

Definição e Modelos para o Problema de Localização de Nós Fog e Designação de Demandas de Sensores IoT

Mayron César de Oliveira Moreira^{1,2}, Samuel Moreira Abreu Araújo³,
Geraldo Robson Mateus²

¹Departamento de Ciência da Computação
Universidade Federal de Lavras (UFLA) – Lavras, MG - Brasil

²Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG - Brasil

³Departamento de Tecnologia em Engenharia Civil, Computação,
Automação, Telemática e Humanidades
Universidade Federal de São João Del-Rei (UFSJ) – Ouro Branco, MG - Brasil

mayron.moreira@ufla.br, sabreu@ufs.j.edu.br,

mateus@dcc.ufmg.br

Abstract. *Fog computing emerges from the need to have technologies to deal with the Internet of Things (IoT) efficiently, contrasting with the consolidated cloud computing model. This study proposed two linear integer mathematical models to solve Fog Nodes Location and Demand Assignment of IoT Sensors (FNLDDAS). Considering the limitation of cores and memory of Fog nodes, the objective consists of minimizing the installation costs of Fog nodes and minimizing the makespan. Through experiments performed in instances inspired by real context, we note that both mathematical models achieve optimal solutions in 99% of the scenarios. This demonstrates the potential to integrate them into other solving methods executed in online environments.*

Resumo. *A Fog computing surge da necessidade de ter tecnologias que trabalhem a Internet das Coisas (IoT) de forma eficiente, contrapondo o modelo consolidado de computação em nuvem. Este estudo propõe dois modelos matemáticos lineares e inteiros para a resolução do Problema de Localização de Nós Fog e Designação de Demandas de Sensores IoT (PLNFDDAS). Considerando a limitação de cores e memória dos nós Fog, o objetivo consiste na minimização de custo de instalação dos nós Fog e na minimização do makespan. Através de experimentos realizados em instâncias inspiradas em um contexto real, nota-se que ambos os modelos matemáticos alcançam 99% de soluções ótimas nos cenários testados. Isso demonstra potencial promissor que se tem ao integrá-los em outros métodos de resolução executados em ambientes online.*

1. Introdução

A Internet das Coisas (IoT, *Internet of Things*) já é realidade em contextos tais como Cidades Inteligentes (*Smart Cities*), Saúde Conectada (*Healthcare*), Indústria 4.0, Agricultura de Precisão, Casas Inteligentes (*Smart Homes*), Veículos Autônomos e Logística

e Cadeia de Suprimentos. Trata-se de uma rede de objetos que incorporados a sensores, *software* e outras tecnologias conectam e trocam informações com outros dispositivos pela internet. De acordo com [Oracle 2024], existem mais de 10 bilhões de dispositivos conectados hoje, e a expectativa é que esse número cresça para 22 bilhões em 2025. O aumento da interação entre dispositivos móveis impulsiona o aprimoramento de tecnologias que forneçam respostas que minimizem atrasos e reduzam os custos de operação. Diante disso, o modelo de comunicação direto com a nuvem (modelo de *cloud computing*) [Erl et al. 2013] torna-se ineficiente em situações cujo sistema e/ou o tomador de decisões deve receber as informações em tempo real. Para contornar esse problema, adota-se uma arquitetura que divide a localização dos dispositivos em três camadas, como ilustrado na Figura 1.

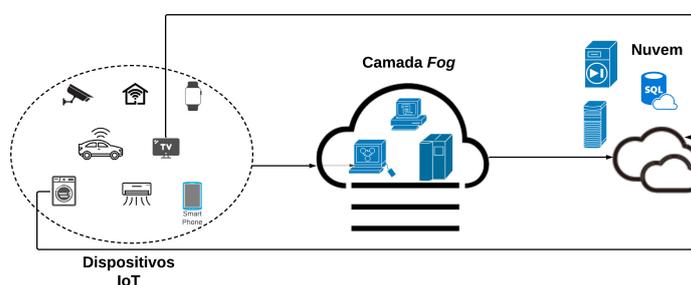


Figura 1. Arquitetura de três níveis: Dispositivos IoT, Nós Fog e Nuvem.

[Bonomi et al. 2012] introduziu o termo *Fog computing*, com o objetivo de criar uma camada complementar à *cloud* que pudesse trazer benefícios ao usuário final. No entanto, de acordo com [Bachiega Jr et al. 2023], não é possível determinar exatamente a delimitação da camada *Fog*, visto que ela pode estar mais próxima dos dispositivos IoT ou da nuvem. Segundo [Iorga et al. 2018], a camada *Fog* possui, em geral, baixa taxa de atraso, distribuição geográfica heterogênea, interoperabilidade, interações em tempo real e escalabilidade. Trata-se de um recurso extremamente útil quando o desempenho da nuvem é insuficiente para o atendimento das demandas dos usuários.

Este estudo aborda um problema de localização de nós *Fog* e designação de recursos aos nós alocados, para o atendimento de requisições advindas de dispositivos IoT. Recentemente, alguns *surveys* tem apresentado o crescente avanço de pesquisas com essa temática, e indicam tendências a serem consideradas em estudos futuros.

[Queiroz et al. 2020] consideram a integração de sensores, nós *Fog* e a nuvem em um problema de designação e alocação de recursos em que sensores enviam requisições à nós *Fog*, e os nós *Fog* repassam tais requisições à nuvem, que por sua vez é representada por um único nó, capaz de receber todas as requisições dos sensores que passam através dos nós *Fog*. Os autores não assumem que haja um envio direto de sensores para a nuvem. De acordo com a descrição do problema, os sensores são mutuamente alcançáveis por qualquer nó *Fog*. Cada sensor envia apenas uma requisição a exatamente um nó *Fog*. Todos os sensores devem ter suas requisições atendidas. O tempo médio de resposta para o atendimento das requisições realizadas pelos sensores é dado pela soma de: (i) atraso médio da designação sensores e nós *Fog*; (ii) atraso médio da alocação das requisições dos nós *Fog* à nuvem; (iii) tempo de processamento das requisições nos nós *fog*. A equação que descreve tal função é não-linear. Um modelo bi-objetivo é proposto, minimizando o

tempo de resposta médio e o número de nós *Fog* ativos, sujeito às restrições de capacidade de processamento dos nós *Fog* e a um limite superior do tempo de resposta levando em conta os Acordos de Nível de Serviço (SLA, *Service Level Agreements*) da rede. Os autores utilizam dados reais da cidade de Modena, na Itália, em instâncias com 50 e 89 sensores, 6 nós *Fog* e um nó da nuvem. Através da resolução via o *software Local Solver*, a otimalidade foi provada em todas as soluções. Todavia, há a necessidade de se redefinir o resolvidor para obter soluções de instâncias maiores.

