Análise de Artigo:
Uma heurística baseada em GRASP-VND para o Problema de Roteamento de Mula de Dados em Redes Mistas

Cainã Penna Mendes[1]**,Kayo Nascimento Xavier Cavalcante Leite**[2]**,Rafael Xavier Ribeiro**[3]

[1]Universidade Federal de Ouro Preto - UFOP

Resumo. *Esta análise apresenta um modelo de programação linear inteira para o Problema de Roteamento de Mula de Dados (PRMD) em redes de sensores mistas O modelo foi desenvolvido para otimizar o roteamento de mulas de dados, considerando as distâncias entre sensores e os raios de comunicação. A abordagem busca minimizar a distância total percorrida pela mula de dados, garantindo que todos os sensores sejam cobertos e que a base seja visitada. As restrições do modelo asseguram a formação de um ciclo viável e a eliminação de subciclos, permitindo uma solução eficiente para a coleta de dados em redes esparsas. Os resultados experimentais demonstram que o modelo alcança um desempenho satisfatório em um tempo computacional aceitável, destacando seu potencial em aplicações de otimização combinatória, especialmente em cenários de cidades inteligentes e Indústria 4.0.*

Palavras Chave: Mula de Dados, ciclo viável, eliminação de subcíclos, redes de sensores mistas, redes esparsas.

Abstract. *This analysis presents an integer linear programming model for the Data Mule Routing Problem (PRMD) in mixed sensor networks. The model was developed to optimize the routing of data mules, considering the distances between sensors and their communication radii. The approach aims to minimize the total distance traveled by the data mule while ensuring that all sensors are covered and the base is visited. The model's constraints ensure the formation of a viable cycle and the elimination of subcycles, allowing for an efficient solution for data collection in sparse networks. Experimental results demonstrate that the model achieves satisfactory performance within an acceptable computational time, highlighting its potential in combinatorial optimization applications, especially in smart city and Industry 4.0 scenarios.*

Keywords: Data Mule, Feasible Cycle, Elimination of Subcycles, Mixed Sensor Networks, Sparse Networks.

1. Contextualização

O **Problema de Roteamento de Mula de Dados** (PRMD) é um desafio significativo em redes de sensores sem fio, especialmente em ambientes esparsos, onde a comunicação direta entre sensores é frequentemente inviável devido à limitação do alcance de transmissão. Redes de sensores sem fio são amplamente utilizadas em aplicações como monitoramento ambiental, segurança e gestão de desastres, onde múltiplos sensores coletam dados de forma distribuída.

Em redes esparsas, a solução clássica de comunicação multi-hop não é suficiente, exigindo a utilização de uma ”mula de dados”— um agente responsável por coletar e transportar dados dos sensores até uma estação base. O papel da mula de dados é crucial, pois

sua eficiência pode impactar diretamente o desempenho da rede e a eficácia na coleta de informações.

O PRMD busca determinar as rotas que a mula de dados deve seguir para minimizar a distância total percorrida, garantindo que todos os sensores sejam visitados e que a base seja alcançada. Este problema é desafiador devido à necessidade de considerar não apenas as distâncias entre sensores, mas também os raios de comunicação de cada um, que definem quais sensores podem se comunicar diretamente. Além disso, a eliminação de subciclos é essencial para evitar rotas redundantes, que aumentariam o tempo e o custo da coleta de dados.

Neste contexto, o artigo propõe um modelo de programação linear inteira que formaliza o PRMD e busca soluções eficientes para o problema, visando melhorar a logística da coleta de dados em redes de sensores mistas e esparsas. A abordagem apresentada demonstra a eficácia do modelo em encontrar rotas viáveis em um tempo computacional aceitável, destacando seu potencial em aplicações práticas, como em cidades inteligentes e na Indústria 4.0, onde a otimização do transporte de informações é vital.

2. Modelo matemático

Variáveis de entrada: Conjuntos e Parâmetros

- n: Número de sensores.
- Sensores: Conjunto de sensores de 1 a n.
- b: Nó base, onde a mula de dados começa e termina seu percurso.
- distancia$[Sensores][Sensores]$: Matriz de distâncias entre cada par de sensores.
- raio$[Sensores]$: Raio de comunicação de cada sensor.
- $C[i \in$ Sensores$]$: Conjunto que define os sensores que estão dentro do raio de comunicação de um sensor i.

Variáveis de Decisão

- x[Sensores][Sensores]: Variável binária que indica se a aresta entre os sensores i e j faz parte da solução (1 se sim, 0 se não).
- y[Sensores]: Variável binária que indica se o sensor i é visitado (1 se sim, 0 se não).
- z[Sensores][Sensores]: Variável auxiliar para evitar a formação de subciclos na rota.

