

PROGRAMAÇÃO E SEQUENCIAMENTO EM SISTEMA *FLOW SHOP* COM TEMPOS DE *SETUP* INDEPENDENTE DA SEQUÊNCIA

Clarissa Tararam de Laurentys

Orientador: Prof. Dr. Marcelo Seido Nagano

Universidade de São Paulo

clarissa.laurentys@usp.br

Objetivos

O problema de sequenciamento da produção ou *scheduling* em ambientes *flow shop* tem sido estudado desde a década de 50. Dada sua importância em ambientes práticos, suas variações e particularidades, têm sido foco de estudos nas últimas décadas. O presente projeto tem como propósito o desenvolvimento de métodos de solução para o problema *flow shop*, considerando que as fábricas possuem tempos de setup independentes da sequência de produção. O objetivo de desempenho adotado é a duração total da programação, chamado de *makespan*. Uma revisão em torno do problema *flow shop* foi realizada de modo a identificar métodos que possam ser adaptados a fim de inserir a análise dos tempos de setup independentes. Os algoritmos foram implementados e experimentações computacionais foram conduzidas. O desempenho desses algoritmos foi avaliado de acordo com sua eficiência e eficácia, buscando-se os melhores métodos para o problema estudado.

Métodos e Procedimentos

A implementação foi efetuada em Python em um PC com CPU Intel Core i5-5200U 2.20 GHz, 8 GB de RAM e operando sob o sistema operacional Windows 10. A base de dados proposta por Rubén Ruiz e Ali Allahverdi (2007) foi adotada para a realização dos experimentos computacionais.

O algoritmo em análise é o NEH, para o qual foram propostas 48 heurísticas a partir da combinação de 12 regras de prioridade, utilizadas na primeira etapa do algoritmo, com 4 estratégias de desempate, utilizadas na quarta etapa.

Essas combinações foram executadas para a base de dados e comparadas entre si. As performances foram avaliadas pelo desvio percentual relativo (Relative Percentage Deviation – RPD). O tempo médio de CPU (The Average CPU Time – ACT) e o Tempo computacional percentual relativo médio (Average Relative Percentage computational Time – ARPT) foram utilizados para avaliar a eficiência computacional das heurísticas.

Resultados

Observou-se que, em todos os casos, a média dos ARPDs obtidos pelas heurísticas compostas pela estratégia de desempate NEH foram as piores. Já as melhores médias variaram, majoritariamente, entre as heurísticas compostas pelas estratégias de desempate FF e RTC, sendo que as diferenças entre estas médias é da ordem da 2ª ou 3ª casa decimal. Além disso, em qualquer caso analisado, a ordem crescente entre os tempos de CPU e de valores ARPT das estratégias de desempate que compuseram as heurísticas foi: NEH, FF, RTC e D.

Dessa forma, entende-se que o impacto nos resultados obtidos é resultante, primordialmente, da escolha da estratégia de desempate. A Figuras 1 torna os resultados

obtidos mais visíveis, sendo evidente a divisão das heurísticas em quatro zonas, cada um representando uma das estratégias de desempate.

A primeira zona, composta pela estratégia NEH, apresenta os maiores valores de ARPT e os menores valores de ARPD; a segunda, composta pela estratégia FF, apresenta os menores valores de ARPT e valores baixos de ARPD, semelhantes aos encontrados na terceira zona, composta pela estratégia RTC, que apresenta valores medianos de ARPT; e, por fim, a quarta zona, composta pela estratégia D, apresenta os maiores valores de ARPT e valores medianos de ARPD.

Porém, não foi possível identificar nenhum padrão entre a utilização da regra e o desempenho da heurística, pois uma mesma regra apresenta impactos diferentes quando combinada com diferentes estratégias de desempate.

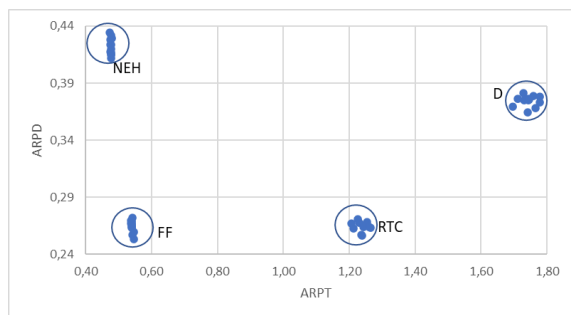


Figura 1: Valores de ARPD e ARPT em relação ao número de atividades para as heurísticas comparadas

Conclusões

A avaliação das 48 heurísticas resultou em quatro grandes conclusões: i) o uso de diferentes estratégias de desempate apresenta grande impacto nos resultados finais; ii) o impacto da regras de prioridade nos resultados finais variam de acordo com a estratégia de desempate utilizada; iii) todas as estratégias de desempate apresentaram melhores valores de ARPD quando comparadas à estratégia original, NEH; iv) as estratégias FF e RTC apresentaram os melhores valores de ARPD, porém a FF se destacou por apresentar menor tempo de CPU.

Dessa forma, evidencia-se a superioridade das heurísticas que utilizam a estratégia de desempate FF, sendo que, entre elas, a variação no uso das regras de prioridade gera mudanças apenas na terceira casa decimal nos valores de ARPD e ARPT. Outra grande contribuição deste trabalho é a adaptação do algoritmo da aceleração de Taillard de forma a incluir os tempos de setup independentes à sequência de atividades.

Referências Bibliográficas

- DONG, X.; HUANG, H.; CHEN, P. An improved NEH-based heuristic for the permutation flowshop problem. *Computers & Operations Research*, v. 35, p. 3962–3968, 2008.
- FERNANDEZ-VIAGAS, V.; FRAMINAN, J. M. A best-of-breed iterated greedy for the permutation flowshop scheduling problem with makespan objective. *Computers and Operations Research*, v. 112, 2019.
- KALCZYNSKI, P. J.; KAMBUROWSKI, J. An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers & Operations Research*, v. 35, p. 3001– 3008, 2008.
- RIBAS, I.; COMPANYS, R.; TORT-MARTORELL, X. Comparing three-step heuristics for the permutation flow shop problem. *Computers & Operations Research*, v. 37, p. 2062–2070, 2010.
- RÍOS-MERCADO, R. Z.; BARD, J. F. Heuristics for the flow line problem with setup costs. *European Journal of Operational Research*, v. 110, p. 76-98, 1998.
- RUIZ, R.; ALLAHVERDI, A. Some effective heuristics for no-wait flowshops with setup times to minimize total completion time. *Springer Science+Business Media*, p. 143–171, 2007.