Atividade para casa #5

Método de Newton, Marquardt e Programação linear
Método SIMPLEX e uso de pacotes

16 de junho de 2025

1 Contextualização

1.1 O método de Newton

Esse é mais um dos métodos numéricos que se constrói com base na ideia de expansão em série de Taylor. Podemos pensar no método de Newton aplicado ao contexto de otimização multidimensional sem restrições como uma variante do método de Newton-Raphson, como desenvolvido para obtenção de zeros de funções. Entretanto, com a diferença que nesse caso estamos buscando zerar um gradiente multidimensional de uma função escalar $f(\mathbf{x})$, em que $\mathbf{x}$ está propositalmente representado em negrito para destacar o fato de que é um vetor e não um escalar, ou seja, $\mathbf{x} = (x_1, x_2, x_3, \cdots, x_n)$.

Nesse contexto multidimensional, podemos representar o valor da função $f(\mathbf{x})$ avaliada no ponto $\mathbf{x}$ em termos de seu valor num ponto próximo qualquer $\mathbf{x}_i$ em termos de uma série de Taylor, como

$$f(\mathbf{x}) = f(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i) \cdot \nabla f|_{\mathbf{x}_i} + \frac{(\mathbf{x} - \mathbf{x}_i)^T \cdot \mathbf{H}_i \cdot (\mathbf{x} - \mathbf{x}_i)}{2!} + \mathcal{O}(|(\mathbf{x} - \mathbf{x}_i)|^3), \quad (1)$$

em que $\nabla f|_{\mathbf{x}_i}$ representa o gradiente da função objetivo avaliada em $\mathbf{x}_i$ e $\mathbf{H}_i$ representa a matriz Hessiana do sistema avaliada no mesmo ponto $\mathbf{x}_i$. O termo $\mathcal{O}(|(\mathbf{x} - \mathbf{x}_i)|^3)$ representa os infinitos termos restantes da série que serão regidos por um termo dominante (*leading order*) de ordem de magnitude igual a $|(\mathbf{x} - \mathbf{x}_i)|^3$. Note que nessa representação multidimensional da série de Taylor, a matriz Hessiana desempenha um papel equivalente às derivadas segundas. Caso quiséssemos representar os demais termos da série podemos utilizar recursos de álgebra tensorial. Só a título de curiosidade, o termo $\mathcal{O}(|(\mathbf{x} - \mathbf{x}_i)|^3)$ seria dado por

$$\mathcal{O}(|(\mathbf{x} - \mathbf{x}_i)|^3) = \frac{(\mathbf{x} - \mathbf{x}_i) \otimes (\mathbf{x} - \mathbf{x}_i) \otimes (\mathbf{x} - \mathbf{x}_i) \cdot^3 \nabla\nabla\nabla f}{3!} + \mathcal{O}(|(\mathbf{x} - \mathbf{x}_i)|^4), \quad (2)$$

em que $\otimes$ denota o produto diático, uma operação que quando realizada entre dois vetores gera um tensor de segunda ordem, ou uma matriz 3×3 e $\cdot^3$ define um produto escalar triplo. Essas operações fazem muito mais sentido quando realizadas em notação indicial. Ao leitor interessado em uma compreensão mais clara sobre o assunto recomenda-se às seguintes aulas sobre o tema, disponibilizadas de maneira gratuita no canal do YouTube Ciência e Brisa:

- Notação indicial - Parte 1;

- Notação indicial - Parte 2;
- Mecânica do contínuo e notação indicial;
- Permutador de Levi-Civita e identidade de permutação.

Para nossa apresentação do método de Newton aplicado ao contexto de otimização multidimensional sem restrições vamos truncar nossa série a partir dos termos $\mathcal{O}(|(\mathbf{x}-\mathbf{x}_i)|^3)$. Sabemos também que no ponto ótimo o gradiente da função objetivo deve ser nulo. Aplicando então o gradiente sobre os dois lados da equação (1)e lembrando que o gradiente de $f(\mathbf{x}_i)$ é nulo, uma vez que o ponto $\mathbf{x}_i$ é um ponto fixo no espaço de busca, obtemos

$$\nabla f = \nabla(\mathbf{x}-\mathbf{x}_i)\cdot\nabla f|_{\mathbf{x}_i} + \frac{1}{2}\left[\nabla(\mathbf{x}-\mathbf{x}_i)^T\cdot\mathbf{H}_i\cdot(\mathbf{x}-\mathbf{x}_i) + (\mathbf{x}-\mathbf{x}_i)^T\cdot\mathbf{H}_i\cdot\nabla(\mathbf{x}-\mathbf{x}_i)\right], \quad (3)$$

mas note que $\nabla\mathbf{x}$ é a própria matriz identidade, uma vez que o termo computa as variações espaciais das próprias componentes do espaço, logo $\nabla\mathbf{x} = \mathbf{I}$. Note também que $\nabla\mathbf{x}_i = \mathbf{0}$, pois novamente, $\mathbf{x}_i$ é um ponto fixo no espaço de busca. Dessa forma, podemos expressar o gradiente de f como

$$\nabla f = \mathbf{I}\cdot\nabla f|_{\mathbf{x}_i} + \frac{1}{2}\left[\mathbf{I}^T\cdot\mathbf{H}_i\cdot(\mathbf{x}-\mathbf{x}_i) + (\mathbf{x}-\mathbf{x}_i)^T\cdot\mathbf{H}_i\cdot\mathbf{I}\right] = \nabla f|_{\mathbf{x}_i} + \mathbf{H}_i\cdot(\mathbf{x}-\mathbf{x}_i). \quad (4)$$

Na equação (4) utilizamos a simetria da matriz identidade e da Hessiana para escrevê-la em sua forma final. Se desejamos então saber o próximo ponto $\mathbf{x}$ para o qual devemos caminhar a partir de um ponto $\mathbf{x}_i$ no sentido de zerarmos o gradiente de f, basta usar a equação (4) para obtermos a seguinte relação de evolução:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}_i^{-1}\cdot\nabla f|_{\mathbf{x}_i}. \quad (5)$$

O salto entre a equação (4) e a equação (5) consiste em zerar ∇f e multiplicar os dois lados pela inversa da Hessiana para que por meio do surgimento do termo $\mathbf{H}_i^{-1}\cdot\mathbf{H}_i = \mathbf{I}$ possamos isolar o termo $\mathbf{x}$ e em seguida definí-lo como o próximo ponto na caminhada rumo ao ótimo.

