

# **SEQUENCIAMENTO DE UM LOTE DE PEÇAS EM UMA ÚNICA MÁQUINA COM DATAS DE ENTREGA E PENALIDADES: UM ESTUDO DE CASO**

**Gustavo Simão Rodrigues**

Instituto Militar de Engenharia – Seção de Ensino de Engenharia Mecânica e de Materiais  
Praça General Tibúrcio, 80, Praia Vermelha – Rio de Janeiro – RJ, Brasil  
[gustavosimao@uol.com.br](mailto:gustavosimao@uol.com.br)

**Nélio Domingues Pizzolato**

Pontifícia Universidade Católica do Rio de Janeiro – Departamento de Engenharia Industrial  
Rua Marquês de São Vicente, 225, Gávea – Rio de Janeiro – RJ, Brasil  
[ndp@puc-rio.br](mailto:ndp@puc-rio.br)

## **RESUMO**

Este artigo analisa o problema do sequenciamento de um conjunto de peças em uma única máquina, com tempos de *setup* dependentes da sequência, mas considerando penalidades de antecipação e atraso sobre as datas prometidas. A metodologia é aplicada a um estudo de caso que considera um processo de usinagem em que todas as tarefas e matérias primas estão disponibilizadas no mesmo momento inicial e todas as peças fazem parte de um produto final mais complexo, devendo atender prazos de entrega pré-estabelecidos. Busca-se encontrar o sequenciamento da produção que minimiza o somatório dos custos das penalidades. Para a solução do problema é aplicada a ferramenta AIMMS, associada a elementos heurísticos, buscando estabelecer a melhor sequência para processamento.

**PALAVRAS-CHAVE.** Sequenciamento em uma Única Máquina, Fabricação por Usinagem, Problema do Caixeiro Viajante

## **ABSTRACT**

This paper examines the single machine scheduling problem with sequence dependent setup times and earliness and tardiness penalties over due dates. The methodology is applied to a case study that deals with a machining process and assumes that all jobs and raw materials are available at the same initial moment. These jobs have to satisfy a given due date since they are parts of a more complex final product. The objective is to sequence the jobs in order to minimize the earliness and tardiness costs. To solve the problem the mathematical tool AIMMS is applied, associated to heuristic elements, to find the best job sequence.

**KEYWORDS.** Single Machine Scheduling, Machining Process, Mixed Integer Programming Models, Traveling Salesman Problem

## 1. INTRODUÇÃO

O Sistema de Produção Just-in-Time (JIT) foi proposto no Japão na década de 70 e muitos afirmam que ele tenha por único objetivo a redução ou eliminação de todo tipo de estoques. Na realidade, o sistema busca atender a outros objetivos igualmente importantes, como a eliminação de desperdícios, a melhor qualidade dos produtos e o aumento na flexibilidade do sistema. O desenvolvimento do JIT tornou-se viável pelo aparecimento das máquinas ferramentas com controle numérico, ou seja, pelos avanços tecnológicos observados nos processos produtivos. Por outro lado, a redução dos estoques, tanto o estoque final quanto o estoque em processo, pode ser creditada em grande parte à otimização do sequenciamento em que as peças são processadas, tema sobre o qual existe uma vasta literatura.

Em um ambiente geral de fabricação, o papel da programação é destacado nas ordens de serviço que são lançadas na configuração da fabricação na forma de trabalhos com datas de entrega associadas. Esses trabalhos normalmente devem ser processados em máquinas em uma dada ordem ou sequência. O caso mais frequente costuma ser a preocupação com o tempo total de produção de todas as peças, ou seja, a minimização do *makespan*. Há situações, entretanto, em que a preocupação maior consiste em atender às datas prometidas, em inglês, *due dates*. As duas abordagens são, tecnicamente falando, muito distintas, especialmente em casos de *setups* dependentes da sequência de produção. No caso em que se busca o menor *makespan*, aplica-se a metodologia do caixeiro viajante, enquanto que no caso de haver datas prometidas distintas o problema se torna muito complexo, não havendo metodologia consolidada, mas métodos desenvolvidos caso a caso, sempre com forte conteúdo heurístico para compensar a incapacidade dos *softwares* processarem problemas de tamanho realista.

Outro fato relevante é que, ao otimizar a sequência de produção em vista do menor *makespan*, otimiza-se também o aproveitamento da máquina, enquanto que, ao minimizar as penalidades, pode-se incorrer em maiores *setups* e reduzir o aproveitamento da máquina. Há, portanto, que levar em conta o custo de oportunidade de ver o equipamento parado, em favor da melhor programação e redução dos custos de penalidades.

Outro complicante do problema é quando os processamentos dos trabalhos podem atrasar se certas máquinas estiverem indisponíveis. Eventos imprevistos no chão-de-fábrica, tais como quebra de máquinas ou tempos de processamento maiores que os previstos, devem ser levados em consideração, desde que esses eventos venham a impactar diretamente a programação. Neste ambiente, o desenvolvimento de uma programação de tarefas detalhadas ajuda a manter a eficiência e o controle das operações.

