



ISSN: 1984-3151

# MODELAGEM MATEMÁTICA DO PROBLEMA DE SEQUENCIAMENTO DA PRODUÇÃO COM SETUP DEPENDENTE E FILOSOFIA *JUST-IN-TIME*

## MATHEMATICAL MODELING OF THE PRODUCTION SEQUENCING PROBLEM WITH DEPENDENT SETUP TIMES AND PHILOSOPHY *JUST-IN-TIME*

Jerlian Soares Araújo<sup>1</sup>; Hélio Yochihiro Fuchigami<sup>2</sup>

- 1 Bacharel em Matemática Industrial. Universidade Federal de Goiás (UFG), 2013. Pesquisador em Matemática Industrial - UFG. Catalão, GO. [jerlian.araujo@bol.com.br](mailto:jerlian.araujo@bol.com.br)
- 2 Doutor em Engenharia de Produção. USP/EESC (2010). Professor da Universidade Federal de Goiás – UFG. Catalão, GO. [helioufg@gmail.com](mailto:helioufg@gmail.com).

Recebido em: 10/09/2013 - Aprovado em: 29/11/2013 - Disponibilizado em: 30/11/2013

**RESUMO:** Este artigo tem como foco o problema de sequenciamento de tarefas em máquina única com setup dependente da sequência e penalidades por antecipação e atraso da produção. Na notação de três campos para problemas de sequenciamento, este caso é representado por  $1|s_{ij}, d_j|\sum(E_j+T_j)$ , indicando no primeiro campo o ambiente de produção (máquina única), no segundo as restrições de setup dependente ( $s_{ij}$ ) e prazos de entrega ( $d_j$ ) e no terceiro a medida de desempenho de minimização da soma dos adiantamentos e atrasos das tarefas. Para o estudo do problema, são apresentadas algumas técnicas e definições utilizadas por alguns autores para a sua resolução. É pesquisado, então, um modelo de programação linear inteira mista para o problema tratado. Este modelo foi implementado usando-se a ferramenta de modelagem MPL e resolvido pelo software Gurobi 1.0.4. Neste trabalho é dada atenção à necessidade de se dispor do material certo na hora certa, isto é, a filosofia *Just-in-Time* (JIT), de forma a utilizar melhor os recursos. Os experimentos computacionais realizados fornecem a solução ótima do problema em tempo de execução aceitável, indicando que a realização do método é eficiente computacionalmente.

**PALAVRAS-CHAVE:** Sequenciamento da Produção. Programação Linear Inteira Mista. *Just-in-Time*. Tempos de setup.

**ABSTRACT:** This research focuses on the problem of task in Single Machine Scheduling with Earliness and Tardiness Penalties and sequence-dependent setup time with production. In the three-fields notation for scheduling problems, this case is represented by  $1|s_{ij}, d_j|\sum(E_j+T_j)$ , indicating the production environment (single machine) on the first field, the restrictions of sequence-dependent setup times ( $s_{ij}$ ) and due dates ( $d_j$ ) on the second and the performance measure minimization of sum of earliness and tardiness on the third one. For the problem studied there are some techniques and definitions used by others authors for the resolution of similar problems. A mixed integer linear programming model is researched to represent the problem. This model was implemented using the modeling tool MPL and solved by software Gurobi 1.0.4. In this research, attention is drawn to the need to dispose material at the right time, i.e., the philosophy *Just-in-Time* (JIT), in order to better utilize the resources. Computational experiments carried out provide the optimal solution at runtime acceptable, indicating that the method is computationally efficient.

**KEYWORDS:** Sequencing of production. Mixed Integer Linear Programming. *Just-in-Time*. Setup Times.

## 1 INTRODUÇÃO

As indústrias vêm crescendo ao longo dos anos, com isso a disputa entre as empresas, quando se trata de

qualidade dos produtos e serviços, satisfação ao cliente, entre outros, vem aumentando também. Se a empresa visa estar de algum modo entre as melhores do mercado é preciso que tenha um ótimo

planejamento, sendo o objetivo do PCP (Planejamento e Controle da Produção) – setor responsável pela coordenação e aplicação dos recursos produtivos, como esclarece Tubino (2008). Segundo Gaither e Frazier (2005), o PCP e a Pesquisa Operacional são temas muito abordados atualmente. Seus métodos buscam substituir as tomadas de decisões intuitivas que com frequência são adotadas nas empresas para problemas complexos, sendo analisadas criticamente baseando-se em critérios racionais.

