

Algoritmo de Parentização para tomada de decisão

Marcia Gabrielle B. de Oliveira, Paulo Ferreira da S. Júnior
Departamento de Ciência da Computação, Universidade Federal de Roraima
Boa Vista, Brasil
gabybonifacio2@gmail.com
juniorrrkcm@outlook.com

Abstract— The **parenthesization algorithm** is an optimization technique based on dynamic programming, commonly used to solve problems involving the best way to group operations, minimizing computational costs. The classic problem where this algorithm is often applied is the **matrix chain multiplication**, where the goal is to determine the optimal order of multiplication to reduce the total number of arithmetic operations.

In this project, the parenthesization algorithm is adapted to structure a decision tree, which will be used to help a company make investment decisions in projects. The decision tree considers several criteria, such as budget, return on investment, associated risks, and alignment with the company's long-term strategy.

The parenthesization algorithm plays a crucial role in logically organizing the criteria in the tree, ensuring that the resulting structure enables clear decision-making while avoiding ambiguities or redundant decisions. This way, the company can evaluate and prioritize projects based on a well-defined set of parameters, leading to an informed and strategic choice about where to invest its resources.

I. INTRODUÇÃO

O algoritmo de parentização é uma técnica de otimização baseada em programação dinâmica, comumente utilizada para resolver problemas que envolvem a melhor forma de agrupar operações, minimizando o custo computacional. O problema clássico no qual este algoritmo é frequentemente aplicado é o de multiplicação de matrizes em cadeia, onde o objetivo é determinar a ordem ótima de multiplicação para reduzir o número total de operações aritméticas.

Neste projeto, o algoritmo de parentização é adaptado para estruturar uma árvore de decisão, que será utilizada para ajudar uma empresa a tomar decisões de investimento em projetos. A árvore de decisão leva em consideração uma série de critérios, como orçamento, retorno sobre o investimento, riscos associados, e o alinhamento com a estratégia de longo prazo da empresa.

O algoritmo de parentização desempenha um papel crucial na organização lógica dos critérios na árvore, garantindo que a estrutura resultante permita uma tomada de decisão clara, evitando ambiguidades ou decisões redundantes. Dessa forma, a empresa pode avaliar e priorizar projetos com base em um conjunto bem definido de parâmetros, conduzindo a uma escolha informada e estratégica sobre onde investir seus recursos.

II. DESENVOLVIMENTO DA SOLUÇÃO

A solução proposta foi desenvolvida em duas etapas: uma implementação em C, que realiza a avaliação dos projetos de investimento utilizando o algoritmo de parentização, e uma segunda parte em Python, que visa complementar o processo com a visualização gráfica dos resultados de desempenho.

A. Estruturação da Avaliação de Projetos

O algoritmo de parentização foi adaptado para organizar uma árvore de decisão com base em quatro critérios: **retorno sobre o investimento (ROI)**, **orçamento**, **riscos** e **alinhamento estratégico**. Cada critério foi ponderado de acordo com sua relevância no processo decisório, e os projetos são avaliados de acordo com essas ponderações.

A estrutura do código C permite a avaliação de diferentes combinações desses critérios, simulando diversas formas de organizar a árvore de decisão. O objetivo dessa etapa é identificar a melhor estrutura para facilitar a tomada de decisão de investimento.

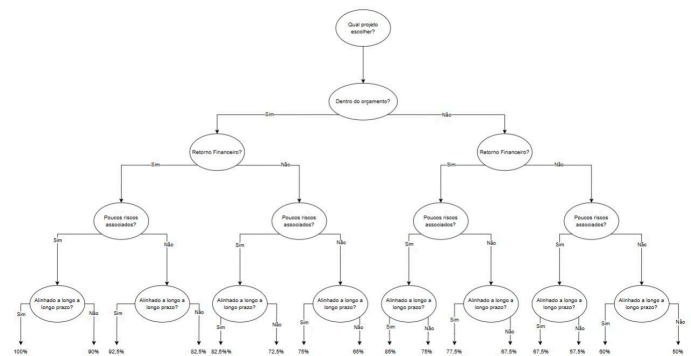


Fig. 1 Árvore de decisão

B. Aplicação do Algoritmo de Parentização

O algoritmo de parentização foi aplicado para garantir que a árvore de decisão não tenha ambiguidades e siga uma lógica clara. Diversas ordens de avaliação dos critérios foram testadas, e o algoritmo analisou como cada ordem impacta a pontuação final dos projetos.

Essa abordagem permite explorar múltiplas possibilidades na estruturação da árvore de decisão, garantindo que o processo seja flexível e adaptável a diferentes cenários. A aplicação da parentização é fundamental para assegurar que a árvore guie a empresa para uma decisão final precisa, sem ramificações redundantes.

