

Aplicação de Métodos de Otimização na Geração de Dietas

Bruno Cesar Pawlak¹, Paola Cristine Carvalho², Wagner W. Á. Bombardelli³,
Mateus Leite Ribeiro da Silva⁴, Lucas Ezequiel Negrini⁵,
Kevin Vinícius Pires dos Santos⁶, Evandro Felipe Silva⁷,
Carlos Eduardo Carli Silva⁷

¹Ciência de Dados – Faculdade Donaduzzi
Paraná – PR – Brazil

Pcrisstine@gmail.com, Brunocesarpawlak@gmail.com

Abstract. *This work presents a system for automated generation of diets based on nutritional goals defined by the user. The problem is treated as an optimization task under constraints. Two approaches were compared: Genetic Algorithm (GA) and Monte Carlo Simulation (MC). The evaluation was based on error metrics, performance and rate of valid solutions. The work highlighted relevant differences between the methods, with GA showing better performance throughout the iterations, with progressive evolution in the quality of the solutions.*

Resumo. *Este trabalho apresenta um sistema para geração automatizada de dietas a partir de metas nutricionais definidas pelo usuário. O problema é tratado como uma tarefa de otimização sob restrições. Foram comparadas duas abordagens: Algoritmo Genético (GA) e Simulação de Monte Carlo (MC). A avaliação foi baseada em métricas de erro, desempenho e taxa de soluções válidas. O trabalho evidenciou diferenças relevantes entre os métodos, com o GA apresentando melhor desempenho ao longo das iterações, demonstrando evolução progressiva da qualidade das soluções.*

1. Contextualização do Problema

Apesar da crescente demanda por alimentação saudável, muitos indivíduos ainda enfrentam desafios para manter uma dieta equilibrada, principalmente no que diz respeito ao controle preciso de nutrientes e à variedade alimentar ao longo do tempo. Nesse contexto, empresas alimentícias enfrentam o desafio de oferecer opções personalizadas que atendam às diferentes necessidades nutricionais de seus clientes.

A elaboração de dietas personalizadas envolve múltiplas variáveis, como o valor calórico total, a distribuição de macronutrientes (proteínas, carboidratos e gorduras) e a combinação entre os alimentos disponíveis. Em larga escala, a definição manual dessas combinações se torna praticamente inviável, o que traz a necessidade de métodos automatizados.

Do ponto de vista computacional, esse problema pode ser visto como uma tarefa de otimização. De acordo com Nocedal e Wright (2006), problemas de otimização envolvem a determinação da melhor solução possível dentro de um conjunto viável de alternativas, frequentemente sujeitos a restrições.

Diante desse contexto, este trabalho tem como objetivo desenvolver um sistema capaz de gerar cardápios personalizados de forma automatizada, atendendo simultaneamente a múltiplas restrições nutricionais, explorando, comparando e avaliando diferentes abordagens de busca e otimização para a obtenção de soluções viáveis.

2. Descrição da Solução

As soluções propostas consistem na aplicação de abordagens computacionais para a geração automatizada de combinações alimentícias. Para isso, modelamos o problema como um caso de otimização, e posteriormente aplicamos técnicas de simulação probabilística (Monte Carlo) e otimização heurística (Algoritmo Genético).

Inicialmente, é considerada uma base de dados contendo os alimentos disponíveis, juntamente com suas respectivas informações nutricionais. A partir dessa base, são geradas combinações de refeições de forma aleatória, respeitando regras básicas, como a presença de alimentos fixos e a seleção de porções adicionais.

No método inspirado em Monte Carlo, o processo ocorre de forma iterativa, produzindo múltiplas combinações. Para cada combinação gerada, são calculados os valores nutricionais totais, verificando se a combinação atende às restrições estabelecidas, como limites de calorias e proporções de macronutrientes, as soluções que atendem aos critérios definidos são armazenadas, enquanto as demais são descartadas. Essa abordagem permite explorar um grande número de possibilidades de forma eficiente, com baixa complexidade de implementação.

Já a aplicação do algoritmo genético adota uma abordagem evolutiva, na qual uma população inicial de refeições é iterativamente refinada. Cada solução é avaliada por uma função de *fitness* responsável por medir o quão próxima a solução está da distribuição ideal de macronutrientes.