Independentemente da indústria, é fundamental o sequenciamento de produção. Obtêm-se ganhos pela redução dos tempos improdutivos, que geram um melhor aproveitamento do sistema, aumenta-se a confiabilidade do processo pela previsibilidade de conclusão das tarefas, além de se reduzirem os custos de produção.

Um problema muito famoso e estudado é o do caixeiro viajante (LAWLER *et al.*, 1985; REINELT, 1994). Antigamente muitos vendedores que distribuíam seus produtos em várias cidades, conhecidos como caixeiros viajantes, saíam de cidade em cidade expondo e vendendo-os, mas a aleatoriedade da rota visitada gerava muitos custos. Para reduzi-los foi preciso que o caixeiro viajante saísse de sua cidade e visitasse exatamente uma única vez cada cidade de uma determinada lista e retornar para a casa de forma que a distância total percorrida fosse a menor possível. Esse problema tem inúmeras aplicações práticas, como minimização de rotas de veículos, confecção de sistemas digitais, sequenciamento de atividades entre outros.

Nos problemas de sequenciamento da produção, no caso em que os tempos de *setup* não são explícitos, o *makespan*, ou seja, a duração total da programação, resulta na soma dos tempos de processamento, não importando a ordem das tarefas. O problema do caixeiro viajante é análogo ao problema de minimização do *makespan*, em que os tempos de

*setup* são dependentes da sequência (denotado por  $1|s_{ij}|C_{max}$ ). Isso ressaltando que *setup* é o tempo decorrido para a troca (ferramenta, programa, equipamento) de um processo em execução até a inicialização do próximo processo.

Pizzolato, Vásquez e D'Ávila (1999) relatam em seu trabalho as opções adotadas referentes aos custos de *setup*, à sequência de produção e à produção de pequenas ordens, as quais exigem pouco tempo produtivo, mas longos tempos de *setup*. Esses autores propõem um método de sequenciamento da produção baseado no problema do caixeiro viajante, seguido de um algoritmo para reduzir as penalidades de antecipação e atraso na produção das várias ordens.

Este artigo investiga um modelo de programação linear inteira mista para o problema denotado por  $(1|s_{ij}, d_j|\sum(E_j+T_j))$ , ou seja, para o sequenciamento da produção em uma máquina com tempos de *setup* dependentes da sequência e visando otimizar a soma dos adiantamentos e atrasos das tarefas. Esse esforço em minimizar tanto adiantamentos como atrasos é parte da filosofia *Just-in-Time*, sobre a qual se discorrerá no próximo capítulo. Esse problema é NP-Difícil, conforme Mendonça (1996), que apresenta em seu trabalho as dificuldades computacionais para o problema.

Dentre os métodos utilizados para a resolução do problema em questão, existem os heurísticos e os exatos. Os métodos heurísticos encontram soluções rápidas em tempo hábil, porém não garantem encontrar a solução ótima. Já os métodos exatos encontram a solução ótima, porém podem gastar tempos com uma complexidade intratável para o seu encontro. Neste trabalho foi utilizado esse último método para a resolução do problema que, dentre outros motivos, foi empregado porque o conflito abordado possui formulação de programação inteira mista.

## 2 FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentados alguns conceitos acerca de problemas que tratam da programação da produção.

### 2.1 PROBLEMAS DE SEQUENCIAMENTO EM UMA MÁQUINA

A programação da produção consiste em alocar uma quantidade de recursos disponíveis (máquinas) que possam processar um conjunto de tarefas e assim obter um produto final. O problema reside na busca pela programação ou sequenciamento que otimize o processo de produção em relação a uma determinada medida de desempenho. A atividade de programação da produção é encontrada em diversos tipos de indústrias, como a de manufatura, metalúrgica, têxtil e de tintas, por exemplo.

No problema de máquina única, considera-se a existência de apenas um recurso para o processamento das tarefas, podendo ser de fato um único equipamento, uma célula de produção ou um conjunto de recursos modelados como apenas um.

Na Figura 1, é mostrado o esquema desse tipo de problema, onde as tarefas de 1 a  $n$  são processadas em uma única máquina, dando origem aos produtos de 1 a  $n$ .

