**Descrição Completa do Problema**

**Contexto**

O problema envolve a otimização do processo de produção em uma fábrica que possui múltiplas linhas de produção, onde diferentes produtos são fabricados. Cada produto tem um tempo específico de produção, e o tempo para trocar de um produto para outro em uma mesma linha (tempo de preparo) também deve ser considerado. A meta é distribuir os produtos pelas linhas de produção de forma que o tempo total de produção seja minimizado.

**Estrutura do Problema**

1. **Entrada de Dados**:
   * **Número de Linhas de Produção (numLinhas)**: Quantidade de linhas disponíveis para produzir os produtos.
   * **Número de Produtos (numProdutos)**: Quantidade de produtos diferentes a serem fabricados.
   * **Tempos de Produção (Tempos)**: Um vetor onde cada elemento representa o tempo necessário para fabricar um determinado produto.
   * **Tempos de Preparação (Preparos)**: Uma matriz onde o elemento na posição [i][j] representa o tempo necessário para ajustar a linha de produção quando se troca a produção do produto i para o produto j.
2. **Objetivo**:
   * Minimizar o tempo total de produção, considerando tanto os tempos de produção quanto os tempos de preparação, ao distribuir os produtos entre as linhas de produção.

**Algoritmo Inicial**

O problema foi inicialmente abordado utilizando um algoritmo guloso, que aloca os produtos às linhas de produção de maneira a minimizar o tempo de produção localmente, ou seja, a cada passo, o produto é alocado na linha que, naquele momento, oferece o menor tempo total de produção.

* **Passos do Algoritmo Guloso**:
  1. Inicializar as linhas de produção.
  2. Para cada produto, determinar a linha onde a adição desse produto resultaria no menor incremento de tempo de produção.
  3. Alocar o produto a essa linha e atualizar o tempo total.

**Melhorias com Heurísticas de Otimização**

Após a execução do algoritmo guloso, foram aplicadas heurísticas adicionais para melhorar a solução encontrada:

1. **Swap Inter-Linhas (SwapInter)**:
   * Esta operação tenta melhorar a solução trocando produtos entre diferentes linhas de produção.
   * Para cada par de linhas e para cada par de produtos dessas linhas, verifica-se se a troca entre esses produtos resulta em uma redução do tempo total de produção.
2. **Swap Intra-Linhas (SwapIntra)**:
   * Esta operação tenta melhorar a solução trocando a ordem dos produtos dentro da mesma linha de produção.
   * Para cada linha e para cada par de produtos adjacentes, verifica-se se a troca desses produtos reduz o tempo total de produção.
3. **VND (Variable Neighborhood Descent)**:
   * Este método aplica uma sequência de operações de SwapInter e SwapIntra, explorando diferentes vizinhanças da solução atual para encontrar melhorias.
   * O processo continua até que nenhuma melhoria adicional seja encontrada.
4. **ILS**
   * O ILS consiste em realizar perturbações em uma solução e melhorar essa solução por uma busca local (que pode ser o VND).
   * A cada iteração, aplica-se uma perturbação na solução corrente e tenta-se melhorá-la, aceitando a nova solução se ela for melhor ou aplicando critérios de aceitação.

**Resultados**

* O algoritmo inicialmente calcula uma solução utilizando o método guloso, e então aplica as heurísticas de otimização para melhorar essa solução.
* O tempo total de produção, tanto antes quanto depois da aplicação das heurísticas, é exibido, juntamente com o tempo de execução do algoritmo.