

A forma normal (ou forma padrão) de um problema de programação linear é uma maneira específica e estruturada de representar o problema, tornando-o mais fácil de ser resolvido por algoritmos, como o Método Simplex. Essencialmente, ela transforma todas as restrições e o objetivo em um formato consistente.

---

## Componentes da Forma Normal

Um problema de programação linear está na forma normal se ele atende às seguintes condições:

- **Função Objetivo de Minimização:** A função objetivo deve ser sempre de minimização. Se o problema original for de maximização, você pode facilmente convertê-lo em minimização multiplicando a função objetivo por -1. Por exemplo, maximizar  $Z=c_1x_1+c_2x_2$  é equivalente a minimizar  $-Z=-c_1x_1-c_2x_2$ .
  - **Restrições de Igualdade:** Todas as restrições devem ser expressas como igualdades. Restrições do tipo "menor ou igual a" ( $\leq$ ) ou "maior ou igual a" ( $\geq$ ) são convertidas em igualdades.
    - Para restrições do tipo  $\leq$ , adicionamos uma variável de folga (slack variable). Por exemplo,  $2x_1+3x_2\leq 10$  se torna  $2x_1+3x_2+s_1=10$ , onde  $s_1\geq 0$ . A variável de folga representa a "folga" ou o recurso não utilizado.
    - Para restrições do tipo  $\geq$ , subtraímos uma variável de excesso (surplus variable). Por exemplo,  $4x_1+x_2\geq 5$  se torna  $4x_1+x_2-e_1=5$ , onde  $e_1\geq 0$ . A variável de excesso representa o "excesso" ou a quantidade que excede o limite mínimo.
  - **Variáveis Não Negativas:** Todas as variáveis de decisão devem ser não negativas ( $x_i\geq 0$ ). Se uma variável for irrestrita em sinal (pode ser positiva, negativa ou zero), ela pode ser substituída pela diferença de duas variáveis não negativas. Por exemplo, se  $x_j$  é irrestrita, você a substitui por  $x_j'-x_j''$ , onde  $x_j'\geq 0$  e  $x_j''\geq 0$ .
  - **Lado Direito das Restrições (RHS) Não Negativo:** O lado direito (constante) de todas as restrições de igualdade deve ser não negativo. Se um lado direito for negativo, basta multiplicar a restrição inteira por -1. Por exemplo,  $-x_1+2x_2=-7$  se torna  $x_1-2x_2=7$ . Lembre-se que multiplicar por -1 também inverte o sinal de uma desigualdade, então você precisaria ajustar a adição/subtração de variáveis de folga/excesso.
-

## Por que a Forma Normal é Importante?

A forma normal é crucial por várias razões:

- **Consistência para Algoritmos:** Algoritmos de otimização, como o Método Simplex, são projetados para trabalhar com problemas em uma forma padronizada. A forma normal fornece essa padronização, simplificando a lógica do algoritmo.
- **Identificação de Soluções Básicas:** A forma normal facilita a identificação de soluções básicas viáveis, que são os pontos de partida para a busca da solução ótima no Método Simplex.
- **Matriz de Identidade:** Ao introduzir variáveis de folga e excesso, muitas vezes é possível obter uma submatriz identidade nas restrições, o que é fundamental para a inicialização do Método Simplex.

---

## Exemplo de Conversão para a Forma Normal

Considere o seguinte problema de programação linear:

Maximizar  $Z=3x_1+2x_2$

Sujeito a:

$2x_1+x_2\leq 10$

$x_1+3x_2\geq 6$

$x_1\geq 0, x_2$  irrestrita

de computação:

---

## Enunciados de Problemas de Programação Linear - Contexto de Computação

Problema 1: Otimização de Uso de Servidores em um Data Center

Uma empresa de cloud computing possui um data center com dois tipos de servidores: Servidores A e Servidores B.

- **Servidor A:** Custa \$100/dia para operar, pode processar 500 requisições/segundo e consome 10 kWh de energia.
- **Servidor B:** Custa \$150/dia para operar, pode processar 800 requisições/segundo e consome 15 kWh de energia.

A empresa precisa garantir uma capacidade de processamento mínima de 10.000 requisições/segundo e tem um orçamento diário máximo de \$3.000 para operação de servidores. Além disso, o consumo total de energia não deve exceder 200 kWh por dia.

