Problema da alocação de viaturas de polícia
Trabalho Prático SIN 480

Igor Lucio (3902)[1], Douglas Boaventura (5144)[1]

[1]UFV – Universidade Federal de Viçosa
Rio Paranaíba – Minas Gerais – Brasil

{igor.lucio,douglas.boaventura}@ufv.br

Abstract. *This work aims to propose the implementation of a mathematical model location of police vehicles, which is able to minimize the cost of installing vehicles in order to reduce the incidence of criminal occurrences. In order to solve the problem, the model based on the p-median problem was used, considering the characteristics of the problem. The model was implemented using a C ++ language and solved via solver CPLEX 20.1. The results prove the model works with solutions applied to small and medium instances to reduce the cost of distance between the vertices.*

Resumo. *Este trabalho tem como objetivo propor a implementação de um modelo matemático de localização de viaturas policiais, que seja capaz de minimizar o custo de instalação de viaturas afim de reduzir a incidência de ocorrências criminais. Objetivando resolver o problema, foi utilizado o modelo baseados no problema de p-medianas, considerando as caraterísticas do problema. O modelo foi implementado utilizando a linguagem C++ e resolvido via solver CPLEX 20.1. Os resultados permitiram comprovar o funcionamento do modelo com soluções aplicadas a pequenas e médias instâncias de forma a reduzir o custo de distância entre os vértices.*

1. Descrição do problema

É fato notório que a segurança pública tem um papel muito importante na sociedade civil. Baseado nisso, o Estado através das forças de segurança traça estratégias para reduzir a quantidade de ocorrências policiais. A operacionalização dos meios de tomada decisão é importante principalmente para os serviços públicos emergenciais, que tem como objetivo atender a população de forma rápida, salvar vidas e/ou garantir a segurança da população. [Ribeiro 2019]

Uma das estratégias está na instalação de postos policiais nos bairros, ou seja, uma distribuição geográfica eficiente da presença policial. Sendo assim, cobrindo áreas de maiores impactos criminais e reduzindo o custo no deslocamento da distância entre as viaturas e os pontos de interesse.

2. Modelo Matemático

O modelo matemático do problema de localização de facilidades com limites para o número de instalações abertas é considerado como um problema de programação inteira binária. [Dantrakul et al. 2014]

O problema de p-medianas tem como objetivo determinar p locais (medianas) em uma rede de nós **N** com a finalidade de minimizar a soma das distâncias entre cada nó até a sua mediana mais próxima. [Lorena et al. 1999]

Sabendo disso, os dados relevantes para o problema são:

- *N*: número finito de pontos de demanda (pontos estratégicos)
- *P*: número finito de candidatos a instalação de facilidades (viaturas policiais)
- c_{ij}: distâncias entre os pontos de demanda e o número P de viaturas (ou postos policiais) a serem alocados
- x_{ij}: matriz de alocação (variável de decisão binária)

O problema é elaborado a partir do conjunto finito de pontos N $\{1, \ldots, n\}$ e tendo também como parâmetros c_{ij} que é uma matriz simétrica que representa os custos (ou distâncias). A variável de decisão x_{ij} é binária e corresponde a uma matriz de alocação, onde:

- se vértice x_j é alocado ao vértice x_i então $x_{ij} = 1$; caso contrário $x_{ij} = 0$
- o vértice x_i é considerado mediana se $x_{ii} = 1$; caso contrário $x_{ii} = 0$

A seguir é descrito, através das equações matemáticas 1 até 5 o modelo matemático do problema. Sendo 1 a função objetivo e as demais expressões são restrições.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i \in V \tag{2}$$

$$\sum_{j=1}^{n} x_{jj} = p \tag{3}$$

$$x_{ij} \leq x_{jj}, \forall i, j \in V, i \neq j \tag{4}$$

$$x_{ij} \in \{0, 1\} \ \forall i, j \in V \tag{5}$$

2.1. Restrições

- Restrição 2: assegurar que cada vértice x_j seja atendido por apenas uma mediana.
- Restrição 3: as medianas escolhidas são identificadas por 1 na variável de decisão quando os índices são iguais. Logo, visa garantir que existe exatamente p vértices medianas.
- Restrição 4: certifica que cada vértice j seja designado a um vértice i, ou seja, a variável de decisão $x_{jj} = 1$.
- Restrição 5: corresponde a restrições de integralidade, em outras palavras, x_{ij} é uma variável binária assumindo apenas valores 0 ou 1.

3. Implementação

O primeiro passo foi implementar os métodos de leitura de arquivos texto. É através destes que conseguimos trazer os dados das instâncias para dentro do contexto do código. As instâncias disponíveis são identificadas pela extensão *.txt* e também é possível passar o nome de uma instância via parâmetro para execução do algoritmo; as instâncias e instruções de como executar o algoritmo está nas subseções 3.1 e 3.2 respectivamente.

