

# Equação de Bellman Aplicada a Redes Mesh

Ian dos Anjos Melo Aguiar

**Resumo**— Este trabalho apresenta uma adaptação da *Equação de Bellman* aplicada a redes *mesh* unidirecionais, com o objetivo de modelar a propagação e atualização de custos de comunicação de forma distribuída. A formulação proposta emprega heurísticas locais para estimar o custo esperado entre nós, permitindo que cada agente da rede ajuste iterativamente seu valor de custo com base nas informações dos vizinhos, reproduzindo o comportamento de aprendizado típico de métodos de Aprendizado por Reforço.

O código foi inteiramente desenvolvido do zero, garantindo controle total sobre as variáveis, parâmetros e estrutura de atualização. Ao longo do texto são discutidas as modificações implementadas e analisada a influência dos parâmetros  $M$  (penalidade de distância) e  $G$  (ganho por salto) sobre a convergência e o equilíbrio entre número de saltos e distância média entre nós. Os resultados demonstram que a abordagem proposta reproduz de forma estável e interpretável o comportamento adaptativo esperado em redes de comunicação descentralizadas.

## I. DESCRIÇÃO DO PROBLEMA

A diminuição do custo de sensores, aliado ao avanço de *hardwares* acessíveis de telecomunicações, tem viabilizado o monitoramento contínuo de diversas aplicações, desde iluminação pública [1] até previsão meteorológica [2]. Para grandes áreas de sensoriamento, um dos métodos mais eficientes de comunicação é a formação de uma *rede mesh* [3], que permite a transmissão ponto a ponto de pequenos volumes de dados até um destino. Esse destino pode ser um ponto central (*gateway*) ou outro nó da própria rede, através do repasse distribuído de mensagens.

Neste contexto, este trabalho investiga uma abordagem inspirada na **Equação de Bellman** [4] para descrever a propagação de custos em redes *mesh*. Cada nó é tratado como um agente autônomo capaz de ajustar iterativamente seu custo local com base nas informações de seus vizinhos, permitindo a convergência da rede para um estado de custo mínimo equivalente às rotas mais eficientes até o *gateway*. Essa formulação caracteriza um processo de aprendizado distribuído e dinâmico, em que o equilíbrio entre distância e número de saltos emerge naturalmente da interação local entre os nós.

## II. MÉTODOS E IMPLEMENTAÇÃO

A metodologia proposta neste trabalho é dividida em quatro etapas principais: [II-A, II-B] formulação baseada na *Equação de Bellman* e sua adaptação ao contexto de redes *mesh*, [II-C] definição da heurística e da função de custo, e [II-E] aplicação das métricas de avaliação.

Ian dos Anjos Melo Aguiar Faculdade de Engenharia Elétrica e de Computação (FEEC), Universidade Estadual de Campinas (UNICAMP), Campinas-SP. E-mail: 172483@dac.unicamp.br.

### A. Equação de Bellman e sua adaptação para o problema

Este trabalho propõe uma adaptação da *Equação de Bellman* para modelar a comunicação em redes *mesh* unidirecionais, nas quais todos os nós propagam informações até um ponto central (*gateway*), considerando ambientes dinâmicos e restrições de conectividade.

Originalmente formulada no contexto de *Dynamic Programming* [4], a *Equação de Bellman* descreve a relação recursiva entre o valor de um estado e os valores esperados de seus estados sucessores, sendo amplamente utilizada em métodos de *Aprendizado por Reforço* para estimar funções de valor. Na formulação proposta, esse conceito é reinterpretado: o “valor de estado” passa a representar o **custo acumulado de comunicação** associado a cada nó da rede. Cada nó (agente) atualiza iterativamente seu custo local a partir das estimativas de seus vizinhos, buscando minimizar o esforço necessário para alcançar o *gateway*. Esse processo configura uma **propagação inversa de custo**, na qual a informação se difunde a partir do nó base e converge gradualmente até um estado de custo mínimo.

### B. Equação de Atualização

A formulação original da *Equação de Bellman* define o valor de um estado  $V(s)$  como o retorno esperado ao executar uma ação ótima  $a$  e seguir uma política ótima  $\pi^*$ :

$$V(s) = \max_a \left[ R(s, a) + \gamma \sum_{s'} P(s'|s, a) V(s') \right], \quad (1)$$

onde  $R(s, a)$  representa a recompensa imediata,  $\gamma$  é o fator de desconto temporal e  $P(s'|s, a)$  é a probabilidade de transição para o estado sucessor  $s'$ . Essa equação expressa o princípio de otimalidade: o valor de um estado depende da melhor escolha de ação e do valor esperado de seus estados futuros.

