Projeto de Otimização de Planejamento e Controle de Produção

Aluno: Daniel Amorim Monteiro Barbosa

RA: 12319235

Professor: Alex Montanha

Data: 06/06/2024

Universidade - UNI BH - BURITIS

Curso - CIÊNCIAS DA COMPUTAÇÃO

**Resumo**

O projeto visa criar um sistema para melhorar o planejamento e controle de produção em uma fábrica, buscando maximizar a eficiência, reduzir o tempo de espera por matérias-primas e evitar desperdícios. O problema é desafiador e classificado como complexo, exigindo a avaliação de muitas opções para encontrar a melhor solução. Entre os métodos possíveis, optou-se pela Programação Linear devido à sua capacidade de resolver eficientemente problemas de otimização. A escolha da linguagem Rust para implementação foi motivada por sua segurança, desempenho e expressividade. Essas decisões visam criar um sistema robusto e eficaz para lidar com os desafios do planejamento e controle de produção.

Parte superior do formulário

Parte inferior do formulário

**Apresentação do Problema**

O planejamento e controle de produção são aspectos fundamentais para a eficiência e a competitividade de uma fábrica. Neste contexto, o problema abordado neste projeto é o desenvolvimento de um sistema de otimização para o planejamento e controle de produção em uma fábrica que produz diversos produtos, cada um composto por diferentes subconjuntos de matérias-primas. O objetivo é maximizar a eficiência na produção, minimizando o tempo de espera das matérias-primas em estoque, evitando desperdícios e atendendo às demandas dos clientes de forma oportuna.

**Complexidade do Problema**

Dada a natureza combinatória e as restrições envolvidas no planejamento e controle de produção, é razoável classificar esse problema como NP-Completo. Isso ocorre porque, embora seja fácil verificar uma solução proposta (se ela atende a todas as restrições), encontrar a solução ótima pode exigir a avaliação de todas as possíveis combinações de produção, o que leva a um crescimento exponencial do tempo de execução à medida que o número de produtos e matérias-primas aumenta.

As restrições de capacidade de produção, disponibilidade e tempo de entrega das matérias-primas, limitações de armazenamento, qualidade e custo das matérias-primas, bem como a conformidade com regulamentações ambientais, todas contribuem para a complexidade do problema.

Algoritmos comumente utilizados para problemas de otimização de planejamento e controle de produção:

1. **Algoritmos Genéticos (AG)**:
   * Os AGs são algoritmos de otimização baseados na teoria da evolução biológica. Eles operam em uma população de soluções candidatas, aplicando operadores genéticos como seleção, crossover e mutação para evoluir as soluções ao longo de gerações. Os AGs são adequados para problemas de otimização combinatória, como o planejamento de produção.
2. **Algoritmos de Colônia de Formigas (ACO)**:
   * Os ACOs são inspirados no comportamento de formigas em busca de comida. Eles são usados para resolver problemas de otimização combinatória, onde as formigas constroem soluções candidatas iterativamente, deixando feromônios nas trilhas que levam às boas soluções. Os ACOs são aplicáveis a problemas de roteamento e escalonamento, que são comuns no planejamento de produção.
3. **Programação Linear e Inteira (PLI)**:
   * A PLI é uma técnica de otimização que lida com problemas de maximização ou minimização de uma função linear sujeita a restrições lineares. No contexto do planejamento de produção, a PLI pode ser usada para modelar e resolver problemas de alocação de recursos, como a atribuição de matérias-primas a diferentes produtos para maximizar o lucro ou minimizar os custos.
4. **Métodos de Programação Dinâmica**:
   * A programação dinâmica é uma técnica de otimização que quebra um problema em subproblemas menores e resolve cada subproblema apenas uma vez, armazenando suas soluções para evitar recalculá-las. É útil para problemas onde as decisões são tomadas em estágios sequenciais, como o planejamento de produção ao longo do tempo.

### Algoritmo Escolhido: Programação Linear (PL)

A programação linear é uma técnica que lida com problemas de maximização ou minimização de uma função linear sujeita a restrições lineares. No contexto do planejamento de produção, podemos formular o problema como um modelo de programação linear, onde as variáveis de decisão representam as quantidades a produzir de cada produto, e as restrições representam as capacidades de produção, disponibilidade de matérias-primas, datas de entrega dos pedidos, entre outros.

### Complexidade do Algoritmo Escolhido

A complexidade computacional da programação linear pode variar dependendo do tamanho e da estrutura do problema. Em geral, a resolução de problemas de programação linear é eficiente e pode ser realizada em tempo polinomial.