A indústria bélica que originou os dados do presente estudo de caso é uma Organização Militar (OM) composta por aproximadamente 500 funcionários, semelhante a uma indústria civil de médio porte. O principal armamento fabricado nesta OM é um Morteiro Pesado com 120 milímetros de calibre, cuja fabricação das peças mais importantes será o foco deste trabalho. Para a fabricação dessas peças, será considerada somente a operação de usinagem por torneamento, que constitui a operação mais significativa da fabricação das peças.

A máquina utilizada para o processamento (torneamento) das peças é um torno com Controle Numérico Computadorizado (CNC), sem troca automática de ferramentas e com fixação manual da matéria-prima pelas castanhas, onde foi feita a simulação de 10 peças, onde cada peça, com uma única exceção, possui duas fases de usinagem, formando um conjunto de 19 trabalhos.

Neste ambiente de fabricação, foi verificada a inexistência de uma ordenação na fabricação das peças, embora as matérias-primas estejam disponíveis ao mesmo instante para serem processadas. Acrescente-se que é oferecido um tempo útil muito apertado para a fabricação do armamento, o que sugere que seja necessário determinar o sequenciamento na fabricação das peças com o intuito de eliminar os gargalos e estoques e atender os prazos de entrega.

Na sequência, a Seção 2 caracteriza o problema a ser estudado; a Seção 3 estabelece a modelagem matemática para sua solução por método exato; a Seção 4 apresenta os resultados do estudo de caso, separando o problema com datas prometidas quando é usado o AIMMS; e a Seção 5 apresenta as conclusões.

## 2. O Problema Estudado

### 2.1. Definição de Parâmetros

As informações básicas associadas à descrição de trabalhos de sequenciamento em uma única máquina são as seguintes:

- Tempo de processamento ( $t_j$ ): tempo de processamento requerido pelo trabalho  $j$ ;
- Tempo de início do processamento ( $s_j$ ): o ponto no tempo em que o trabalho  $j$  é iniciado;
- Tempo de Preparação de Máquina ou Tempo de *set up* ( $S_{ij}$ ): Tempo necessário para deixar a máquina em condições de realizar o trabalho  $j$ , após a conclusão do trabalho  $i$ ;
- Data prometida ( $d_j$ ): é o ponto no tempo em que o processamento do trabalho  $j$  está prometido para ser concluído.

As datas prometidas podem não ser pertinentes a todos os problemas, mas estabelecer prazos (*deadlines*) é uma prática comum na indústria e o problema básico pode auxiliar na determinação do prazo de entrega. As informações resultantes do sequenciamento são:

- Data de Conclusão ( $C_j$ ). O tempo no qual o processamento do trabalho  $j$  é terminado.

Os critérios quantitativos para escolher uma sequência são geralmente funções dos tempos de conclusão. Uma quantidade importante é:

- Defasagem (*Lateness*) –  $L_j$  – Diferença entre a data de conclusão e a data prometida do trabalho  $j$ , podendo assumir valores positivos ou negativos:

$$L_j = C_j - d_j \quad (1)$$

A defasagem,  $L_j$ , mede a conformidade do sequenciamento em relação ao prometido. É importante notar que a defasagem terá valor negativo quando um trabalho é finalizado antecipadamente. Defasagens negativas podem representar serviços melhores do que solicitados, enquanto atrasos positivos representam, quase sempre, serviços piores do que requisitados. Em muitas situações, penalidades distintas e outros custos serão associados para defasagens positivas e para defasagens negativas. Dessa forma, tem-se as definições de Atraso (*Tardiness*) e Antecipação (*Earliness*):

- Atraso (*Tardiness*) –  $T_j$  – é quanto o trabalho  $j$  é terminado com atraso em relação à sua data prometida, caso contrário será considerado zero:

$$T_j = \max\{0, L_j\} = (C_j - d_j)^+ \quad (2)$$

- Antecipação (*Earliness*) –  $E_j$  – é quanto o trabalho  $j$  é terminado com antecipação em relação à sua data prometida, caso contrário será considerado zero:

$$E_j = \max\{d_j - C_j, 0\} = (d_j - C_j)^+ \quad (3)$$

Outra quantidade muito importante é o *makespan*:

*Makespan* –  $C_{\max}$  – O *makespan*, definido como o maior dentre as datas de conclusão ( $C_1, \dots, C_n$ ), ou seja, o tempo de conclusão do último trabalho a sair do sistema ( $C_{[n]}$ ). Minimizar o *makespan* usualmente implica em uma boa utilização dos recursos (máquinas).

