possibilitando que o último trajeto volte à origem e encerre a viagem do caixeiro.

A estrutura do modelo inicial descrita na seção 2.3 é conservada, em que são somadas as restrições de inibição dos subciclos definidas a seguir:

2.4.2 FORMULAÇÃO DANTZIG-FULKERSON-JOHNSON (DFJ)

*Dantzig, Fulkerson, Johnson* (1954), definem a ideia de que para um subconjunto qualquer de nós pertencentes as cidades avaliadas, há uma quantidade máxima de grafos possíveis. De forma geral, para todos os *tours* possíveis em um conjunto de *n* cidades define-se ; Para cada subconjunto de deverá avaliar se a quantidade de trajetos *C* selecionados são insuficientes para formar um subciclo.

Para ilustrar a abordagem *DFJ* de restrição dos subciclos, foram selecionados os municípios de Itaboraí, Maricá, Niterói e São Gonçalo; numerados respectivamente de 1 a 4. Esses municípios formarão o conjunto *K* para avaliação.

Figura - Abordagem DFJ

Mapa

Descrição gerada automaticamente

Fonte: Autor

No caso desse conjunto de 4 cidades, para impedir que haja a formação de subciclos é necessário fazer todas as possíveis combinações entre os elementos de *K*.

A seguir serão detalhados três casos, definindo o comportamento das restrições:

1. Para os elementos de dimensão *D =* 1, a restrição gerada impede que se saia e volte para o mesmo ponto. As restrições dessa dimensão já foram verificadas pelo modelo inicial, ao fixar que o somatório de será igual a 1 para *i* ≠ *j*. Por essa razão, as restrições são consideradas pelos autores somente a partir da formação prática de *tours*.
2. Para os elementos com dimensão entre (Ex. [1, 2] ): será fixado o número máximo de *D* - 1 grafos interligando os nós. A restrição impede que se complete um ciclo entre os subconjuntos de *K*.
3. Haverá um único elemento de dimensão *n*, quando todos os nós são avaliados. Nesse momento, é necessário que a restrição seja descartada para que se assuma a solução objetivo com um ciclo interligando todos os nós.

Tabela - Restrições DFJ

Fonte: Autor

Com isso, chega-se no modelo geral de restrição da formulação *DFJ*, definida por:

O cálculo do número de restrições avaliadas pela formulação *DFJ* pode ser obtido da seguinte maneira:

Para n grande, é possível aproximar o crescimento do número de restrições a uma taxa exponencial, fazendo com que a formulação ultrapasse rapidamente os limites de capacidade computacional. O gráfico abaixo relaciona a quantidade de nós avaliados com o do número de variáveis; o seu comportamento retilíneo ratifica o crescimento exponencial.

Figura - Log(Qtd. Variáveis) Vs. n

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Fonte: Autor

Para o caso dos municípios do Rio de Janeiro (92 nós), a ordem de restrições já passa de , tornando a aplicação da formulação *DFJ* inviável isoladamente. O benefício do método *DFJ* é a maior facilidade de se aplicar métodos de relaxação de restrições, como o caso do *Lazy Constraint* apresentado na seção subsequente. Por esse motivo, a formulação será utilizado para o restante do trabalho.

2.5 *LAZY CONSTRAINTS*

O método de *Lazy Constraints* (*LC)* se apresenta como candidato para tornar os problemas de programação linear mais eficientes a partir do momento em que se avalia que muitas restrições serão dificilmente violadas. Na prática, as restrições são arbitrariamente ignoradas até que sua necessidade seja realmente comprovada. O benefício é a possibilidade de se avaliar o modelo com um conjunto menor de restrições, em que há um critério de julgamento para a viabilidade prática da solução, conferindo o status de decisão ótima a partir do momento em que a mesma se apresente possível.

Além disso, por mais que com o uso do *LC* não se resolva o problema a partir da modelagem teórica integral, se há a garantia de um algoritmo com solução ótima. Sendo R e o conjunto de restrições para a modelagem teórica e sua solução ótima; r e as restrições do modelo aplicado com *LC* e sua solução, têm se que:

Sabe-se que r é um subconjunto contido em R (1) e o PPL para o *TSP* tem orientação de minimização; Logo, a solução ótima da modelagem teórica só pode ser maior ou igual que a solução encontrada através da relaxação (2).

No caso específico do *TSP*, as restrições que impedem a formação de subciclos não serão declaradas previamente, mesmo sabendo da grande possibilidade de se encontrar uma solução com subciclos. Conhecendo os subciclos gerados na solução inicial, o problema é novamente resolvido, mas dessa vez com a inserção das restrições que justamente impedem a formação dos subciclos constituídos originalmente. Desta maneira, a solução do problema se dará de maneira iterativa; A cada solução novas restrições são inseridas, mais subciclos são impedidos e verifica-se que a quantidade desses diminui ou se mantêm a cada iteração. Enquanto não seja encontrada uma solução interligando todos os nós com uma só *tour*, o problema continuará sendo resolvido. A tabela a seguir mostra o resultado das iterações para otimizar a rota do caixeiro viajante que passa por todos os municípios do Rio de Janeiro.

Tabela - Resultado das Iterações do PPL



Fonte: Autor

Referências

**C. E. Miller, A. W. Tucker, R. A. Zemlin.**  *In:* “*Integer Programming Formulation of Traveling Salesman Problems.”,* J. ACM 7, 4 (Oct. 1960), pp. 326–329. Disponível em: <https://doi.org/10.1145/321043.321046>. Acesso em: 18/09/2022

**G. Dantzig, R. Fulkerson, S. Johnson.** *In*: “Solution of a Large-Scale Traveling-Salesman Problem”, Journal of the Operations Research Society of America, 1954, pp. 393-410, Disponível em: <https://pubsonline.informs.org/doi/abs/10.1287/opre.2.4.393>. Acesso em: 18/09/2022