

Ótimo Posicionamento de Sensores em uma Rede de Distribuição de Água com Uso de Programação Linear.

Lázaro Robert da Silva Cunha

Universidade Federal de São Paulo

São Paulo, São José dos Campos, Av. Cesare Monsueto Giulio Lattes, 1201

robert.silva10@unifesp.br

1. Definição do Problema

Redes de distribuição de água são essenciais para o funcionamento da sociedade. No entanto, possíveis ataques por contaminantes ou a qualidade geral da água com o passar do tempo podem prejudicar esse sistema e as pessoas que o usam. Nesse sentido, este trabalho busca soluções para o posicionamento ótimo de sensores em uma rede de distribuição de água levando em consideração restrições de quantidade de sensores com o fito de diminuir a quantidade de contaminantes consumidos pela população. Os dados do problema serão simulados no *software* EPANET da *United States Environmental Protection Agency*, *software* gratuito, simples, robusto e bastante conhecido e usado na área de modelagem de redes de distribuição de água.

2. Revisão Bibliográfica

Os primeiros esforços sistemáticos para definir metodologias de posicionamento ótimo de sensores em redes de distribuição de água remontam ao trabalho de [Kessler et al. [1998]]. Os autores formularam uma metodologia baseada na definição de um nível de serviço, medido pelo volume máximo de água contaminada consumida antes da detecção. A proposta consistia em transformar a rede hidráulica em uma rede auxiliar representando direções de fluxo médias de um ciclo típico de demanda, aplicar algoritmos de caminhos mínimos para identificar os domínios de propagação da contaminação e, por fim, resolver um problema de cobertura de conjuntos para selecionar a configuração mínima de estações de monitoramento. Os resultados mostraram que, mesmo em redes de médio porte, era possível reduzir significativamente o número de sensores sem comprometer o nível de serviço, além de incorporar critérios de invulnerabilidade para aumentar a resiliência do sistema frente a falhas em estações específicas.

Na sequência, [Berry et al. [2004]] introduziram um arcabouço de programação inteira mista para tratar a alocação de sensores em redes municipais com o objetivo de minimizar a exposição populacional em cenários de contaminação. O modelo integrava simulações hidráulicas e de qualidade de água com o EPANET e considerava distribuições probabilísticas de ataques em diferentes nós e períodos. Os testes realizados em redes de benchmark e em uma rede real evidenciaram que a formulação era eficiente do ponto de vista computacional, permitindo obter soluções em minutos, e robusta sob incertezas de dados populacionais e de risco. Os resultados mostraram que a abordagem era capaz de proteger grandes parcelas da população com um número relativamente reduzido de sensores, consolidando a programação inteira mista como uma ferramenta poderosa para apoiar decisões estratégicas nesse campo.

Ampliando essa linha, [Berry et al. [2005]] evoluíram o modelo anterior ao incorporar custos de instalação e restrições orçamentárias na formulação. A metodologia combinou programação inteira dinâmica com simulações de cenários de contaminação, permitindo comparar soluções ótimas com estratégias manuais de alocação. Os experimentos em uma grande rede de distribuição nos Estados Unidos demonstraram que a abordagem de otimização superava significativamente a seleção manual: mesmo com um número menor de sensores, as soluções otimizadas obtinham menor volume esperado de consumo de contaminantes. Os autores ressaltaram que, embora a modelagem

matemática seja central para orientar o processo decisório, a experiência prática e considerações políticas permanecem fundamentais na definição final das estratégias de monitoramento.

Outro avanço relevante foi apresentado por [Ostfeld e Salomons [2004]], que propuseram uma estrutura para projetar sistemas de detecção precoce (EWDS) com o objetivo de minimizar o volume de água contaminada consumida acima de níveis perigosos. A metodologia utilizou o conceito de matriz de poluição randomizada, que representava múltiplos cenários possíveis de injeção de contaminantes, em conjunto com simulações hidráulicas não estacionárias no EPANET. Para otimização da localização de sensores, aplicou-se um algoritmo genético, permitindo explorar soluções em diferentes níveis de cobertura e redundância. Os resultados, testados em redes-padrão, mostraram que poucas estações já eram suficientes para garantir elevada probabilidade de detecção, ainda que com aumento de redundância entre sensores. Além disso, análises de sensibilidade confirmaram a robustez da metodologia frente a variações nas probabilidades de injeção e nos parâmetros de operação do sistema.

3. Apresentação dos Dados

Os dados para resolver o problema são advindos da simulação em um período de 24 horas da rede de distribuição de água *Net3*, rede de benchmark da própria EPANET.