## 2.2. Penalidades por Antecipação e Atraso

Para descrever um modelo genérico com antecipação e atraso, ou modelo E/T, toma-se um número de trabalhos,  $n$ , a ser sequenciado. O trabalho  $j$  é descrito por um tempo de processamento  $t_j$  e uma data prometida  $d_j$ . Conforme indicado, assume-se que os trabalhos estão disponíveis simultaneamente. Como resultado das decisões de programação, ao trabalho  $j$  será atribuído um tempo de conclusão,  $C_j$ .

Baker e Scudder (1990) associam a cada trabalho uma unidade de penalização por antecipação  $\alpha_j > 0$  e uma unidade de penalização por atraso  $\beta_j > 0$ . Assumindo que as funções de penalização são lineares, a função objetivo E/T de programação de uma sequência  $S$  pode ser escrita como  $f(S)$ , onde

$$f(S) = \sum_{j=1}^n [\alpha_j (d_j - C_j)^+ + \beta_j (C_j - d_j)^+] \quad (4)$$

ou

$$f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j) \quad (5)$$

Este modelo E / T com diferentes datas prometidas para o conjunto dos trabalhos é tipicamente de alta complexidade. Garey et al (1988) foram os primeiros a mostrar que este problema é NP-completo. Neste modelo, a sequência ideal pode não ser em forma de V e a inserção de tempo ocioso pode ser desejável. A busca por uma programação ótima pode, no entanto, ser decomposto em dois subproblemas: encontrar uma sequência ótima de trabalhos e programar a inserção de tempos ociosos. No caso das datas prometidas serem tratadas como variáveis de decisão, ou seja, quando as datas prometidas são consideradas como incógnitas a serem determinadas pela solução do modelo, o problema acaba por ser relativamente simples.

Abdul-Razaq e Potts (1988) resolvem este tipo de problema, com uma penalidade para a data prometida incluída, mas eles consideram somente as programações sem tempo ocioso. O seu método de solução é um esquema de *branch-and-bound*, e eles usam um processo de programação com relaxação dinâmica para obter bons limites. Seus resultados computacionais sugerem que os problemas com mais de 20 trabalhos podem levar a tempos de solução excessivos.

Embora existam outros trabalhos que não levam em consideração os tempos ociosos, o modelo que permita a utilização destes tempos é mais realista para a minimização de  $f(S)$ .

Dada uma sequência de trabalho, a sequência ótima para os tempos ociosos pode ser resolvida através da resolução de um problema de programação linear. No entanto, Garey et. al. (1988) fornecem detalhes sobre um procedimento mais simples que pode ser implementado para ser executado em tempo  $O(n \log n)$ .

## 3. Modelagem Matemática para Solução por Método Exato

Os primeiros modelos a tratar do Problema de Sequenciamento da Produção datam das décadas de 50 e 60. Desses modelos, destacam-se os propostos por Manne (1960) e Wagner (1959). Ambos são modelos de Programação Linear Inteira Mista, tendo em mente abordar o problema mais geral de *job shop*.

O modelo de Wagner (1959) usa o problema clássico de alocação de tarefas para assinalar trabalhos a posições na sequência de produção. Já o modelo de Manne (1960) utiliza um par de restrições dicotômicas para controlar a ordem relativa dos trabalhos dentro da sequência de produção.

Nesta Seção será apresentado o problema a ser estudado e as características levadas em consideração para fins de modelagem e em seguida serão apresentados os resultados obtidos com a resolução do modelo através do *software* de otimização AIMMS. Posteriormente, será feita uma análise sem levar em conta as datas prometidas, com o objetivo de se minimizar o *makespan* da sequência a partir da minimização dos Tempos de Preparação de Máquina, utilizando a metodologia do Problema do Caixeiro Viajante.

### 3.1. O Problema

O problema descrito é o problema do sequenciamento em uma máquina com penalidades por antecipação e atraso da produção com tempo de preparação da máquina dependente da sequência de produção e datas prometidas para cada trabalho, possuindo as seguintes características: a) Uma máquina deve processar um conjunto de  $n$  trabalhos; b) Cada trabalho  $i$  possui um tempo de processamento  $t_i$  e uma data prometida,  $d_i$ , desejada para o término do processamento; c) A máquina executa no máximo um trabalho por vez e, uma vez iniciado o processamento de um trabalho, o mesmo deve ser finalizado, ou seja, não é permitida a interrupção do processamento; d) Todos os trabalhos estão disponíveis para processamento no tempo 0; e) Quando um trabalho  $j$  é sequenciado imediatamente após um trabalho  $i$ , é necessário um tempo  $S_{ij}$  para a preparação da máquina. Assume-se, ainda, que a máquina não necessita de tempo de preparação inicial, ou seja, o tempo de preparação da máquina para o processamento do primeiro trabalho na sequência é igual a 0; f) É permitido tempo ocioso entre a execução de dois trabalhos consecutivos; g) Os trabalhos devem ser finalizados o mais próximo da data prometida. Se o trabalho  $i$  for finalizado antes de  $d_i$  então há um custo de manutenção de estoque. Caso o trabalho seja finalizado após  $d_i$ , então há associado um custo por atraso. Os trabalhos que forem finalizados na data prometida não proporcionarão nenhum custo adicional; h) Os custos unitários por antecipação e atraso da produção são dependentes dos trabalhos, ou seja, cada trabalho possui um custo de antecipação  $\alpha_i$  e um custo de atraso  $\beta_i$ ; (i) O objetivo a ser alcançado com a resolução deste problema é a minimização do somatório dos custos de antecipação e atraso da produção.

