

PROBLEMA DE SEQUENCIAMENTO DE MÁQUINAS ABORDAGEM DE DATA DE ENTREGA COMUM

Bruno Alberto Soares Oliveira¹; Diego Rodrigues Andrade Simão²;

1 Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Minas Gerais, Belo Horizonte – MG

2 Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Minas Gerais, Belo Horizonte – MG

RESUMO

Este trabalho tem como objetivo a resolução de um problema de sequenciamento de máquinas com data de entrega em comum com penalidade por antecipação e atraso da produção. São consideradas duas instâncias com 100 e 200 tarefas e suas respectivas penalidades. Para resolução deste problema foi construído um algoritmo utilizando como solução inicial um critério guloso proposto e aprimorando-se a solução por meio das técnicas de Meta-Heurística Busca em Vizinhança Variável (VNS) e Descida em Vizinhança Variável (VND), auxiliadas por uma tabela hash para eliminar repetições de tentativas de soluções. Os resultados encontrados foram satisfatórios e melhores do que referência imposta, comprovando a eficiência das técnicas utilizadas

INTRODUÇÃO:

A busca constante por maior eficiência e aspectos como qualidade e prazo de entrega requeridos pelo mercado faz com que empresas se destaquem diante de tamanha competitividade segundo [1]. Tendo em vista essa necessidade, foi criada a filosofia *Just-in-Time* (JIT), baseada em produzir apenas o volume indispensável no momento ideal [2]. Nesse contexto é de extrema relevância eliminar estoques intermediários (produtos e subprodutos adiantados ou atrasados) para que o custo de estoque seja reduzido e a rotatividade do capital de giro aumentada. Caso a metodologia seja implantada e funcione de forma correta, setores como galpões, almoxarifados e salas de estocagem serão desnecessárias de acordo com [3]. Para que essa metodologia seja eficiente em um ambiente com inúmeras máquinas e tarefas a serem processadas em um prazo de entrega em comum, surge o problema de definição ótima da ordem de execução com penalidades para atrasos e adiantamentos como estudado por [4]. Partindo desse princípio o objetivo deste trabalho é o de estudar o problema de programação de várias tarefas com data comum de entrega num ambiente com uma máquina. Trata-se de um problema combinatório de difícil resolução na otimalidade. Procura-se definir a atribuição de um conjunto de n tarefas independentes a uma única máquina, bem como a sequência de processamento das tarefas dado o instante de início de cada tarefa, visando a minimização da soma de custos de penalidades de atraso e adiantamento em relação à data de entrega.

Na literatura há três variações desse problema quanto a entrega: janelas de entrega distintas (*distinct due windows*), data de entrega comum (*common due date*) e datas de entrega distintas (*distinct due date*). No primeiro tipo, há uma flexibilidade maior no prazo de entrega, adotando-se janelas de tempo sem onerar custo caso a tarefa tenha sido entregue dentro do intervalo de tempo pré-estabelecido. No segundo, as tarefas devem ser entregues na mesma data. No último, há datas de entrega distintas para cada tarefa a ser executada.

Além de relevância prática, esse problema é considerado em sua essência como NP-difícil [5][6][7][8] e grande parte das abordagens encontradas utilizam de estratégias com heurísticas ou meta-heurísticas. Métodos exatos são encontrados em menor número por resolverem problemas desta natureza em dimensão reduzida, não conseguindo abstrair toda complexidade prática encontrada nas indústrias e/ou em outros ambientes. Um exemplo de resolução exata foi proposto por [9] ao utilizar o algoritmo de *Branch-and-Bound*. Os métodos heurísticos apesar de serem considerados como métodos aproximados são capazes de encontrar boas soluções em tempo computacional reduzido, mas sem poder concluir se a solução encontrada é ótima ou não. Mas como em problemas reais o importante é encontrar uma solução factível e relevante e não necessariamente uma solução ótima, métodos heurísticos são ideais para representar problemas de dimensões elevadas e bem próximos da realidade.