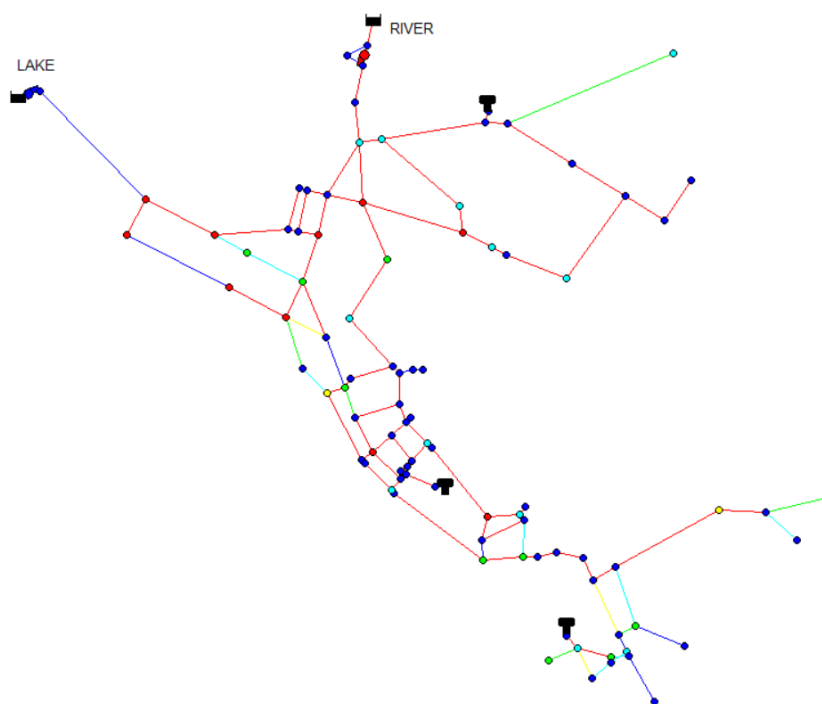


Figura 1: Topologia da Rede Net3

Nesse sentido, os principais componentes de dados que serão extraídos da simulação são: nós (reservatórios, tanques e junções), arestas (canos) e intensidade do fluxo de água em cada aresta para 25 padrões diferentes de 00:00 até 24:00 horas de uma em uma hora. Assim, com esses dados, serão criadas as estruturas de dados para resolver os problemas de programação linear inteira mista e têmpera simulada. Nesse contexto, a rede *Net3* tem 92 junções, 2 reservatórios, 3 tanques e 119 canos.

A Tabela 1 mostra como a matriz de fluxos é estruturada. Nas linhas temos os *pipes* e cada coluna define um padrão de fluxo em que, se existe fluxo positivo no *pipe*, então 1, se não, 0.

Tabela 1: Matriz de Padrões de Fluxos

Pipe	Flow Pattern 0	Flow Pattern 1	...	Flow Pattern 24
0	1	1	0	0
1	1	1	0	1
2	1	0	1	0
...
118	1	1	0	1

Já a Tabela 2 mostra como a matriz de probabilidades de ataque em um dado nó é estruturada. A matriz contém uma distribuição de probabilidade simples calculada como: $\frac{1}{\text{Número de Cenários}}$ em que o Número de Cenários é calculado pela quantidade de nós multiplicado pela quantidade de padrões de fluxo.

Tabela 2: Matriz Alpha de Probabilidades de Ataque em um Nó

Node	Flow Pattern 0	Flow Pattern 1	...	Flow Pattern 24
0	0.000412	0.000412	0.000412	0.000412
1	0.000412	0.000412	0.000412	0.000412
2	0.000412	0.000412	0.000412	0.000412
...
96	0.000412	0.000412	0.000412	0.000412

Agora, a Tabela 3 mostra a estruturação da matriz de densidade populacional em cada nó da rede dado em cada padrão. A matriz leva em consideração uma população de 500 pessoas em cada nó e 0 pessoas nos reservatórios e tanques. É feita uma normalização para que proporções da população sejam atribuídas a cada nó. Isso é calculado como: $\frac{\text{População no Nó}}{\text{População Total}}$

Tabela 3: Matriz Delta de Densidade Populacional em um Nó

Node	Flow Pattern 0	Flow Pattern 1	...	Flow Pattern 24
0	0.010870	0.010870	0.010870	0.010870
1	0.010870	0.010870	0.010870	0.010870
2	0.010870	0.010870	0.010870	0.010870
...
96	0.000000	0.000000	0.000000	0.000000