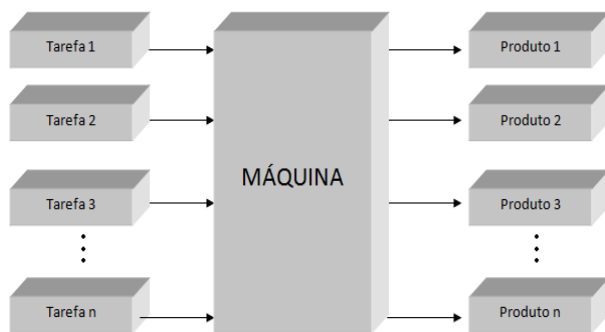


Figura 1 - Esquema do processamento em máquina

Uma ferramenta muito utilizada, tanto no planejamento como no acompanhamento da execução das operações, é o Gráfico de Gantt, construído por Gantt (1910), com a finalidade de verificar o andamento das tarefas e atividades das indústrias. O gráfico foi elaborado com barras de tarefas e marcadores da situação atual, os quais indicam a sequência e a duração de todas as tarefas num processo. Um exemplo é mostrado na Figura 2.

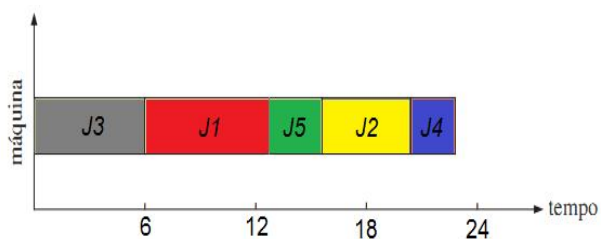


Figura 2 - Gráfico de Gantt para problema de máquina única

Na Figura 2,  $J_3$  é a primeira tarefa a ser processada e inicia seu processamento na data zero, terminando na data seis. A tarefa  $J_1$  tem seu início na data seis e seu término na data treze e assim sucessivamente com as outras tarefas.

### 2.2 MÁQUINA ÚNICA COM TEMPOS DE *SETUP*

Tempo de *setup* é o período dispendido para preparar a máquina ou o ambiente para a fabricação do próximo produto (ou lote de produtos), incluindo operações de posicionamento, ajustes, inspeção, obtenção de ferramentas e materiais a serem processados, limpeza, entre outros. O tempo de *setup* está diretamente relacionado com as variações do produto e o planejamento da produção realizado pela indústria.

Segundo Allahverdi, Gupta e Aldowaisan (1999), existem dois tipos de problemas que requerem que os tempos de preparação das máquinas sejam explícitos ou separados do tempo de processamento das

tarefas: tempos de preparação das máquinas independentes da sequência de produção, os quais se subordinam somente à operação a ser processada, e tempos de preparação das máquinas dependentes da sequência de produção, que se espelham tanto na operação a ser processada quanto naquela que foi processada imediatamente antes na mesma máquina.

Nesse artigo, consideram-se os tempos de preparação das máquinas dependentes da sequência de produção. O exemplo prático mais conhecido deste tipo de *setup* é a indústria de tinta, em que os processos de limpeza são diferenciados de acordo com a cor que foi produzida anteriormente e com aquela que será fabricada em seguida. O *setup* dependente da sequência também pode ser encontrado em indústrias químicas e farmacêuticas e em processos que exigem ajustes de temperatura.

### 2.3 MÁQUINA ÚNICA E *JUST-IN-TIME*

Para uma melhor compreensão sobre o termo *Just-in-Time* (JIT), buscou-se primeiro entender sua origem e sua utilização nas organizações. A filosofia JIT é uma ferramenta que a organização moderna utiliza na busca da melhoria contínua dos seus processos.

A ideia do JIT caracteriza-se em adquirir materiais apenas a tempo de serem transformados em peças fabricadas, em fazer peça apenas a tempo de entrar nas sub-montagens, em montá-las apenas a tempo de montar os produtos acabados e, finalmente, em entregar produtos apenas a tempo de ser vendidos ao consumidor final (SCHONBERGER, 1984).

Sob o enfoque de uma das linhas de programação da produção, Baker e Scudder (1990) mostram que o JIT pode ter sua dimensão de programação caracterizada por meio da penalização do término adiantado ou atrasado de ordens de produção.

Maiores discussões acerca da aplicabilidade e definição desses sistemas podem ser encontradas em

Silver, Pyke e Peterson (1998) e Vollmann, Berry e Whybark (1997).