Objetivo: Minimizar o custo diário total de operação, atendendo às demandas de processamento e restrições de energia.

---

## Problema 2: Alocação de Tarefas para Processadores Heterogêneos

Um sistema de processamento distribuído tem três processadores com diferentes capacidades e custos de utilização por unidade de tempo:

- Processador 1: Custo de \$5/minuto, capacidade de processar 10 tarefas/minuto.
- Processador 2: Custo de \$7/minuto, capacidade de processar 15 tarefas/minuto.
- Processador 3: Custo de \$4/minuto, capacidade de processar 8 tarefas/minuto.

Um conjunto de 500 tarefas precisa ser processado em um período de 60 minutos. Devido a licenças de software, o Processador 1 não pode ser utilizado por mais de 40 minutos, e o Processador 2 deve ser utilizado por pelo menos 30 minutos.

Objetivo: Minimizar o custo total de utilização dos processadores, garantindo que todas as tarefas sejam processadas dentro do tempo limite e respeitando as restrições de uso de cada processador.

---

## Problema 3: Otimização de Largura de Banda para Transmissão de Dados

Uma rede de comunicação tem capacidade total de 100 Mbps. Três tipos de dados precisam ser transmitidos:

- Dados de Prioridade Alta (A): Requerem 2 Mbps por unidade de dado, geram \$10 de receita por unidade.
- Dados de Prioridade Média (B): Requerem 1.5 Mbps por unidade de dado, geram \$7 de receita por unidade.
- Dados de Prioridade Baixa (C): Requerem 1 Mbps por unidade de dado, geram \$4 de receita por unidade.

A gerência da rede estabeleceu que a quantidade de dados de prioridade alta deve ser pelo menos 20 unidades, e a quantidade total de dados de prioridade média e baixa não pode exceder 50 unidades. Além disso, a quantidade de dados de prioridade alta não pode ser mais do que o dobro da quantidade de dados de prioridade média.

Objetivo: Maximizar a receita total gerada pela transmissão dos dados, respeitando a capacidade da rede e as prioridades estabelecidas.

---

#### Problema 4: Escalonamento de Jobs em um Sistema Batch

Um sistema batch precisa executar três tipos de jobs: Job X, Job Y e Job Z. Cada job requer diferentes quantidades de tempo de CPU e memória:

- Job X: 5 minutos de CPU, 2 GB de memória, gera 8 pontos de "valor" para o sistema.
- Job Y: 3 minutos de CPU, 3 GB de memória, gera 6 pontos de "valor".
- Job Z: 6 minutos de CPU, 1 GB de memória, gera 10 pontos de "valor".

O sistema tem disponível 120 minutos de tempo de CPU e 60 GB de memória. A gerência determinou que pelo menos 5 Jobs X e 3 Jobs Y devem ser executados. O número total de Jobs Z executados não deve exceder a soma do número de Jobs X e Y.

Objetivo: Maximizar o "valor" total dos jobs executados, respeitando as restrições de recursos e as políticas de execução.

---

#### Problema 5: Configuração de Rede Wi-Fi em um Campus

Um campus universitário precisa instalar pontos de acesso Wi-Fi para cobrir três áreas: A, B e C. Existem dois modelos de roteadores disponíveis: Modelo Alpha e Modelo Beta.

- Modelo Alpha: Custa \$200, cobre 50 m<sup>2</sup> na Área A, 30 m<sup>2</sup> na Área B, e 20 m<sup>2</sup> na Área C.
- Modelo Beta: Custa \$300, cobre 40 m<sup>2</sup> na Área A, 60 m<sup>2</sup> na Área B, e 30 m<sup>2</sup> na Área C.

As áreas mínimas a serem cobertas são: Área A (200 m<sup>2</sup>), Área B (180 m<sup>2</sup>), Área C (150 m<sup>2</sup>). O orçamento total para roteadores é de \$3.000. O número de roteadores Modelo Alpha não pode ser mais do que o dobro do número de roteadores Modelo Beta.

Objetivo: Minimizar o custo total de aquisição dos roteadores, garantindo a cobertura mínima em todas as áreas e respeitando o orçamento.

---

## Problema 6: Otimização de Armazenamento de Dados em Nuvem

Uma empresa precisa armazenar três tipos de dados em um serviço de nuvem: dados de clientes, dados de projetos e backups. O serviço de nuvem oferece dois planos de armazenamento: Padrão e Premium.