Neste trabalho propõe-se modelos matemáticos para a resolução do Problema de Localização de Nós *Fog* e Designação de Demandas de Sensores IoT (PLNFDDS). A proposta se baseia no problema reportado por [Queiroz et al. 2020], com as seguintes diferenças e contribuições, apresentadas na sequência.

1. Em contextos reais, é comum que a camada *Fog* seja complementar à nuvem, sendo incapaz de processar todas as requisições. Então, assume-se que algumas requisições podem não ser processadas por nós *Fog* e serão enviadas diretamente à nuvem.
2. Estabelece-se, assim, que as camadas de interesse do problema em questão são as camadas de dispositivos IoT e a *Fog*. Trabalhos que abordam a interação entre tais camadas correspondem a 9,3% do total das publicações entre 2012 e 2022, de acordo com [Bachiega Jr et al. 2023].
3. Levando em conta que todas as requisições serão processadas (na *Fog* ou na nuvem), o foco do estudo proposto consiste em otimizar os recursos da cada *Fog*, visto que o poder computacional dos servidores localizados nessa camada é inferior aos dispositivos da nuvem.
4. Considerando que em redes de tecnologia 5G e 6G (ambiente típico de aplicações IoT) o alcance dos dispositivos não é mútuo, leva-se em conta que os sensores não são capazes de alcançar qualquer nó *Fog*.
5. Considera-se, como limite de capacidade, o número de processadores e a quantidade de memória que cada nó *Fog* possui. Além disso, os nós *Fog* são heterogêneos quanto aos custos e capacidades.
6. Trata-se o problema a partir de duas abordagens mono-objetiva: (i) minimização da soma dos custos dos nós *Fog* ativos e uma penalização da quantidade de sensores cuja demanda não foi atendida; (ii) utilizando como referência o valor obtido em (i), minimiza-se o maior tempo de processamento (conhecido como *makespan*) de uma requisição por um nó *Fog*, penalizando ainda a função objetivo com um termo relativo à não designação de sensores a nós *Fog* ou a nuvem. Os dois objetivos (custo e tempo) são adotados em aproximadamente 50% dos trabalhos, de acordo com [Bachiega Jr et al. 2023].
7. Este estudo trabalha com a minimização do *makespan* tratando-a por meio de uma formulação linear inteira, contornando a não-linearidade inserida no modelo de [Queiroz et al. 2020], o que facilita a resolução do problema.

Com isso, este artigo está em consonância com as recentes tendências de trabalhos futuros indicadas por [Bachiega Jr et al. 2023]. O restante do documento está organizado da seguinte forma. A Seção 2 apresenta recentes *surveys* que abordam a temática de *Fog computing*, além de estudos que utilizam modelos matemáticos aplicados a esse contexto. A Seção 3 define formalmente o PLNFDDS. Os modelos matemáticos propostos

são apresentados na Seção 4. A Seção 5 descreve o ambiente de simulação elaborado, assim como os resultados computacionais acerca das duas formulações implementadas. As conclusões e trabalhos futuros são discutidos na Seção 6.

2. Trabalhos relacionados

[Costa et al. 2023] propõem uma revisão sistemática considerando aspectos relacionados a orquestração em *Fog computing*. O compilado de artigos corresponde ao período 2015-2021. Entre os assuntos pesquisados, constam: (i) os objetivos da orquestração; (ii) as entidades orquestradas; (iii) a topologia de controle do orquestrador; (iv) a arquitetura de camadas cobertas; (v) as funcionalidades do orquestrador. As oportunidades de se avançar no estado da arte envolvendo orquestração se concentram em três contextos: privacidade e segurança de dados; avaliação de orquestradores por meio de simuladores; e padronização do ambiente de execução dos estudos. A revisão sistemática conduzida por [Bachiega Jr et al. 2022] sumariza aspectos referentes à alocação de recursos. O período da busca é de 2012-2022, e os autores sumarizam: (i) métricas; (ii) algoritmos; (iii) arquiteturas; (iv) abordagens de virtualização; (v) ambientes de avaliação dos métodos; (vi) contextos aplicados. Pesquisas futuras contribuirão com o estado da arte através de estudos que agreguem aos nós *Fog* aspectos de mobilidade, outras limitações que impactem na capacidade de processamento, maior heterogeneidade dos dispositivos, limitação da conectividade dos nós *fog*, e testes computacionais com uma maior quantidade de cenários reais.

[Costa et al. 2022] introduzem nova taxonomia considerando aspectos de monitoramento utilizado por orquestradores no ambiente de *Fog computing*. Dez soluções de monitoramento são analisadas. Os autores indicam, como sugestões de próximos passos, pesquisas envolvendo a integração de monitoramento nos níveis de *Fog* e com a nuvem, além da necessidade do desenvolvimento de mais simuladores que consideram a questão do monitoramento. Já [Das and Inuwa 2023] apresentam um compilado de trabalhos que consideram a evolução de arquiteturas de redes, considerando *Mobile edge computing*, *Mobile cloud computing*, *Edge computing*, *Cloud computing* e *Fog computing*. Os autores indicam possibilidades de melhorias em métodos, métricas e em quais contextos cada tipo de rede é mais aplicada. Os dois trabalhos apresentados na sequência se alinham com o PLNFDDS em relação à metodologia proposta para sua resolução (programação linear inteira) e o contexto ao qual se situa (*Fog computing*).

