ESTUDO E APLICAÇÃO DE MÉTODOS MULTICRITÉRIO DE TOMADA DE DECISÃO PARA RANQUEAMENTO E SELEÇÃO DE PROVEDORES PARA APLICAÇÕES SERVERLESS

Kelwin Efrain Bagnhuk da Silva, Adriano Fiorese

INTRODUÇÃO

A computação em nuvem revolucionou o desenvolvimento de software por meio do paradigma *serverless*, que abstrai o gerenciamento de infraestrutura de servidores que, apesar do nome, continuam existindo, mas com suas operações sendo responsabilidade inteiramente do provedor desse serviço e não do desenvolvedor da aplicação. Esse paradigma também otimiza os custos operacionais sob o modelo financeiro *pay-as-you-go* (CASTRO et al., 2019), eliminando custos desnecessários com tempos ociosos, uma vez que o provedor desse serviço cobra apenas quando a aplicação é executada. Contudo, a expansão do mercado gerou uma vasta oferta de provedores com diferentes capacidades, tornando a seleção da plataforma *serverless* ideal um problema decisório. Para solucionar esse problema, a literatura utiliza os Métodos de Análise de Decisão Multicritério (MCDA). No entanto, o uso isolado de algoritmos tradicionais frequentemente esbarra em limitações estruturais, como alta sobrecarga computacional ao lidar com inúmeras alternativas e a dificuldade em traduzir com exatidão as preferências e prioridades do cliente. Diante disso, este trabalho tem como objetivo desenvolver e validar um método multicritério híbrido estruturado em três fases, visando otimizar o tempo de processamento computacional e garantir alta precisão no ranqueamento de infraestruturas *serverless*. Para tanto, a solução proposta combina três métodos multicritérios complementares: uma filtragem baseada em Indicadores de Desempenho (PIs) para eliminar previamente opções *serverless* inviáveis; o método *Analytic Hierarchy Process* (AHP), responsável pela ponderação dos critérios; e o PROMETHEE II para gerar o ranqueamento final dos provedores.

DESENVOLVIMENTO

O método proposto opera sobre Indicadores de Desempenho (PIs), classificados conforme a expectativa de otimização ideal (JAIN, 1991): Maior é Melhor (HB), quando buscamos maximizar o valor; Menor é Melhor (LB), quando buscamos minimizar o valor; Valor Nominal (NB), quando queremos um valor ideal específico com uma tolerância. Os critérios avaliados refletem métricas reais de provedores *serverless* do mercado, como Custo Base por Milhão de Execuções, Custo de Requisição Adicional, Arredondamento de Duração, Memória Máxima, Requisições Gratuitas por Mês, Tempo Máximo de Execução, Tempo Computacional Gratuito, Tempo de Cold Start, Escalabilidade e Limite de Concorrência (AMAZON WEB SERVICES, 2026; MICROSOFT AZURE, 2026; GOOGLE CLOUD, 2026).

Dessa forma, a Figura 1 apresenta o fluxograma da arquitetura híbrida proposta, que é dividida em três fases. Na primeira, uma filtragem lógica inspirada no estudo de Moraes, Fiorese e Matos (2017) elimina alternativas que violam restrições essenciais definidas pelo decisor, reduzindo o espaço de busca antes das etapas seguintes que são mais custosas computacionalmente. Na segunda fase, o método *Analytic Hierarchy Process* (AHP) é aplicado exclusivamente sobre os critérios, por meio de comparações de valores par a par na escala de Saaty, que foram decididos previamente pelo usuário, para extrair os pesos globais de importância de cada critério (SAATY, 1990). Nessa extração dos pesos, o AHP normaliza a matriz de comparações e deriva o vetor de prioridades utilizando o método da média aritmética das linhas. Por fim, na terceira fase, o algoritmo PROMETHEE II processa apenas as alternativas aprovadas no filtro inicial, calculando, para cada par de alternativas em cada critério, a diferença de desempenho. Essas diferenças são convertidas em graus de preferência pela Função de Preferência Gaussiana (Tipo VI), parametrizada automaticamente pelo desvio padrão da

distribuição dos dados, e agregadas conforme os pesos fornecidos pelo AHP no Índice de Preferência Multicritério. A partir disso, os índices são utilizados para calcular os fluxos de superação (positivo e negativo), cuja diferença resulta no fluxo líquido, que determina a posição e gera o ranqueamento final das alternativas (BRANS et al., 1986).