Adiante isto, é declarado variáveis que tem relação com o problema tais como:

- *IloEnv* ambiente: ambiente do modelo matemático;
- *IloModel* modelo: representa o modelo matemático e recebe o ambiente por parâmetro;
- *IloArray<IloNumArray>* c: matriz de distâncias e recebe o ambiente por parâmetro;
- *IloArray<IloNumVarArray>* x: matriz de alocação (variável de decisão binária) e recebe o ambiente e a quantidade de vértices (*n*) por parâmetro.

Importante salientar que a quantidade de vértices (nós) não é lida da instância visto que pode ser obtida dinamicamente via método *getSize()* (nativo da linguagem de programação C++) que retorna o tamanho de qualquer *array*.

Com isso é lido do arquivo a matriz de distâncias *c* e a quantidade de medianas *p*. A variável de decisão *x* é inicializada com a seguinte configuração:

IloNumVarArray(ambiente, n, 0, 1, ILOBOOL)

Primeiro parâmetro é a variável que representa o ambiente do modelo matemático, seguido pela quantidade de vértices, menor valor que a matriz pode assumir no caso é zero pois é binária, maior valor que pode assumir e por fim o tipo destes valores que é *boolean* mas que também pode ser inteiro; neste caso não há diferença real para o solver.

Em seguida temos a declaração de uma expressão *IloExpr* que representa a função objetivo do problema. São dois laços de repetição de i,j até n onde há dois somatórios com a multiplicação da matriz de distâncias nas posições i e j com a variável de decisão x também nas posições i e j. Em seguida a função objetiva é adicionada ao modelo matemático. A função objetivo também pode ser encontrada pela equação 1 na seção 2.

As restrições são implementadas e adicionadas ao modelo seguindo o que foi descrito na seção 2.

```
/*
    Restrição: 1b
    assegurar que cada vertice xj seja atendido por apenas uma mediana
*/
for (int i = 0; i < n; i++) {
    IloExpr restricao_1b(ambiente);
    for (int j = 0; j < n; j++) {
        restricao_1b += x[i][j];
    }
    modelo.add(restricao_1b == 1);
}

/*
    Restrição: 1c
    Garante que a quantidade de medianas é igual a p
*/
IloExpr restricao_1c(ambiente);

for (int j = 0; j < n; j++) {
    restricao_1c += x[j][j];
}
modelo.add(restricao_1c == p);
restricao_1c.end();


/*
    Restrição: 1d
    certifica que cada v ertice j seja designado a um vertice i
    desde que xjj = 1
*/
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        modelo.add(x[i][j] <= x[j][j]);
    }
}
```

Figura 1. Implementação das restrições
Fonte: Autoria Própria

3.1. Instâncias

Nesta seção será evidenciado os dados de entrada do problema. As figuras 2 e 3, respectivamente, foram criadas objetivando facilitar a validação do algoritmo.

A figura 2 apresenta uma matriz de distâncias 4×4 onde N = 4 e P = 2. Ou seja, temos 4 nós para alocação porém somente 2 viaturas policiais disponíveis para atender estes nós.

A seguir, a figura 3 apresenta uma matriz de distâncias 6×6 onde N = 6 e P = 3. Ou seja, temos 6 nós para alocação porém somente 3 viaturas policiais disponíveis para atender estes nós.

Para fins de comparação, usamos a instância do artigo de base onde nos foi cedido pelos autores através de contato via e-mail. Esta instância tem estrutura semelhante com as das figuras 2 e 3; porém com a matriz de alocação c_{ij} maior; sendo N = 20 e P = 8. A imagem não foi anexada devido ao seu tamanho, se trata de uma matriz 20×20.

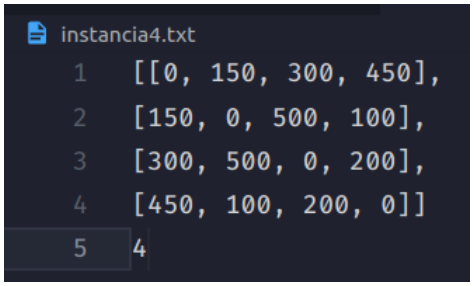

Figura 2. Instância 4×4 com N = 4 e P = 2
Fonte: Autoria Própria

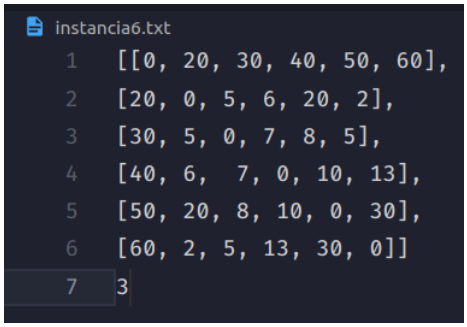

Figura 3. Instância 6×6 com N = 6 e P = 3
Fonte: Autoria Própria

3.2. Testes computacionais

A execução do algoritmo é feita através de terminal (linha de comando) usando o comando *make* concatenado com o executável do arquivo compilado (*main*) juntamente com o nome do arquivo texto que representa a instância, que é o primeiro argumento.