METODOLOGIA:

Segundo [10] o problema de sequenciamento em máquina única está sujeito a algumas condições:

1. As n tarefas estão disponíveis de maneira simultânea para serem realizadas no instante $t=0$;
2. Cada máquina só pode realizar uma tarefa de cada vez;
3. Os tempos de preparação das tarefas são independentes da sequência escolhida e estão incluídos nos tempos de processamento;
4. As características das tarefas são determinísticas e, portanto, conhecidas a priori;
5. Não há intervalo entre tarefas;
6. As máquinas sempre estão funcionando, não há tempo para setup ou não funcionamento delas;
7. Uma vez que uma tarefa foi iniciada a mesma será concluída sem interrupção

Abaixo dessas premissas há uma relação de um-a-um entre a permutação de tarefas a serem executadas. O total de combinações possíveis no caso de máquina única é, portanto, $n!$. Pode-se adotar como um exemplo uma instância com 5 tarefas para ser definida a sequência ótima de execução. O total de combinações possíveis para tomar essa decisão é de 120. Já no exemplo proposto no desenvolvimento deste trabalho, que são instâncias de 100 e 200 tarefas os números ascendem a grandezas de ordem de 10^{157} e 10^{374} , respectivamente. Como já citado anteriormente, para tratar situações de dimensões tão elevadas recomenda-se a utilização de abordagens heurísticas, que serão discutidas no próximo tópico.

Modelagem Matemática

Para o problema descrito, foi desenvolvido um modelo de programação linear inteira mista.

São parâmetros do modelo:

- K : conjunto de posições na fila da máquina;
- N : conjunto de tarefas;
- D_D : prazo comum para execução de todas as tarefas;
- p_i : tempo de processamento de cada tarefa do conjunto N ;
- α_i : penalidade por adiantamento de cada tarefa do conjunto N ;
- β_i : penalidade por atraso de cada tarefa do conjunto N ;
- M : somatório do tempo de processamento de todas as tarefas + 1;

São variáveis:

- y_{ik} : variável binária que informa se a tarefa i será executada na ordem k ;
- d_i : variável inteira que informa a data fim da tarefa i ;
- s_k : variável inteira que contabiliza o tempo transcorrido após a execução das tarefas das k primeiras posições;
- e_i : variável inteira que informa em quanto tempo a tarefa i terminou adiantada;
- t_i : variável inteira que informa em quanto tempo a tarefa i terminou atrasada;

A função objetivo, FO, definida para a solução do problema é

$$\min \text{penalidades} = \sum_{i \in N} \alpha_i \cdot e_i + \beta_i \cdot t_i$$

sujeito a:

$$e_i = \max(0, D_D - d_i), \quad \forall i \in N$$

$$t_i = \max(0, d_i - D_D), \quad \forall i \in N$$

$$\sum_{k \in K} y_{ik} = 1, \quad \forall i \in N$$

$$\sum_{i \in N} y_{ik} = 1, \quad \forall k \in K$$

$$s_1 = \sum_{i \in N} y_{i1} \cdot p_i$$

$$s_k = s_{k-1} + \sum_{i \in N} y_{ik} \cdot p_i, \quad \forall k \in K$$

$$d_i = \sum_{k \in K} \min(s_k, M_{y_{ik}}), \quad \forall i \in N$$

A função objetivo visa minimizar as penalidades do conjunto de tarefas executadas, somando os atrasos e adiantamentos conforme a ordem da linha de produção.

A restrição (2) calcula o tempo de adiantamento da tarefa i . Caso ela tenha terminado depois do prazo de entrega, a tarefa não foi adiantada e é atribuído valor zero à variável. Caso contrário, a data fim da tarefa é subtraída do parâmetro prazo comum.

A restrição (3) calcula o tempo de atraso da tarefa i . Caso ela tenha terminado antes do prazo de entrega, a tarefa não sofreu atraso e é atribuído valor zero à variável. Caso contrário, o prazo comum de finalização da tarefa é subtraído da data fim da tarefa.

A restrição (4) garante que cada tarefa entrará na fila uma única vez. Já a restrição (5) assegura que cada posição na fila será ocupada por uma única tarefa. Ou seja, a máquina executará uma única tarefa por vez.

As restrições (6) e (7) contabilizam o tempo total de execução após a máquina ter concluído a tarefa da k -ésima posição da fila. Os tempos são cumulativos (restrição(7)) e apenas a primeira tarefa não acumulará tempos anteriores (restrição(6)).

A restrição (8) define o tempo de conclusão de cada tarefa. Como foi atribuído ao parâmetro M um valor superior ao valor de toda variável s_k , caso a tarefa esteja na k -ésima posição, a data de finalização corresponderá ao tempo registrado em s_k , caso contrário, será somado à data fim apenas valores nulos

RESULTADOS E DISCUSSÕES:

O algoritmo implementado foi executado cinco vezes para cada instância (n=100 e n=200) em um computador *Dell Inspiron i14-5458-B40*, com processador Intel Core i5 5200U de 2.2GHz, com 8GB de memória RAM, sistema operacional Windows. O tempo médio de execução foi de 19s e 122s, respectivamente.