Um das consequências mais importantes advindas da filosofia JIT é que, dependendo de certas condições, pode valer a pena manter uma máquina parada, inserindo-se tempo de ociosidade entre a realização de suas ordens. Essa característica, peculiar ao ambiente do tipo JIT, é válida enquanto os custos provenientes de um suposto atraso não forem maiores do que os custos provenientes do adiantamento. Portanto, uma metodologia que leve em consideração ambos os custos – de adiantamento e de atraso – pode ser de grande validade quando se deseja trabalhar em um ambiente desse tipo (COLIN; SHIMIZU, 2000).

### 2.4 MODELAGEM MATEMÁTICA

Quando se trata de modelos, logo vem a ideia de representação, seja de roupas, pessoas, entre outros. Para Goldbarg e Luna (2005), modelos são representações simplificadas da realidade, que preservam, para determinadas situações e enfoques, uma equivalência adequada.

Um modelo matemático pode ser apresentado como uma representação de um sistema real, o que significa que um modelo deve representar um sistema e a forma como ocorrem as suas modificações. Para uma melhor classificação dos problemas, Goldbarg e Luna (2005) subdividem a programação matemática em algumas subáreas, que são: Programação Linear (quando todas as variáveis são contínuas e a função-objetivo e as restrições apresentam comportamento linear); Programação Não-Linear (quando ocorre algum tipo de não-linearidade, seja na função-objetivo ou em qualquer uma das suas restrições); Programação Inteira (quando qualquer variável não pode assumir valores contínuos, ficando condicionada a assumir valores discretos) e Programação Linear

Inteira Mista (neste caso há uma mistura da programação linear com a programação inteira, ou seja, alguma das variáveis assume valores discretos).

### 3 MODELAGEM DO PROBLEMA

#### 3.1 PRESSUPOSTOS DO PROBLEMA

O problema estudado neste trabalho possui as seguintes características: uma única máquina deve processar um conjunto de  $n$  tarefas; cada tarefa possui um tempo de processamento, uma data de entrega (prazo) e tempos de *setup*, todos fixos e conhecidos previamente; a máquina executa uma tarefa por vez e, uma vez iniciado o processamento de uma tarefa, este deve ser finalizado, ou seja, não é permitida a interrupção do processamento; todas as tarefas estão disponíveis para processamento na data 0, isto é, as datas de liberação das tarefas são iguais a zero; quando uma tarefa  $J_j$  é sequenciada imediatamente após uma tarefa  $J_i$ , é necessário um tempo para a preparação da máquina; assume-se, ainda, que a máquina necessita de tempo de preparação inicial, isso significa que existe  $s_{0j}$ ; então é permitido tempo ocioso entre a execução de duas tarefas consecutivas; se a data de término de uma tarefa for menor que a data de entrega, então se tem uma penalidade pela antecipação (decorrente de custos de estoque) e, caso seja maior que a data de entrega, tem-se uma penalidade de atraso; o objetivo a ser alcançado com a resolução deste problema é a minimização do somatório dos tempos de antecipação e atraso da produção.

A seguir é apresentado o modelo matemático para representar o problema e o método de sua resolução.

#### 3.1.1 MODELAGEM MATEMÁTICA

O principal intuito deste trabalho é apresentar um modelo matemático de programação linear inteira mista.

A modelagem matemática para o sequenciamento de produção com *setup* dependente e filosofia *Just-in-Time* é apresentada abaixo e foi baseada no trabalho de Fuchigami (2012):

Considere as seguintes definições:

$E_j$  → adiantamento da tarefa  $J_j$

$T_j$  → atraso da tarefa  $J_j$

$C_j$  → data de término da tarefa  $J_j$

$p_j$  → tempo de processamento da tarefa  $J_j$

$s_{ij}$  → tempo de *setup* entre os pares de tarefas (de  $J_i$  para  $J_j$ )

$d_j$  → data estipulada para a entrega da tarefa  $J_j$

Variáveis de decisão:

$X_{ij} = \begin{cases} 1, & \text{se a tarefa } J_i \text{ precede imediatamente } J_j; \\ 0, & \text{caso contrário} \end{cases}$

Seja  $J_0$  uma tarefa fictícia que precede imediatamente a primeira tarefa e sucede imediatamente a última tarefa da sequência e  $B$  uma constante muito grande.