4. Solução por Programação Inteira Mista

O primeiro modelo para resolver o problema de minimização da contaminação na rede de distribuição de água é dado por [Berry et al. [2004]], uma formulação inteira mista:

$$\begin{aligned}
 &\text{minimize} \quad \sum_{i=1}^n \sum_{p=1}^P \sum_{j=1}^n \alpha_{ip} c_{ipj} \delta_{jp} \\
 &\text{where} \quad \begin{cases} c_{ipi} = 1 & \forall i = 1 \dots n, p = 1 \dots P \\ s_{ij} = s_{ji} & \forall i = 1 \dots n-1, i < j \\ c_{ipj} \geq c_{ipk} - s_{kj} & \forall (k, j) \in E \text{ tal que } f_{kjp} = 1 \\ \sum_{(i,j) \in E, i < j} s_{ij} \leq S_{max} \\ s_{ij} \in \{0, 1\} & \forall (i, j) \in E \end{cases}
 \end{aligned}$$

Em que o primeiro conjunto de restrições define que um nó atacado está contaminado. O segundo conjunto define a simetria do problema, uma aresta detecta fluxo bidirecionalmente. Já o terceiro conjunto mostra a propagação de contaminantes na rede nó a nó formando em conjunto a cadeia de contaminação da rede. A penúltima restrição define que a soma de sensores posicionados não pode ultrapassar a quantidade máxima configurada no modelo. A última restrição define a binariedade das variáveis de decisão de posicionamento dos sensores.

5. Solução por Simulated Annealing

O segundo método de resolução do problema foi feito com a meta-heurística de Simulated Annealing. A estratégia de busca adotada baseia-se no controle de um parâmetro de temperatura T , que decai ao longo das iterações segundo uma taxa de resfriamento α . Em temperaturas elevadas, o algoritmo explora amplamente o espaço de busca, aceitando movimentos que aumentam a função objetivo (pioram a solução) com alta probabilidade. À medida que T se aproxima de T_{min} , o sistema 'congela', aceitando apenas melhorias na função objetivo. O Algoritmo 1 apresenta o pseudocódigo da implementação, detalhando a geração de vizinhos através da perturbação da posição dos sensores e a avaliação baseada no impacto da contaminação.

Algoritmo 1 Simulated Annealing para Posicionamento de Sensores

```

1: Parâmetros:
2:  $T \leftarrow 100$                                 ▷ Temperatura Inicial
3:  $\alpha \leftarrow 0.95$                         ▷ Taxa de Resfriamento
4:  $Iter_{max} \leftarrow 100$                     ▷ Iterações por Temperatura
5:  $Tempo_{max} \leftarrow 30$                     ▷ Tempo Limite (s)
6:  $T_{min} \leftarrow 0.01$                     ▷ Temperatura Mínima

7:  $S \leftarrow \text{GerarSolucaoInicial}(S_{max})$                                 ▷ Bloco 1
8:  $E \leftarrow \text{Avaliar}(S)$                                 ▷ Bloco 2: Custo inicial
9:  $S_{melhor} \leftarrow S$ 
10:  $E_{melhor} \leftarrow E$ 

11: enquanto  $T > T_{min} \wedge \text{TempoDecorrido}() < Tempo_{max}$  faça
12:   para  $i \leftarrow 1$  até  $Iter_{max}$  faça
13:      $S_{vizinho} \leftarrow \text{GerarVizinho}(S)$                                 ▷ Bloco 3: Move sensor
14:      $E_{vizinho} \leftarrow \text{Avaliar}(S_{vizinho})$                     ▷ Bloco 4: Avalia novo custo
15:      $\Delta E \leftarrow E_{vizinho} - E$                                 ▷ Bloco 5: Compara

16:     se  $\Delta E < 0$  then                                ▷ Melhora encontrada
17:        $S \leftarrow S_{vizinho}$ 
18:        $E \leftarrow E_{vizinho}$ 
19:       se  $E < E_{melhor}$  then                                ▷ Bloco 6: Novo Melhor
20:          $S_{melhor} \leftarrow S_{vizinho}$ 
21:          $E_{melhor} \leftarrow E_{vizinho}$ 
22:       end se
23:     senão                                ▷ Piora: Critério de Metropolis
24:        $P \leftarrow e^{-\Delta E/T}$ 
25:       se  $\text{Random}(0, 1) < P$  then
26:          $S \leftarrow S_{vizinho}$ 
27:          $E \leftarrow E_{vizinho}$ 
28:       end se
29:     end se
30:   end para
31:    $T \leftarrow T \times \alpha$                                 ▷ Resfriamento
32: end enquanto
33: retorne  $S_{melhor}$ 

```

6. Resultados

Os resultados foram obtidos executando os algoritmos em uma máquina com especificações: I7 1165G7 e 16 GB de RAM. Nesse sentido, os resultados podem ser vistos na tabela 4. A análise dos resultados apresentados evidencia uma disparidade significativa de desempenho entre o método exato (Gurobi) e a meta-heurística implementada (Simulated Annealing em Python).