O método de Newton então consiste em partir de um ponto inicial e ir caminhando rumo ao ponto ótimo (onde o gradiente é nulo) por meio da equação (5). Esse é um método que possui de um modo geral um desempenho melhor que o método do aclive máximo e pode ser mais simples de ser implementado do que o método dos gradientes conjugados.

Entretanto, o método de Newton apresenta dois problemas fundamentais. O primeiro deles é o alto custo computacional envolvido a cada passo não só na obtenção da matriz Hessiana do sistema como em sua inversão. Já dissemos antes que inversões matriciais costumam ser processos caros. O outro problema é que dependendo do ponto de partida ele pode divergir. Isso ocorre geralmente quando o ponto inicial está muito longe do ponto ótimo.

1.2 O método de Levenberg-Marquardt

Pensando nas problemáticas apresentadas pelo método de Newton, podemos conceber um método híbrido que une o melhor dos mundos. Um método que inicia sua busca de forma cautelosa a passos mais lentos e seguros para evitar a divergência potencial do método de Newton e que conforme vai se aproximando do ponto ótimo vai transicionando para o método de Newton, encontrando um caminho mais direto rumo ao ponto crítico sem passar pelos habituais zigue-zagues do método do aclive máximo. Esse é o método de Levenberg-Marquardt.

Esse método híbrido parte de uma ideia simples, porém genial, começando com a proposta de uma leve modificação no cálculo da matriz Hessiana do sistema por meio de uma Hessiana modificada $\mathbf{H}_i^*$ dada por

$$\mathbf{H}_i^* = \mathbf{H}_i + \alpha_i \mathbf{I}. \tag{6}$$

A ideia é começar com um valor inicial grande do parâmetro α_i e utilizar $\mathbf{H}_i^*$ no lugar de $\mathbf{H}_i$ na equação (5). Conforme caminhamos no sentido do gradiente e o valor de $\nabla f|\mathbf{x}_i$ vai diminuindo podemos ir diminuindo progressivamente o valor de α_i.

Note que se $\alpha_i \gg 1$, pela equação (6) temos

$$\mathbf{H}_i^* \approx \alpha_i \mathbf{I} \Rightarrow \mathbf{H}_i^{*-1} \approx \frac{1}{\alpha_i}\mathbf{I}. \tag{7}$$

Substituindo (7) em (5) obtemos

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\nabla f|_{\mathbf{x}_i}}{\alpha_i}, \tag{8}$$

que representa exatamente a equação de evolução dos pontos de busca de acordo com o método do aclive máximo, porém com um passo de caminhada dado por $1/\alpha_i$. Como no método de Levenberg-Marquardt escolhemos um valor inicial grande de α_i, o mesmo se inicia como um método do aclive máximo com passos curtos e cautelosos.

Conforme o gradiente de f vai diminuindo vamos progressivamente diminuindo o valor de α_i e a equação de busca vai transicionando de um método do aclive máximo e se transformando no método de Newton. Dessa forma, o método de Levenberg-Marquardt evita o cálculo da inversa da Hessiana no início da busca e passa a caminhar mais diretamente rumo ao ponto crítico conforme se aproxima deste.

A figura (1) ilustra uma trajetória típica de busca pelo método de Levenberg-Marquardt, na qual é possível observar um acúmulo de passos curtos no início na região em que o método se comporta como o método do aclive máximo e os passos largos do método de Newton posteriormente. É possível ver também a evolução do parâmetro α_i e os comandos em Python no encarte responsáveis pela atualização do mesmo.

2 Otimização multidimensional com restrições

Entraremos agora numa nova seção do nosso estudo de técnicas voltadas à problemas de otimização. A ideia aqui é acrescentar restrições ao problema. Essas restrições vão mudar radicalmente a forma de abordarmos o problema. Essas mudanças trarão complexidades adicionais que demandariam cursos específicos para serem endereçadas com o respeito que mercebem, de tal sorte que aqui restringiremos o nosso olhar para problemas multidimensionais sujeitos à restrições nos quais tanto a função objetivo quanto suas restrições são lineares. Ficaremos portanto restritos aqui ao universo do que se conhece como *programação linear*.

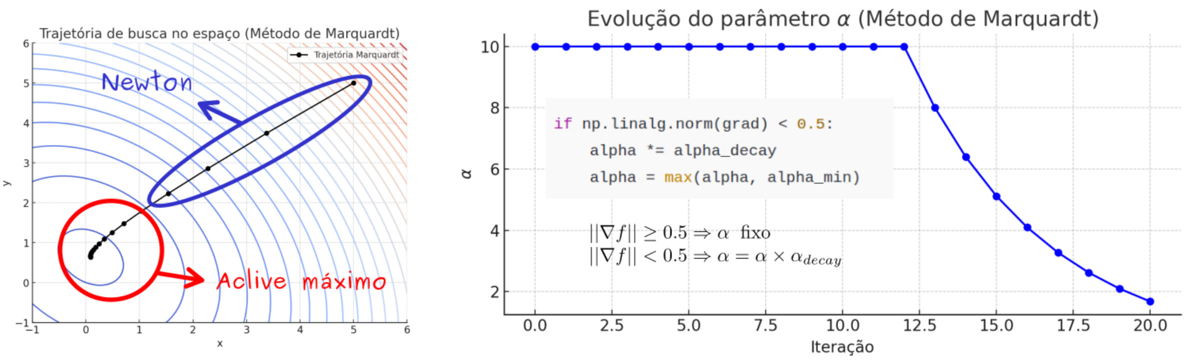

Figura 1: Trajetória de busca no método de Levenber-Marquardt e evolução do parâmetro α_i num exemplo genérico

2.1 Como formular um problema de programação linear?

Antes de adentrarmos no estudo formal de técnicas relacionadas à programação linear é importante definirmos algumas coisas. A primeira delas diz respeito à terminologia. Aqui a palavra *programação* não é utilizada no sentido usual de uma atividade voltada ao desenvolvimento de programas de computador. O termo se refere mais a uma atividade de planejamento.