Função Objetivo

A função objetivo é minimizar a distância total percorrida pela mula de dados:

```
minimize sum(i in Sensores, j in Sensores: i != j) distancia[i][j] * x[i][j];
```

Ou seja, busca-se minimizar a soma das distâncias entre sensores que são conectados.

Restrições

Cobertura dos Sensores:

```
forall(i in Sensores)
    sum(j in C[i]) y[j] >= 1;
```

Essa restrição garante que, se um sensor estiver dentro do raio de comunicação de outro já visitado, ele não precisa ser visitado diretamente pela mula de dados.

Formação do Ciclo:

```
forall(i in Sensores)
    sum(j in Sensores: i != j) x[i][j] == y[i];

forall(i in Sensores)
    sum(j in Sensores: i != j) x[j][i] == y[i];
```

Essas restrições garantem que, se um sensor for visitado, deve haver um fluxo de entrada e saída, formando um ciclo válido.

Visita à Base:

```
y[b] == 1;
```

Garante que a mula de dados sempre visite a base.

Fluxo dos nós

```
forall(i in Sensores: i != b)
    sum(j in Sensores: (i != j)) z[i][j] == sum(j in Sensores: (i != j)) z[j][i] + y[i];
```

Essa restrição garante que o número de nós visitados na entrada seja o mesmo que o número na saída, exceto pelo nó base, que precisa ser sempre visitado.

Eliminação de Subciclos:

```
forall(i in Sensores, j in Sensores: i != j)
    z[i][j] <= n * x[i][j];

forall(i in Sensores, j in Sensores: i != j)
    z[i][j] <= sum(k in Sensores) y[k];
```

Essas restrições garantem que não haja subciclos na solução, utilizando as variáveis auxiliares z.
A de cima é a responsável por definir que o numero de nós visitados em z não possa nunca ser maior que o numero n de nós.
A de baixo é a responsável por basicamente a rota final de visitação dos nos, ela permite que a variavel auxiliar seja visitada se, somente se os nos estiverem na rota de visitação

Fluxo da base

```
sum(j in Sensores: j != b) z[b][j] == y[b];
```

Essa restrição garante que a variavel auxiliar z para base seja igual ao numero de nós visitados para garantir que a rota retorne a base, sem essa restrição, não necessariamente a base estaria conectada nos ciclos encontrados, podendo resultar em itens erroneos.

3. Instancias utilizadas

Foram usados um total de 3 exemplos para rodar este programa como teste e são eles

Exemplo 1: Neste caso foram utilizados 5 sensores gerando a seguinte matriz de distâncias. A base para este exemplo foi definida como o Sensor 2.

$$\begin{pmatrix} 0.0 & 5.39 & 5.83 & 4.47 & 9.22 \\ 5.39 & 0.0 & 4.24 & 7.81 & 5.39 \\ 5.83 & 4.24 & 0.0 & 4.47 & 5.66 \\ 4.47 & 7.81 & 4.47 & 0.0 & 9.22 \\ 9.22 & 5.39 & 5.66 & 9.22 & 0.0 \end{pmatrix}$$

E com os seguintes valores para os raios:

$$\mathbf{raio}^{\mathbf{T}} = [5.0, 9.0, 4.0, 7.0, 8.0]$$

Exemplo 2: Neste caso foram utilizados 10 sensores gerando a seguinte matriz de distâncias. A base para este exemplo foi definida como o Sensor 1.

$$\begin{pmatrix} 0.0 & 4.2 & 5.5 & 7.8 & 6.1 & 8.9 & 9.2 & 5.3 & 4.9 & 6.4 \\ 4.2 & 0.0 & 3.6 & 6.7 & 8.2 & 5.5 & 7.9 & 9.3 & 4.6 & 7.5 \\ 5.5 & 3.6 & 0.0 & 4.9 & 6.8 & 7.2 & 5.1 & 6.4 & 9.1 & 8.7 \\ 7.8 & 6.7 & 4.9 & 0.0 & 5.3 & 9.0 & 6.7 & 4.8 & 6.9 & 9.2 \\ 6.1 & 8.2 & 6.8 & 5.3 & 0.0 & 4.7 & 6.9 & 5.1 & 7.8 & 4.3 \\ 8.9 & 5.5 & 7.2 & 9.0 & 4.7 & 0.0 & 5.8 & 6.6 & 8.4 & 7.3 \\ 9.2 & 7.9 & 5.1 & 6.7 & 6.9 & 5.8 & 0.0 & 4.5 & 5.7 & 7.9 \\ 5.3 & 9.3 & 6.4 & 4.8 & 5.1 & 6.6 & 4.5 & 0.0 & 6.1 & 8.2 \\ 4.9 & 4.6 & 9.1 & 6.9 & 7.8 & 8.4 & 5.7 & 6.1 & 0.0 & 5.3 \\ 6.4 & 7.5 & 8.7 & 9.2 & 4.3 & 7.3 & 7.9 & 8.2 & 5.3 & 0.0 \end{pmatrix}$$