- Plano Padrão: Custa \$0.05/GB/mês, oferece 100 IOPS (operações de entrada/saída por segundo) por GB.
- Plano Premium: Custa \$0.10/GB/mês, oferece 200 IOPS por GB.

A empresa precisa armazenar um total de 500 GB de dados. Os dados de clientes exigem pelo menos 150 GB de armazenamento, e os dados de projetos exigem pelo menos 100 GB. A performance total (IOPS) deve ser de no mínimo 80.000 IOPS.

Objetivo: Minimizar o custo mensal de armazenamento, atendendo às necessidades de capacidade e performance.

---

## Problema 7: Design de Arquitetura de Microserviços

Uma equipe de desenvolvimento está projetando uma nova arquitetura de microserviços e precisa decidir quantos microserviços de cada tipo (Tipo 1 e Tipo 2) serão implantados.

- Microserviço Tipo 1: Requer 2 unidades de CPU, 4 GB de RAM, e tem um custo de desenvolvimento de \$500. Aumenta a performance em 10%.
- Microserviço Tipo 2: Requer 3 unidades de CPU, 2 GB de RAM, e tem um custo de desenvolvimento de \$700. Aumenta a performance em 15%.

A equipe tem um limite de 30 unidades de CPU e 40 GB de RAM disponíveis para a nova arquitetura. O custo total de desenvolvimento não pode exceder \$5.000. A equipe deseja que a performance total seja de pelo menos 100%.

Objetivo: Maximizar o aumento total de performance do sistema, respeitando os recursos de hardware e o orçamento de desenvolvimento.

---

## Problema 8: Otimização de Publicidade Online (PPC)

Uma agência de marketing digital está gerenciando uma campanha de Pay-Per-Click (PPC) e precisa alocar seu orçamento entre três plataformas: Google Ads, Facebook Ads e LinkedIn Ads.

- Google Ads: Custo de \$0.50/clique, gera 0.1 conversões/clique.

- Facebook Ads: Custo de \$0.30/clique, gera 0.05 conversões/clique.
- LinkedIn Ads: Custo de \$1.00/clique, gera 0.2 conversões/clique.

O orçamento total é de \$1.000. A agência deseja gerar pelo menos 50 conversões. O número de cliques no Google Ads deve ser no mínimo o dobro do número de cliques no Facebook Ads. O número de cliques no LinkedIn Ads não pode exceder 500.

Objetivo: Maximizar o número total de cliques obtidos, respeitando o orçamento e as metas de conversão.

---

### Problema 9: Balanceamento de Carga em Cluster de Containers

Um cluster Kubernetes possui dois tipos de nós (nodes) para executar containers: Nós de Alta Memória e Nós de Alta CPU.

- Nó de Alta Memória: Pode hospedar 10 containers, tem 32 GB de RAM e 4 CPUs.
- Nó de Alta CPU: Pode hospedar 8 containers, tem 16 GB de RAM e 8 CPUs.

O cluster precisa hospedar um total de pelo menos 100 containers. A demanda total por RAM é de 400 GB, e a demanda total por CPU é de 200 CPUs. O custo de manutenção diária de um Nó de Alta Memória é \$20, e de um Nó de Alta CPU é \$25.

Objetivo: Minimizar o custo total diário de manutenção dos nós, garantindo a capacidade de hospedagem de containers e os recursos de RAM e CPU.

---

### Problema 10: Otimização de Backups de Dados

Uma empresa precisa realizar backups de seus dados em dois locais: Servidor Local e Nuvem. Três tipos de dados (Críticos, Importantes, Auxiliares) precisam ser salvos.

- Servidor Local: Custo de armazenamento \$0.01/GB, velocidade de backup 100 MB/s.
- Nuvem: Custo de armazenamento \$0.05/GB, velocidade de backup 50 MB/s.

A empresa tem um orçamento máximo de \$200 para armazenamento. Os dados críticos totalizam 500 GB e devem ter uma cópia local. Os dados importantes totalizam 300 GB e podem ser armazenados em qualquer local. Os dados auxiliares totalizam 200 GB e também podem ser armazenados em qualquer local. A velocidade total de backup deve ser de pelo menos 1000 MB/s.

Objetivo: Minimizar o custo total de armazenamento dos backups, garantindo as capacidades e velocidades mínimas de backup para cada tipo de dado.

---