No contexto de redes *mesh*, o problema é reformulado de modo inverso. Em vez de maximizar recompensas acumuladas, busca-se **minimizar o custo total de comunicação**. Assim, cada nó é tratado como um agente autônomo que ajusta seu custo local com base nos custos de seus vizinhos, procurando o caminho de menor custo até o *gateway*.

A partir dessa reinterpretação, a Equação 1 é adaptada para o domínio de custo, substituindo o operador de maximização por uma minimização e a recompensa por uma penalidade heurística. A forma resultante é:

$$C_i(t+1) = \alpha C_i(t) + (1 - \alpha) \min_j [C_j(t) + f(d_{ij})], \quad (2)$$

onde:

- $C_i(t)$  é o custo acumulado do nó  $i$  na iteração  $t$ ;
- $\alpha$  é o fator de suavização temporal (SMOOTH\_ALPHA);
- $C_j(t)$  representa o custo do vizinho  $j$ ;
- $f(d_{ij})$  é a heurística que expressa o custo incremental de comunicação entre  $i$  e  $j$ .

A Equação 2 é análoga à atualização de valor (*Value Iteration*), mas invertida: enquanto a forma clássica busca maximizar retornos positivos, aqui o sistema reduz progressivamente o custo acumulado, propagando o valor mínimo a partir do nó base. O fator  $\alpha$  regula a inércia temporal da atualização, e o termo  $f(d_{ij})$ , definido na Seção II-C, orienta a convergência local, determinando a influência da distância e do ganho por salto sobre o custo total.

### C. Função de Custo (Heurística)

Na implementação da classe *PointMeshUnidirectional*, a função `_cost` define a forma concreta da heurística  $f(d_{ij})$ , responsável por orientar o processo de aprendizado local. Essa função foi projetada empiricamente para representar o esforço relativo de comunicação entre dois nós, penalizando conexões longas e favorecendo vizinhos mais próximos.

A equação utilizada é:

$$cost_{i \rightarrow j} = \alpha C_i(t) + (1 - \alpha) \left[ C_j(t) + 1 + \left( \frac{d_{ij}}{M} \right)^2 - G \right], \quad (3)$$

onde:

- $M$  é o fator de escala espacial (METERS\_TO\_POINT), que regula a penalização de distância;
- $G$  é o ganho aplicado a cada salto na rede (GAIN\_JUMP);
- o termo quadrático  $\left( \frac{d_{ij}}{M} \right)^2$  define a penalização não linear da distância.

O termo  $f(d_{ij}) = 1 + \left( \frac{d_{ij}}{M} \right)^2 - G$  constitui o núcleo da heurística de atualização. Embora sua forma não derive de um modelo físico exato, ela reflete o comportamento observado em redes reais, nas quais o custo de comunicação cresce de modo não linear com a distância. Essa escolha garante estabilidade numérica ao processo iterativo e assegura positividade da heurística sempre que  $\left( \frac{d_{ij}}{M} \geq G \right)$ .

### D. Aprendizado Iterativo e Dinâmico

O processo de atualização ocorre de forma contínua e distribuída, caracterizando um modelo de *aprendizado iterativo*, no qual os custos se ajustam dinamicamente conforme a topologia da rede se altera. Essa característica permite que a rede se reorganize automaticamente diante da adição ou remoção de nós, reproduzindo comportamentos adaptativos típicos de sistemas de *Aprendizado por Reforço*. O resultado é uma malha autônoma que converge para estados estáveis, nos quais os menores custos correspondem às rotas de comunicação mais eficientes.

### E. Métricas de Avaliação

Após as simulações, são calculadas métricas que avaliam a eficiência e a estabilidade do processo de convergência. As principais medidas analisadas são:

- **Tempo de convergência:** número de iterações necessárias até que as variações de custo entre iterações sucessivas se tornem desprezíveis;
- **Distância média entre saltos:** média das distâncias entre saltos consecutivos;
- **Saltos médios:** número médio de saltos necessários para alcançar o *gateway*;
- **Pontos conectados:** quantidade de nós efetivamente conectados à rede *mesh*.

Essas métricas permitem quantificar o comportamento global da rede e avaliar a influência dos parâmetros  $M$ ,  $G$  e  $\alpha$  sobre a convergência e a estabilidade do sistema.

