

Método multicritério híbrido utilizando AHP e PROMETHEE II para o problema de seleção de provedores *serverless*

Kelwin Efrain Bagnhuk da Silva¹, Adriano Fiorese¹

¹Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina (UDESC)
Joinville, Brasil

kelwinbagnhuk@gmail.com, adriano.fiorese@udesc.br

Abstract. *The serverless paradigm has become increasingly popular, expanding provider offerings and complicating the choice of the ideal platform. This paper proposes a three-layered hybrid method for selecting serverless infrastructures. The approach integrates filtering by Performance Indicators (PIs), the AHP method to translate personal preferences into global weights, and PROMETHEE II for final ranking. Time and accuracy experiments validated the model, ensuring fast, adaptable performance with high precision.*

Resumo. *O paradigma serverless tem-se popularizado cada vez mais e isso ampliou a oferta de provedores, dificultando a escolha de uma plataforma ideal. Este artigo propõe um método híbrido de três camadas na seleção de infraestruturas serverless, integrando uma filtragem por Indicadores de Desempenho (PIs), o método AHP para traduzir preferências pessoais em pesos globais e o PROMETHEE II para ordenação final. Experimentos de tempo e acurácia validaram o modelo, garantindo desempenho computacional rápido, adaptável e com alta precisão.*

1. Introdução

A consolidação da computação em nuvem juntamente com a crescente adoção do modelo *serverless* transformaram as estratégias de desenvolvimento de software. No entanto, essa mudança arquitetural derivou um cenário complexo no mercado devido à vasta quantidade de provedores e plataformas *serverless*. Frequentemente, clientes enfrentam o problema da seleção de um provedor ideal que atenda aos seus requisitos técnicos e estruturais.

Como uma ferramenta matemática de suporte para a resolução de problemas com múltiplas opções conflitantes (mais adequado provedor *serverless*), a literatura acadêmica e científica faz o uso frequente dos Métodos de Análise de Decisão Multicritério (MADM), ou popularmente conhecidos pelo termo em inglês *Multi Criteria Decision Analysis* (MCDA). Alguns dos algoritmos mais populares são o *Analytic Hierarchy Process* (AHP) e o *Preference Ranking Organization Method for Enrichment Evaluations* (PROMETHEE) [Basílio et al. 2022].

Dessa maneira, com o objetivo de mitigar o desafio de selecionar o provedor *serverless* ideal e evitar a sobrecarga diante de múltiplas opções, este trabalho propõe um método de decisão híbrido estruturado em três camadas lógicas. O fluxo de execução inicia-se com uma filtragem adaptável, baseada em Indicadores de Desempenho (PIs), que remove todos os provedores que violam as preferências e restrições arquiteturais do

cliente. Em seguida, o método AHP é executado de forma parcial, dedicado exclusivamente à extração das preferências e dos pesos do decisor sobre os critérios que estão sendo analisados. Por fim, o algoritmo PROMETHEE II é empregado sobre as alternativas que restaram da etapa inicial, gerando o ranqueamento final e exato.

Dessa forma, este artigo está estruturado e organizado nas seguintes seções: A Seção 2 apresenta a metodologia do método híbrido proposto. A Seção 3 apresenta e discute os testes computacionais envolvendo a implementação híbrida. Por fim, a Seção 4 apresenta as considerações finais e sugere temas para trabalhos futuros.

2. Metodologia do Método Proposto

A arquitetura divide-se em três etapas operacionais. A camada inicial atua como um mecanismo de filtragem inspirado em [de Moraes and Fiorese 2017], utilizando Indicadores de Desempenho (PIs) para analisar e eliminar as alternativas que não satisfazem as restrições mínimas exigidas pelo projeto e pelo decisor. Esse processo é adaptável e pode se adequar às condições de diversos cenários de uso. Na etapa seguinte, a modelagem aplica o *Analytic Hierarchy Process* (AHP) de modo restrito aos critérios analisados, garantindo uma extração consistente e coerente dos pesos globais de importância, refletindo as preferências do decisor. Finalmente, o modelo processa apenas o subconjunto de opções aprovadas na primeira fase, utilizando os pesos extraídos na segunda, submetendo-as ao algoritmo PROMETHEE II para o cálculo das relações de superação (*outranking*) e a formulação do ranqueamento final.

2.1. Estruturando o Problema

Para este trabalho, os critérios técnicos que definem a capacidade de um provedor *serverless* foram modelados a partir de intervalos de valores providos por meio de *benchmarks* [Copik et al. 2021], juntamente com a estruturação da fundamentação teórica consolidada em [Rittes and Fiorese 2025]. Entre os atributos determinantes, destacam-se as severas limitações arquiteturais (como o teto de Memória Máxima e o Limite de Concorrência por função) e os gargalos de latência, como o tempo de *Cold Start* [Eismann et al. 2021].