A função *fitness* é baseada no erro absoluto (*Manhattan distance* ou norma L1) entre os macronutrientes obtidos e os desejados. E formalmente, pode ser descrita como:

$$fitness(x) = - \sum_{i=1}^3 |m_i - t_i| \quad (1)$$

Onde m_i representa a proporção do macronutriente obtido na solução gerada e t_i representa a proporção desejada. O uso do sinal negativo permite transformar o problema de minimização do erro em um problema de maximização da função de *fitness*, característica típica dos algoritmos genéticos.

Cada indivíduo do algoritmo genético representa uma refeição personalizada composta por cinco genes, sendo dois destinados às opções de alimentos fixos e três correspondentes aos demais alimentos da refeição.

Ao longo das gerações, são aplicadas operações de seleção, cruzamento e mutação, responsáveis pela recombinação e variação das soluções. O método de seleção utilizado foi o *Tournament Selection* com $k=3$. O operador de cruzamento realizou a troca das posições dos genes correspondentes aos alimentos da refeição, enquanto a mutação consistiu na substituição aleatória de genes com taxa de 0.1.

Para evitar redundância no espaço de busca, os genes correspondentes aos alimentos foram ordenados, garantindo que combinações equivalentes não fossem tratadas como soluções distintas.

Em ambos os métodos, as soluções válidas são definidas com base em um intervalo de 10 % de tolerância para cada macronutriente, sendo armazenadas de forma única. Destaca-se o uso de paralelismo para múltiplos clientes, permitindo a execução simultânea das buscas e reduzindo significativamente o tempo total de processamento.

Logo de início, é possível observar que, enquanto o processo do Monte Carlo prioriza a exploração das soluções por meio de geração aleatória, o algoritmo genético faz uma busca mais direcionada, buscando melhorar progressivamente as combinações.

3. Ferramentas e Tecnologias

Para o desenvolvimento da solução, foi utilizada a linguagem de programação Python.

Foram utilizadas bibliotecas para manipulação e pré-processamento de dados, bem como para operações numéricas e suporte ao paralelismo. Também foram adotadas abordagens vetorizadas para otimização de cálculos, reduzindo gargalos associados a processos iterativos.

A geração de valores aleatórios foi realizada com o uso de uma *seed* fixa (42), fundamental para a implementação da abordagem baseada em simulação probabilística.

Como base de dados, foi utilizada a Tabela Brasileira de Composição de Alimentos (TACO), contendo mais de 500 alimentos e suas respectivas informações nutricionais, como valor energético e composição de macronutrientes.

Os dados da TACO passaram por um processo de pré-processamento, incluindo a limpeza e padronização das informações nutricionais, como a substituição de valores não numéricos e a remoção de alimentos incompatíveis com o contexto proposto. Esse tratamento foi necessário para garantir a consistência dos dados utilizados.

O código-fonte da solução está disponível no: Repositório [otimizacao-nutricional](#)

4. Resultados Práticos Obtidos

Foram considerados 100 perfis de clientes, cada um com proporções alvo distintas de macronutrientes. Para cada perfil, o método de Monte Carlo realizou 100.000 amostragens de dietas em lotes de 100 indivíduos por iteração, e o algoritmo genético operou com uma população de 100 indivíduos ao longo das gerações, totalizando igualmente 100.000 avaliações.

A avaliação dos métodos considerou quatro métricas principais: taxa de soluções viáveis (com tolerância de 10%), erro médio absoluto (MAE), erro médio quadrático (MSE) e tempo de execução (latência).

Tabela 1. Comparação entre Algoritmo Genético (GA) e Monte Carlo (MC)

Métrica	GA	MC
MAE	0,0106	0,0449
MSE	0,0003	0,0027
Latência	147.7 s	14.6 s

A seguir, o gráfico demonstra visualmente essas informações, após normalização das métricas, indicando o desempenho dos métodos.

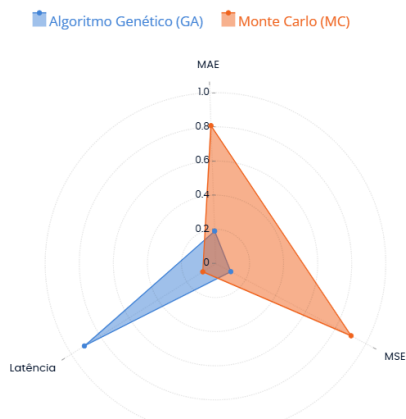


Figura 1. Comparação visual de desempenho (MAE, MSE e latência)

Esses valores indicam que o algoritmo genético não apenas apresenta menor erro nas soluções geradas, mas também gera soluções significativamente mais próximas das metas nutricionais, evidenciando maior precisão na otimização.