[Asensio et al. 2020] definem o problema de agrupamento de nós *fog* para a nuvem (FCC, do inglês: *Fog to Cloud Clustering problem*). Os nós *fog* recebem requisições de dispositivos da camada IoT, e agrupados para o melhor atendimento às demandas, repassando possivelmente algumas requisições à nuvem. Os nós *Fog* podem ser estáticos ou móveis. A interação entre os nós *Fog* de cada agrupamento é definida pela alocação de um nó *líder*, que gerencia o recebimento da demanda de outros nós *fog*, e as repassa à nuvem. Existe a possibilidade do nó móvel ser aquele ao qual o *backup* dos dados é realizado. Porém, essa possibilidade deve ser evitada na medida do possível. A função objetivo minimiza a soma ponderada dos custos relacionados a: (i) latência; (ii) potência gasta na comunicação de nós *Fog*; (iii) número de nós *Fog* móveis utilizados como *líderes* em seu respectivo agrupamento. Os autores propõem um modelo matemático linear inteiro e um algoritmo para a geração de limite superior baseado no algoritmo *K-means*. Os resultados computacionais executados em instâncias artificiais mostram que a resolução do modelo

matemático pelo *solver* CPLEX é dificultada quando o número de nós *Fog* atinge 80. Por outro lado, a heurística proposta foi superior em todos os cenários. O trabalho indica a necessidade de incorporação ao modelo matemático da escolha de qual nó da nuvem a requisição de cada nó *Fog líder* deve ser alocada.

[Santos et al. 2021] estudam o problema de alocação de Encadeamento de Funções de Serviço (SFC, *Service Function Chaining*) no nível da *Fog*. Os autores levam em conta diferentes tecnologias de Redes de Longo Alcance de Baixa Potência (LPWANs, *Low Power Wide Area Networks*) e múltiplas funções objetivo, a saber: (i) maximização do número de requisições aceitas por usuários; (ii) minimização do número de migrações entre serviços, no decorrer do processo de otimização pelo algoritmo proposto; (iii) minimização do número de nós *Fog* ativos; (iv) minimização do número de *gateways* ativos; (v) minimização da latência da rede; (vi) minimização do tempo de transferência de dados; (vii) minimização do atraso com relação ao usuário. Um modelo matemático foi proposto, e resolvido pelo *solver* CPLEX, utilizando em cada iteração um objetivo a ser minimizado, limitando superiormente os outros objetivos já considerados nas iterações anteriores através do melhor valor encontrado de cada um deles. Os testes foram realizados simulando uma cidade inteligente localizada na Antuérpia, Bélgica. Os resultados do modelo são eficazes na medida que apresentam diferentes soluções para que o tomador de decisões analise a mais apropriada, considerando a função objetivo mais relevante em cada contexto. Os autores sugerem que o desenvolvimento de heurísticas é uma linha de pesquisa adicional a ser considerada para o problema.

Este estudo é similar aos trabalhos de [Asensio et al. 2020, Santos et al. 2021] ao considerar a alocação de recursos em uma arquitetura IoT/*Fog* no contexto de cidades inteligentes. No entanto, ao contrário de [Asensio et al. 2020], não há a premissa de classificação de nós como estáticos e móveis, nem a presença de nós líderes. Além disso, diferentemente de [Santos et al. 2021], o presente estudo não aborda diferentes serviços configurados em SFCs. Desta forma, como mencionado na Seção 1, as contribuições deste artigo se aproximam do artigo de [Queiroz et al. 2020].

3. Definição formal

A arquitetura da rede modelada considera as interações entre sensores e nós *Fog*. Cada sensor possui um tipo de requisição, em quantidade unitária, e pode enviá-la a no máximo um nó *Fog* ativo. Considera-se a possibilidade de que sensores não tenham as suas requisições atendidas pelos nós *Fog* ativos. Nesse caso, assume-se que a nuvem atenderá tais demandas. Seja \mathcal{S} o conjunto de sensores e \mathcal{F} o conjunto de nós *Fog*. Cada requisição de um sensor $s \in \mathcal{S}$ possui diferentes tipos de demanda. Sejam $d_s \in \mathbb{Z}_+$ a quantidade de *cores* demandada pelo sensor s , e $m_s \in \mathbb{R}_+$ a capacidade de memória (em *Mb*) de acesso aleatório demandada por s . Define-se por $\mathcal{A}_s \subseteq \mathcal{F}$ o conjunto de nós *Fog* que podem receber dados do sensor s .

Os sensores e nós *Fog* possuem suas posições geográficas definidas por suas respectivas longitude e latitude. O atraso de comunicação do sensor $s \in \mathcal{S}$ ao nó *Fog* $f \in \mathcal{F}$ é dado por $t_{s,f}^a \in \mathbb{R}_+$. Além disso, considera-se $t_f^p \in \mathbb{R}_+$ como o tempo de processamento gasto no nó *Fog* f . As capacidades de processamento e de memória de acesso aleatório oferecidas no nó *Fog* f são definidas por $\mathcal{D}_f^p \in \mathbb{Z}_+$ (quantidade de *cores*) e $\mathcal{D}_f^m \in \mathbb{R}_+$ (em *Mb*), respectivamente. Por fim, considera-se $\alpha_f \in \mathbb{R}_+$ como o custo da instalação do nó

Fog f . A descrição abaixo sumariza o Problema de Localização de Nós *Fog* e Designação de Demandas de Sensores IoT (PLNFDDS).

Instância: Todos os parâmetros definidos acima.

Solução: Definição de quais nós *Fog* serão localizados, e quais sensores terão sua demanda processada pelos nós ativos, sujeita às restrições de capacidade e cobertura dos sensores.

Objetivos: Minimização de: (i) o custo de instalação dos nós *Fog*; (ii) o tempo máximo de processamento das requisições dos sensores atendidos pela *Fog*; (iii) número de sensores não atendidos pela *Fog*.

