



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Trabalho Prático de Otimização de Sistemas*

João Pedro Oliveira Batisteli¹

Resumo

Implementação de uma sistema para cálculo de macronutrientes, utilizando modelagem matemática e estratégias de otimização.

Link dos repositórios:

Palavras-chave: Otimização. Cálculo de dieta.

*Trabalho apresentado à disciplina de Otimização de Sistemas.

¹ Aluno do Programa de Graduação em Engenharia de Computação, Brasil – joão.batisteli@sga.pucminas.br.

1 INTRODUÇÃO

A dieta é um planejamento alimentar de acordo com o seu objetivo de físico e saúde. Manter uma dieta equilibrada é uma das coisas mais importantes para a saúde. Grande parte das pessoas associam alimentação saudável a comer alimentos orgânicos, comer bastante legumes, verduras e frutas, se alimentar várias vezes ao dia (BOTTURA, 2017) e, conseqüentemente, perder peso ou ganhar massa magra, de acordo com o objetivo de cada um.

Para alcançar seus objetivos as pessoas podem adotar diferente protocolos como *carb backloading*, *low carb*, jejum intermitente, dieta paleo, entre outras.

A base de uma dieta é o balanço calórico, que ajuda no planejamento da dieta de acordo com o objetivo de cada um.

Nós, seres humanos, precisamos de energia para sobreviver e a fonte de energia que usamos são as calorias contidas nos alimentos. Caloria é uma unidade de medida de energia referente a quantidade de energia necessária para elevar a temperatura de 1g de água em 1 grau Celsius.

2 METODOLOGIA

Nesta seção serão apresentados os passos para o desenvolvimento do trabalho. Todos os arquivos associados que serão mostrados estarão em anexo, junto a este relatório

2.1 Modelagem Matemática

De acordo com Bottura (2017), existem duas classes de nutrientes: os macronutrientes e os micronutrientes. Os macronutrientes incluem: proteínas, carboidratos, gorduras e em alguns casos fibras e álcool. Já os micronutrientes incluem: vitaminas, minerais, fitoquímicos e água. A diferença entre os dois grupos é que os macronutrientes liberam calorias quando ingeridos, enquanto os micronutrientes, não.

A quantidade de calorias que cada macronutriente gera por grama pode ser visto na figura 1.

Figura 1 – Calórias em 1g de Cada Macronutriente

Calorias em 1g de cada Macronutriente	
Proteína	4kcal/g
Carboidrato	4kcal/g
Gordura	9kcal/g
Fibra	1,5-2kcal/g
Álcool	7kcal/g

Fonte: (BOTTURA, 2017)

Podemos então modelar a quantidade de calorias ingeridas em função de cada macronutriente da seguinte maneira:

$$4Q_p + 4Q_c + 9Q_g + 1.5Q_f + 7Q_a$$

Onde:

- Q_p - Quantidade de proteína (em gramas);
- Q_c - Quantidade de carboidratos (em gramas);
- Q_g - Quantidade de gordura (em gramas);
- Q_f - Quantidade de fibra (em gramas);
- Q_a - Quantidade de álcool (em gramas).

Ao se pensar em uma dieta, o objetivo seria *maximizar* essa função até o limite das calorias que o usuário deseja consumir diariamente ou *minimizar* a diferença entre as calorias calculadas através da função e a quantidade de calorias que o usuário deseja consumir.

Existem várias maneiras de se calcular os macronutrientes e devido a estratégias tomadas os cálculos podem se modificar, por isso o modelo possui restrições fixas e outras que podem ser utilizadas dependendo da estratégia que está sendo adotada pelo usuário.

2.2 Restrições

As restrições podem ser em relação ao percentual de cada tipo de nutriente em que o usuário deseja consumir. Como exemplo, dietas ricas em proteína são uma estratégia que o usuário pode assumir, o trabalho de Gheorghe et al. (2005) mostra que esse tipo de dieta pode ajudar na preservação de massa magra durante o processo de perda de peso, a figura 2 mostra a divisão de macronutrientes em dietas desse tipo.

Figura 2 – Dietas ricas em proteína

DIET	CHO	Fat	Protein	g/kg/d Protein*
USDA recommend	45-65%	20-35%	10-35%	0.8
Atkins [4]	6%	59%	35%	2.3
South Beach [5]	28%	33%	39%	2.6
Stillman [5]	3%	33%	64%	4.3
Zone [5]	36%	29%	34%	2.3
High Protein, normal CHO [6]	50%	30%	20%	1.3

Fonte: (GHEORGHE et al., 2005)

Usando como exemplo a dieta de Atkins (ATKINS, 2002), teríamos as seguintes restrições:

- $Qp = 0,35 * Pkg$;
- $Qc = 0,06 * Pkg$;
- $Qg = 0,59 * Pkg$.