## III. RESULTADOS

Os experimentos foram conduzidos variando os parâmetros  $M$  e  $G$ , que controlam, respectivamente, a penalidade quadrática da distância e o ganho associado ao número de saltos. O parâmetro  $M$  regula a influência da distância física na heurística  $f(d_{ij})$ : quanto maior o valor de  $M$ , menor é a penalização de distâncias longas. Já o parâmetro  $G$  atua como um fator de incentivo para redes com maior número de saltos, reduzindo o custo efetivo por salto adicional.

Foram testadas combinações de valores de  $M$  e  $G$ , mantendo o fator de suavização  $\alpha = 0.5$  fixo, e aplicando o modelo a uma topologia fixada com 200 pontos e 1 *Gateway*, apresentadas na Figura 4 (ver Apêndice).

Foram testadas diversas combinações para os parâmetros  $G$  e  $M$  como pode ser visto na Tabela I. Essas combinações permitem avaliar a influência conjunta dos parâmetros  $M$  e  $G$  na convergência da rede e no número médio de saltos até o *gateway*. Espera-se que valores maiores de  $M$  produzam topologias mais diretas (menos saltos e distâncias maiores entre nós), enquanto valores mais altos de  $G$  incentivem formações mais densas e robustas, com maior número de saltos e melhor redundância de comunicação.

TABELA I  
RESULTADOS EXPERIMENTAIS PARA DIFERENTES COMBINAÇÕES DE  $M$  (PENALIDADE) E  $G$  (GANHO DE SALTO).

$M$	$G$	Distância Média	Saltos Médios	Tempo de Convergência	Pontos Conectados
25	0	57.36	5.09	27	103
50	0	71.33	6.31	33	200
75	0	85.24	4.79	28	200
100	0	100.86	3.96	26	200
25	1	52.02	7.19	34	122
50	1	57.25	9.48	41	200
75	1	57.12	8.91	38	200
100	1	58.17	8.62	37	200

## IV. DISCUSSÃO

Os resultados obtidos (Tabela I) revelam dois comportamentos distintos. Primeiro, quando o parâmetro  $G = 0$  (Figura 4),

a rede não se beneficia diretamente da redução no número de saltos até o *gateway*. Nesse caso, o termo heurístico  $f(d_{ij}) = 1 + \left(\frac{d_{ij}}{M}\right)^2$  atua unicamente como penalizador de distância: quanto menor o valor de  $M$ , maior a penalização relativa entre nós. Consequentemente, redes com  $M$  pequeno tendem a apresentar trajetórias mais fragmentadas, com múltiplos saltos curtos que priorizam vizinhos próximos em detrimento de conexões longas.

Por outro lado, quando  $G = 1$  (Figura 5), observa-se o surgimento de um ponto de equilíbrio entre o número de saltos e a distância média entre nós. Nesse regime, o termo  $G$  introduz um ganho associado a cada salto adicional, o que reduz o custo acumulado e favorece a formação de rotas mais densas e conectadas. A análise dos resultados (Tabela I) mostra uma inflexão consistente (Figura 1) para  $M = 75$ , sugerindo que esse valor representa uma condição de equilíbrio entre a penalização de distância e o ganho por salto, produzindo redes que conciliam eficiência de comunicação e estabilidade de convergência.

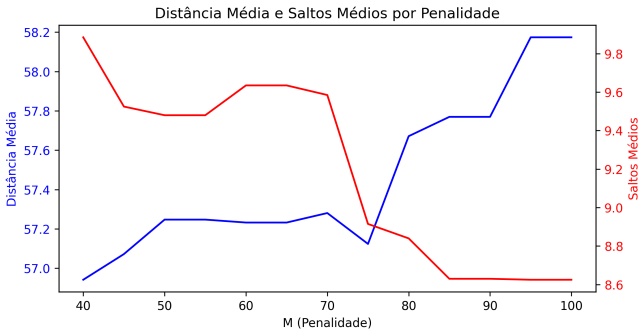


Fig. 1. Relação entre penalidade ( $M$ ) e métricas.

A escolha dessa heurística é justificada por fundamentos físicos observados em redes *mesh*: distâncias menores entre nós implicam menores perdas de sinal e menores taxas de erro de transmissão. Assim, permitir que o sinal percorra múltiplos saltos curtos é, na prática, mais eficiente do que utilizar poucos saltos longos, que tendem a apresentar maior perda e degradação da comunicação.