E com os seguintes valores para os raios:

$$\mathbf{raio^T} = [4.0, 5.5, 4.2, 6.0, 4.8, 5.9, 4.5, 5.0, 6.2, 4.7]$$

Exemplo 3: Neste caso foram utilizados 20 sensores gerando a seguinte matriz de distâncias.A base para este exemplo foi definida como o Sensor 10.

$$\begin{pmatrix}
[0.0 & 10.5 & 12.3 & 8.7 & 6.5 & 9.8 & 7.3 & 11.2 & 5.9 & 14.1 \\
10.2 & 13.4 & 9.6 & 7.1 & 15.4 & 13.6 & 16.3 & 18.9 & 11.0 & 9.8 \\
10.5 & 0.0 & 8.4 & 7.9 & 11.6 & 14.2 & 9.3 & 10.1 & 12.7 & 6.2 \\
15.5 & 9.1 & 8.8 & 7.0 & 9.4 & 13.5 & 6.8 & 12.4 & 10.6 & 15.3 \\
12.3 & 8.4 & 0.0 & 11.2 & 13.9 & 10.7 & 8.9 & 12.5 & 9.7 & 14.8 \\
11.3 & 9.9 & 10.8 & 9.4 & 7.6 & 14.9 & 12.8 & 13.7 & 8.4 & 16.9 \\
8.7 & 7.9 & 11.2 & 0.0 & 9.5 & 10.2 & 9.1 & 11.6 & 7.7 & 14.3 \\
6.9 & 13.0 & 10.1 & 7.4 & 15.7 & 12.3 & 13.4 & 10.6 & 14.0 & 11.7 \\
6.5 & 11.6 & 13.9 & 9.5 & 0.0 & 10.9 & 9.2 & 14.5 & 7.1 & 12.3 \\
11.4 & 8.6 & 12.5 & 6.8 & 13.7 & 15.1 & 16.0 & 10.4 & 11.9 & 9.9 \\
9.8 & 14.2 & 10.7 & 10.2 & 10.9 & 0.0 & 13.8 & 12.4 & 9.6 & 7.8 \\
11.7 & 14.6 & 9.3 & 11.5 & 8.4 & 15.0 & 9.9 & 13.2 & 12.5 & 10.8 \\
7.3 & 9.3 & 8.9 & 9.1 & 9.2 & 13.8 & 0.0 & 7.9 & 6.4 & 11.9 \\
9.4 & 8.7 & 10.6 & 7.5 & 9.7 & 12.0 & 14.2 & 11.1 & 10.0 & 15.4 \\
11.2 & 10.1 & 12.5 & 11.6 & 14.5 & 12.4 & 7.9 & 0.0 & 13.2 & 10.5 \\
11.8 & 8.5 & 15.4 & 10.0 & 12.7 & 7.3 & 14.4 & 9.6 & 12.3 & 11.9 \\
5.9 & 12.7 & 9.7 & 7.7 & 7.1 & 9.6 & 6.4 & 13.2 & 0.0 & 8.6 \\
12.5 & 10.2 & 9.8 & 8.9 & 11.0 & 15.3 & 9.5 & 10.1 & 9.7 & 12.8 \\
14.1 & 6.2 & 14.8 & 14.3 & 12.3 & 7.8 & 11.9 & 10.5 & 8.6 & 0.0 \\
15.7 & 9.3 & 12.1 & 10.8 & 13.9 & 12.7 & 11.6 & 13.0 & 12.4 & 14.2 \\
10.2 & 15.5 & 11.3 & 6.9 & 11.4 & 11.7 & 9.4 & 11.8 & 12.5 & 15.7 \\
0.0 & 14.8 & 13.5 & 10.2 & 7.8 & 12.4 & 13.3 & 9.5 & 10.7 & 11.1 \\
13.4 & 9.1 & 9.9 & 13.0 & 8.6 & 14.6 & 8.7 & 8.5 & 10.2 & 9.3 \\
14.8 & 0.0 & 12.9 & 9.4 & 10.6 & 14.0 & 15.4 & 11.9 & 13.8 & 12.1 \\
9.6 & 8.8 & 10.8 & 10.1 & 12.5 & 9.3 & 10.6 & 15.4 & 9.8 & 12.1 \\
13.5 & 12.9 & 0.0 & 11.3 & 10.0 & 14.1 & 9.5 & 12.6 & 13.4 & 9.8 \\
7.1 & 7.0 & 9.4 & 7.4 & 6.8 & 11.5 & 7.5 & 10.0 & 8.9 & 10.8 \\
10.2 & 9.4 & 11.3 & 0.0 & 13.0 & 14.7 & 10.9 & 8.5 & 12.4 & 10.0 \\
15.4 & 9.4 & 7.6 & 15.7 & 13.7 & 8.4 & 9.7 & 12.7 & 11.0 & 13.9 \\
7.8 & 10.6 & 10.0 & 13.0 & 0.0 & 11.9 & 10.7 & 14.9 & 12.8 & 9.7 \\
13.6 & 13.5 & 14.9 & 12.3 & 15.1 & 15.0 & 12.0 & 7.3 & 15.3 & 12.7 \\
12.4 & 14.0 & 14.1 & 14.7 & 11.9 & 0.0 & 13.4 & 15.8 & 10.9 & 13.1 \\
16.3 & 6.8 & 12.8 & 13.4 & 16.0 & 9.9 & 14.2 & 14.4 & 9.5 & 11.6 \\
13.3 & 15.4 & 9.5 & 10.9 & 10.7 & 13.4 & 0.0 & 15.2 & 10.3 & 14.8 \\
18.9 & 12.4 & 13.7 & 10.6 & 10.4 & 13.2 & 11.1 & 9.6 & 10.1 & 13.0 \\
9.5 & 11.9 & 12.6 & 8.5 & 14.9 & 15.8 & 15.2 & 0.0 & 12.9 & 11.7 \\
11.0 & 10.6 & 8.4 & 14.0 & 11.9 & 12.5 & 10.0 & 12.3 & 9.7 & 12.4 \\
10.7 & 13.8 & 13.4 & 12.4 & 12.8 & 10.9 & 10.3 & 12.9 & 0.0 & 11.4 \\
9.8 & 15.3 & 16.9 & 11.7 & 9.9 & 10.8 & 15.4 & 11.9 & 12.8 & 14.2 \\
11.1 & 12.1 & 9.8 & 10.0 & 9.7 & 13.1 & 14.8 & 11.7 & 11.4 & 0.0]
\end{pmatrix}$$