Onde Pkg é o peso do usuário em kilogramas.

No a dieta modelada anteriormente não teríamos nenhum trabalho de otimização a ser feito, os valores de Qp , Qc e Qg seriam definidos baseados no peso do usuário.

Essa modelagem não é a mais ideal pois quando o usuário for realizar ajustes nas calorias da dieta todos os seus macros serão alterados.

Tendo em vista os problemas, neste trabalho será adotado a modelagem baseada em grama por quilo de macronutriente. Precisamos que o usuário informe seu peso e o número de calorias que deseja consumir por dia, a partir disso seria definido limites de g/kg para cada macronutriente e então calculado a quantidade de cada um aproximando-se ao máximo da quantidade em calorias desejado.

A imagem 3 mostra valores recomendados para esse problema.

Figura 3 – Recomendação de Macronutrientes

Tabela de Recomendações de Macronutrientes	
Proteínas	2-3g/kg
Gorduras	0,6-1g/kg
Carboidratos	Resto das Calorias ÷ 4
Fibras	10-15g a cada 1000 calorias

Fonte: (BOTTURA, 2017)

Então podemos modelar o nosso problema da seguinte maneira:

$$\text{Maximizar : } (Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9)$$

Sujeito a:

- $(Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9) \leq Cdu$;
- $2 \leq Qp \leq 3$;
- $0,6 \leq Qg \leq 1$.
- $Qc = Cdu - ((Pkg * Qp * 4) + (Pkg * Qg * 9))$;
- $Qf = 10 * (Cdu / 1000)$.

Onde:

- Cdu - Calorias diárias do usuário;
- Pkg - Peso em kilogramas do usuário;
- Qp - g/kg de proteína;
- Qc - g/kg de carboídrato;
- Qg - g/kg de gordura;
- Qf - g/kg de fibra.

Também pode-se modelar o problema dessa outra maneira:

Minimizar : $((Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9)) - Cdu$

Sujeito a:

- $(Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9) \leq Cdu$;
- $2 \leq Qp \leq 3$;
- $0,6 \leq Qg \leq 1$.
- $Qc = Cdu - ((Pkg * Qp * 4) + (Pkg * Qg * 9))$;
- $Qf = 10 * (Cdu / 1000)$.

A função objetivo $((Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9)) - Cdu$ permite uma melhor interpretabilidade do problema proposto, onde queremos achar um valor de Qp , Qc e Qg que minimize a diferença entre as calorias calculadas pelo modelo e as que seriam prescritas ao usuário.

Na modelagem inicial será levado em conta a quantidade de fibra no cálculo de calorias diárias. Os limites inferiores e superiores das variáveis Qp , Qc , Qg e Qf poderão ser modificados com o objetivo de atender estratégias diferentes mas sempre mantendo-se em níveis que não afetarão a saúde do usuário, novas restrições também podem ser aplicadas dependendo da estratégia.

2.3 AMPL

AMPL é uma linguagem de modelagem algébrica para programação matemática, ela foi projetado e implementado por Fourer et al. (1990) em 1985.

Foi realizada uma prototipagem do problema proposto nesse trabalho em AMPL com o intuito de verificar se as restrições estavam bem definidas e se o problema tinha solução. A figura 4 mostra a prototipagem inicial do problema em AMPL.

Figura 4 – Prototipagem Inicial AMPL

```
param Pkg = 60;
param Cdu = 2500;
var Qp >= 0;
var Qc >= 0;
var Qg >= 0;
var Qf >= 0;
maximize Qcal: (Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9);

subject to total_cal: (Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9) <= Cdu;
subject to Limite_Qp_1: Qp<=5;
subject to Limite_Qp_2: Qp>=3;
subject to Limite_Qg_1: Qg<=0.6;
subject to Limite_Qg_2: Qg>=0.3;
subject to Valor_Qf: Qf = 15 * (Cdu div 1000);

solve;
display Qp;
display Qc;
display Qg;
display Qf;
display Qcal;
```

Fonte: Elaborado pelo autor

A Figura 5 mostra uma modelagem modular onde pode-se testar diferentes entradas para o problema, a Figura 6 exemplifica valores que podem ser fornecidos para o modelo.