O processo de aprendizado iterativo tende, portanto, a estabilizar a rede em um estado no qual o custo local é minimizado tanto em termos topológicos quanto de confiabilidade.

Esse comportamento pode ser observado de forma prática nas Figuras 2 e 3, obtidas para  $M = 50$  e  $G = 0$ . Na Figura 2, a distância média entre saltos apresenta instabilidade inicial, seguida por um período de reorganização e posterior estabilização, indicando a convergência do processo de aprendizado. Já a Figura 3 mostra que o número médio de saltos até o *gateway* tende a se estabilizar, confirmando a formação de uma estrutura de comunicação estável e coerente com o comportamento esperado da *Equação de Bellman* adaptada.

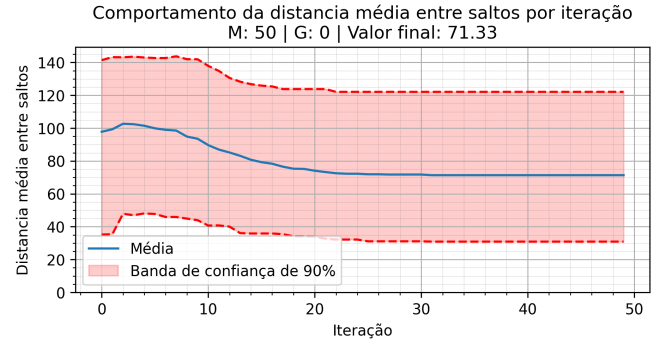


Fig. 2. Evolução da distância média entre saltos para  $M = 50$  e  $G = 0$ .