E com os seguintes valores para os raios:

$$\mathbf{raio^T} = [10, 14.5, 13, 15, 14.8, 10, 12.2, 11, 11.3, 9, 4, 17, 22, 9, 13.2, 15.7, 14.4, 16.3, 10, 9.8]$$

Exemplo 4 e 5: Neste caso foram utilizados 30 e 40 sensores gerando matrizes de distâncias 30x30 e 40x40 respectivamente. Entretanto, o como a versão estudantil disponibilizada do CPLEX possui limitação de variáveis de decisão, o programa não consegue executar tais dados.

4. Resultados

4.1. *Resultados do primeiro exemplo:*

No primeiro caso as Mulas de Dados passariam pelas seguintes posições de x e teria como solução otima encontrada igual a 15,46 percorrendo a rota 2 -¿ 3 -¿ 1 -¿ 2 :

Tabela 1. Posiçoes de X

0	1	2	3	4	5
1	0	0	1	0	0
2	0	0	0	0	0
3	1	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

Sendo assim, visitaria os seguintes nós em Y:

Tabela 2. Posições de y

Sensores	Posição
1	1
2	1
3	1
4	0
5	0

4.2. *Resultados do segundo exemplo:*

No segundo caso as Mulas de Dados passariam pelas seguintes posições de x e teria como solução otima encontrada igual a 22,7 percorrendo a rota 1 -¿ 3 -¿ 5 -¿ 8 -¿ 1 :

Tabela 3. Posiçoes de x

0	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Sendo assim, visitaria os seguintes nós em Y:

Tabela 4. Posição em y

Sensores	Posição
1	1
2	0
3	1
4	0
5	1
6	0
7	0
8	1
9	0
10	0

4.3. ***Resultados do terceiro exemplo:***

No terceiro caso as Mulas de Dados passariam pelas seguintes posições de x e teria como solução otima encontrada igual a 40,4 percorrendo a rota 10 -¿ 11 -¿ 1 -¿9 -¿ 10:

Tabela 5. Posição em x

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Sendo assim, visitaria os seguintes nós em Y:

Tabela 6. Posição em y

Sensores	Posiçao
1	1
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	1
10	1
11	1
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

5. Dificuldades Encontradas

Compilar e obter resultados de um artigo que aborda problemas complexos, como o Problema de Roteamento de Mula de Dados em Redes Mistas, sem a utilização de meta-heurísticas pode apresentar várias dificuldades, especialmente em relação à eficiência e à escalabilidade do método de resolução escolhido. Vamos explorar algumas dessas dificuldades de forma mais detalhada:

Complexidade do Problema

O problema em questão é intrinsecamente complexo, frequentemente classificado como NP-difícil. Isso significa que, à medida que o número de sensores e a complexidade da rede aumentam, o tempo necessário para encontrar uma solução ótima através de métodos tradicionais de programação linear ou de busca exaustiva cresce exponencialmente. Compilar um modelo que lide com essa complexidade sem recorrer a meta-heurísticas pode resultar em algoritmos que não conseguem fornecer soluções em um tempo razoável, especialmente em cenários práticos com um número elevado de sensores. Outra coisa díficil foi para tomar decisões em relação ao entendimento prático de algumas caracteristicas específicas do mesmo, tal como o uso do raio de comunicação

Limitações dos Métodos Clássicos

Métodos clássicos, como algoritmos de busca ou programação linear, são muitas vezes insuficientes para lidar com a natureza dinâmica e a variabilidade dos dados em um problema de roteamento. Sem a flexibilidade das meta-heurísticas, que exploram soluções aproximadas através de técnicas como busca local, simulação, ou algoritmos genéticos, a abordagem pode ser rígida e incapaz de se adaptar a mudanças nos dados ou na estrutura da rede.

Tempo de Cálculo

Quando se tenta resolver um problema complexo com métodos tradicionais, o tempo de cálculo pode se tornar um grande obstáculo. Ao buscar soluções exatas, é comum enfrentar longos períodos de espera para a conclusão do processo de compilação e execução, especialmente quando se tenta testar diferentes cenários ou parâmetros. Essa limitação pode resultar em uma experiência frustrante e na incapacidade de realizar análises robustas e abrangentes.

Dificuldade em Obter Soluções Práticas

A obtenção de resultados práticos e aplicáveis é uma das principais metas de qualquer pesquisa. Sem a implementação de meta-heurísticas, é possível que as soluções encontradas sejam apenas teóricas ou muito distantes das condições reais de operação. Isso pode limitar a relevância do trabalho e sua aplicação em contextos do mundo real, onde soluções rápidas e eficazes são frequentemente necessárias.

A Necessidade de Avaliação de Desempenho

Ao utilizar algoritmos tradicionais, a avaliação de desempenho torna-se mais desafiadora. Com meta-heurísticas, é comum fazer uso de métricas de desempenho como a qualidade da solução, tempo de execução e robustez. Sem essas ferramentas, a análise se torna limitada, dificultando a comparação dos resultados obtidos com os de outros métodos ou trabalhos anteriores, o que é crucial em pesquisas para validar a eficácia de novas abordagens.

Falta de Flexibilidade

A rigidez dos métodos tradicionais pode dificultar a personalização do algoritmo para atender às necessidades específicas do problema. Por exemplo, a implementação de restrições e critérios adicionais pode ser mais desafiadora em métodos clássicos do que em meta-heurísticas, que são projetadas para serem adaptáveis.

Ferramenta CPLEX

Além dos problemas ocasionados pela caracterização do problema, o maior problema foi o uso do CPLEX studio.A ferramenta se mostrou muitas vezes incapaz de ser simples ou fácil de ser utilizada, em vários cenarios a mesma não conseguia nem exemplificar os seus erros de maneira intuitiva, um exemplo sendo a dificuldade de lidar com diretorios que possuissem caracteres especiais (â, ã,ç, etc) mesmo estando instalado em português. Outra barreira imposta pela IDE é por seu uso comunitário, onde as variáveis de decisão são limitadas, o que não permitiu a execução de instancias de dados maiores (30 e 40 sensores).

Conclusão

Diante dessas dificuldades, fica evidente que compilar e obter resultados sem a utilização de meta-heurísticas em um problema como o Problema de Roteamento de Mula de Dados em Redes Mistas pode não apenas aumentar a complexidade do processo, mas também limitar a eficácia e a aplicabilidade das soluções. Para abordar essas questões, a adoção de técnicas de otimização que considerem a natureza multifacetada do problema pode ser essencial para se alcançar resultados significativos e práticos