### 3.2. Modelo

O modelo matemático desenvolvido foi baseado no trabalho de Manne (1960). Sejam  $n$  o número de trabalhos a serem processados,  $t_i$  o tempo de processamento do trabalho  $i$ ,  $s_i$  o tempo de início do processamento do trabalho  $i$  ( $s_i \geq 0$ ) e  $S_{ij}$  o tempo de preparação da máquina necessário para processar o trabalho  $j$  depois do trabalho  $i$ .

Diferentemente de Bustamante (2006), foram utilizados dois trabalhos fictícios, 0 (zero) e  $n + 1$ , de tal forma que 0 antecede imediatamente a primeira tarefa e  $n + 1$  sucede imediatamente o último trabalho na sequência de produção. Admite-se que  $t_0$  e  $t_{n+1}$  são iguais a zero e que  $S_{0i} = 0$  e  $S_{in+1} = 0$ ,  $\forall i = 1, \dots, n$ .

Para garantir que haja um tempo suficiente para completar um trabalho  $i$  antes de começar um trabalho  $j$ , caso este último trabalho  $j$  seja processado imediatamente após um trabalho  $i$ , sem nenhum trabalho intermediário, é necessário impor as restrições (6) e (7). Dados dois trabalhos  $i$  e  $j$ , pode-se estabelecer uma restrição dependendo da precedência: caso  $i$  preceda  $j$ , aplica-se (6) e caso  $j$  preceda  $i$ , aplica-se (7).

$$s_j \geq s_i + t_i + S_{ij} \quad \forall i = 0, 1, 2, \dots, n, \forall j = 0, 1, 2, \dots, n+1 \text{ e } i \neq j \quad (6)$$

$$\text{ou}$$
$$s_i \geq s_j + t_j + S_{ji} \quad \forall i = 1, 2, \dots, n+1, \forall j = 0, 1, 2, \dots, n \text{ e } i \neq j \quad (7)$$

Como estas restrições são disjuntas, faz-se necessária a introdução de uma variável  $y_{ij} \in [0; 1]$  para que esta disjunção não apareça no modelo matemático. A variável  $y_{ij}$  é definida da seguinte forma:  $y_{ij} = 1$ , se o trabalho  $j$  é sequenciado imediatamente após o trabalho  $i$  e  $y_{ij} = 0$ , caso contrário.

Desta maneira, as restrições (6) e (7) podem ser substituídas pelas restrições (8). Neste novo conjunto de restrições,  $M$  é um valor muito grande.

$$s_j - s_i - (M + S_{ij}) y_{ij} \geq t_i - M \quad \forall i = 0, 1, 2, \dots, n, \forall j = 1, 2, \dots, n+1 \text{ e } i \neq j \quad (8)$$

Desta forma, quando  $y_{ij} = 1$ , a restrição (8) torna-se:

$$s_j \geq s_i + t_i + S_{ij} \quad (9)$$

Por outro lado, quando  $y_{ij} = 0$ , a restrição (8) torna-se:

$$s_j - s_i - t_i \geq -M \quad (10)$$

A restrição (10) é, desta forma, desativada, pois a equação (10) é inócua, ou seja, a parcela  $(s_j - s_i - t_i)$  será sempre maior que  $-M$ .

Estas restrições garantem que cada trabalho tenha somente um trabalho imediatamente antecessor e um trabalho imediatamente sucessor, respectivamente. Além disso, estas restrições eliminam o problema da desigualdade triangular.

$$\sum_{i=0, i \neq j}^n y_{ij} = 1 \quad \forall j = 1, 2, \dots, n+1 \quad (11)$$

$$\sum_{j=1, i \neq j}^{n+1} y_{ij} = 1 \quad \forall i = 0, 1, 2, \dots, n \quad (12)$$

Seja  $d_i$  a data prometida do trabalho  $i$ ,  $E_i$  o tempo de antecipação do trabalho  $i$  e  $T_i$  o tempo de atraso do trabalho  $i$ . As restrições (13) a (16) garantem que o tempo de antecipação  $E_i$  seja o máximo entre 0 e  $d_i - t_i - s_i$  e que o atraso  $T_i$  seja o máximo entre 0 e  $s_i + t_i - d_i$ .