Quando pensamos em um plano para realizar determinada tarefa precisamos traçar uma estratégia, definir um programa a ser executado. Esse termo faz sentido quando pensamos na origem da programação linear como ciência. Veremos aqui que o método mais famoso de solução de problemas de programação linear é o método SIMPLEX, desenvolvido por George Bernard Dantzig, um cientista e matemático norte-americano, que juntamente com Tjalling Koopmans e Leonid Kantorovich é conhecido com um dos pais da programação linear. Em 1975 Koopsman e Kantorovich dividiram o prêmio nobel de economia por suas contribuições para a teoria da alocação ótima de recursos. Nesse sentido, a programação linear surge como técnica para solução de problemas associados a temas que envolvem planejamento de recursos visando otimizar cenários econômicos. Por isso faz sentido pensarmos aqui na palavra *programação* em termos de planejamento.

Uma outra questão que precisamos definir diz respeito à forma de enunciar um problema relacionado à programação linear. Para isso, vamos a um exemplo genérico associado à estimativa de lucro de uma empresa em função da comercialização de suas atividades econômicas, que podem ser tanto produtos quanto serviços. Nesse caso, vamos definir a nossa função objetivo Z como sendo a receita total pela comercialização dessas atividades. Essa receita é definida matematicamente por

$$Z = c_1 x_1 + c_2 x_2 + \cdots c_n x_n, \tag{9}$$

em que $c_1, c_2, \cdots, c_n$ representam respectivamente os valores de venda vinculados às atividades econômicas (produtos ou serviços) $1, 2, \cdots, n$ e $x_1, x_2, \cdots, x_n$ representam as quantidades comercializadas das atividades $1, 2, \cdots, n$. A equação (9) representa portanto a receita total obtida pela venda dos n produtos. Agora, todos sabemos que a produção dessas atividades tem um custo que está associado à disponibilidade finita dos recursos disponíveis. Dessa forma, precisamos introduzir inequações adicionais associadas à disponibilidade desses recursos. Essas restrições podem ser expressas pela seguinte equação compacta:

$$a_{i1} x_1 + a_{i2} x_2 + \cdots a_{in} x_n \leq b_i, \tag{10}$$

em que a_{ij} representa a quantidade do i-ésimo recurso disponível para a atividade j e b_i representa a quantidade total do i-ésimo recurso disponível. Aqui interpretamos a equação (10) como um conjunto de equações que representa todas as restrições de todos os tipos de recursos vinculados à todas as atividades. Imagine por exemplo que no contexto de uma fábrica qualquer, teremos restrições com relação à quantidade de matéria prima disponível, quantidade de tempo de produção disponível por semana e mesmo de espaço de armazenamento para composição do estoque de produtos que será vendido. Esses três tipos de restrições podem ser representados genericamente pela equação (10). Além disso, temos um segundo tipo bem geral de restrição que diz simplesmente que $x_i \geq 0$, ou seja, não podemos produzir atividades negativas.

Para entendermos melhor como formular um problema de programação linear vamos a um exemplo. Considere que você é o gerente de uma indústria de processamento de gás. A sua fábrica recebe um gás bruto (não processado) semanalmente a uma quantidade fixa. Esse gás é então processado pela sua fábrica em dois tipos possíveis, um gás regular e um gás prêmio. Esses dois produtos possuem valores de venda diferentes, tempos de produção diferentes, características físicas diferentes e estão sujeitos à restrições de espaço de armazenamento distintas também. A pergunta central aqui é: *como descobrir a quantidade ótima de produção de cada tipo de gás que gere o maior lucro para a empresa atendendo todas as restrições impostas?*

Para darmos corpo à formalização deste problema, considere os dados da tabela (1).

Recurso	**Gás regular**	**Gás premium**	**Disponibilidade**
Gás bruto	7 m^3/ton	11 m^3/ton	77 m^3/semana
Tempo de produção	10 h/ton	8 h/ton	80 h/semana
Armazenamento	9 tons/semana	6 tons/semana	-
Lucro	150 R$/ton	175 R$/ton	-

Tabela 1: Informações para montagem de um problema-exemplo em programação linear

Dessa forma, chamando de x_1 a quantidade de gás regular produzida por semana e de x_2 a quantidade de gás premium produzida no mesmo período, temos que a nossa função objetivo é dada por

$$Z = 150x_1 + 175x_2, \tag{11}$$

além disso podemos expressar as restrições, uma a uma, por

1. Disponibilidade de gás $\longrightarrow 7x_1 + 11x_2 \leq 77$;
2. Disponibilidade de tempo de produção $\longrightarrow 10x_1 + 8x_2 \leq 80$;
3. Restrição de armazenamento 1 $\longrightarrow x_1 \leq 9$;
4. Restrição de armazenamento 2 $\longrightarrow x_2 \leq 6$;
5. Restrição de positividade 1 $\longrightarrow x_1 \geq 0$;
6. Restrição de positividade 2 $\longrightarrow x_2 \geq 0$.

Nosso problema passa a ser então descobrir a combinação x_1 e x_2 que maximiza Z atendendo simultaneamente as 6 restrições do problema. O problema está formulado. Resta resolvermos.

2.2 Solução pelo método gráfico

Uma primeira forma de solução consiste na aplicação do bom e velho método gráfico. A ideia seria traçar retas que expressem $x_2 = f(x_1)$ a partir das restrições, utilizando os pontos limites das mesmas por meio da transformação das desigualdades em igualdades. Esse seria o primeiro passo do processo. Nesse passo ainda não incorporamos a função objetivo, pois estamos primeiramente tentando definir o nosso espaço de busca. A figura (2) ilustra de maneira clara o processo de delimitação de um espaço de busca em termos da sobreposição das 6 restrições associadas ao nosso problema.

O próximo passo consiste em incorporar a esse espaço de busca a nossa função objetivo. Para isso, podemos isolar $x_2 = f(x_1, Z)$ a partir equação (11) para obter

$$x_2 = \frac{Z}{175} - \frac{150}{175}x_1. \tag{12}$$

A ideia aqui consiste em testarmos graficamente diferentes valores de Z visando identificar a reta limite que toque um ponto extremo do espaço de busca com o maior valor possível da função objetivo. A figura (3) ilustra esse procedimento. Nela é possível notar que o ponto C representa o ponto extremo onde a função objetivo é maximizada e ainda atende as restrições impostas pelo problema. Nesse ponto temos a nossa solução pelo método gráfico, que fornece $x_1 = 4.9$ e $x_2 = 3.9$.