Modelo:

$$\text{Min } \sum_{j=1}^n (E_j + T_j)$$

$$\text{s.a } \sum_{i=0, i \neq j}^n X_{ij} = 1, \quad j=0, 1, \dots, n \quad (3.1)$$

$$\sum_{j=0, j \neq i}^n X_{ij} = 1, \quad i=0, 1, \dots, n \quad (3.2)$$

$$C_j - C_i + B(1 - X_{ij}) \geq p_j + s_{ij}, \\ i=0, 1, \dots, n, j=1, \dots, n \quad (3.3)$$

$$E_j \geq d_j - C_j \quad j=1, \dots, n \quad (3.4)$$

$$T_j \geq C_j - d_j \quad j=1, \dots, n \quad (3.5)$$

$$E_j \geq 0, T_j \geq 0, \quad j=1, \dots, n \quad (3.6)$$

$$C_j \geq 0 \quad j=1, \dots, n \quad C_0 = 0 \quad (3.7)$$

$$X_{ij} \in \{0,1\} \quad i=0, 1, \dots, n, j=0, 1, \dots, n, X_{j0}=0 \quad (3.8)$$

As restrições (3.1) e (3.2) garantem que cada tarefa terá somente uma predecessora e uma sucessora, respectivamente.

A restrição (3.3) assegura a consistência das datas de término das tarefas em relação ao tempo de processamento e tempo de setup, de acordo com a tarefa anterior. Se  $X_{ij} = 1$ , implica que  $C_j \geq C_i + p_j + s_{ij}$  e se  $X_{ij} = 0$ , tem-se que  $C_j - C_i \geq -B$ , assim a restrição será desativada.

As restrições (3.4), (3.5) e (3.6) calculam os valores dos atrasos e adiantamentos, garantindo que não haja valores negativos.

As restrições (3.7) e (3.8) indicam o domínio das variáveis. Não há valores negativos para data de término das tarefas, sendo que a fictícia terá data de término "zero".

A Tabela 1, a seguir, mostra o número de variáveis e restrições do modelo para  $n$  tarefas.

Tabela 1  
Tamanho do problema em relação ao modelo

Variáveis	Binárias	$(n+1)^2$
	Inteiras	$(n+1)$
		$(n+1)$
		$(n+1)$
<b>Total</b>	<b><math>(n+1)^2+3(n+1)</math></b>	
Restrições	(3.1)	$(n+1)$
	(3.2)	$(n+1)$
	(3.3)	$n(n+1)$
	(3.5)	$N$
	(3.6)	$N$
	(3.7)	$2n$
	(3.8)	$(n+1)$
	(3.9)	$(n+1)$
	<b>Total</b>	<b><math>n^2+9n+4</math></b>

Exemplo para  $n=5$  tarefas:  $(5+1)^2+3(5+1) = 54$  variáveis;  $5^2+9(5)+4 = 74$  restrições.

### 3.1.2 APLICAÇÃO DO PROBLEMA

Para melhor entendimento com relação ao objetivo deste estudo, deve-se considerar o seguinte exemplo de um problema de máquina única, com *setup* dependente e duas tarefas ( $J_1$  e  $J_2$ ), cujos dados de tempos de processamentos ( $p_j$ ), das datas de entregas ( $d_j$ ) e tempos de *setup* ( $s_{ij}$ ) são apresentados na Tabela 2.

Tabela 2  
Dados do problema exemplo

$J_j$	$J_1$	$J_2$
$p_j$	7	5
$d_j$	11	17
$s_{0j}$	3	1
$s_{1j}$	0	2
$s_{2j}$	3	0

Neste exemplo, existem apenas duas programações possíveis para o sequenciamento dessas tarefas, a primeira é a sequência  $J_1-J_2$  e a segunda é a sequência  $J_2-J_1$ , conforme Tabela 3:

Tabela 3  
Medidas de desempenho do problema exemplo

Sequência	$J_1 - J_2$		$J_2 - J_1$	
$C_j$	10	17	6	16
$E_j$	1	0	11	0
$T_j$	0	0	0	5
<b><math>\Sigma(E_j + T_j)</math></b>	1		16	

Assim, a soma dos adiantamentos e atrasos deste exemplo mostraram que neste caso o melhor sequenciamento é  $J_1-J_2$ , uma vez que este valor foi de uma unidade de tempo, que se comparado com a outra sequência é muito menor. Tendo em vista que, em muitos casos, é necessário que os produtos sejam entregues o mais próximo possível da data combinada, para evitar excesso de estoques ou a sua

falta, um simples sequenciamento pode resolver este problema.