$$s_i + t_i + E_i \geq d_i \quad \forall i = 1, 2, \dots, n \quad (13)$$

$$s_i + t_i - T_i \leq d_i \quad \forall i = 1, 2, \dots, n \quad (14)$$

$$E_i \geq 0 \quad \forall i = 1, 2, \dots, n \quad (15)$$

$$T_i \geq 0 \quad \forall i = 1, 2, \dots, n \quad (16)$$

Sejam  $\alpha_i$  e  $\beta_i$  os custos de antecipação e atraso da produção do trabalho  $i$  por unidade de tempo, respectivamente. O custo total por antecipação é dado por  $\sum_{i=1}^n \alpha_i E_i$  e o custo por atraso é dado por  $\sum_{i=1}^n \beta_i T_i$ . A função objetivo, que consiste em minimizar o somatório dos custos totais de antecipação e atraso é dada pela equação (17).

$$\min Z = \sum_{i=1}^n (\alpha_i E_i + \beta_i T_i) \quad (17)$$

Resumindo, as variáveis de decisão do modelo proposto são:

- $s_i$ : tempo de início do processamento do trabalho  $i$ .
- $y_{ij}$ : variável que determina a sequência de produção, se  $y_{ij}=1$  o trabalho  $j$  é processado depois do trabalho  $i$  e 0 caso contrário.
- $E_i$ : tempo de antecipação do trabalho  $i$ .
- $T_i$ : tempo de atraso do trabalho  $i$ .

E o modelo correspondente de Programação Linear Inteira Mista é:

$$\text{minimizar } Z = \sum_{i=1}^n (\alpha_i E_i + \beta_i T_i) \quad (18)$$

$$\text{sujeito a:} \quad s_j - s_i - (M + S_{ij}) y_{ij} \geq t_i - M \quad \forall i, j = 1, 2, \dots, n \text{ e } i \neq j \quad (19)$$

$$\sum_{i=0, i \neq j}^n y_{ij} = 1 \quad \forall j = 1, 2, \dots, n+1 \quad (20)$$

$$\sum_{j=1, i \neq j}^{n+1} y_{ij} = 1 \quad \forall i = 0, 1, 2, \dots, n \quad (21)$$

$$s_i + t_i + E_i \geq d_i \quad \forall i = 1, 2, \dots, n \quad (22)$$

$$s_i + t_i - T_i \leq d_i \quad \forall i = 1, 2, \dots, n \quad (23)$$

$$s_i \geq 0 \quad \forall i = 1, 2, \dots, n \quad (24)$$

$$E_i \geq 0 \quad \forall i = 1, 2, \dots, n \quad (25)$$

$$T_i \geq 0 \quad \forall i = 1, 2, \dots, n \quad (26)$$

$$y_{ij} \in \{0,1\} \quad \forall i, j = 0, 1, 2, \dots, n+1 \quad (27)$$

A função objetivo, representada pela equação (18), tem como critério de otimização a minimização dos custos de antecipação e atraso. A restrição (19) definem a sequência de operações sobre o recurso (máquina) utilizado. As restrições (20) e (21) garantem que cada trabalho tenha somente um trabalho imediatamente antecessor e um trabalho imediatamente sucessor, respectivamente. As restrições (22) e (23) definem os valores do atraso e da antecipação de acordo com a data prometida desejada para o término do processamento do trabalho  $i$ , caso estes existam. As restrições (24) a (27) definem o domínio das variáveis do problema.

#### 4. Resultados

Os resultados aqui apresentados utilizam o modelo proposto na Seção 3.2, resolvido por meio do *software* de otimização AIMMS para os problemas com datas prometidas. Esse tipo de problema pertence à classe NP-difícil, (SOUZA, 2011). A propósito, foi constatado que o programa AIMMS ao nosso alcance tem limites de capacidade bem modestos para processar problemas desse tipo.

No caso do problema analisado com 19 trabalhos, não se tem um tempo de preparação simétrico, ou seja,  $S_{ij} \neq S_{ji}$ . Sendo assim, o número total de sequências possíveis é de  $(n - 1)!$ , ou seja,  $18!$ . Supondo que um computador avalie cada sequência de trabalhos em  $10^{-8}$  segundos, seriam necessários 741 dias para analisar todas as sequências!

Ao se implementar o modelo para 19 trabalhos no *software* AIMMS, comprovou-se que devido à complexidade do problema, o *software* não encontraria a solução ótima em tempo razoável. Esse fato propiciou a adoção de uma heurística, qual seja, a divisão dos trabalhos em grupos, de acordo com suas datas prometidas.