4. Modelo matemático

O método de solução proposto para o PLNFDDS consiste na resolução de dois modelos lineares inteiros, que utilizam duas funções objetivo distintas. A Seção 4.1 apresenta a primeira formulação matemática, que minimiza os custos de instalação da rede. Na sequência, a Seção 4.2 mostra o segundo modelo que trabalha com a minimização do tempo máximo (*makespan*), dado o valor da função obtida previamente.

4.1. Modelo 1 (M1): minimização dos custos de localização de nós *Fog*

Seja $x_{sf} \in \{0, 1\}$ uma variável binária igual a 1 se o sensor $s \in \mathcal{S}$ envia requisição para o nó *Fog* $f \in \mathcal{A}_s$. Define-se $z_f \in \{0, 1\}$ como variável binária igual a 1 se o nó *Fog* $f \in \mathcal{F}$ está localizado (ou ativo). A variável auxiliar $0 \leq w_s \leq 1$ assume o valor igual a 1 se o sensor $s \in \mathcal{S}$ não teve sua requisição atendida por algum nó *Fog*. Por fim, considere que $M = \max_{f \in \mathcal{F}} \alpha_f$ seja uma constante suficientemente grande, que corresponde ao custo máximo de um nó *Fog*. O modelo matemático M1 é apresentado a seguir.

$$v^* = \min \sum_{f \in \mathcal{F}} \alpha_f z_f + M \sum_{s \in \mathcal{S}} w_s \quad (1)$$

sujeito a:

$$\sum_{f \in \mathcal{A}_s} x_{sf} = 1 - w_s, \quad \forall s \in \mathcal{S} \quad (2)$$

$$\sum_{s \in \mathcal{S} | f \in \mathcal{A}_s} d_s x_{sf} \leq \mathcal{D}_f^p z_f, \quad \forall f \in \mathcal{F} \quad (3)$$

$$\sum_{s \in \mathcal{S} | f \in \mathcal{A}_s} m_s x_{sf} \leq \mathcal{D}_f^m z_f, \quad \forall f \in \mathcal{F} \quad (4)$$

$$x_{sf} \in \{0, 1\}, \quad \forall s \in \mathcal{S}, \forall f \in \mathcal{A}_s \quad (5)$$

$$z_f \in \{0, 1\}, \quad \forall f \in \mathcal{F} \quad (6)$$

$$0 \leq w_s \leq 1, \quad \forall s \in \mathcal{S}. \quad (7)$$

A função objetivo representada pela equação (1) minimiza o custo total de localização (ou ativação) dos nós *Fog*, acrescida de uma penalização pelo não atendimento a alguma demanda de um sensor. Note que a definição do valor de M é uma forma de forçar

que uma maior quantidade de sensores seja atendida pela camada *Fog*. As Restrições (2) indicam que cada sensor s pode enviar sua requisição a , no máximo, um nó *Fog*. Além disso, caso o sensor s não tenha sua requisição atendida, então a variável w_s assume o valor 1. As limitações da capacidade de *cores* e memória, referentes aos nós *Fog* ativos, são descritas pelas Restrições (3) e (4), respectivamente. As Restrições (5)-(7) estabelecem o escopo das variáveis de decisão. O modelo M1 possui $\mathcal{O}(|\mathcal{S}||\mathcal{F}|)$ variáveis e $\mathcal{O}(|\mathcal{S}|)$ restrições, visto que, em geral, $|\mathcal{F}| < |\mathcal{S}|$.

4.2. Modelo 2 (M2): minimização do tempo máximo de atendimento dos sensores

Considere $\alpha_{ref} \in \mathbb{R}_+$ o custo total máximo permitido para a localização dos nós *Fog*. Além disso, seja $\mathcal{TR}_{max} \in \mathbb{R}_+$ uma variável que mensura o tempo máximo para o processamento de todas as requisições entre sensores e nós *Fog*. O modelo M2 é apresentado a seguir.

$$\min \mathcal{TR}_{max} + M \sum_{s \in \mathcal{S}} w_s \quad (8)$$

sujeito a:

$$(2) - (7), \mathbf{e} \quad (9)$$

$$(t_{sf}^a + t_f^p)x_{sf} \leq \mathcal{TR}_{max}, \quad \forall s \in \mathcal{S}, \forall f \in \mathcal{A}_s \quad (10)$$

$$\sum_{f \in \mathcal{F}} \alpha_f z_f \leq \alpha_{ref} \quad (11)$$

$$\mathcal{TR}_{max} \in \mathbb{R}_+. \quad (12)$$

A função objetivo (8) minimiza o *makespan* referente as designação de sensores a nós *Fog* da rede. O termo adicional na função objetivo indica, novamente, uma priorização no atendimento da demanda dos sensores pela camada *Fog*. Além das Restrições (2)-(7) advindas do modelo M1, considera-se as Restrições (10), que estabelecem o cálculo do *makespan*, levando em conta o tempo de processamento gasto pelo nó *Fog* e o atraso na comunicação com os sensores. As Restrições (11) garantem que o custo da nova rede não ultrapasse um valor pré-estabelecido. Este valor pode ser obtido considerando a solução do modelo M1 ou meramente por um orçamento estabelecido pelo provedor. O domínio da variável \mathcal{TR}_{max} é indicado através da Restrição (12). De forma análoga ao modelo M1, o modelo M2 possui $\mathcal{O}(|\mathcal{S}||\mathcal{F}|)$ variáveis e $\mathcal{O}(|\mathcal{S}|)$ restrições.