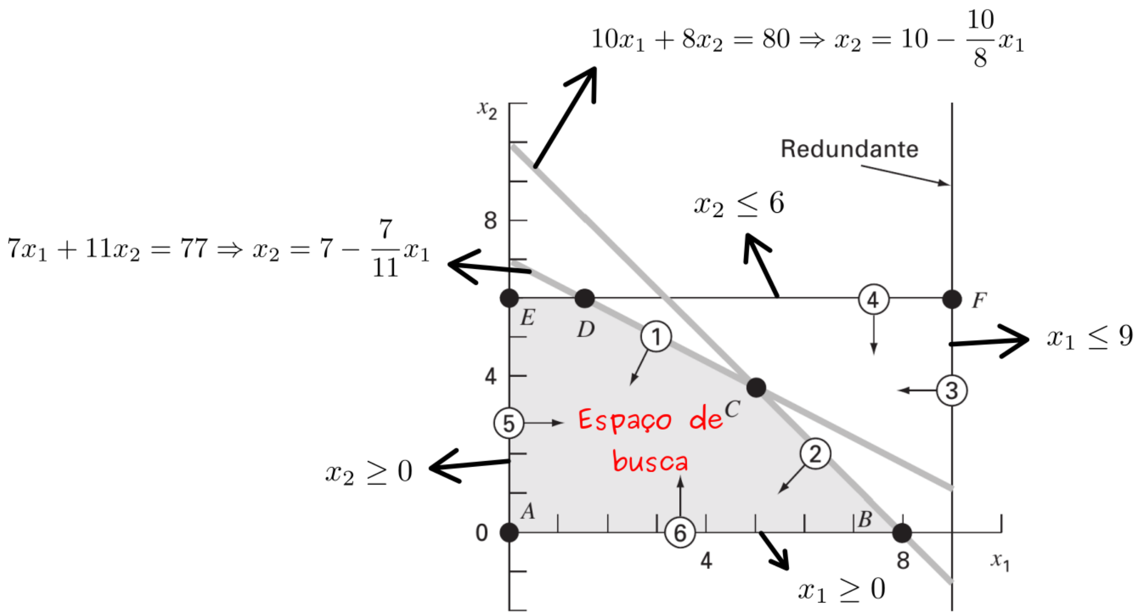

Figura 2: Definição de um espaço de busca para solução de um problema simples de programação linear usando o método gráfico

2.3 O algoritmo SIMPLEX

Uma desvantagem clara do método gráfico é justamente sua impossibilidade de resolver problemas de dimensões maiores. Como representar espaços de busca multidimensionais em problemas de programação linear para os quais a função objetivo depende de várias variáveis? No caso de uma função objetivo $f(x_1, x_2, x_3)$ até conseguimos obter uma representação visual de uma possível região de busca, como mostrada na figura (4). Nesse

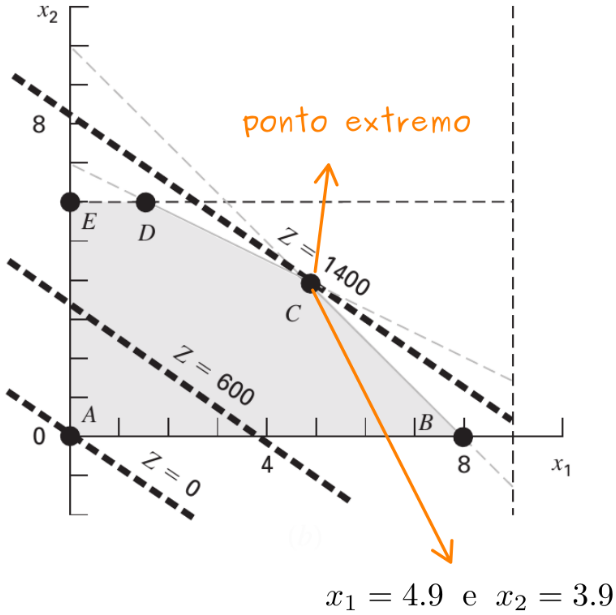

Figura 3: Sobreposição da função objetivo ao nosso espaço de busca para solução de um problema simples de programação linear pelo método gráfico

cenário teríamos que sobrepor o plano da função objetivo a esse poliedro irregular e identificar um ponto do espaço no qual esse plano cruza o poliedro em um de seus pontos extremos com seu valor máximo.

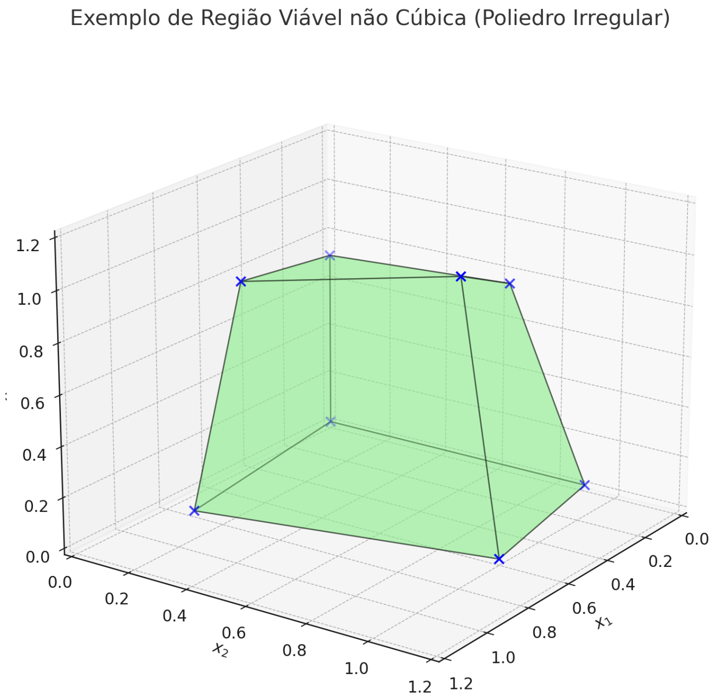

Figura 4: Representação genérica de uma região de busca tridimensional para um problema de programação linear de 3 variáveis