Os resultados para as cinco execuções são apresentados nas Tabelas 1 e 2. Em todas as execuções foram obtidos valores de FO melhores, se comparados com o da solução de referência. Para a instância com n=100, em média, o ganho em valor de FO ficou em torno de 2,5%. Para n=200, o ganho obtido foi em torno de 1% em relação ao valor de referência.

TABELA 1 – VALORES DE FO OBTIDOS PARA CINCO EXECUÇÕES DO ALGORITMO IMPLEMENTADO PARA INSTÂNCIAS COM N=100 E N=200

	Instância n=100	Instância n=200
1	87712	300078
2	86213	298828
3	88575	297679
4	87143	298917
5	87127	298800

TABELA 2 – VALORES DE FO DE REFERÊNCIA E OBTIDOS COM A EXECUÇÃO DO ALGORITMO IMPLEMENTADO PARA INSTÂNCIAS COM N=100 E N=200

	Instância n=100	Instância n=200
FO de referência	89588	301449
Melhor valor de FO	86213	297679
Pior valor de FO	88575	300078
Valor médio de FO	87354	298860

CONCLUSÕES:

Os resultados obtidos por meio do algoritmo implementado comprovaram que as técnicas de meta-heurísticas utilizadas (VNS e VND) possibilitam a melhora de uma solução inicial obtida por um critério guloso, tanto para uma instância com 100 tarefas (melhora média de cerca de 2,5% em relação ao valor de referência) quanto para uma com 200 tarefas (melhora média de cerca de 1% em relação ao valor de referência). Isso mostra que, em um Problema de Sequenciamento de Máquinas com Abordagem de Data de Entrega em Comum, é possível se determinar sequências de execução de tarefas que possibilitem ganhos significativos de redução de custos em um processo industrial.

REFERÊNCIAS BIBLIOGRÁFICAS:

VIII Seminário de Iniciação Científica do IFMG – 12 a 14 de agosto de 2019, Campus Ribeirão das Neves.

- [1] Feldmann, M. & Biskup, D. (2003) Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches. *Computers and Industrial Engineering*, 44, 307-323.
- [2] Kimura, O. & Terada, H. Design and analysis of pull system, a method of multistage production control, *International Journal Production Research*, v.19, n.3, p.241-253, 1981.
- [3] Brown, S.; Lamming, R.; Bessant, J.; Jones, P. *Administração da produção e operações: um enfoque estratégico na manufatura e nos serviços*. Rio de Janeiro: Elsevier, 2005
- [4] Gordon, V., Proth, J.M. & Chu (2002). A survey of the state-of-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139,1-25.
- [5] Du, J.; Leung, J. Y. T. Minimizing total tardiness on one machine is NP-hard. *Mathematics of Operations Research*, v. 15, p. 483-495, 1990. <http://dx.doi.org/10.1287/moor.15.3.483>
- [6] Liaw, C. F. A Branch-and-Bound Algorithm for the Single Machine Earliness and Tardiness Scheduling Problem. *Computers and Operations Research*, v. 26, p. 679-693, 1999. [http://dx.doi.org/10.1016/S0305-0548\(98\)00081-1](http://dx.doi.org/10.1016/S0305-0548(98)00081-1)
- [7] Wan, G.; Yen, B. P. C. Tabu Search for Single Machine Scheduling with Distinct Due Windows and Weighted Earliness/Tardiness Penalties. *European Journal of Operational Research*, v. 142, p. 271-281, 2002. [http:// dx.doi.org/10.1016/S0377-2217\(01\)00302-2](http://dx.doi.org/10.1016/S0377-2217(01)00302-2)
- [8] Hall, N., Kubiak, G.W. Sethi, S.P. (1991). Earliness-tardiness scheduling problems, II: deviations of completion times about a restrictive common due date. *Operations Research*, 39, 847-856.
- [9] Kawamura, Marcio Seiti. Aplicação do método de Branch-and-Bound na programação de tarefas em uma única máquina com data de entrega comum sob penalidade de adiantamento e atraso. 2006
- [10] Baker, Keneth; Trietsch, Dan – *Principles of Sequencing and Scheduling*, New Jersey 2009.