5. Ambiente de simulação e experimentos computacionais

Essa seção apresenta um estudo experimental a respeito dos modelos propostos. A Seção 5.1 propõe um esquema de geração de instâncias baseado em dados reais de uma cidade italiana. A Seção 5.2 reporta a qualidade das soluções encontradas na resolução do modelo M1. Na sequência, a Seção 5.3 apresenta a comparação entre custos de localização e tempo máximo para o atendimento da demanda de sensores pela camada *Fog*. Os modelos matemáticos foram implementados em Python 3.10, através da interface do resolvidor comercial Gurobi 10.03. As configurações do Gurobi fixaram sua resolução com 1 *thread*

e tempo limite de 300s (valor comumente utilizado quando se quer testar a eficiência de modelos matemáticos em cenários de grande magnitude, com limite de tempo restrito). A máquina utilizada possui processador Intel Core i7-4770 CPU 3,40 GHz de 8 *cores*, com 8 Gb RAM, sistema operacional Ubuntu 22.04.3 LTS. Os resultados abaixo não reportam estatísticas referentes ao uso da capacidade de memória nos nós *Fog* (Restrição (4)), visto que a alocação de sensores a estes ocupou uma quantidade pequena (menos de 1%) de memória do nó *Fog* mais carregado. Assim, desconsideramos tal informação da análise final. Os resultados detalhados estão disponíveis em [Moreira et al. 2024].

5.1. Configuração de ambiente de simulação

As instâncias geradas são inspiradas na base de dados proposta por [Queiroz et al. 2020]. Trata-se de um projeto de cidade inteligente proposto para Modena, localizada na província de Emilia-Romagna, com população aproximada de 180 mil habitantes. Os sensores são dispositivos sem-fio que coletam informações da movimentação de pedestres e de veículos automotivos, e enviam imagens a serem processadas pelos nós *Fog*.

Assume-se que o atraso entre um dado sensor e um nó *Fog* depende da distância física entre eles. Particularmente, os autores utilizam a distância *Haversine* como referência para o cálculo do atraso. Tais dados alinham-se com a formulação proposta neste trabalho, visto que [Queiroz et al. 2020] consideram apenas um único servidor na nuvem, preocupando-se, de fato, com a otimização de recursos dos nós *Fog*. Considera-se 20 instâncias de cada combinação de quantidade de sensores e nós *Fog*, classificando-as a partir de seu número de sensores em “Baixa”, “Média” e “Alta” demandas. Tais combinações são apresentadas a seguir, por meio da notação $(|\mathcal{S}|, |\mathcal{F}|)$: (i) “Baixa” - (50, 5), (89, 6); (ii) “Média” - (100, 5), (100, 10), (100, 20), (200, 4), (200, 10), (200, 20), (500, 5), (500, 10), (500, 25), (500, 50); (iii) “Alta” - (1000, 10), (1000, 20), (1000, 50), (1000, 100). Cada combinação possui 20 problemas teste, totalizando 320 instâncias.

Os valores de latitude e longitude dos sensores e nós *Fog* foram extraídos de [Queiroz et al. 2020]. Além disso, na geração das novas instâncias, utilizou-se os mesmos valores referência de atrasos entre os dispositivos. A demanda de processamento de cada sensor foi medida em termos do número de *cores* necessários para processar uma imagem. Cada sensor pode enviar uma requisição de um único tipo, que requer um *core* do processador do nó *Fog* que a recebe. A requisição enviada por um sensor exige 10 Mb de espaço no nó *Fog*. Tal valor se baseou na geração de dados do trabalho de [Silva et al. 2019], que também utilizam o processamento de imagens em cidades inteligentes como cenário de testes. Os nós *Fog* gastam 50ms para o processamento da imagem recebida. As possibilidades de escolha de servidores *Fog*, adotadas nos experimentos, são apresentadas na Tabela 1.

Nas instâncias em que a quantidade de nós *Fog* é menor ou igual a quantidade de opções de servidores, sorteia-se, por distribuição uniforme, $|\mathcal{F}|$ configurações distintas da Tabela 1. Caso contrário, escolhe-se com reposição de configuração a quantidade necessária de servidores *Fog*. A cobertura (ou alcançabilidade) do envio de requisições de um sensor $s \in \mathcal{S}$ para nós *Fog*, denotada por \mathcal{A}_s , é definida de três formas.

Cobertura 1 - Cob1 Indica que $\gamma \in [1, |\mathcal{F}|]$ nós *Fog* serão escolhidos aleatoriamente como cobertos para cada sensor s .

Tabela 1. Especificação de servidores fog utilizados nos testes. Fonte: [CISCO 2023, Intel 2023, Ito 2023].

Produto	Freq. (GHz)	Cores	Memória (Gb)	Custo (US\$)
HX-CPU-I6326	2.9	16	256	2.790,95
HX-CPU-I6314U4	2.3	32	256	4.500,00
HX-CPU-I6312U5	2.4	24	256	2.868,24
HX-CPU-I5320T	2.3	20	256	3.710,65
HX-CPU-I5320	2.2	26	256	3.313,77
HX-CPU-I5318N	2.1	24	256	2.997,10
HX-CPU-I5318S	2.1	24	256	2.803,81
HX-CPU-I5318Y	2.1	24	256	2.803,81
HX-CPU-I5317	3.0	12	256	2.089,75
HX-CPU-I5315Y	3.2	8	256	1.968,93
HX-CPU-I4316	2.3	20	256	2.203,35
HX-CPU-I4314	2.4	16	256	1.559,21
HX-CPU-I4310T	2.3	10	256	1.218,88
HX-CPU-I4310	2.1	12	256	1.100,13
HX-CPU-I4309Y	2.8	8	256	1.100,13

Observação: HX-CPU-I6314U4: preço estimado; HX-CPU-I6312U5: preço definido pela média dos preços de servidores Fog da tabela, com 24 cores.

Cobertura 2 - Cob2 Indica que os $\gamma \in [1, |\mathcal{F}|]$ nós Fog com a menor distância *Haversine* em relação ao sensor s serão escolhidos como cobertos em relação a s .

Cobertura 3 - Cob3 Indica que os $\gamma \in [1, |\mathcal{F}|]$ nós Fog com a menor distância *Euclidiana* em relação ao sensor s serão escolhidos como cobertos em relação a s .

Portanto, variando para cada instância original três formas de se gerar uma cobertura, tem-se o total de $320 \times 3 = 960$ instâncias¹.

