

Relatório Final de Pesquisa Operacional: Resolução do Problema de Blocking Flowshop

Gabriel Davi / Giordani Andre

Agosto de 2025

1 Links

1. [Clique aqui para acessar o Código](#)
2. [Clique aqui para acessar o Slide](#)
3. [Clique aqui para acessar o Artigo](#)

2 Introdução ao artigo

O artigo aborda o Problema de Sequenciamento em Flowshop com Bloqueio (do inglês, Blocking Flowshop Scheduling Problem - BFSP), um problema de otimização combinatória NP-completo de grande relevância em sistemas de produção onde não existem pulmões (buffers) intermediários entre as máquinas. Neste cenário, uma tarefa, ao finalizar seu processamento em uma máquina, a bloqueia até que a máquina subsequente esteja livre. O objetivo do trabalho é minimizar o tempo de fluxo total (do inglês, Total Flowtime), que corresponde à soma dos tempos de completção de todas as tarefas na última máquina.

3 Revisão da Literatura

O artigo de referência, "Minimização do Tempo Total de Fluxo no Problema de Blocking Flowshop com Uso de GRASP Reativo" de Gelli et al., apresenta uma abordagem de meta-heurística para resolver o problema BFSP com o objetivo de minimizar o tempo de fluxo total. Sendo um problema de classificação NP-difícil, o uso de heurísticas é justificado para encontrar boas soluções em tempo computacional viável.

A abordagem proposta pelos autores é um GRASP (Greedy Randomized Adaptive Search Procedure) Reativo. A principal característica desta variação do GRASP é a sua capacidade de autoajuste do parâmetro α , que controla o nível de aleatoriedade na construção da solução, adaptando-se ao longo da execução com base na qualidade das soluções geradas.

O procedimento é composto por duas fases principais:

- **Fase Construtiva:** Utiliza uma versão aleatorizada da heurística FF, proposta originalmente por Fernandez-Viagas e Framinan.
- **Busca Local:** Após a construção, uma solução é refinada através de buscas locais do tipo *Interchange* (troca de duas tarefas) e *Insertion* (re-alocação de uma tarefa).

Os autores validaram seu método utilizando as instâncias de benchmark de Taillard e compararam seus resultados com os do estado da arte, obtidos por Tasgetiren et al., demonstrando a alta performance de sua abordagem.

4 Desenvolvimento da Solução

Para a resolução do problema, foi desenvolvido um conjunto de algoritmos em Python, utilizando as bibliotecas ‘numpy’ para análise estatística e ‘tabulate’ para a exibição de resultados. A estratégia adotada seguiu uma abordagem em duas fases: uma fase construtiva para gerar uma solução inicial de qualidade, seguida por uma fase de refinamento utilizando duas técnicas de meta-heurística distintas.

4.1 Heurística Construtiva: NEH

Como ponto de partida, foi implementada a clássica e eficiente heurística de Nawaz, Ensore e Ham (NEH). O funcionamento da função `heuristica_neh` consiste em:

1. Calcular o tempo total de processamento para cada tarefa em todas as máquinas.
2. Ordenar as tarefas em ordem decrescente com base nessa soma.
3. Inserir iterativamente cada tarefa da lista ordenada na sequência parcial, testando todas as posições possíveis e escolhendo aquela que resulta no menor tempo de fluxo total (calculado pela função `calcular_total_flowtime_BLOCKING`).

Esta heurística foi escolhida por sua simplicidade de implementação e sua conhecida eficácia em gerar boas soluções iniciais para problemas de flowshop.

4.2 Fase de Refinamento: Meta-heurísticas

A solução gerada pela NEH foi então utilizada como entrada para duas abordagens de refinamento distintas, permitindo uma comparação de eficácia.

4.2.1 Busca Local com Interchange

A função `busca_local_interchange` implementa uma busca local iterativa. Partindo da solução NEH, ela explora sistematicamente a vizinhança de troca (swap) de todas as tarefas. A cada iteração, ela avalia todas as trocas possíveis

e aplica a que gera a maior melhoria (estratégia *Best Improvement*). O processo se repete até que nenhuma troca possa melhorar a solução atual, atingindo assim um ótimo local.