Figura 5 – Prototipagem Modular AMPL

```
param Pkg ;
param Cdu ;
param Qg_s ;
param Qg_i ;
param Qp_s ;
param Qp_i ;
var Qp >= 0;
var Qc >= 0;
var Qg >= 0;
var Qf >= 0;
maximize Qcal: (Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9);

subject to total_cal: (Pkg * Qp * 4) + (Pkg * Qc * 4) + (Pkg * Qg * 9) <= Cdu;
subject to Limite_Qp_1: Qp<=Qp_s;
subject to Limite_Qp_2: Qp>=Qp_i;
subject to Limite_Qg_1: Qg<=Qp_s;
subject to Limite_Qg_2: Qg>=Qp_i;
subject to Valor_Qf: Qf = 15 * (Cdu div 1000);
```

Fonte: Elaborado pelo autor

Figura 6 – Dados da Prototipagem Modular

```
data;

param Pkg := 60;
param Cdu := 2500;
param Qg_s := 1;
param Qg_i := 0.6;
param Qp_s := 3;
param Qp_i := 2;
```

Fonte: Elaborado pelo autor

2.4 Implementação em Python

Com o objetivo de atender as especificações do trabalho, também foi implementado a solução para o problema proposto na linguagem Python, junto com uma interface gráfica visando uma interação mais simples do usuário com o modelo.

O modelo foi implementado utilizando o módulo *linprog* da biblioteca *Scipy.optimize*, o código referente a modelagem proposta pode ser conferido na Figura 7.

Figura 7 – Função Solver_diet

```
def solve_diet(Qp_s=3, Qp_i=2, Qg_s=1, Qg_i=0.6, Cdu=2000, Pkg=72):
    #ordem Qp Qc Qg
    c = [-(Pkg*4), -(Pkg*4), -(Pkg*9)] # coeficientes da função objetivo
    A = [[(Pkg*4), (Pkg*4), (Pkg*9)]] #matriz de coeficientes das restrições
    b = [Cdu] #termos independentes

    qp_bounds = (Qp_i, Qp_s)
    qc_bounds = (None, None)
    qg_bounds = (Qg_i, Qg_s)

    #Método padrão: simplex.
    res = scipy.optimize.linprog(c, A_ub=A, b_ub=b, bounds=(qp_bounds, qc_bounds, qg_bounds),
    | | | options={"disp": True})

    #print(res)
    x=res.x
    print(x[0], x[1], x[2])
    Qp=round(x[0], 3)
    Qc=round(x[1], 3)
    Qg=round(x[2], 3)
    Qcal=(Qp*Pkg*4)+(Qc*Pkg*4)+(Qg*Pkg*9)
    Qf = 15*(Cdu//1000)
    return round((Qp*Pkg),3), round((Qc*Pkg),3), round((Qg*Pkg),3), Qcal, Qf
```

Fonte: Elaborado pelo autor

A interface foi implementada utilizando a biblioteca *PySimpleGUI* e o seu código pode ser visto na figura 8.

Figura 8 – Código da Interface

```
def interface():
    sg.theme('Reddit')
    layout = [
        [sg.Text('Peso em kg do usuário', size=(35,1)), sg.Input(key='Pkg', size=(10,1))],
        [sg.Text('Calorias diárias a serem ingeridas', size=(35,1)), sg.Input(key='Cdu', size=(10,1))],
        [sg.Text('Quantidade mínima de proteína por kg', size=(35,1)), sg.Input(key='Qp_i', size=(10,1))],
        [sg.Text('Quantidade máxima de proteína por kg', size=(35,1)), sg.Input(key='Qp_m', size=(10,1))],
        [sg.Text('Quantidade mínima de gordura por kg', size=(35,1)), sg.Input(key='Qg_i', size=(10,1))],
        [sg.Text('Quantidade máxima de gordura por kg', size=(35,1)), sg.Input(key='Qg_s', size=(10,1))],
        [sg.Button('Calcular Macronutrientes')]
    ]
    #sg.Output(size=(75,3))

janela=sg.Window('Trabalho Prático Otimização', layout)

while True:
    eventos, valores = janela.read()
    if(eventos == sg.WINDOW_CLOSED):
        break

    if(eventos == 'Calcular Macronutrientes'):
        pkg_user = valores['Pkg']
        cdu_user = valores['Cdu']
        qp_i_user = valores['Qp_i']
        qp_s_user = valores['Qp_m']
        qg_i_user = valores['Qg_i']
        qg_s_user = valores['Qg_s']
        Qp_user, Qc_user, Qg_user, total_calories, Qf_user = solve_diet(float(qp_s_user), float(qp_i_user), float(qg_s_user), float(qg_i_user), float(cdu_user), float(pkg_user))
        sg.Print('Quantidade de proteína (em gramas) a ser consumida: ', Qp_user)
        sg.Print('Quantidade de carboidrato (em gramas) a ser consumida: ', Qc_user)
        sg.Print('Quantidade de gordura (em gramas) a ser consumida: ', Qg_user)
        sg.Print('Quantidade de fibras (em gramas) a ser consumida: ', Qf_user)
        sg.Print('Total de calorias a ser consumida: ', total_calories)
```