O solver **Gurobi** demonstrou superioridade absoluta tanto em qualidade da solução quanto em tempo computacional para as instâncias testadas. Observa-se que o método exato encontrou o ótimo global em tempos reduzidos (variando entre 0.78s e 5.17s), enquanto a meta-heurística, configurada com um tempo fixo de execução próximo a 30 segundos, não obteve convergência para o

ótimo global na maioria dos casos.

O *gap* de otimalidade do método aproximativo variou consideravelmente, atingindo um pico de **+131.20%** para o cenário com 6 sensores. Isso sugere que, para a topologia de rede e os parâmetros de resfriamento utilizados, o Simulated Annealing teve dificuldades em escapar de ótimos locais.

Conclui-se que, para instâncias desta dimensão, o uso de solvers exatos é preferível. O uso da meta-heurística justificaria-se apenas em cenários de escala massiva onde a complexidade combinatorial tornasse inviável a execução do Gurobi, o que não ocorreu neste experimento.

Tabela 4: Resultados Comparativos: Gurobi vs. Python (SA)

# Sensores	Gurobi: 12.0.3		Python: 3.11.4		Gap (SA vs AMPL)
	Objetivo AMPL	AMPL Tempo [s]	Objetivo SA	SA Tempo [s]	
0	0.170387	0.78	0.170387	30.02	0.00%
1	0.088373	5.17	0.116404	30.06	+31.71%
2	0.069729	4.41	0.080979	30.10	+16.13%
4	0.051421	3.91	0.064203	30.10	+24.86%
6	0.041260	3.17	0.095377	30.06	+131.20%
8	0.034856	2.98	0.058865	30.01	+68.87%
10	0.031412	2.81	0.048439	30.01	+54.22%
20	0.022343	2.69	0.033638	30.02	+50.55%

7. Conclusão

Este trabalho abordou o problema de otimização do posicionamento de sensores em redes de distribuição de água, visando minimizar o impacto de contaminações na rede Net3. A metodologia comparou uma abordagem exata, utilizando Programação Inteira Mista via solver Gurobi, com uma abordagem meta-heurística baseada em Simulated Annealing.

A análise dos experimentos computacionais revelou uma superioridade expressiva do método exato para a instância avaliada. O solver Gurobi foi capaz de identificar o ótimo global em todos os cenários, com tempos de execução reduzidos (entre 0,78s e 5,17s). Em contrapartida, o algoritmo de Simulated Annealing, configurado com tempo fixo de aproximadamente 30 segundos, não obteve convergência para o ótimo global, apresentando *gaps* de otimalidade que variaram de +16,13% a +131,20%.

Conclui-se que, para redes de dimensão similar à Net3, a utilização de solvers comerciais é preferível, pois garante a qualidade da solução com baixo custo computacional. A aplicação da meta-heurística justifica-se, portanto, apenas como uma alternativa para cenários de escala massiva ou alta complexidade não linear, onde os métodos exatos podem se tornar inviáveis computacionalmente devido ao crescimento exponencial do espaço de busca.

Como trabalhos futuros, sugere-se a investigação de métodos híbridos (Math-heuristics) ou o ajuste fino (tuning) dos parâmetros de resfriamento e vizinhança do Simulated Annealing para melhorar sua competitividade em instâncias de maior porte.

Referências

- Berry, J., Fleischer, L., Hart, W., Phillips, C., e Watson, J. (2004). A general integer-programming-based framework for sensor placement in municipal water networks. *Research Gate*.
- Berry, J., Hart, W., C., P., Uber, J., e Walski, T. (2005). Water quality sensor placement in water networks with budget constraints. *ASCE Library*.

Kessler, A., Ostfeld, A., e Sinai, G. (1998). Detecting accidental contaminations in municipal water networks. *ASCE Library*.

Ostfeld, A. e Salomons, E. (2004). Optimal layout of early warning detection stations for water distribution systems security. *ASCE Library*.