Entretanto, para dimensões superiores à 3 já não temos mais como recorrer ao método gráfico para a solução do problema. É nesse ponto que o algoritmo SIMPLEX ou método SIMPLEX se torna necessário, pois ele navega pelas faces dessa entidade matemática conhecida como politopo buscando o ponto ótimo. Aqui nos referimos a um politopo simplesmente como uma generalização matemática de um polígono ou poliedro de dimensão arbitrária. No caso 2D chamamos o politopo de polígono, uma região limitada por segumentos de reta. No caso 3D o politopo recebe o nome de poliedro, uma região limitada por faces planas. Em dimensões maiores que três dizemos simplesmente que o politopo é uma região convexa limitada por hiperplanos. No contexto da programação linear os hiperplanos são definidos pelas restrições do problema e o espaço de busca é definido pelo politopo formado pela intersecção de todos esses hiperplanos.

Para entendermos o método SIMPLEX, precisamos definir inicialmente o conceito de variáveis de folga. Esse conceito é fundamento no método e consiste em um passo crucial para transformarmos as inequações relacionadas às restrições em equações. Vamos então prosseguir por meio do nosso exemplo anterior. A aplicação das variáveis de folga s_1, s_2, s_3, s_4 para as quatro primeiras restrições do nosso problema exemplo fornece:

$$7x_1 + 11x_2 \leq 77 \Rightarrow 7x_1 + 11x_2 + s_1 = 77 \tag{13}$$

$$10x_1 + 8x_2 \leq 80 \Rightarrow 10x_1 + 8x_2 + s_2 = 80 \tag{14}$$

$$x_1 \leq 9 \Rightarrow x_1 + s_3 = 9 \tag{15}$$

$$x_2 \leq 6 \Rightarrow x_2 + s_4 = 6 \tag{16}$$

Note que poderíamos num primeiro momento pensar na possibilidade de expressarmos as equações relacionadas às restrições acima em termos de um sistema linear. Porém, o nosso sistema ficaria subespecificado, uma vez que essas 4 restrições levam à 4 equações, mas se relacionam com 6 icógnitas, que incluem as duas variáveis estruturais x_1 e x_2 além das quatro variáveis de folga s_1, s_2, s_3, s_4. Note também que no nosso contexto, qualquer ponto extremo do nosso espaço de busca é composto apenas por um par de informações, justamente o número de variáveis estruturais do problema. Voltando à figura (2), é fácil perceber que os 5 pontos extremos, limitantes do polígono que compreende o espaço de busca, ocorrem sempre para um par de variáveis nulas, entre as seis variáveis totais que compõem o nosso sistema $(x_1, x_2, s_1, s_2, s_3, s_4)$. Essa informação é mais fácil de ser visualizada na tabela (2).

Ponto extremo	**Variáveis nulas**
A	x_1, x_2
B	x_2, s_2
C	s_1, s_2
D	s_1, s_4
E	x_1, s_4

Tabela 2: Relações do par de variáveis nula em cada ponto extremo do espaço de busca para o nosso problema-exemplo

Nesse contexto, quando zeramos uma variável de folga relacionada a uma restrição específica estamos definindo um ponto extremo sobre a reta no espaço de busca (para um caso 2D) que define aquela restrição. Faça o exercício de tentar relacionar os pontos expressos na tabela (2) com a figura (2).

A ideia do método SIMPLEX começa com a proposta de calcular inicialmente todos os pontos extremos simplesmente zerando duas variáveis por vez e resolvendo os sistemas lineares bem especificados, resultantes desse processo. Ao final dessa etapa teríamos todos os pontos extremos relacionados à intersecção de todas as restrições impostas. Em seguida, aplica-se um processo de filtragem para eliminarmos os pontos extremos que gerem soluções inviáveis, pois nem todos os pontos extremos associados às intersecções das restrições cairão no espaço de busca. Isso ficará mais claro logo mais com um exemplo numérico. Finalmente, entre os pontos extremos viáveis, caminha-se entre eles buscando aquele que gere o maior valor da função objetivo.

Para entendermos melhor como esses passos são feitos, voltemos ao nosso exemplo modelo. Vamos estimar por exemplo as coordenadas do ponto E da figura (2). Para esse ponto extremo, temos pela tabela (2) que $x_1 = 0$ e $s_4 = 0$, o que nos permite escrever o sistema de inequações referente às restrições do problema utilizando as variáveis de folga s_1, s_2, s_3 como

$$\begin{aligned} 11x_2 + s_1 &= 77 \\ 8x_2 + s_2 &= 80 \\ s_3 &= 9 \\ x_2 &= 6. \end{aligned} \tag{17}$$

As equações expressas em (17) podem ser representadas na forma de um sistema linear como

$$\begin{pmatrix} 11 & 1 & 0 & 0 \\ 8 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{Bmatrix} x_2 \\ s_1 \\ s_2 \\ s_3 \end{Bmatrix} = \begin{Bmatrix} 77 \\ 80 \\ 9 \\ 6 \end{Bmatrix}, \tag{18}$$

cuja solução fornece $x_2 = 6$, $s_1 = 11$, $s_2 = 32$ e $s_3 = 9$ com $x_1 = s_4 = 0$. Esses valores representam de fato o ponto E que aparece na figura (2). Antes de prosseguirmos com a organização da estrutura do algoritmo SIMPLEX, precisamos entender por que nem todos os pontos extremos calculados dessa forma constituem possibilidades viáveis de solução para o nosso problema.