Para que o método híbrido proposto consiga adaptar todos os comportamentos das diversas métricas utilizadas, a implementação converte essas avaliações singulares em Indicadores de Desempenho (PIs), modelados com base na literatura de [Jain 1991]. Essa lógica segmenta o problema em três tipos de processamento, categorizando cada critério com base em como a sua utilidade é maximizada: [de Moraes and Fiorese 2017]:
a) Maior é Melhor (*Higher is Better* ou HB): A meta analítica é a maximização do valor. No contexto *serverless*, são métricas em que, quanto maior a oferta, mais vantajosa será para o cliente.
b) Menor é Melhor (*Lower is Better* ou LB): Tipificam métricas cujo valor menor é melhor sob um determinado ponto de vista (nesse caso do cliente).
c) Valor Nominal (*Nominal is Better* ou NB): Tipifica métrica cujo valor é determinado e específico.

Assim, é o usuário ou o cliente (decisor) que irá definir as importâncias globais de cada critério durante a avaliação e a análise das plataformas *serverless* disponíveis. Nesse contexto, certos critérios podem ser classificados como estritamente *essenciais*, admitindo-se uma porcentagem fixa de tolerância em relação aos valores oferecidos por cada alternativa. São esses requisitos iniciais que estipulam as regras da avaliação e do

ranqueamento, preparando os dados brutos para alimentar a primeira camada de triagem do método.

2.2. Filtragem

O estágio inicial do método híbrido foi projetado para funcionar como um mecanismo adaptável de corte lógico das alternativas disponíveis. Dessa maneira, esta camada elimina as opções incompatíveis com os desejos do decisor antes que o algoritmo avance para a etapa de cálculos densos de avaliação, evitando, assim, uma sobrecarga computacional desnecessária.

Essa estratégia opera de maneira flexível sobre as entradas fornecidas na etapa anterior (relativa aos critérios). O filtro é executado exclusivamente sobre os Indicadores de Desempenho que o decisor classificou como essenciais, isto é, aqueles critérios que são vitais e cruciais para o desenvolvimento da avaliação. Para realizar a filtragem, o algoritmo exige duas variáveis estipuladas pelo usuário para cada critério essencial: o valor ideal desejado (y) e a margem de tolerância de desvio aceitável (t). Durante o funcionamento dessa fase, o valor real ofertado por cada alternativa (x) é comparado com os requisitos do decisor. Para critérios de maximização (tipo Maior é Melhor), uma alternativa é descartada quando não atinge o limite mínimo aceitável, definido como ($y - t$). Nos critérios de minimização (tipo Menor é Melhor), o descarte ocorre quando o valor da alternativa ultrapassa o limite máximo ($y + t$). Já para critérios nominais, a exigência é que o valor esteja contido no intervalo $[y - t, y + t]$, penalizando tanto a insuficiência quanto o excesso.

Por fim, ao término dessa camada, todos os provedores *serverless* que não atendem as equações e aos parâmetros exigidos em seus PIs caracterizados como essenciais são removidos da base de dados ativa em análise.

2.3. Extração de Pesos via AHP

O estágio intermediário da implementação híbrida introduz o *Analytic Hierarchy Process* (AHP) [Saaty 1990], um dos métodos MADM mais utilizados na literatura acadêmica. A implementação híbrida proposta adapta o uso do AHP, executando-o de maneira estrita e isolada sobre a camada de critérios. O decisor é solicitado a utilizar a escala de Saaty (valores de 1 a 9) para julgar a importância relativa de cada Indicador de Desempenho (PI), transferindo para o AHP a responsabilidade de estabelecer o vetor de pesos globais dos requisitos específicos do projeto. A partir dessas entradas diretas, o sistema constrói uma matriz quadrada e recíproca. Na extração matemática dos pesos dos critérios (PIs dos provedores *serverless*), o método proposto adota a aproximação por normalização de colunas seguida da média aritmética das linhas, procedimento amplamente aceito na literatura como simplificação do método do *eigenvector* original, com resultados equivalentes para matrizes consistentes.

2.4. PROMETHEE II

A terceira e última camada do método híbrido representa a etapa principal de avaliação do sistema, introduzindo o método *Preference Ranking Organization Method for Enrichment Evaluations* (PROMETHEE) [Brans et al. 1986]. Enquanto a etapa anterior utiliza o AHP com o intuito de modelar a subjetividade de um usuário e extrair os pesos dos critérios,

o PROMETHEE assume uma postura analítica e quantitativa para avaliar o desempenho dos provedores *serverless* restantes da filtragem inicial. A abordagem desenvolvida adota estritamente a Função de Preferência Gaussiana (Tipo VI), do método PROMETHEE, para todos os critérios, na qual o seu único parâmetro é o desvio padrão populacional (σ). Ao final adota-se a abordagem PROMETHE II para cálculo do ranqueamento e ordenação determinística dos provedores *serverless*.