## 4 ANÁLISE DOS RESULTADOS

### 4.1 EXPERIMENTAÇÃO COMPUTACIONAL

Neste capítulo são apresentados e analisados os resultados obtidos pela modelagem matemática do problema estudado. A implementação foi realizada em um computador Intel *Pentium*® *Dual-Core*, com 2,50 GHz de frequência e 2 GB de memória RAM, sistema operacional de 32 Bits, sob plataforma *Windows 7 Ultimate*.

Para a realização da experimentação computacional, foram definidos diferentes valores para as tarefas, um intervalo uniformemente distribuído para os tempos de processamento e um intervalo para os tempos de *setup*, conforme consta na Tabela 4. Os valores utilizados seguem uma distribuição uniforme no intervalo  $[P(1-T-R/2), P(1-T+R/2)]$ , onde  $T$  e  $R$  são dois parâmetros denominados fator de atraso e faixa de dispersão, respectivamente, e  $P$  é um limitante inferior para o *makespan*, definido aqui como  $P = \sum_{j=1}^n (p_j + \min_i s_{ij})$ . Os critérios utilizados para a obtenção dos valores foram retirados de Ventura e Radhakrishan (2003), e os valores foram gerados aleatoriamente, utilizando-se a função de gerar números aleatórios disponível em planilhas eletrônicas.

Tabela 4  
Parâmetros do problema para experimentação computacional

Parâmetros	Símbolo	Valores
Tempos de processamentos	$p_j$	U[1-99]
Tempos de <i>setup</i>	$s_j$	U[1-49]
Datas de entrega ( <i>due dates</i> )	$d_j$	U $[P(1-T-R/2), P(1-T+R/2)]$
Fator de atraso	$T$	0,1
Faixa de dispersão	$R$	0,5

### 4.2 RESULTADOS DO MODELO

Foi utilizado o *software* modelador MPL 2.0 e o otimizador GUROBI 1.0.4. A título de ilustração, serão apresentadas detalhadamente as soluções de dois problemas resolvidos na experimentação (com 5 e 10 tarefas). As Tabelas 5 e 6 apresentam os dados do problema com cinco tarefas.

Tabela 5  
Tempos de processamento e data de entrega do problema com cinco tarefas

$J_j$	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$
$p_j$	32	67	34	68	90
$d_j$	224	281	193	237	258

Tabela 6  
Tempos de *setup* do problema com cinco tarefas

$s_{ij}$	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$
$J_0$	14	9	6	41	47
$J_1$	0	31	33	7	35
$J_2$	35	0	49	11	10
$J_3$	44	48	0	18	15
$J_4$	2	46	39	0	36
$J_5$	37	18	8	18	0

A Tabela 7 mostra os resultados do problema com 5 tarefas, indicando os valores dos adiantamentos e atrasos das tarefas e a função-objetivo com valor ótimo de 341.

Tabela 7  
Resultados do problema com cinco tarefas

$J_j$	$C_j$	$E_j$	$T_j$
$J_1$	224	0	0
$J_2$	322	0	41
$J_3$	104	89	0
$J_4$	190	47	0
$J_5$	422	0	164
$\sum_{j=1}^n (E_j + T_j)$			$Z = 341$

A sequência ótima obtida:  $J_3 - J_4 - J_1 - J_2 - J_5$ .

A Figura 3 apresenta o gráfico com a programação ótima obtida pelo modelo matemático.

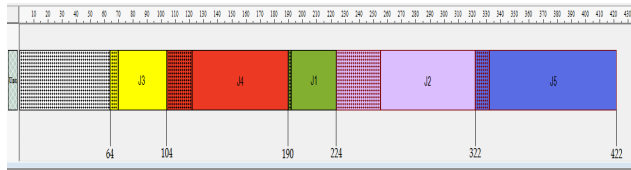


Figura 3 - Programação ótima do problema com cinco tarefas

Observando-se a Figura 3, verifica-se que se faz necessário incluir um tempo de máquina ociosa igual a 64 unidades de tempo antes de começar o processamento das tarefas, para se obter a menor soma dos adiantamentos e atrasos da produção.

O próximo problema resolvido contém 10 tarefas, e os seus dados encontram-se nas Tabelas 8 e 9.

Tabela 8  
Tempos de processamento e datas de entrega do problema com dez tarefas.