Note que no nosso exemplo temos 6 variáveis no total e precisamos zerar 2 por vez para estimarmos os pontos extremos. Dessa forma, ao zerarmos 2 variáveis sobram 4 por grupo de solução. Podemos estimar o número total de grupos contendo 4 elementos de um total de 6 em termos de uma combinação. Imagine que o número total de elementos nesse caso constitui 6 letras possíveis para formarmos palavras. A pergunta é: quantos grupos podemos formar contendo 4 letras extraídas de um total de 6, nas quais a ordem não importa? A resposta está nos nossos estudos de análise combinatória e fornece a seguinte fórmula

$$N_G = \frac{N_{vt}!}{N_{vpg}!(N_{vt} - N_{vpg})!}, \tag{19}$$

em que N_G é o número total de grupos, N_{vt} é o número total de variáveis (nesse caso 6) e N_{vpg} é o número de variáveis por grupo (nesse caso 4). Essa relação pode ser representada em termos do número de variáveis estruturais e de folga do problema como

$$N_{pe} = \frac{(N_{ve} + N_{vf})!}{N_{vf}!V_{ve}!}, \tag{20}$$

em que N_{pe} é o número de pontos extremos do problema, N_{ve} representa o número de variáveis estruturais e N_{vf} o número de variáveis de folga. No nosso exemplo, teríamos 15 pontos extremos, cada um estimado por meio da solução de um sistema linear 4×4. Mais ainda, pela figura (2) sabemos que apenas 5 desses 15 pontos são viáveis. A figura (5) ilustra em vermelho diversos pontos extremos que constituem candidatos inviáveis para o ponto ótimo que buscamos. Apenas como título de curiosidade, se quiséssemos resolver um problema mais complexo contendo 10 restrições e 6 variáveis estruturais precisaríamos resolver 8008 sistemas lineares 10×10 apenas para descobrir os pontos extremos que atendam às restrições, sendo que vários destes não serão pontos viáveis.

Precisamos então de um algoritmo eficiente para resolver esse problema. Para isso, mostraremos então por meio da solução do nosso problema-exemplo, a aplicação do método SIMPLEX. O ponto de partida do método consiste na obtenção de um ponto extremo viável que seja óbvio, mesmo que claramente não seja o ponto ótimo que procuramos. No nosso exemplo, esse ponto seria a origem do nosso sistema de coordenadas estrututrais, ou seja, $x_1 = x_2 = 0$. Esse ponto é óbvio no sentido que atende à restrição de positividade para as duas variáveis estruturais. É claro que nossa receita máxima não ocorrerá quando a quantidade de nossas atividades econômicas for nula. Entretanto, precisamos de um ponto de partida que atenda às restrições impostas e pelo menos em termos das intersecções das restrições seja viável.

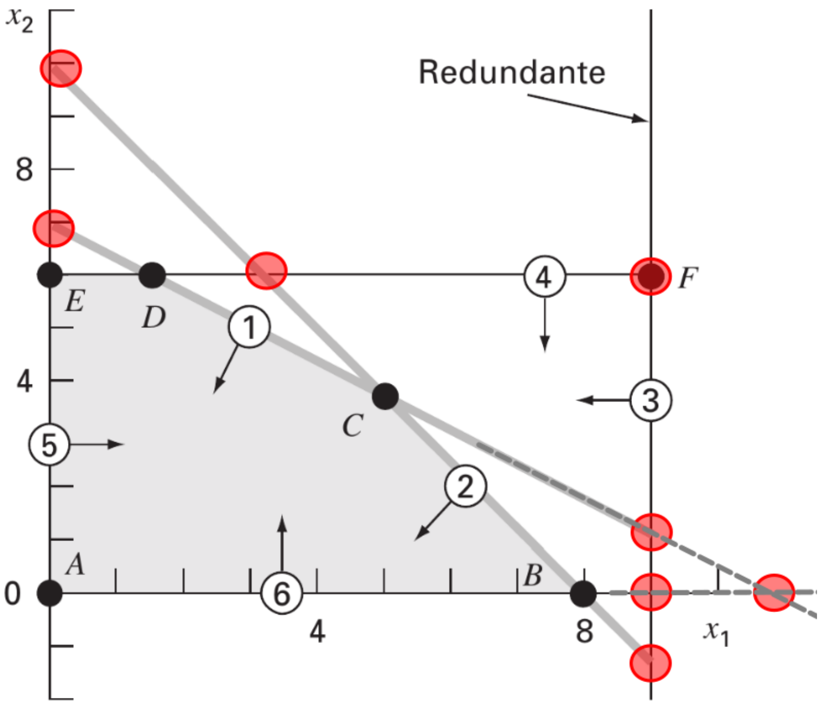

Figura 5: Representação gráfica de pontos extrermos não-viáveis para ilustração da etapa de filtragem do algoritmo SIMPLEX

A determinação das variáveis de folga para esse ponto inicial é feita pela solução do seguinte sistema linear

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{Bmatrix} = \begin{Bmatrix} 77 \\ 80 \\ 9 \\ 6 \end{Bmatrix}. \tag{21}$$

O que o método SIMPLEX propõe não é necessariamente a solução direta desse sistema, mas a organização dessas informações na forma de uma tabela conhecida no jargão da programação linear como *tableau.* Essa tabela é resumida em (3).

Note que a primeira linha do tableau compreende a inclusão da função objetivo, que até o momento não havia aparecido na descrição do método SIMPLEX. Além disso, a primeira coluna representa a seleção das variáveis não nulas da rodada atual. Lembre-se que a ideia do método SIMPLEX consiste em caminhar entre pontos extremos viáveis do espaço de busca a fim de encontrar aquele que otimize a função objetivo. As linhas subsequentes representam de fato as restrições impostas. Note que a linha referente à restrição 1, ou seja, a linha de s_1 possui coeficientes não nulos que multiplicam x_1 e x_2 (7 e 11 respectivamente) e que vem da imposição da restrição de quantidade total de matéria prima. O interessante aqui é que estamos fazendo x_1 e x_2 nulos nessa primeira rodada, de tal sorte que esses coeficientes (7 e 11) na linha de s_1 não terão qualquer impacto sobre o sistema linear representado pela equação (21). Note também que a coluna relacionada ao que chamamos de intersecção ainda não foi preenchida. Abordaremos esse ponto logo mais.

Básicas	Z	x_1	x_2	s_1	s_2	s_3	s_4	Solução	Intersecção
Z	1	-150	-175	0	0	0	0	0	?
s_1	0	7	11	1	0	0	0	77	?
s_2	0	10	8	0	1	0	0	80	?
s_3	0	1	0	0	0	1	0	9	?
s_4	0	0	1	0	0	0	1	6	?