3. Experimentos e Resultados

Os experimentos utilizaram uma base de dados simulada, estruturada com base no escopo de métricas proposto por [Rittes and Fiorese 2025], utilizando valores inseridos em intervalos realísticos de provedores *serverless*. A escalabilidade foi testada em cenários contendo de 5 a 1.000 alternativas simultâneas. Para cada conjunto populacional, o sistema processou 100 requisições aleatórias distintas. Estas requisições foram modeladas para simular dois possíveis perfis de decisores: I) Perfil Flexível (Desenvolvedor Comum): Projeto com poucos critérios essenciais e margens de tolerância mais altas. Os testes com este perfil foram parametrizados com 30% dos critérios marcados como essenciais e tolerância de 10%, II) Perfil Estrito (Sistemas Críticos): Planejamento restrito, no qual a maioria dos critérios é marcada como essencial e com tolerâncias baixas. Os testes com este perfil utilizaram 70% de essencialidade e tolerância de 5% sobre os critérios.

3.1. Desempenho do Modelo

O desempenho e a eficiência do modelo foram medidos pelo Tempo de CPU e pela alocação de memória. A Figura 1 compara o tempo de CPU de todo o *pipeline* do método híbrido (filtragem, AHP e ordenação) com a execução nativa do PROMETHEE II pela biblioteca pyDecision. Os gráficos exibem os testes dos Perfis Estrito (à direita) e Flexível (à esquerda) lado a lado, adotando uma escala logarítmica em ambos os eixos para evidenciar visualmente a diferença de tempo de processamento, desde pequenos conjuntos até 1.000 provedores. A análise demonstra que, enquanto o tempo do pyDecision cresce de forma acentuada devido à sobrecarga quadrática intrínseca ao método tradicional, a abordagem híbrida mantém estabilidade operando em patamares muito inferiores. Essa eficiência computacional é maximizada no Perfil Estrito, onde o rigor da filtragem inicial reduz drasticamente a carga da última fase, mas também se prova superior no Perfil Flexível ao mitigar com sucesso a sobrecarga computacional.

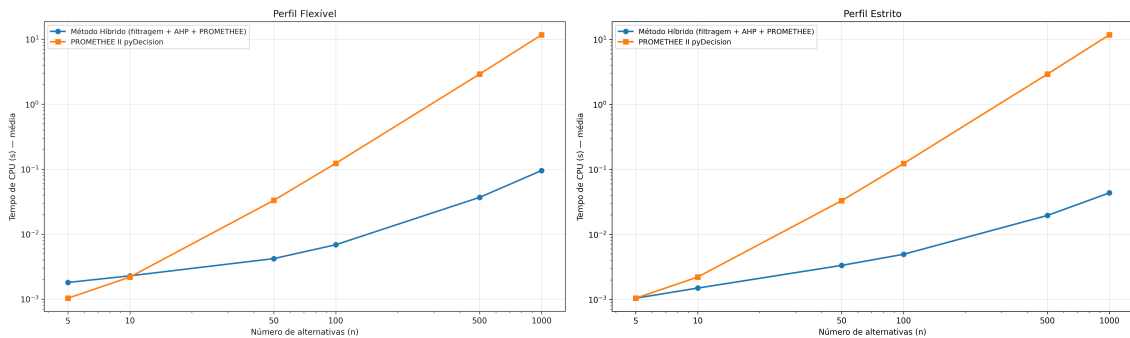


Figura 1. Comparação do tempo médio de CPU entre a arquitetura híbrida e a linha de base (pyDecision).

A Figura 2 detalha a fração de consumo de memória (RSS) e tempo de CPU por cada etapa do algoritmo, no Perfil Estrito. Como se observa a filtragem inicial elimina a maioria dos provedores, reduzindo significativamente a carga do PROMETHEE II, efeito especialmente visível em 1000 alternativas. Embora não apresentado, no Perfil Flexível, mais provedores sobrevivem ao filtro, fazendo com que a fase de ordenação final concentre a maior parte do processamento. Em ambos os perfis, o consumo de memória acompanha proporcionalmente o número de alternativas restantes.