A complexidade computacional do algoritmo de programação linear depende principalmente do método utilizado para resolver o problema. Métodos diretos, como o Método Simplex, têm uma complexidade média de O(m^2 \* n), onde m é o número de restrições e n é o número de variáveis de decisão. No entanto, em casos de problemas grandes, o Método Simplex pode se tornar ineficiente.

Para problemas com um grande número de variáveis e restrições, algoritmos de otimização mais avançados, como os métodos de pontos interiores, podem ser mais eficientes. Esses métodos podem ter uma complexidade média de O(n^3) ou melhor em alguns casos.

**Trecho de Maior Complexidade**

O trecho de maior complexidade do algoritmo de programação linear é a fase de resolução do problema, que envolve a aplicação iterativa de operações para encontrar a solução ótima. A complexidade exata dessa fase pode variar dependendo do método utilizado e das características específicas do problema.

### Paradigma e Estratégia do Algoritmo Escolhido

A programação linear segue o paradigma de otimização, onde o objetivo é encontrar a melhor solução possível para um problema de acordo com critérios definidos. A estratégia do algoritmo de programação linear é explorar de forma sistemática o espaço de soluções viáveis para encontrar aquela que otimiza a função objetivo, respeitando todas as restrições do problema.

### Comparação entre Métodos de Programação Dinâmica e Programação Linear

**Complexidade:**

* **Métodos de Programação Dinâmica (MPD)**:
  + A complexidade dos MPD pode variar dependendo da estrutura do problema e da abordagem específica utilizada. Em geral, os MPD buscam reduzir a complexidade de um problema que pode ser dividido em subproblemas sobrepostos, evitando recalculos desnecessários através do armazenamento de soluções intermediárias. A complexidade típica dos MPD é polinomial, com uma complexidade de tempo que varia de O(n^2) a O(n^3) ou mais, onde n é o tamanho do problema.
* **Programação Linear (PL)**:
  + A complexidade da PL depende principalmente do método utilizado para resolver o problema. Métodos diretos, como o Método Simplex, têm uma complexidade média de O(m^2 \* n), onde m é o número de restrições e n é o número de variáveis de decisão. Em casos de problemas grandes, o Método Simplex pode se tornar ineficiente. Métodos mais avançados, como os métodos de pontos interiores, podem ter uma complexidade média de O(n^3) ou melhor em alguns casos.

**Paradigma e Estratégia:**

* **Métodos de Programação Dinâmica (MPD)**:
  + Os MPD seguem o paradigma de divisão e conquista, onde um problema é dividido em subproblemas menores que podem ser resolvidos independentemente. A estratégia principal é resolver cada subproblema apenas uma vez e armazenar suas soluções para evitar recalculos.
* **Programação Linear (PL)**:
  + A PL segue o paradigma de otimização, onde o objetivo é encontrar a melhor solução possível para um problema de acordo com critérios definidos. A estratégia da PL é explorar de forma sistemática o espaço de soluções viáveis para encontrar aquela que otimiza a função objetivo, respeitando todas as restrições do problema.

### Linguagem

O Rust foi escolhido como linguagem de implementação para o algoritmo de otimização de planejamento e controle de produção por várias razões:

1. **Segurança e Confiabilidade**: Rust é conhecido por sua forte ênfase em segurança e prevenção de erros, graças ao seu sistema de tipos robusto e ao controle de memória sem garbage collector. Isso é crucial em aplicações críticas, como sistemas de controle de produção, onde erros podem ter consequências significativas.
2. **Performance**: Rust oferece um desempenho comparável ao de linguagens de baixo nível, como C e C++, o que é essencial para lidar com grandes conjuntos de dados e cálculos intensivos, comuns em problemas de otimização.
3. **Expressividade e Concorrência**: Rust possui um sistema de tipos poderoso e expressivo, o que facilita a expressão de ideias de forma clara e concisa. Além disso, sua capacidade de lidar de forma eficiente com a concorrência por meio do modelo de propriedade de mutabilidade contribui para a implementação de algoritmos eficientes em paralelo.
4. **Ecossistema e Ferramentas**: Rust possui uma comunidade ativa e um ecossistema em crescimento, com bibliotecas e ferramentas úteis para uma ampla gama de aplicações. Isso facilita o desenvolvimento e a manutenção de projetos em Rust.

Para a implementação do algoritmo de otimização de planejamento e controle de produção em Rust, aproveitaremos essas características da linguagem