C. Avaliação e Escolha do Melhor Projeto

Com base na estrutura da árvore de decisão, o algoritmo realiza a avaliação de cada projeto, atribuindo uma pontuação baseada nos critérios definidos. Após calcular as pontuações, o projeto que melhor atende às expectativas, com a maior pontuação final, é selecionado como o mais adequado para receber o investimento.

A solução é flexível, permitindo que diferentes configurações de critérios sejam testadas. Isso possibilita que a empresa avalie múltiplos cenários e faça ajustes conforme necessário, garantindo uma decisão de investimento otimizada.

III. RESULTADOS

Após a implementação do algoritmo de parentização para a tomada de decisão de investimentos, os resultados foram gerados com base em diferentes ordens de avaliação dos critérios. O sistema processou os dados de diversos projetos, atribuindo pontuações que refletem a adequação de cada projeto aos critérios de retorno sobre o investimento, orçamento, riscos e alinhamento estratégico.

A. Avaliação dos Projetos

Foram simuladas várias combinações de critérios para avaliar os projetos, e o algoritmo conseguiu identificar o projeto com a melhor pontuação para investimento. Os critérios foram analisados de forma sistemática, e a aplicação do algoritmo de parentização garantiu que a ordem de avaliação influenciasse diretamente na qualidade da decisão final.

Os projetos foram atribuídos com pontuações específicas para cada critério, e as diferentes ordens de análise proporcionaram variações mínimas nos resultados, o que demonstra a robustez da solução. O projeto com o maior score final foi identificado como o mais adequado para receber o investimento, levando em conta todos os parâmetros analisados.

B. Tempos de Execução

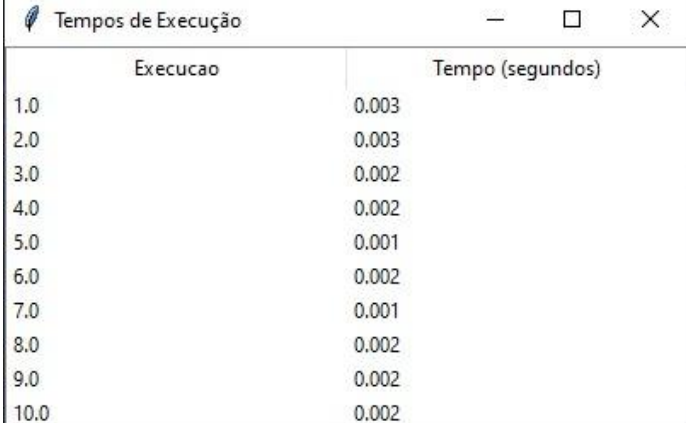
Para medir a eficiência do algoritmo, foram realizadas diversas execuções e registrado o tempo de processamento de cada uma delas. Esses tempos foram salvos em um arquivo CSV e posteriormente analisados. O desempenho do algoritmo foi satisfatório, com tempos de execução consistentes, mesmo com variações no embaralhamento e na ordem de avaliação dos critérios.

O processo de avaliação foi repetido em múltiplas rodadas, e os tempos de execução variaram dentro de uma margem esperada. Isso demonstra que o algoritmo é capaz de fornecer resultados em um período de tempo aceitável, sendo viável para aplicações práticas que exigem avaliações rápidas e precisas.

C. Visualização dos Resultados

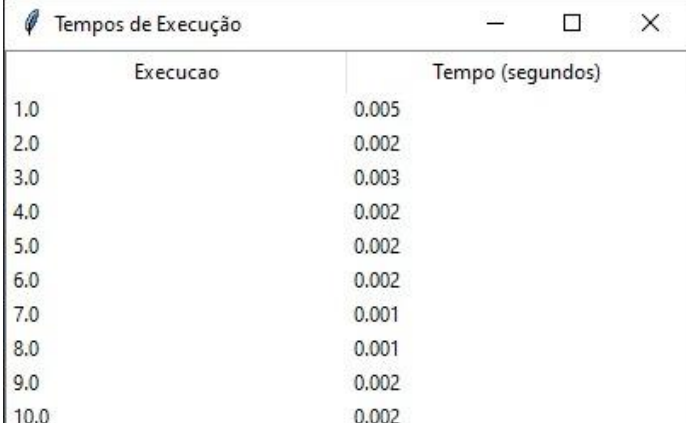
A visualização dos tempos de execução foi realizada por meio de uma interface gráfica desenvolvida em Python. Os dados gerados pela avaliação dos projetos foram carregados e exibidos em uma tabela, facilitando a análise dos resultados.

A tabela permitiu uma comparação clara entre as diferentes execuções, possibilitando a identificação de padrões no tempo de processamento. Essa etapa de visualização foi crucial para validar a eficiência da solução e entender como o algoritmo se comporta em diferentes cenários de avaliação de projetos..