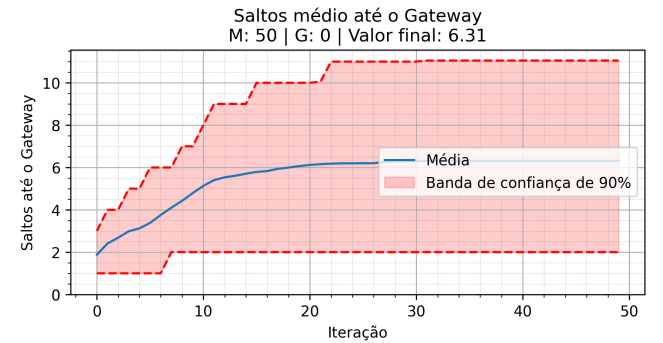


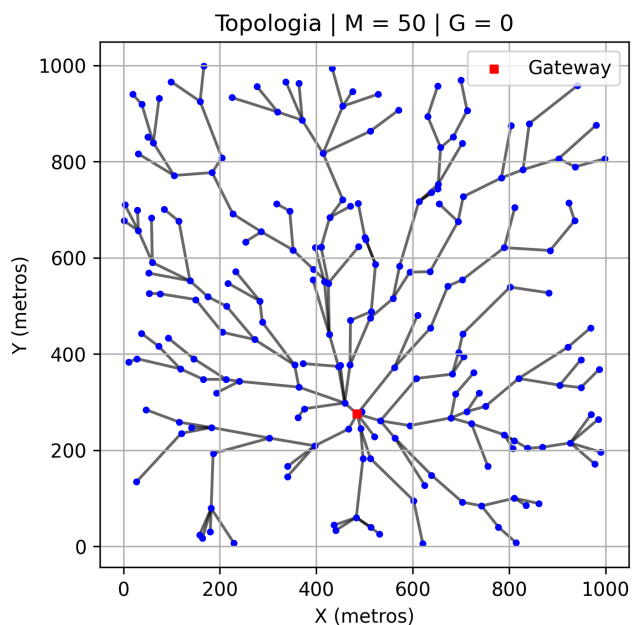
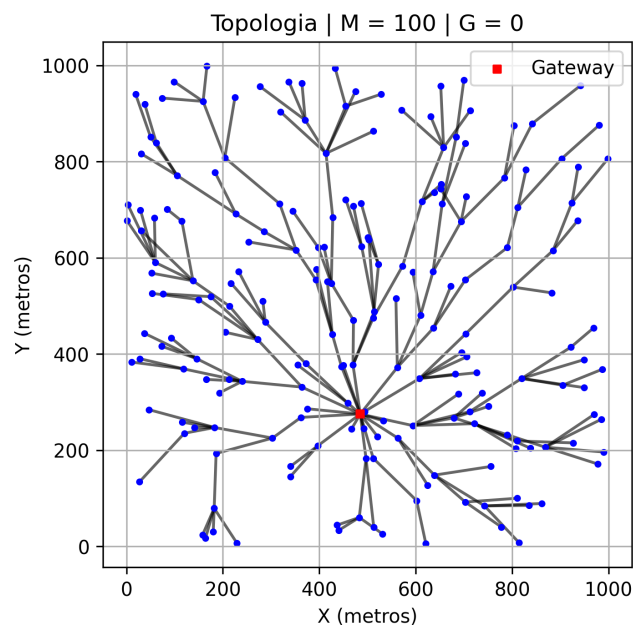
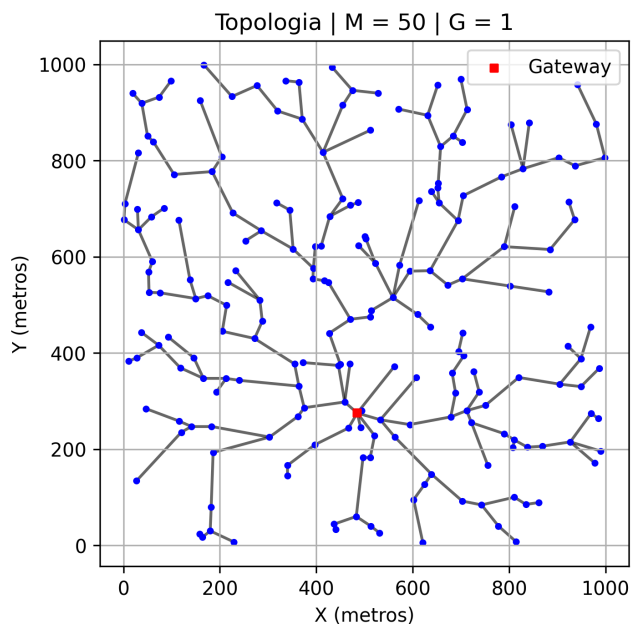
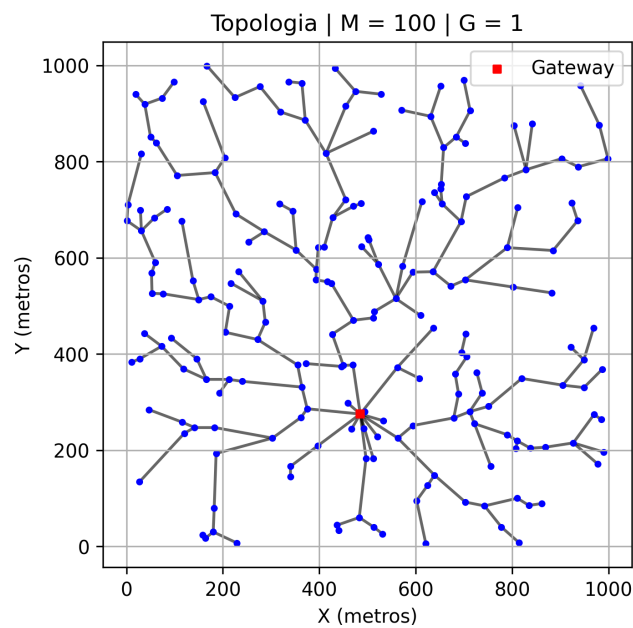
Fig. 3. Evolução do número médio de saltos até o *gateway* para  $M = 50$  e  $G = 0$ .

- [2] T. Akilan and K. Baalamurugan, "Enhanced iot-based weather forecasting and field monitoring system utilizing multiple cnn classification models," in *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. IEEE, 2023, pp. 961–966.
- [3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11 b mesh network," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, 2005, pp. 31–42.
- [4] R. Bellman, "Dynamic programming," *science*, vol. 153, no. 3731, pp. 34–37, 1966.

## REFERÊNCIAS

- [1] M. H. Kabir, A. Al Noman, A. Al Afiq, R. H. Raju, M. N. Hasan, and Ahmad, "Design and implement iot-based intelligent manageable smart street lighting systems for future smart city," *Engineering Proceedings*, vol. 56, no. 1, p. 147, 2023.

## APÊNDICE I

Fig. 4. Topologia e conexões com parâmetros  $M = 50$  e  $G = 0$ .Fig. 6. Topologia e conexões com parâmetros  $M = 100$  e  $G = 0$ .Fig. 5. Topologia e conexões com parâmetros  $M = 50$  e  $G = 1$ .Fig. 7. Topologia e conexões com parâmetros  $M = 100$  e  $G = 1$ .