O arquivo *Makefile* garante a compilação do algoritmo e é encarregado de encontrar a biblioteca CPLEX na versão 20.1 na máquina. Caso o CPLEX não tenha sido localizado no caminho padrão (*/opt/ibm/ILOG/CPLEX_Studio201*), sua configuração deve ser feita manualmente diretamente no arquivo *Makefile*.

A figura 4 evidencia como executar o algoritmo diretamente pelo terminal usando a instância 4×4. Para as outras instâncias deve-se alterar o nome enviado por argumento, no caso trocar ”instancia4.txt”pelo nome do arquivo da instância a ser utilizada.

Figura 4. Execução via terminal usando *make*
Fonte: Autoria Própria

3.2.1. Instância 4×4

Considerando p = 2 e n = 4, o menor custo encontrado foi 250.

0	1	0	0
0	1	0	0
0	0	1	0
0	1	0	0

Tabela 1. Matriz de alocação x_{ij} para instância 4×4
Fonte: Autoria Própria

3.2.2. Instância 6×6

Considerando p = 2 e n = 4, o menor custo encontrado foi 250.

1	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	0	0	0	1	0
0	1	0	0	0	0

Tabela 2. Matriz de alocação x_{ij} para instância 6×6
Fonte: Autoria Própria

3.2.3. Instância 20×20

Considerando p = 8 e n = 20, o menor custo encontrado foi 3394.08.

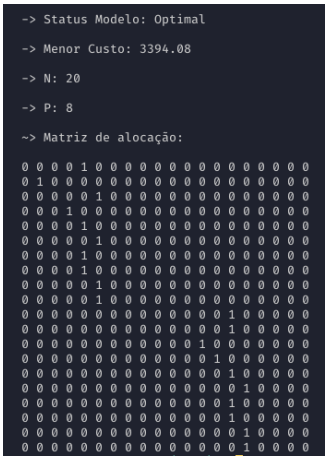

Figura 5. Saída para instância 20×20
Fonte: Autoria Própria

As instâncias 6×6 e 4×4 foram criadas objetivando a comprovação do funcionamento do algoritmo; além de auxiliar a implementação na medida em que o algoritmo este avançava.

4. Resultados

A solução deste trabalho foi implementado sobre a linguagem de programação C++ juntamente com o solver CPLEX 20.1, e executados em uma máquina composto por processador AMD Ryzen 7 3700X, 16GB memória RAM e 240GB SSD M2.

Para as instâncias menores (2×2 e 4×4) os valores podem ser, facilmente, confirmados apenas com a observação das matrizes de distâncias declaradas na seção 3.1. Com não obtivemos resposta do resultado de menor custo dos autores a tempo e nem era especificado no trabalho original, a instância 20x20 com p = 8 e n = 20, teve como resultado de menor custo o valor de 3394.08. O resultado é o semelhante com o algoritmo implementado por [Borges 2020] na qual utiliza a mesma instância (20×20).

5. Dificuldades Encontradas

A maior adversidade encontrada foi implementar a restrição 3 pois no artigo de base a mesma estava incorreta; e isto dificultou o entendimento da mesma. Se a mesma é implementada incorretamente gera erros que ficam difíceis de serem rastreados tais como vazamento de memória, além de dificultado o entendimento da restrição 4 pois, num contexto geral, para entender uma a restrição 4 necessita entender a 3.

O artigo original também não especificava os resultados obtidos através do modelo CPLEX; por mais que o objetivo do trabalho era comparação dos métodos; nossa conclusão é de que o foco dos autores foi maior no algoritmo *k-means* e na visualização dos pontos em um mapa.

Referências

Borges, P. R. (2020). Resolução do problema das p-medianas utilizando julia e cplex. https://github.com/rmaxflo/p-mediana-with-Julia-and-CPLEX/blob/main/dados.dat.

Dantrakul, S., Likasiri, C., and Pongvuthithum, R. (2014). Applied p-median and p-center algorithms for facility location problems. *Expert Systems with Applications*, 41(8):3596–3604.

Lorena, L. A. N., Senne, E. L. F., Paiva, J. d. C., and Marcondes, S. P. B. (1999). Integração de um modelo de p-medianas a sistemas de informações geográficas. *In31º Simpósio Brasileiro de Pesquisa Operacional, Juiz de Fora, MG. Anais. Rio de Janeiro, RJ: SOBRAPO*, pages 635–647.

Ribeiro, W. C. (2019). Modelo de localização de postos policiais auxiliares para a minimização do tempo de atendimento das ocorrências criminais da cidade de joão monlevade.