4.2.2 Simulated Annealing (SA)

A função `simulated_annealing` implementa a meta-heurística de Recozimento Simulado. Assim como a busca local, ela utiliza a vizinhança de *Interchange* para gerar novas soluções. No entanto, sua principal característica é a capacidade de aceitar soluções piores com uma certa probabilidade, que diminui conforme a "temperatura" do sistema baixa. Este mecanismo permite que o algoritmo escape de ótimos locais e explore uma área mais ampla do espaço de soluções. Os parâmetros de resfriamento (temperatura inicial, final e fator alfa) foram definidos com valores padrão da literatura para garantir um número consistente de iterações.

5 Resultados Encontrados

Para avaliar o desempenho dos algoritmos implementados, os resultados foram comparados com os valores de BKS (Best Known Solutions) da literatura através do Desvio Percentual Relativo (RPD), cuja fórmula é dada por:

$$RPD = \frac{(CustoObtido - CustoBKS)}{CustoBKS} \times 100\%$$

Os testes foram executados com um tempo limite de 480 segundos (8 minutos) por instância, o mesmo utilizado no artigo de referência, para garantir uma comparação justa. A seguir, apresentamos um resumo dos resultados para o grupo de instâncias 50x5.

Table 1: Resumo dos resultados para o grupo 50x20

Métrica	NEH + Interchange	NEH + SA
RPD (%)	27.40	26.20
Iterações (Média)	14	1919

5.1 Análise Comparativa

A partir dos resultados, observamos que:

- **NEH + Interchange:** Este método continua se mostrando extremamente rápido, encontrando um ótimo local com um número muito baixo de iterações (média de 14). Ele consistentemente melhora a solução da heurística NEH, porém seu RPD médio de 27,40, embora seja uma melhora substancial em relação a testes anteriores, ainda é considerado alto. Isso indica que a busca local ainda converge para ótimos locais de qualidade intermediária, distantes das melhores soluções conhecidas (BKS).

- **NEH + Simulated Annealing:** Este método se consolidou como a abordagem mais eficaz, superando a busca local. Atingiu um RPD médio de 26,20, o que demonstra a capacidade do Simulated Annealing de escapar de ótimos locais e explorar o espaço de busca de forma mais abrangente. O número de iterações foi determinado pelo cronograma de resfriamento do algoritmo, executando uma média de 1919 movimentos por instância.
- **Comparação com o Artigo:** Apesar da melhora significativa nos resultados, ambos os métodos implementados ainda apresentam um desempenho inferior ao do GRASP Reativo do artigo de referência (que obteve um RPD próximo de 0). A principal razão para essa diferença permanece a mesma: nossa abordagem é single-start, refinando uma única solução inicial. Em contraste, o GRASP é multi-start, gerando e aprimorando milhares de soluções diferentes dentro do tempo limite de 8 minutos, o que aumenta drasticamente a probabilidade de encontrar soluções de elite.

6 Conclusão

O objetivo deste trabalho foi implementar e avaliar o desempenho de heurísticas para a resolução do Problema de Sequenciamento em Flowshop com Bloqueio (BFSP), visando minimizar o tempo total de fluxo. Para isso, foram implementadas com sucesso a heurística construtiva NEH e duas estratégias de refinamento: uma Busca Local baseada em Interchange e a meta-heurística Simulated Annealing.

Os resultados obtidos, quando comparados com o estado da arte, demonstraram que, embora os algoritmos funcionem corretamente e melhorem a solução inicial, eles não são suficientes para atingir a qualidade das soluções encontradas por meta-heurísticas mais complexas, como o GRASP Reativo do artigo de referência.

A análise confirma a complexidade do problema e evidencia o clássico trade-off entre tempo de execução e qualidade da solução em problemas NP-difíceis. A superioridade do GRASP se deve à sua estratégia multi-start e ao uso intensivo de um grande orçamento de tempo (8 minutos por instância), permitindo uma exploração muito mais ampla do espaço de soluções. O trabalho prático validou que, para problemas desta natureza, meta-heurísticas mais robustas e com maior investimento de tempo são essenciais para se aproximar dos melhores resultados conhecidos na literatura.