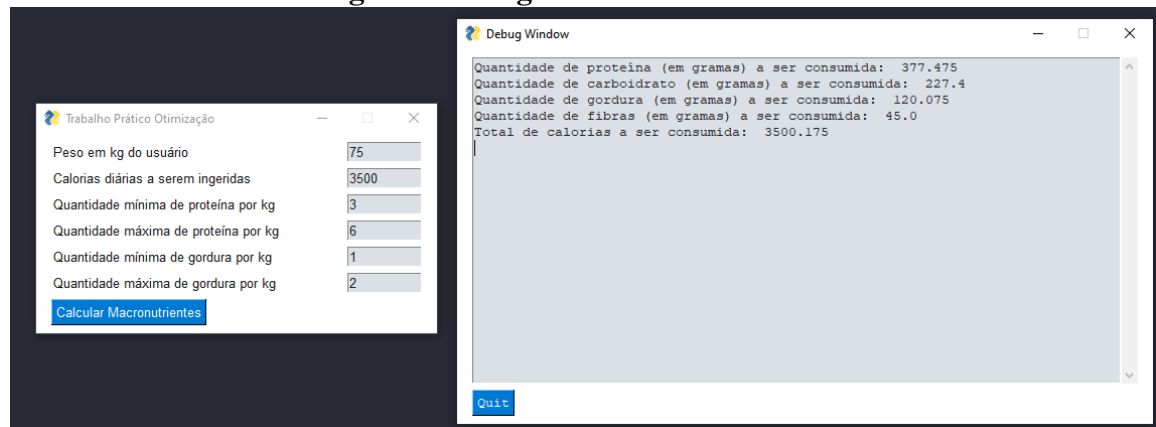
Fonte: Elaborado pelo autor

Ao executar o código a interface da Figura 9 é apresentada ao usuário. Os dados podem ser inseridos em seus respectivos campos e após serem devidamente preenchidos e o botão "*Calcular Macronutrientes*" for pressionado os cálculos são feitos e os resultados apresentados na interface da Figura 10.

Figura 9 – Primeira Parte da Interface

Fonte: Elaborado pelo autor

Figura 10 – Segunda Parte da Interface



Fonte: Elaborado pelo autor

3 CONCLUSÃO

Nesse trabalho, foi apresentado a implementação de uma ferramenta para cálculo de macronutrientes utilizando-se estratégias de otimização. Ao final foi possível identificar um problema do mundo real, apresentar as variáveis envolvidas, o objetivo e as restrições e solucionar o mesmo com ajuda de ferramentas que foram mostradas durante a disciplina.

Os resultados obtidos foram satisfatórios, mas foi possível identificar alguns cenários em que a modelagem em AMPL, em função *solver*, não apresentava uma distribuição equilibrada entre as variáveis Q_p , Q_c , Q_g . A implementação na linguagem Python não apresentou o mesmo problema.

REFERÊNCIAS

ATKINS, CDC. **Dr. Atkins' new diet revolution**. [S.l.]: Government Institutes, 2002.

BOTTURA, Caio Cintra. **Dieta Flexível e Nutrição**. [S.l.: s.n.], 2017.

FOURER, Robert; GAY, David M; KERNIGHAN, Brian W. A modeling language for mathematical programming. **Management Science**, INFORMS, v. 36, n. 5, p. 519–554, 1990.

GHEORGHE, Liana et al. Improvement of hepatic encephalopathy using a modified high-calorie high-protein diet. **Romanian journal of gastroenterology**, v. 14, n. 3, p. 231–238, 2005.