##### 4.1. Resultados para 19 Trabalhos com Datas Prometidas

Os dados do problema analisado podem ser resumidos nas seguintes informações: a) Dados 19 trabalhos numerados sequencialmente 1 a 19; b) Oito trabalhos (1, 2, 8, 9, 10, 11, 12 e 13) têm data prometida comum,  $d = 600$  min; c) Nove trabalhos (3, 4, 5, 6, 7, 16, 17, 18 e 19) têm datas prometidas que variam entre 100, 150 e 250 minutos; e d) Dois trabalhos, 14 e 15, têm data prometida de 400 minutos. Os tempos de *setup* são dependentes da sequência e exibidos nas Tabelas 1 e 2 para todos os trabalhos. Aproveitando as diferentes datas prometidas, vamos fazer uso de um raciocínio heurístico que sugere a divisão do problema em três grupos distintos e menores, a saber: datas prometidas 600; 400; e aqueles com datas 100, 150 e 250 minutos.

Tabela 1 – Tempos de setup para 11 trabalhos com  $d_i = 100, 150, 250$  e  $400$

	3	4	5	6	7	14	15	16	17	18	19
3	0	47,22	40,22	38,22	32,22	32,22	32,22	46,22	52,22	41,22	41,22
4	29,16	0	28,16	14,16	20,16	20,16	20,16	34,16	28,16	29,16	29,16
5	37,64	43,64	0	34,64	28,64	28,64	28,64	26,64	32,64	37,64	37,64
6	22,74	16,74	21,74	0	13,74	13,74	13,74	27,74	21,74	22,74	22,74
7	24,48	30,48	23,48	21,48	0	15,48	15,48	29,48	35,48	24,48	24,48
14	21,90	27,90	20,90	18,90	12,90	0	12,90	26,90	32,90	21,90	21,90
15	14,16	20,16	13,16	11,16	5,16	5,16	0	19,16	25,16	14,16	14,16
16	33,32	39,32	16,32	30,32	24,32	24,32	24,32	0	16,32	33,32	33,32
17	41,90	35,90	24,90	26,90	32,90	32,90	32,90	18,90	0	41,90	41,90
18	30,90	36,90	29,90	27,90	21,90	21,90	21,90	35,90	41,90	0	30,90
19	23,16	29,16	22,16	20,16	14,16	14,16	14,16	28,16	34,16	23,16	0

Tabela 2– Tempos de set up para 8 trabalhos com  $d_i = 600$

	1	2	8	9	10	11	12	13
1	0	36,90	29,90	41,90	29,90	41,90	29,90	35,90
2	36,90	0	35,90	35,90	35,90	35,90	35,90	29,90
8	24,74	30,74	0	19,74	7,74	19,74	7,74	13,74
9	39,32	33,32	30,32	0	22,32	10,32	22,32	16,32
10	22,16	28,16	13,16	17,16	0	17,16	5,16	11,16
11	36,74	30,74	27,74	7,74	19,74	0	19,74	13,74
12	27,32	33,32	18,32	22,32	10,32	22,32	0	16,32
13	33,32	27,32	24,32	16,32	16,32	16,32	16,32	0

Como mencionado, será feita uma divisão dos trabalhos de acordo com suas datas prometidas. Inicialmente, são programados os trabalhos com data prometida de 600 min; depois, aqueles com datas prometidas de 100, 150 e 250 minutos; por fim, analisam-se os dois trabalhos restantes com datas prometidas igual a 400 minutos.

A Figura 1 apresenta o gráfico de Gantt para os 8 trabalhos com data prometida de 600. A Figura 2, construída a partir dos dados de saída gerados pelo AIMMS, apresenta os resultados para os inícios dos trabalhos e os respectivos desvios (antecipações e atrasos), bem como o valor da função objetivo, que vale 3.948,1.

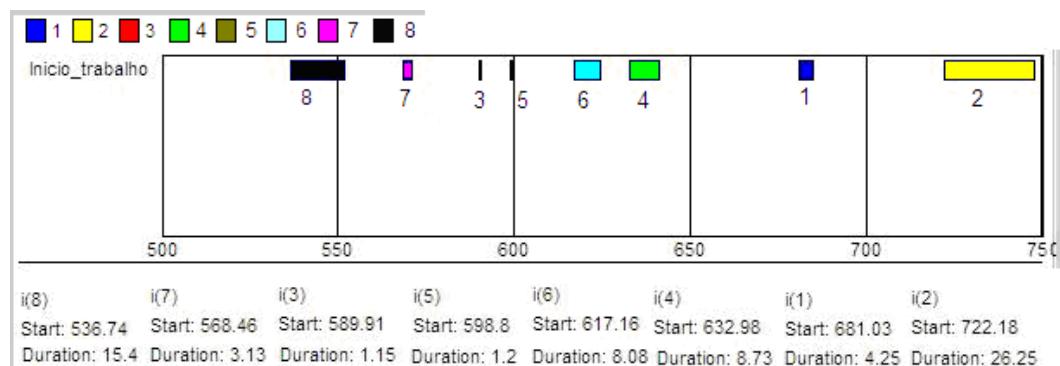


Figura 1 – Gráfico de Gantt para 8 trabalhos com  $d_i = 600$

Fonte: Autor



Os trabalhos numerados de 1 a 8 nas Figuras 1 e 2 correspondem aos trabalhos 1, 2, 8, 9, 10, 11, 12 e 13, respectivamente.