RESULTADOS

Os experimentos computacionais avaliaram o tempo de CPU e a acurácia do método em cenários de 5 a 1.000 provedores em dois tipos de perfil de usuário, comparando-o com a execução nativa do PROMETHEE II por meio da biblioteca *pyDecision* (PEREIRA; BASILIO; SANTOS, 2025). Os dois perfis de decisor continham uma rigorosidade diferente e critérios essenciais variados, para simular dois comportamentos comuns durante a busca de uma infraestrutura *serverless*: o Perfil Flexível determinava aleatoriamente 30% dos critérios como "essenciais", buscando uma métrica ideal com a aceitação de uma tolerância com desvio de 10% do valor estipulado; o Perfil Estrito considerava sistemas mais críticos, delimitando 70% como critérios essenciais e tolerância de 5%. Os resultados demonstraram que a arquitetura híbrida apresentou um desempenho adaptável conforme a exigência do decisor, com um tempo computacional inferior ao do método isolado, especialmente no Perfil Estrito. Essa eficiência e escalabilidade podem ser observadas na Figura 2, que ilustra o comportamento da arquitetura híbrida nos dois perfis testados. Para o Perfil Flexível, a Figura 2a apresenta o tempo de CPU do método híbrido comparado à linha de base implementada pela biblioteca *pyDecision*, enquanto a Figura 2b faz a análise detalhada do consumo de tempo em cada etapa. De forma semelhante, para o Perfil Estrito, a Figura 2c apresenta a mesma comparação de tempo de CPU e a Figura 2d detalha o consumo nas etapas. Em ambos os perfis, é observável um ganho expressivo de tempo em comparação à biblioteca externa, pois a filtragem inicial elimina uma grande parte das alternativas inviáveis antes do processamento matemático na fase final. Esse efeito é ainda mais perceptível no Perfil Estrito, no qual a maior porcentagem de critérios essenciais reduz significativamente o número de alternativas sobreviventes. A validação de precisão seguiu duas etapas. Inicialmente, o método foi submetido a uma validação fundamental (*ground-truth)*, composta por 16 cenários controlados de pequena escala, cada um com ranqueamento ideal previamente definido. O resultado gerado pelo método híbrido foi verificado por meio da correspondência dos ranqueamentos esperados, alcançando 100% de acerto em todos casos. Em seguida, para os cenários de maior escala, o ranqueamento final foi comparado à execução nativa do PROMETHEE II, confirmando acurácia posicional quase perfeita até 1.000 provedores, com raros *outliers* apenas nos maiores volumes testados.

CONSIDERAÇÕES FINAIS

Esta pesquisa apresentou um método multicritério híbrido para o problema de seleção de provedores *serverless*, organizado inicialmente com uma filtragem das alternativas por Indicadores de Desempenho, extração das preferências do decisor pela ponderação com o algoritmo AHP e a ordenação final via PROMETHEE II, entregando a opção que mais atende aos requisitos definidos. Os experimentos confirmam que a abordagem em fases reduziu a sobrecarga computacional sem prejudicar a precisão do ranqueamento, considerando cenários desde pequeno até amplo volume de alternativas. Como trabalhos futuros, propõe-se expandir o estudo de análise de sensibilidade dos dados em cada fase, além da investigação do desempenho com outros métodos determinísticos como TOPSIS, almejando um algoritmo mais eficiente e preciso.

Palavras-chave: *serverless*; análise multicritério; AHP; PROMETHEE II.

ANEXO(S)