$J_j$	$p_j$	$d_j$
$J_1$	89	370
$J_2$	75	401
$J_3$	94	478
$J_4$	12	566
$J_5$	84	540
$J_6$	4	341
$J_7$	32	423
$J_8$	57	573
$J_9$	9	567
$J_{10}$	50	528

Tabela 9  
Tempos de *setup* do problema com dez tarefas

$S_{ij}$	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$	$J_6$	$J_7$	$J_8$	$J_9$	$J_{10}$
$J_0$	24	6	45	9	16	43	23	42	5	1
$J_1$	0	43	42	34	18	30	14	33	7	8
$J_2$	7	0	16	37	28	12	18	22	15	46
$J_3$	32	47	0	15	47	26	33	19	38	8
$J_4$	43	17	28	0	16	8	16	32	25	44
$J_5$	22	28	45	28	0	24	29	13	11	20
$J_6$	22	3	17	3	4	0	1	2	41	44
$J_7$	23	5	38	7	14	28	0	49	1	27
$J_8$	36	24	31	39	36	38	7	0	43	14
$J_9$	44	35	23	42	35	36	5	40	0	20
$J_{10}$	41	42	1	48	33	22	45	40	47	0

Os resultados do problema com dez tarefas são apresentados na Tabela 10, incluindo o valor ótimo da função-objetivo de 888.

Tabela 10  
Resultados do problema com dez tarefas

$J_j$	$C_j$	$E_j$	$T_j$
$J_1$	280	90	0
$J_2$	392	90	0
$J_3$	159	319	0
$J_4$	614	0	48
$J_5$	540	0	0
$J_6$	314	27	0
$J_7$	442	0	19
$J_8$	703	0	130
$J_9$	560	7	0
$J_{10}$	767	0	239
$\sum_{j=1}^n (E_j + T_j)$		<b>Z = 888</b>	



A sequência ótima para este problema é:  $J_3 - J_1 - J_6 - J_2 - J_7 - J_5 - J_9 - J_4 - J_8 - J_{10}$ .

A Figura 4 mostra o gráfico de Gantt com a programação ótima do problema com 10 tarefas.

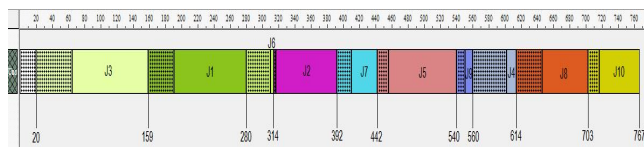


Figura 4 - Programação ótima do problema com dez tarefas

Tabela 11  
Resultados dos problemas resolvidos

Problemas	Nº de Tarefas	Nº de variáveis	Nº de restrições	Nº de iterações	Tempo de execução	Solução obtida
P1	5	54	74	1256	0,08s	241
P2	7	88	116	37514	0,84s	482
P3	10	154	194	17731032	6m, 24s	888
P4	11	180	224	57987993	23m, 36s	907
P5	12	208	256	155568634	1h, 10m, 46s	735
P6	13	238	290	230145195	2h*	1209**
P7	20	504	584	90744848	2h*	2982**
P8	30	1054	1174	35187831	2h*	8345**
P9	50	2754	2954	13087549	2h*	22149**
P10	80	6804	7124	5906870	2h*	58753**
P11	100	10504	10904	1940680	55m, 32s***	122109**

\*Tempo de execução limitado por 2h.  
 \*\*Melhor solução encontrada durante a execução.  
 \*\*\*Algoritmo finalizado por memória 'insuficiente'.

Analisando os resultados obtidos durante a execução, percebe-se que o modelo executa perfeitamente fornecendo a solução ótima com até doze tarefas em um tempo inferior a 2 horas. Então o tempo foi limitado em duas horas, encontrando-se uma solução viável,

Faz-se necessário, então, analisar o custo computacional para a execução do modelo para determinados números de tarefas, conforme mostrado na Tabela 11.

mas não ótima. O algoritmo foi interrompido por falta de memória na execução do modelo com cem tarefas, caso ocorrido no problema P11.

Os problemas que tratam de sequenciamento da produção são de grande aplicabilidade, principalmente

industrial. Por isso, é importante o seu estudo e a proposição de métodos de solução adequados e viáveis.

Foram considerados problemas testes com diferentes números de tarefas no experimento. A solução ótima foi encontrada em tempo computacional viável para problemas com até doze tarefas. Para quantidades maiores, foi determinado um tempo máximo de execução do programa igual a duas horas, obtendo a melhor solução neste intervalo.