Início_trabalho	Antecipação	Atraso	t(i)
1		1	1
2		2	2
3	8.94	3	3
4		4	4
5		5	5
6		6	6
7	28.41	7	7
8	47.86	8	8
9		9	9
0			0
9			9

Funcao_Objeto	3948.10
---------------	---------

Figura 2 – Inícios e desvios para 8 trabalhos com  $d_i = 600$

Fonte: Autor

A Figura 3 mostra o gráfico de Gantt para 9 trabalhos com as datas prometidas de 100, 150 e 250. Os trabalhos numerados entre 1 a 9 na Figura 3, correspondem, respectivamente, aos trabalhos 3, 4, 5, 6, 7, 16, 17, 18 e 19.

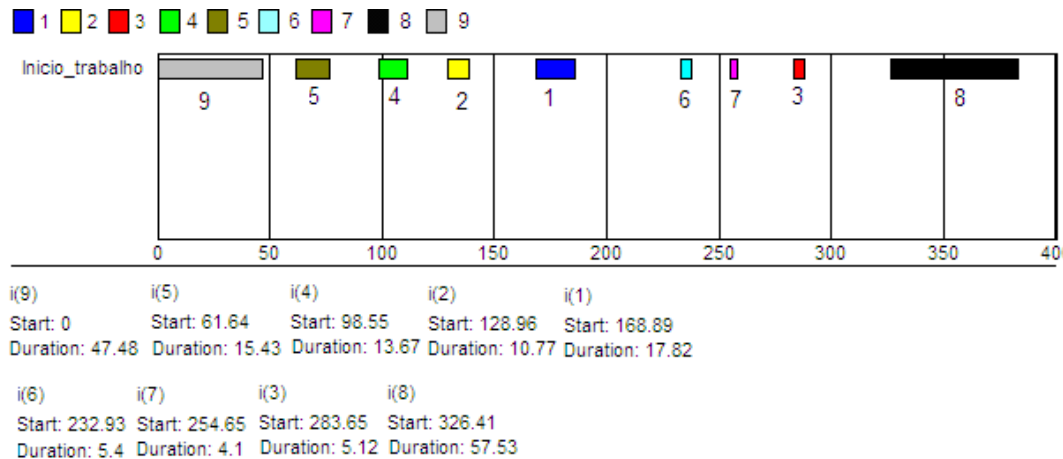


Figura 3 – Resultado para 9 trabalhos com  $d_i = 100, 150, 250$

Fonte: Autor

A Figura 4 é construída a partir das variáveis de saída do AIMMS, e mostra os resultados para os inícios dos trabalhos e os respectivos desvios (antecipações e atrasos) e os tempos de processamentos,  $t(i)$ , bem como o valor da função objetivo para 9 trabalhos.

Para tratar dos trabalhos com data prometida de 400 minutos, nota-se que o tempo de conclusão máximo ocorrerá em 383,84 minutos, ao fim do trabalho 18 (correspondente ao trabalho 8 na Figura 5). Sendo assim, para inserir os trabalhos 14 e 15, deve-se levar em consideração o tempo de *setup* do trabalho 18 para o trabalho 14 ou para o trabalho 15, dependendo da sequência que minimiza a função objetivo.

Como observado na Tabela 1, o tempo de *setup* do trabalho 18 para o trabalho 14 e 15 é o mesmo, com o valor de 21,9. O tempo de *setup* do trabalho 14 para o trabalho 15 é de 12,9 e o tempo de *setup* do trabalho 15 para o trabalho 14 é de 5,16. Os tempos de processamento dos trabalhos 14 e 15 são, respectivamente, 16,58 e 0,51.

Início trabalho	Antecipação	Atraso	t(i)
1 168.89	1	1 36.71	1 17.8
2 128.96	2 10.27	2	2 10.8
3 283.65	3	3 38.77	3 5.1
4 98.55	4	4 12.22	4 13.7
5 61.64	5 22.93	5	5 15.4
6 232.93	6 11.67	6	6 5.4
7 254.65	7	7 8.75	7 4.1
8 326.41	8	8 283.94	8 57.5
9	9 52.52	9	9 47.5
0	0	0	0
10 383.94	10	10	10