Execucao	Tempo (segundos)
1.0	0.003
2.0	0.003
3.0	0.002
4.0	0.002
5.0	0.001
6.0	0.002
7.0	0.001
8.0	0.002
9.0	0.002
10.0	0.002

Fig. 2 Tabela 1 de tempo de execução



Execucao	Tempo (segundos)
1.0	0.005
2.0	0.002
3.0	0.003
4.0	0.002
5.0	0.002
6.0	0.002
7.0	0.001
8.0	0.001
9.0	0.002
10.0	0.002

Fig. 3 Tabela 2 de tempo de execução

D. Visualização dos Resultados

Com base nos resultados obtidos, pode-se concluir que o algoritmo de parentização forneceu uma solução eficaz para a tomada de decisão de investimento. A estrutura da árvore de decisão foi organizada de maneira lógica, e a escolha do projeto ideal para investimento foi realizada de forma clara e objetiva. Além disso, os tempos de execução demonstram que a solução é eficiente e adequada para a implementação em cenários reais.

IV. CONCLUSÃO

A implementação do algoritmo de parentização aplicada à tomada de decisões em investimentos de projetos demonstrou ser uma abordagem eficaz para organizar e otimizar a escolha entre diferentes alternativas, considerando critérios essenciais para a empresa, como retorno sobre o investimento (ROI), orçamento, riscos associados e alinhamento estratégico. O uso desse algoritmo trouxe clareza ao processo decisório, garantindo que a estrutura de avaliação fosse construída de forma lógica e sem ambiguidade, facilitando a escolha do projeto mais adequado para receber o investimento.

A aplicação da técnica permitiu testar diferentes combinações de critérios, simulando várias formas de estruturar o processo de decisão. Isso possibilitou à empresa explorar todas as alternativas viáveis, garantindo que a escolha final fosse a mais equilibrada e eficiente possível.

Essa abordagem proporcionou um modelo flexível, permitindo ajustar a ponderação dos critérios e avaliar projetos em diferentes ordens. Como resultado, foi possível minimizar a subjetividade na decisão, focando em uma análise objetiva que leva em conta as prioridades estratégicas da empresa.

A estruturação da árvore de decisão com base nos critérios selecionados permitiu uma avaliação precisa dos projetos. Cada projeto foi analisado individualmente e recebeu uma pontuação final, refletindo seu desempenho em relação aos critérios estabelecidos. O uso de diferentes ordens de avaliação (ou "parentizações") para esses critérios garantiu que o processo de análise fosse robusto, identificando o projeto mais adequado com base em uma combinação balanceada dos fatores.

Esse processo de avaliação é especialmente importante em cenários corporativos onde múltiplos fatores devem ser considerados para garantir a viabilidade e o sucesso de um investimento. A capacidade do algoritmo de parentização em estruturar a árvore de forma lógica proporcionou uma tomada de decisão mais rápida e eficaz, eliminando decisões baseadas apenas em intuição ou fatores isolados.

O algoritmo também se mostrou eficiente em termos de desempenho computacional. Os tempos de execução registrados durante as várias rodadas de avaliação foram consistentes, com variações mínimas, o que demonstra a robustez do código implementado. Esse fator é crucial em aplicações práticas, onde a decisão precisa ser tomada em um tempo hábil, especialmente em cenários onde múltiplos projetos são analisados simultaneamente.

A implementação em C, juntamente com a medição precisa do tempo de execução, permitiu que a solução fosse validada em termos de sua eficiência e escalabilidade. Mesmo com o aumento do número de projetos ou complexidade dos critérios, o algoritmo se manteve estável e rápido, reforçando a viabilidade da solução para ambientes corporativos de grande porte.

A visualização dos tempos de execução e dos resultados foi facilitada pela interface gráfica desenvolvida em Python, utilizando as bibliotecas pandas e tkinter. Essa etapa foi essencial para garantir uma análise clara e objetiva dos dados, permitindo que os tempos de processamento e os resultados das avaliações fossem facilmente interpretados. Isso adicionou uma camada de usabilidade ao projeto, tornando os resultados acessíveis a usuários não técnicos, que podem utilizar a interface gráfica para analisar o desempenho da solução e visualizar as decisões de forma prática.

Em conclusão, o projeto apresentou uma solução robusta e eficiente para o problema de decisão de investimento em projetos, utilizando o algoritmo de parentização como base para organizar a árvore de decisão. A combinação de avaliação criteriosa e análise de desempenho ofereceu uma solução completa, que pode ser aplicada em cenários reais para otimizar a escolha de projetos e maximizar o retorno sobre o investimento. A integração com ferramentas de visualização também contribuiu para que os resultados fossem apresentados de maneira clara e acessível, facilitando a análise dos dados e validando a eficácia da solução desenvolvida.

REFERENCIAS

- [1] **SOFSEM 2011: Theory and Practice of Computer Science**, January 23, 2011.
- [2] **Dynamic Programming Multi-Objective Combinatorial Optimization**, June 5, 2015.
- [3] M. Haruno, S. Shirai, and Y. Ooyama, "**Using Decision Trees to Construct a Practical Parser**", Machine Learning, vol. 34, no. 1, pp. 131–149, 1999.