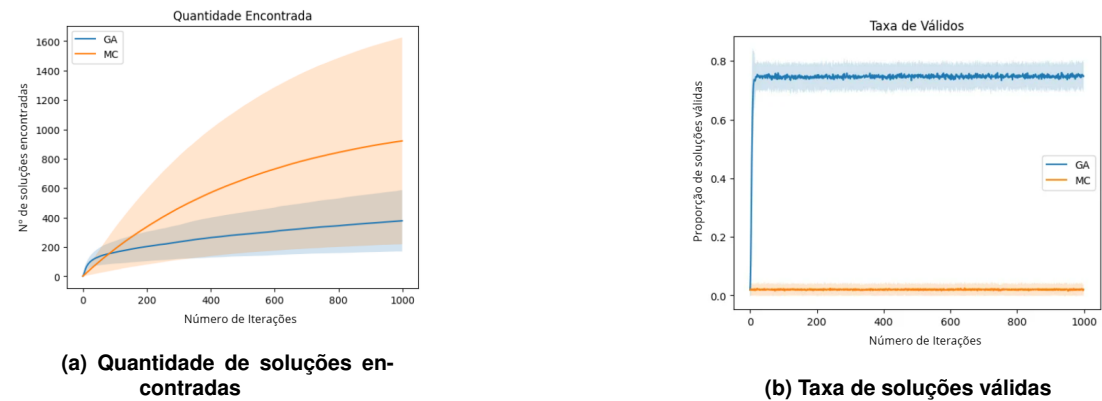


Figura 2. Comparação entre os métodos

Ao analisar os gráficos acima, notamos que o método de Monte Carlo explora amplamente o espaço de soluções e encontra um número absoluto maior de soluções distintas. No entanto apresenta limitações em termos de eficiência, pois necessita de uma quantidade elevada de tentativas para encontrar soluções viáveis, indicando que grande parte do esforço computacional é direcionada à geração de combinações inválidas.

Identificamos também que o algoritmo genético apresenta menor diversidade em comparação ao método de Monte Carlo, com redução gradual desses ao longo das iterações, uma vez que sua busca tende a se concentrar em regiões mais promissoras do espaço de soluções.

Esse comportamento reflete um dilema entre exploração e convergência e está alinhado com a natureza exploratória dos métodos probabilísticos, em contraste com a abordagem direcionada de algoritmos evolutivos.

5. Conclusão

O objetivo deste estudo foi propor uma estratégia de otimização para criar dietas personalizadas, levando em conta as limitações nutricionais e os objetivos de macronutrientes estabelecidos para diferentes perfis de usuários.

Com base nos resultados alcançados, conclui-se que o algoritmo genético é mais adequado para o problema proposto, exibiu um desempenho superior em relação à precisão e à qualidade das soluções produzidas. Por outro lado, o método de Monte Carlo pode ser utilizado como abordagem inicial, ou utilizado como uma ferramenta estratégica de exploração do espaço de busca.

Como trabalhos futuros, destaca-se a possibilidade de utilização de abordagens híbridas, combinando a capacidade exploratória do método de Monte Carlo com o refinamento proporcionado por algoritmos evolutivos.

Referências

- Alba, E. and Tomassini, M. (2002). Parallelism and evolutionary algorithms. *IEEE transactions on evolutionary computation*, 6(5):443–462.
- IBM (2026). Optimization model. Acesso em: 02 mar. 2026.
- Moffitt, C. (2019). Monte carlo simulation with python. Acesso em: 05 abr. 2026.
- Namen, A. A. and Bornstein, C. T. (2004). Uma ferramenta para avaliação de resultados de diversos modelos de otimização de dietas. *Pesquisa Operacional*, 24(3):445–465.
- NEPA - Núcleo de Estudos e Pesquisas em Alimentação (2011). *Tabela Brasileira de Composição de Alimentos (TACO)*. UNICAMP, Campinas, 4 edition.
- Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer.
- Rodríguez-Mier, P. (2017). A tutorial on differential evolution with python. Acesso em: 16 abr. 2026.
- Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo Method*. Wiley.
- Silva, A. L. M. (2014). Algoritmo baseado em evolução diferencial para solução de problemas de otimização combinatória.