Tabela 3: Tableau para a primeira rodada do algoritmo SIMPLEX

O próximo passo no algoritmo SIMPLEX envolve o uso da função objetivo. A ideia aqui é avançarmos rumo a um outro ponto viável que faça essa função crescer. Por enquanto x_1 e x_2 são nulos. Precisamos fazer com que um dos dois deixe de ser nulo e entre para a lista das variáveis não nulas do problema. Sabemos que a função objetivo é dada por $Z = 150x_2 + 175x_2$, de tal sorte que x_2 gera um crescimento mais rápido de Z do que x_1, sendo portanto a escolha lógica para entrar no problema. Entretanto, para tornar esse exemplo mais breve, vamos usar uma informação que já temos da solução gráfica. No nosso contexto se fizermos $x_2 = 0$ e $x_1 \neq 0$ cairemos no próximo ponto extremo viável B, enquanto se fizermos $x_1 = 0$ e $x_2 \neq 0$ cairemos no ponto E. Sabemos que a solução que buscamos está no ponto C, que no nosso exemplo está mais perto B do que de E, portanto, pela brevidade do exemplo faremos $x_2 = 0$ e $x_1 \neq 0$.

Assim, chamaremos agora x_2 de variável de saída, pois continuará sendo nula, não pertencendo portanto à categoria de variáveis que passarão a entrar no problema, diferentemente de x_1 que por ser agora não-nulo entrará ativamente no problema, constituindo uma variável de entrada. Uma questão importante é que para que x_1 possa entrar no problema precisamos escolher uma das variáveis anteriores para sair. Temos portanto 4 opções: s_1, s_2, s_3, s_4. Avaliaremos cada uma dessas 4 opções individualmente.

- Se fizermos $s_1 = 0$ com $x_2 = 0$ cairíamos no ponto G da figura (6), o que é um ponto inviável;

- Se fizermos $s_2 = 0$ com $x_2 = 0$, cairíamos no ponto B da figura (6), o que é um ponto viável;

- Se fizermos $s_3 = 0$ com $x_2 = 0$, cairíamos no ponto H da figura (6), o que é um ponto inviável;
- Se fizermos $s_4 = 0$ com $x_2 = 0$, esses pontos nunca se encontrariam, o que leva a uma intersecção em ∞, o que certamente é um ponto inviável;

A questão central é: como fazer essa escolha matematicamente, sem precisarmos olhar graficamente para o espaço de busca? É aqui que entra aquela coluna intersecção, ainda não explorada no tableau que montamos para fazer a primeira rodada de aplicação do método SIMPLEX. O procedimento consiste em verificar para cada variável da rodada onde a mesma intercepta o eixo x_2, ou seja, a última variável de saída do problema. A versão estendida do tableau apresentado em (3) é mostrada na tabela (4), agora com a coluna referente à intersecção devidamente preenchida. A escolha da variável que sairá da rodada é baseada no menor valor dessa intersecção, pois a mesma seria a primeira a ser atingida no processo de varredura dos pontos extremos.

Básicas	Z	x_1	x_2	s_1	s_2	s_3	s_4	Solução	Intersecção
Z	1	-150	-175	0	0	0	0	0	-
s_1	0	7	11	1	0	0	0	77	11
s_2	0	10	8	0	1	0	0	80	8
s_3	0	1	0	0	0	1	0	9	9
s_4	0	0	1	0	0	0	1	6	∞

Tabela 4: Tableau com a adição da coluna **intersecção**

O próximo passo consiste na montagem desse novo sistema, com $x_2 = s_2 = 0$, temos

$$7x_1 + s_1 = 77 \tag{22}$$
$$10x_1 = 80 \tag{23}$$
$$x_1 + s_3 = 9 \tag{24}$$
$$s_4 = 6, \tag{25}$$

cuja solução fornece $x_1 = 8, s_1 = 21, s_3 = 1, s_4 = 6$. Essa combinação de valores representa justamente o ponto B. Precisamos agora atualizar o tableau para o ponto B. Esse tableau irá nortear o próximo movimento dentro da nossa caminhada pelas arestas do polígono associado à zona de busca.

Lembre-se que começamos o processo no ponto A, com $x_1 = x_2 = 0$. Em seguida, fizemos $x_1 \neq 0$, mantendo $x_2 = 0$ e caminhamos rumo ao ponto B. Para esse ponto, precisamos refazer o tableau, com as novas variáveis básicas, dadas por Z, s_1, x_1, s_3, s_4. Em outras palavras, temos que documentar no tableau a retirada de s_2 com a respectiva inclusão de x_1 no elenco das novas variáveis básicas. Esse tableau é representado, ainda sem a coluna de intersecção na tabela (5).

Note que a linha anteriormente associada à s_2 e agora relacionada à x_1 foi dividida pelo elemento pivô. Esse procedimento foi intencional, uma vez que o próximo passo consiste numa estratégia de eliminação. A ideia agora é eliminar todos os valores da coluna relacionada à x_1 nas linhas acima e abaixo da linha referente à x_1. Se multiplicarmos por exemplo a linha referente à x_1 no tableau (5) por 150 e subtrairmos esse valor da primeira linha, teremos uma nova linha 1 referente à função objetivo expressa por:

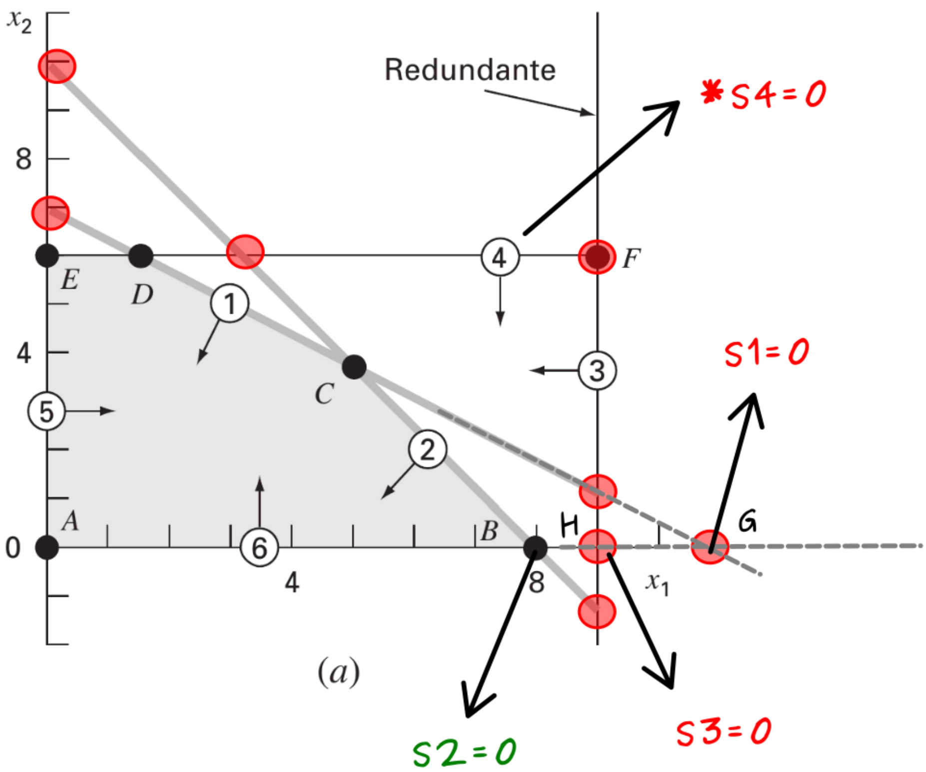

Figura 6: Pontos extremos inviáveis formados pela eliminação de uma ou mais restrições no algoritmo SIMPLEX

Básicas	Z	x_1	x_2	s_1	s_2	s_3	s_4	Solução	Intersecção
Z	1	-150	-175	0	0	0	0	0	-
s_1	0	7	11	1	0	0	0	77	?
x_1	0	1	0.8	0	0.1	0	0	8	?
s_3	0	1	0	0	0	1	0	9	?
s_4	0	0	1	0	0	0	1	6	?

Tabela 5: Tableau inicial para a segunda rodada do algoritmo SIMPLEX

Básicas	Z	x_1	x_2	s_1	s_2	s_3	s_4	Solução	Intersecção
Z antiga	1	-150	-175	0	0	0	0	0	-
Z nova	1	0	-55	0	15	0	0	1200	-

Tabela 6: Processo de eliminação para a primeira linha do tableau

2.4 Para casa: algoritmo SIMPLEX

1. Aplique o processo de eliminação para as demais linhas do tableau e mostre que o novo tableau é dado por:

2. Para esse novo tableau quais são as novas variáveis de entrada e de saída?

3. Repita os passos seguintes e mostre como a solução avança para o ponto C;

Básicas	Z	x_1	x_2	s_1	s_2	s_3	s_4	Solução	Intersecção
Z	1	0	-55	0	15	0	0	1200	-
s_1	0	0	5.4	1	-0.7	0	0	21	3.889
x_1	0	1	0.8	0	0.1	0	0	8	10
s_3	0	0	-0.8	0	-0.1	1	0	1	-1.25
s_4	0	0	1	0	0	0	1	6	6

Tabela 7: Tableau final para a segunda rodada do algoritmo SIMPLEX

4. Mostre por meio do tableau que a solução final é dada por $Z = 1413.889$, $x_1 = 4.889$ e $x_2 = 3.889$;

5. Que informações nesse novo tableau indicam que esta é a solução final?

2.5 Programação linear com pacotes

Apesar de extremamente eficiente, o método SIMPLEX pode ficar muito complicado de ser aplicado manualmente, especialmente para problemas de dimensões mais elevadas. Por conta desse motivo, outras alternativas mais robustas e eficientes tem surgido e ganhado espaço no campo da programação linear. Hoje em dia, com a disseminação em escala global do uso de linguagens de programação como o Python, onde uma extensa comunidade de usuários disponibiliza bibliotecas com subrotinas prontas para os mais diversos problemas matemáticos, é possível resolver problemas complexos de programação linear de maneira eficaz.

Um script simples em Python que resolve o nosso problema exemplo por meio da aplicação de um conjunto de algoritmos open-source, denominado HIGHS (disponível em Python na biblioteca Scipy que utiliza a função linprog do sub-pacote optimize) é apresentado abaixo. O conjunto de algoritmos HIGHS inclui versões otimizadas de variantes do método SIMPLEX que conseguem caminhar não só pelos vértices do politopo de busca, mas podem cruzara própria região de busca, encontrando mais rapidamente os pontos ótimos quando comparada ao método SIMPLEX clássico que apresentamos aqui. Além disso, seus algoritmos permitem tanto processamento em série quanto em paralelo e podem ser aplicados também em problemas não-lineares no contexto da programação quadrática.

```
import numpy as np
from scipy.optimize import linprog

# Exemplo didatico de Programacao Linear com metodo SIMPLEX
# Problema em 2 variaveis, 4 restricoes lineares

# Objetivo: Max z = c^T x  -->  min -z = -c^T x
c = np.array([-150, -175])  # Max 150x1 + 175x2 --> min -150x1 -175x2

# Restricoes (A_ub x <= b_ub)
# Exemplo de 4 restricoes
A_ub = np.array([
    [7, 11],
    [10, 8],
    [1, 0],
    [0, 1]
```

```
])

b_ub = np.array([
    77,
    80,
    9,
    6
])

# Chamando o solver linprog com metodo SIMPLEX
# OBS: Em versoes recentes do scipy, o metodo 'simplex' foi descontinuado.
# Podemos usar 'highs' (que tambem navega vertices, mas de forma mais robusta).
# Para visualizacao didatica aqui, usaremos 'highs'.
res = linprog(c, A_ub=A_ub, b_ub=b_ub, method='highs')

# Resultados
print("Status da otimizacao:", res.message)
print("Valor otimo da funcao objetivo:", -res.fun)
print("Variaveis estruturais (x1, x2):", res.x)
```

Listing 1: Código simplificado para solução de um problema de programação linear usando HIGHS

2.6 Para casa: programação linear usando pacotes

Elabore um problema prático de programação linear, contendo pelo menos 4 variáveis estruturais e um número maior ou igual de restrições. Tente formular o seu problema de maneira criativa. Faça uma pesquisa para encontrar valores realistas associados à modelagem do problema proposto. Enuncie de maneira clara o problema a ser resolvido, incluindo tabelas com as informações referentes às restrições e estruture todo o seu equacionamento com o formalismo matemático aqui apresentado. Em seguida, resolva o seu problema utilizando um código semelhante ao que apresentamos aqui que utilize o método highs da função linprog em Python.