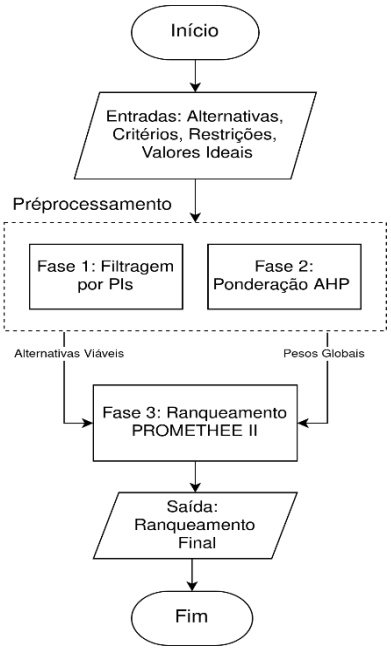


Figura 1. *Fluxograma da Arquitetura Híbrida.*

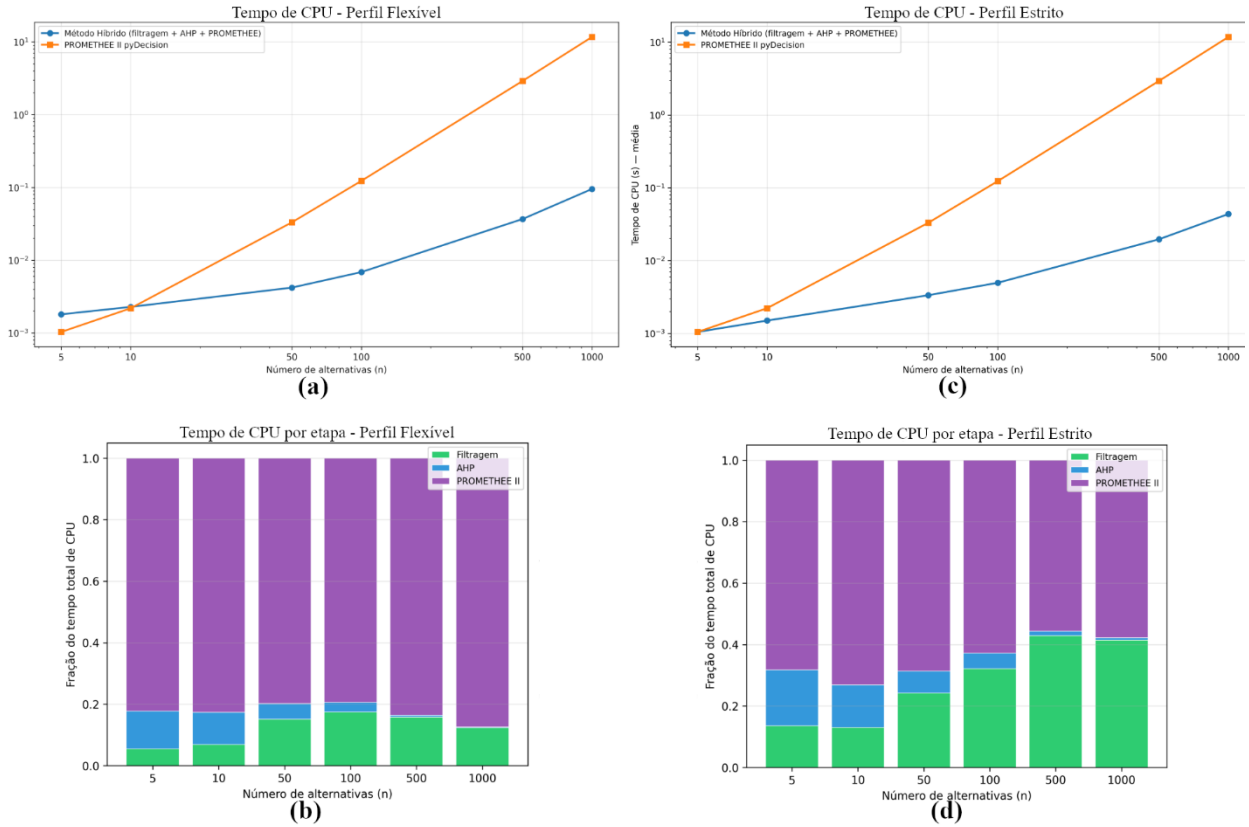


Figura 2. *Tempo de CPU da arquitetura híbrida em comparação à biblioteca pyDecision e fração de tempo de execução por etapa, nos perfis Flexível (a, b) e Estrito (c, d).*

REFERÊNCIAS BIBLIOGRÁFICAS

AMAZON WEB SERVICES. **AWS Lambda Pricing**. 2026. Disponível em: https://aws.amazon.com/lambda/pricing/. Acesso em: 11 de ago. 2026.

BRANS, J. P.; VINCKE, P.; MARESCHAL, B. How to select and how to rank projects: the PROMETHEE method. **European Journal of Operational Research**, v. 24, n. 2, p. 228-238, 1986.

CASTRO, P.; ISHAKIAN, V.; MUTHUSAMY, V.; SLOMINSKI, A. The rise of serverless computing. **Communications of the ACM**, v. 62, n. 12, p. 44-54, 2019.

GOOGLE CLOUD. **Cloud Functions Pricing Overview**. 2026. Disponível em: https://cloud.google.com/functions/pricing. Acesso em: 11 de ago. 2026.

JAIN, R. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. New York: John Wiley & Sons, 1991.

MICROSOFT AZURE. **Azure Functions Pricing**. 2026. Disponível em: https://azure.microsoft.com/pricing/details/functions/. Acesso em: 11 de ago. 2026.

MORAES, L. B. de; FIORESE, A.; MATOS, F. A multi-criteria scoring method based on performance indicators for cloud computing provider selection. In: INTERNATIONAL CONFERENCE ON ENTERPRISE INFORMATION SYSTEMS, 19., 2017. **Proceedings** [...]. [S. l.]: INSTICC; SciTePress, 2017. v. 2, p. 588-599.

PEREIRA, V.; BASILIO, M. P.; SANTOS, C. H. T. Enhancing decision analysis with a large language model: pyDecision, a comprehensive library of MCDA methods in Python. **Journal of Modelling in Management**, 2025.

SAATY, T. L. How to make a decision: the analytic hierarchy process. **European Journal of Operational Research**, v. 48, n. 1, p. 9-26, 1990.