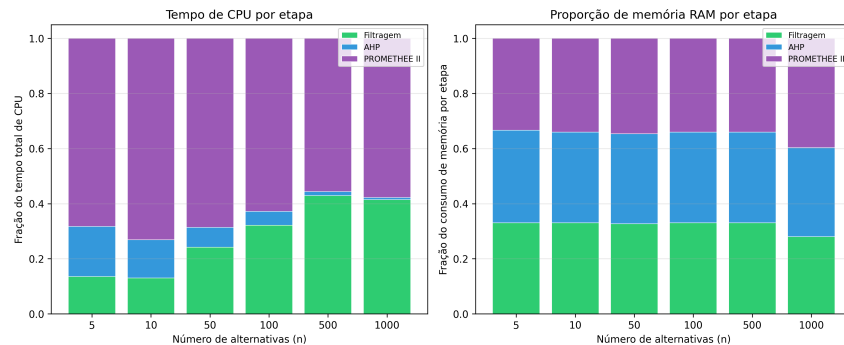


Figura 2. Fração de consumo de Tempo de CPU e Memória (RSS) por fase do algoritmo híbrido em Perfil Estrito.

3.2. Acurácia

Para garantir que a implementação está correta e precisa, os ranqueamentos gerados pelo método proposto foram comparados a resultados considerados previamente consolidados na literatura.

A Figura 3 apresenta uma validação de acurácia, ilustrada a partir de um mapa de calor (*heatmap*) com 16 cenários de 5 alternativas (provedores *serverless*) cada, previamente conhecidos e induzidos matematicamente a gerar um ranqueamento específico. O experimento realiza diferentes testes envolvendo as exigências do decisor e atesta sua confiabilidade de uso (100% de precisão em todos os cenários) para refletir as preferências do usuário de forma determinística.

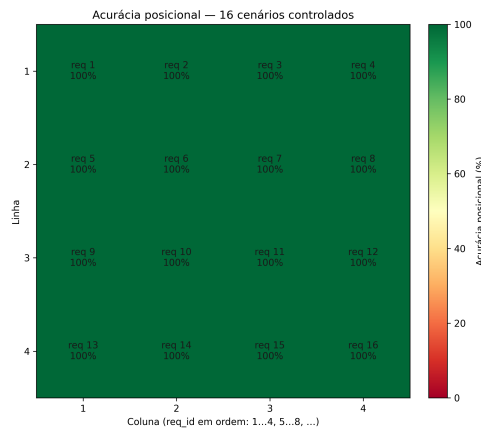


Figura 3. Mapa de calor (heatmap) da acurácia posicional validada em 16 cenários de requisição controlada.

4. Considerações Finais

Esse trabalho desenvolveu uma ferramenta para ranquear provedores *serverless* de forma automatizada e sem perda de precisão matemática. O modelo híbrido desenvolvido e validado demonstrou que parte da sobrecarga computacional pode ser contornada por meio da divisão do problema de seleção em camadas. Uma característica que se destacou nos testes é que o custo computacional do método não segue um padrão de crescimento. Além disso, nos experimentos com ambos os perfis (Estrito e Flexível), a acurácia manteve-se alta.

Como trabalhos futuros propõe-se a investigação de novas abordagens matemáticas na estrutura do método. Além disso, propõe-se substituir camadas de ordenação, visando outros métodos de análise multicritério que podem ser mais eficientes.

Agradecimentos

Os autores gostariam de agradecer à UDESC pela bolsa PROBIC, bem como ao suporte da Coordenação de Aperfeiçoamento de Pessoal do Ensino Superior (CAPES), Brasil (PROAP/AUXPE).

Referências

- Basílio, M. P., Pereira, V., Costa, H. G., Santos, M., and Ghosh, A. (2022). A systematic review of the applications of multi-criteria decision aid methods (1977–2022). *Electronics*, 11(11).
- Brans, J. P., Vincke, P., and Mareschal, B. (1986). How to select and how to rank projects: The promethee method. *European Journal of Operational Research*, 24(2):228–238.
- Copik, M., Kwasniewski, G., Besta, M., Podstawski, M., and Hoefler, T. (2021). Sebs: a serverless benchmark suite for function-as-a-service computing. In *Proceedings of the 22nd International Middleware Conference*, pages 64–78, New York, NY, USA. Association for Computing Machinery.
- de Moraes, L. B. and Fiorese, A. (2017). Um método de seleção de provedores de computação em nuvem baseado em indicadores de desempenho. In *Workshop em Clouds e Aplicações (WCGA)*. SBC.
- Eismann, S., Scheuner, J., van Eyk, E., Schwinger, M., Grohmann, J., et al. (2021). Serverless applications: Why, when, and how? *IEEE Software*, 38(1):32–39.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, New York, NY.
- Rittes, L. and Fiorese, A. (2025). Using multicriteria decision method to score and rank serverless providers:. In *Proceedings of the 27th International Conference on Enterprise Information Systems*, pages 27–36, Porto, Portugal. SCITEPRESS - Science and Technology Publications.
- Saaty, T. L. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26.