5.2. Resultados do Modelo M1

O modelo matemático M1 foi testado considerando as instâncias descritas na Seção 5.1, com as três estratégias para a geração de cobertura. Para a cobertura *Cob1*, o resolvidor Gurobi conseguiu provar 100% de otimalidade em todos os grupos de instâncias, a exceção do grupo $(|\mathcal{S}|, |\mathcal{F}|) = (1000, 100)$, em que em 30% dos casos (6 instâncias), não foi possível comprovar a otimalidade das soluções em 300s. Nesses exemplos, o *gap* médio obtido foi igual a 0,12% e desvio padrão do *gap* equivalente a 0,07%. Considerando a cobertura *Cob2*, o desempenho para a resolução do modelo foi similar. A exceção de 4 instâncias (20%) do grupo $(|\mathcal{S}|, |\mathcal{F}|) = (1000, 100)$, todos os outros exemplos tiveram a otimalidade provada pelo Gurobi. Os 4 casos em que a otimalidade não foi provada em 300s alcançaram *gap* médio de otimalidade igual a 0,10% e desvio padrão do *gap* equivalente a 0,04%. Os testes com a *Cob3* reportaram apenas 3 instâncias (15%) sem a otimalidade provada, pertencentes ao mesmo grupo citado anteriormente. Nesses casos, *gap* médio foi igual a 0,08% e desvio padrão do *gap* equivalente a 0,03%.

¹Instâncias disponíveis em: <https://acesse.one/Ugaw6>.

Em todas as variações de cobertura, os tempos gastos para a resolução das instâncias foram menores que 1 segundo, exceto para os grupos $(|\mathcal{S}|, |\mathcal{F}|) = (500, 50)$ e $(|\mathcal{S}|, |\mathcal{F}|) = (1000, 100)$, cuja execução média variou de 10s e 137s. Diante de tempos reduzidos na grande parte dos cenários, pode-se afirmar que a resolução via programação linear inteira possui potencial como método de solução ou parte de um algoritmo em alguns contextos de alocação de recursos que envolvem *Fog computing*, onde a escalabilidade é um requisito importante. As Figuras 2-3 analisam outros aspectos quantitativos a respeito das soluções, referente às três coberturas consideradas. Pelo que será visto a seguir, nota-se que as soluções obtidas tiveram valores próximos, independente da cobertura analisada. Na Figura 2, mensura-se o percentual médio de sensores cuja requisição foi atendida pelos nós *Fog* alocados. Como na função objetivo (1) existe um termo que prioriza o atendimento dos sensores pelos nós *Fog*, devido à definição do valor de M , o gráfico mostra que na maioria dos testes, esse percentual se situa acima dos 60%. O cenário que apresenta o menor atendimento das demandas (de 35% a 40%) dos sensores pela *Fog* consiste na combinação $(|\mathcal{S}|, |\mathcal{F}|) = (200, 4)$, a qual a maior parte das requisições dos sensores vai para a nuvem.

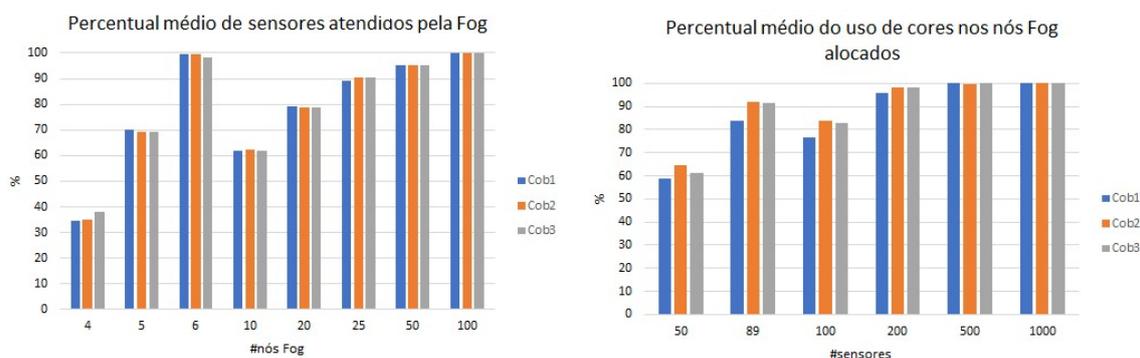


Figura 2. Atendimento da demanda de sensores e percentual do uso de cores nos nós *Fog*.

A mesma figura ilustra o percentual médio de ocupação dos *cores* dos nós *Fog* alocados em relação a designação das demandas dos sensores. Nos cenários de baixa demanda (50 e 89 sensores), nota-se uma folga de aproximadamente 40% do número de *cores* utilizados. Isso indica que, nesses casos, existe uma concentração maior de carga de trabalho em alguns nós *Fog* da rede. Todavia, a medida que o tamanho da rede fica maior, esse valor vai se aproximando a 100% do uso dos processadores, visto que a abordagem adotada pelo modelo M1 tenta alocar, ao máximo, as requisições na *Fog*. A Figura 3 mostra um comportamento esperado nos cenários testados: a medida que a quantidade de sensores aumenta, o custo da rede cresce, devido a um maior número de nós *Fog* alocados. O crescimento da função de custo é praticamente a mesma para as diferentes formas de alcançabilidade aos nós *Fog* (*Cob1*, *Cob2* e *Cob3*).

Analisa-se na Figura 3 a quantidade de nós *Fog* ativados. Nos cenários de baixa demanda, utiliza-se em torno de 60% a 70% dos candidatos a serem servidores na *Fog*. Com o aumento do número de sensores, é esperado que a quantidade necessária de localização de nós *Fog* cresça, englobando valores de 90% a 100%. Assim, de acordo com a abordagem considerada, assume-se um custo maior da instalação da rede em troca de taxas menores de atrasos na transferência de dados entre sensores e nós *Fog*.