A restrição da utilização de modelos matemáticos na resolução de problemas é justamente o tempo computacional, que se torna inaceitável para problemas com tamanhos a partir de médio porte. Portanto, aconselha-se a utilização de métodos heurísticos, principalmente para problemas de grande porte.

Nota-se também que quanto maior for o número de tarefas, de variáveis e de restrições, maior será o número de iterações e o tempo de execução. O problema com números de iterações maiores foi interrompido, pois faltou memória suficiente no computador.

## 5 CONCLUSÃO

Foi proposto neste trabalho um modelo matemático para se resolver problemas de sequenciamento com uma máquina, tempos de *setup* dependentes e filosofia *Just-in-Time*. O modelo de programação linear inteira mista investigado foi implementado por meio do *software* MPL e resolvido pelo *software* Gurobi 1.0.4.

Foi considerada a filosofia *Just-in-Time*, isto é, os produtos devem ser entregues o mais próximo possível dos prazos estabelecidos, definindo a medida de desempenho como a minimização da soma dos adiantamentos e atrasos das tarefas.

Para trabalhos futuros, sugere-se o estudo das propriedades estruturais e a implementação de problemas de programação da produção em outros ambientes, como máquinas paralelas e *flow shop*, incluindo restrições diferentes das tratadas nesta pesquisa, como *setup* independente da sequência, janelas de entrega e penalidades individuais (distintas por tarefas) por antecipação e atraso.

---

## REFERÊNCIAS

ALLAHVERDI, A.; GUPTA, J. N. D.; ALDOWAISAN, T. **A review of scheduling research involving setups considerations**. *Omega – The international Journal of Management Science*, 27, p. 219-239, 1999.

BAKER, K. R.; SCUDDER, G. D. **Sequencing with Earliness and Tardiness Penalties: A Review**. *Operation Research*, v. 38, p.22-36, 1990.

COLIN, E. C.; SHIMIZU, T. **Algoritmo de programação de máquinas individuais com penalidades distintas de adiantamento e atraso**. *Pesquisa Operacional*, v.20, n.1, 2000.

FUCHIGAMI, H. Y. **Introdução ao Sequenciamento da Produção**. Material Didático. Catalão: UFG, 2012.

GAITHER, N.; FRAZIER, G. **Administração da produção e operações**. 8ª ed. São Paulo: Pioneira, 2005.

GANTT, H. L. **Work, Wages and Profits**. The Engineering Magazine, New York, 1910.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear: modelos e algoritmos**. 2ª Ed. Editora Campus, Rio de Janeiro – RJ, 2005.

LAWLER, E. L. *et al.* **The Traveling Salesman Problem. Wiley-Interscience series in discrete mathematics and optimization**. John Wiley & Sons, New York. 1985.

MENDONÇA, F. **O Problema da Programação do Lote Econômico Aplicado a uma Indústria por Processo**. Dissertação de Mestrado, Departamento de Engenharia Industrial, PUC-Rio de Janeiro, 1996.

PIZZOLATO, N. D.; VÁSQUEZ, S. G. G.; D'ÁVILA, S. L. G. **O problema do sequenciamento da produção em uma indústria química: avaliação de uma aplicação real**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 1999.

REINELT, G. The Traveling Salesman: Computational Solutions for TSP Applications. **Lecture Notes in Computer Science**, Springer Verlag, Berlin Heideberg, v.840, 1994.

SCHONBERGER, R. J. **Técnicas industriais japonesas: nove lições ocultas sobre simplicidade**.

Tradução de Oswaldo Chiquetto. São Paulo: Saraiva, 1984.

SILVER, E. A.; PYKE, D. F.; PETERSON, R. **Inventory management and production planning and scheduling** 3.ed. New York: Wiley, 1998.

TUBINO, D. F. **Planejamento e Controle da Produção: teoria e prática**. 1. ed. São Paulo: Atlas, 190p. 2008.

VENTURA, J. A.; RADHAKRISHAN, S. **Single machine scheduling with symmetric earliness and tardiness penalties**. European Journal of Operational Research, v.144, p.598-612, 2003

VOLLMANN, T. E.; BERRY, W. L.; WHYBARK, D. C. **Manufacturing planning and control systems**.4.Ed. Mc Graw : Hill, New York. 1997.