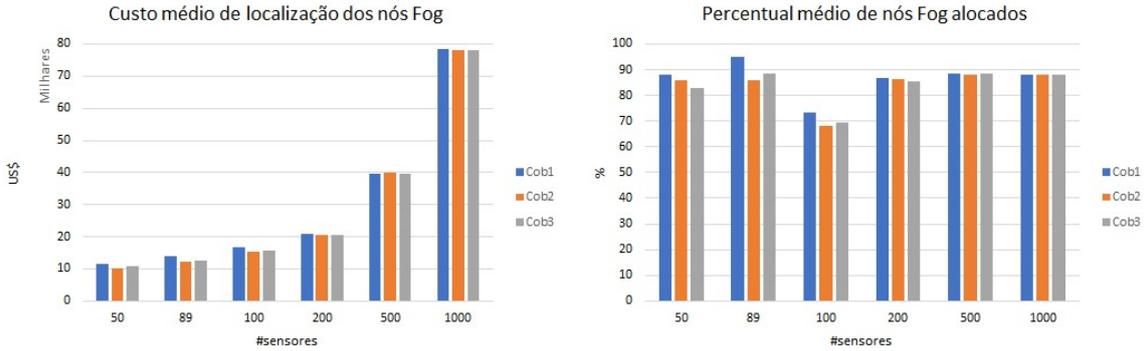


Figura 3. Percentual de uso de nós Fog e custo médio de localização.

5.3. Resultados do Modelo M2

A formulação proposta M2 visa a minimização do *makespan* em relação ao processamento das demandas dos sensores. Para que a nova rede a ser planejada conserve um custo reduzido ao priorizar a redução dos atrasos, adota-se o valor de referência $\alpha_{ref} = \beta v'$ para cada instância testada, em que v' é o valor de v^* (função (1)) sem o termo de penalidade relativa à não designação de sensores a nós Fog. Nos experimentos realizados, considera-se $\beta \in \{1, 0; 1, 05; 1, 10\}$. Assim, para $\beta = 1, 0$, pretende-se minimizar o *makespan* e obter um custo de rede igual ao obtido pela melhor solução da resolução do modelo M1. Por outro lado, ao adotar $\beta = 1, 10$, o tomador de decisões paga o preço de aumentar em 10% o custo de localização de nós Fog em detrimento de se reduzir ainda mais o atraso entre sensores e servidores da Fog.

Nos experimentos realizados, reporta-se tal compromisso entre esses dois objetivos através da Figura 4 e a Tabela 2. Primeiramente, compara-se o aumento do custo das soluções considerando a média dos resultados de todas as formas de coberturas, com os modelos M1, M2 com $\beta = 1, 05$ (M2(5%)) e M2 com $\beta = 1, 10$ (M2(10%)). Como esperado, as abordagens do modelo M2 utilizam da liberdade de perder a redução quanto ao custo para reduzirem o *makespan*. A diferença de custo entre as soluções do modelo M1 e do modelo M2(10%) fica em torno de US\$ 1.000,00, com um pequeno decréscimo nos valores absolutos em relação à Cob2.

A média dos valores de *makespan* são apresentadas na Tabela 2. As métricas *md* e *sd* são as médias e desvio padrão do *makespan*, respectivamente. O tempo de processamento das requisições enviadas pelos sensores (50 milissegundos) é dominante no resultado final. Ao analisar a parcela fracionária das médias indicadas, nota-se que diferentes designações de sensores a nós Fog foram realizadas, visto que houve uma redução do atraso em relação ao desvio padrão dos tempos. O modelo M2 encontrou solução ótima em quase todas as instâncias, a exceção de uma instância com alta demanda de sensores, cujo *gap* de otimalidade resultante, após 300s, foi de aproximadamente 90%. À exceção desse exemplo, o tempo computacional de execução foi de até 1s. O desvio padrão dos atrasos foram menores que a proposta M1, apesar da média se manter igual, no geral. Isso pode ser explicado pela pequena diferença de atraso nas instâncias originais de [Queiroz et al. 2020], visto que os pontos de localização dos sensores e nós Fog são relativamente próximos, e a rede de Modena é densa. Mesmo com valores de média e desvio padrão próximos, a hipótese de nulidade estatística das médias dos atrasos de ambos

os modelos foi descartada, através do Wilcoxon teste com 95% de nível de confiança.

Assim, pode-se afirmar que a abordagem do modelo M2 é promissora para projetos de redes em que, em duas etapas, planeja-se redução de custos e aumento da eficiência do processamento das informações.

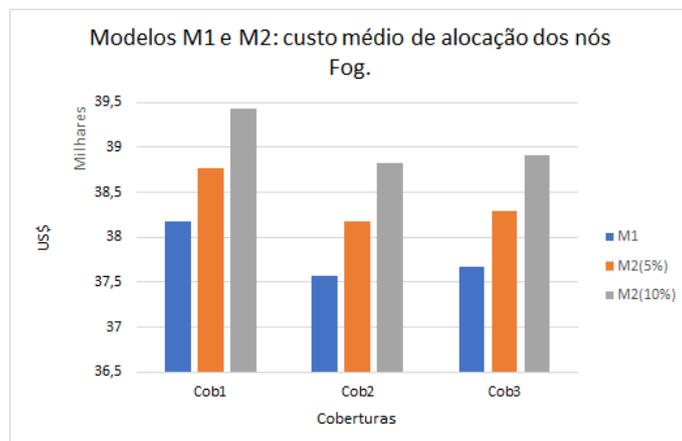


Figura 4. Modelos M1 e M2: custo médio de localização dos nós *Fog*.

Tabela 2. *Makespan*: comparação dos modelos M1 e M2.

Cobertura	Tol. (β)	Modelo M1 <i>md</i> \pm <i>sd</i>	Modelo M2 <i>md</i> \pm <i>sd</i>
Cob1	0,0	50,9 \pm 3,0	50,9 \pm 3,0
	0,05	50,9 \pm 3,3	50,8 \pm 2,9
	1,0		50,8 \pm 2,8
Cob2	0,0	50,6 \pm 2,4	50,6 \pm 2,4
	0,05	50,6 \pm 2,5	50,6 \pm 2,1
	0,1		50,5 \pm 2,0
Cob3	0,0	50,6 \pm 2,3	50,6 \pm 2,3
	0,05	50,6 \pm 2,5	50,5 \pm 2,1
	0,1		50,5 \pm 2,0

6. Conclusões e trabalhos futuros

Este artigo estuda o Problema de Localização de Nós *Fog* e Designação de Demandas de Sensores IoT (PLNFDDS), encontrado em contextos que lidam com IoT e tecnologias recentes, como 5G e 6G. Neste estudo, considera-se a interação entre as camadas de dispositivos IoT (representada por sensores) e a camada *Fog*. Cada sensor envia uma requisição unitária de um tipo, que pode ser atendida pela *Fog* ou tratada na nuvem. O problema é resolvido através de programação matemática, com a formulação de dois modelos matemáticos: (i) M1, minimizando os custos de localização dos nós *Fog*; (ii) M2, minimizando o *makespan* da localização dos sensores na *Fog*, dado pelo tempo de processamento de cada demanda por cada nó *Fog* e o atraso de comunicação entre sensor e o seu respectivo nó *Fog* designado. Em ambas as modelagens, considera-se um termo

adicional à função objetivo, como forma de priorizar designação de todas as requisições dos sensores aos servidores da *Fog* a fim reduzir atrasos.

Para a avaliação das propostas, foram geradas instâncias baseadas nos dados reais de [Queiroz et al. 2020], mensurados a partir de um projeto inteligente da cidade italiana de Modena. O modelo M1 foi resolvido com muita eficiência e, mesmo nos casos em que não foi possível provar a otimalidade em 300s, os *gaps* foram baixos (menores que 0,12%). Isso indica que, possivelmente, a melhor solução inteira encontrada é a ótima. A partir dos custos ótimos obtidos da resolução de M1, executou-se o modelo M2, minimizando o *makespan* e permitindo uma perda de até 10% do custo ótimo para a minimização dos atrasos advindos da designação de sensores a nós *Fog*. A formulação em questão foi resolvida de forma eficiente pelo Gurobi.

Existem três linhas de investigação possíveis como trabalhos futuros: (i) extensão do PLNFDDS para que considere um ambiente com diferentes tipos de requisições para cada sensor, e que cada sensor possua diferentes quantidades de requisição. Além disso, o cenário estudado pode ser dividido em vários períodos. Essa variante também poderia incluir dinamicidade e heterogeneidade dos entes envolvidos na modelagem, o que exigiria que a restrição de memória fosse mais utilizada; (ii) estudar a integração de modelos de simulação com processos decisórios baseados em tomadas de decisão via otimização combinatória no contexto de *Fog computing*. (iii) incluir diretamente a possibilidade da designação de requisições de sensores a um conjunto de nós da nuvem, e parâmetros como largura de banda e taxa de transferência na formulação.

Agradecimentos

Os autores agradecem ao CNPq, CAPES e FAPEMIG.

Referências

- Asensio, A., Masip-Bruin, X., Durán, R. J., de Miguel, I., Ren, G., Daijavad, S., and Jukan, A. (2020). Designing an efficient clustering strategy for combined fog-to-cloud scenarios. *Future Generation Computer Systems*, 109:392–406.
- Bachiega Jr, J., Costa, B., and Araujo, A. P. F. (2022). Computational perspective of the fog node. *arXiv preprint arXiv:2203.07425*, pages 1–8.
- Bachiega Jr, J., Costa, B., Carvalho, L. R., Rosa, M. J., and Araujo, A. (2023). Computational resource allocation in fog computing: A comprehensive survey. *ACM Computing Surveys*, 55:336:1–336:31.
- Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–15.
- CISCO (2023). Cisco edge servers. <https://11nq.com/YNVhf>. Acessado em 12 de janeiro de 2024.
- Costa, B., Bachiega Jr, J., Carvalho, L. R., Rosa, M., and Araujo, A. (2022). Monitoring fog computing: A review, taxonomy and open challenges. *Computer Networks*, 215:1–19.
- Costa, B., Bachiega Jr, J., Carvalho, L. R. D., and Araujo, A. P. (2023). Orchestration in fog computing: A comprehensive survey. *ACM Computing Surveys*, 55:29:1–29:34.

- Das, R. and Inuwa, M. M. (2023). A review on fog computing: Issues, characteristics, challenges, and potential applications. *Telematics and Informatics Reports*, 10:1–20.
- Erl, T., Puttini, R., and Mahmood, Z. (2013). *Cloud Computing: Concepts, Technology & Architecture*. Pearson Education, 1 edition.
- Intel (2023). Intel processors. <https://11nq.com/89NIV>. Acessado em 12 de janeiro de 2024.
- Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N., and Mahmoudi, C. (2018). Fog computing conceptual model. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf>. Acessado em 12 de janeiro de 2024.
- Ito (2023). Ito solutions - cisco edge servers. <https://www.itosolutions.net/Cisco-Intel-Xeon-Gold-3rd-Gen-5317-Dodeca-core-p/hx-cpu-i5317.htm>. Acessado em 12 de janeiro de 2024.
- Moreira, M. C. O., Araújo, S. M. A., and Mateus, G. R. (2024). Resultados detalhados do artigo "definição e modelos para o problema de localização de nós fog e designação de demandas de sensores iot". <https://acesse.one/Ugaw6>. Acessado em 12 de janeiro de 2024.
- Oracle (2024). O que é iot? <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acessado em 08 de janeiro de 2024.
- Queiroz, T. A. D., Canali, C., Iori, M., and Lancellotti, R. (2020). A location-allocation model for fog computing infrastructures. In *Proceedings of CLOSER*, pages 1–10.
- Santos, J., Wauters, T., Volckaert, B., and Turck, F. D. (2021). Towards end-to-end resource provisioning in fog computing over low power wide area networks. *Journal of Network and Computer Applications*, 175.
- Silva, P., Costan, A., and Antoniu, G. (2019). Investigating edge vs. cloud computing trade-offs for stream processing. In *Proceedings of IEEE International Conference on Big Data*, pages 469–474.