Funcao_Objetivo	21018.20
-----------------	----------

Figura 4 – Resultados para 9 trabalhos

Fonte: Autor

Dessa forma, os valores que resultarão nos desvios mínimos para os trabalhos com  $d_i = 400$ , trabalhos 14 e 15, serão formados a partir da sequência  $15 \rightarrow 14$ :

$$S_{18,15} + t_{15} = 21,9 + 0,51 = 22,41 \quad (28)$$

$$S_{18,15} + t_{15} + S_{15,14} + t_{14} = 21,9 + 0,51 + 5,16 + 16,58 = 44,15 \quad (29)$$

Como o tempo de conclusão do trabalho 18 é em 383,94, o tempo de conclusão do trabalho 15 será  $383,94 + 22,41 = 406,35$  minutos, ou seja, o trabalho 15 será entregue com atraso de 6,35 minutos. Analogamente, o tempo de conclusão do trabalho 14 será  $383,94 + 44,15 = 428,09$ , isto é, o trabalho 14 será concluído com atraso de 28,09 minutos. Dessa forma, a parcela dos custos de atraso dos trabalhos cuja data prometida é de 400 minutos é:

$$f(S) = 6,35 \cdot 20 + 28,09 \cdot 20 = 688,80 \quad (30)$$

Conclui-se que o menor valor para a soma dos desvios para 19 trabalhos é constituída por três parcelas: o referente aos trabalhos com datas prometidas de 600 min (3.498,1); aos com 100, 150 e 200 minutos (21.018,2); e aos com 400 min (688,8).”

$$f(S) = 3.498,1 + 21.018,2 + 688,80 = 25.205,1 \quad (31)$$

A sequência ótima para esse valor da função objetivo é:  $19 \rightarrow 7 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 16 \rightarrow 17 \rightarrow 5 \rightarrow 18 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 8 \rightarrow 10 \rightarrow 11 \rightarrow 9 \rightarrow 1 \rightarrow 2$

A Figura 5 apresenta o gráfico de Gantt para os 11 trabalhos (datas prometidas de 100, 150, 250 e 400 minutos).

A Figura 6 mostra os inícios dos trabalhos com seus respectivos desvios e tempos de processamentos,  $t(i)$ . Nota-se que todos os desvios dos trabalhos, menos os trabalhos 6 e 7 (que correspondem aos trabalhos 14 e 15 e possuem data prometida igual a 400 minutos) são idênticos aos desvios para 9 trabalhos como apresentado na Figura 3. Além disso, comprova-se que os desvios dos trabalhos 6 e 7 (que correspondem aos trabalhos 14 e 15) são os mesmos que os desvios calculados acima. Além disso, a soma das funções objetivo para os 9 trabalhos analisados (com datas prometidas iguais a 100, 150 e 250) e os dois trabalhos com data prometida igual à 400 minutos é igual à função objetivo para os 11 trabalhos analisados simultaneamente:

$$f(S) = 21.018,2 + 688,8 = 21.707 \quad (32)$$

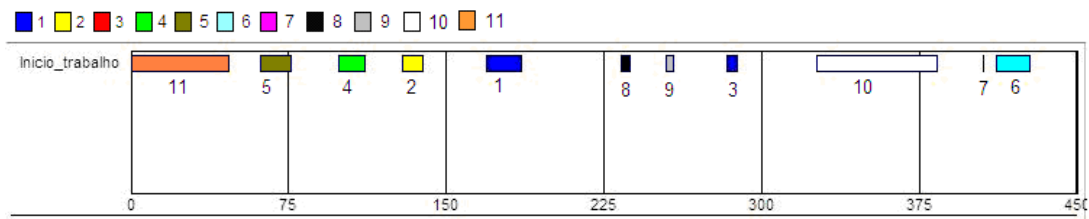


Figura 5 – Gráfico de Gantt para 11 trabalhos

Fonte: Autor

Inicio_trabalho		Antecipação		Atraso		t(i)	
1	168.89	1		1	36.71	1	17.8
2	128.96	2	10.27	2		2	10.8
3	283.65	3		3	38.77	3	5.1
4	98.55	4		4	12.22	4	13.7
5	61.64	5	22.93	5		5	15.4
6	411.51	6		6	28.09	6	16.6
7	405.84	7		7	6.35	7	0.5
8	232.93	8	11.67	8		8	5.4
9	254.65	9		9	8.75	9	4.1
10	326.41	10		10	283.94	10	57.5
11		11	52.52	11		11	47.5
0		0		0		0	
12	428.09	12		12		12	

Funcao_Objetoivo	21707.00
------------------	----------

Figura 6 – Resultados para 11 trabalhos

Fonte: Autor

Para comprovar o resultado acima, foi feita a análise dos 11 trabalhos agrupados (com datas prometidas de 100, 150, 250 e 400 minutos). Como se pode constatar na janela de resumo do *software* AIMMS, Figura 7, a função objetivo tem o valor de 21.707 (*Best Solution*).

Progress Window	
READY	
AIMMS	: Sequenciamento da Produção.amb
Executing	: MainExecution
Line number	: 3 [body]
Math.Program	: FO
# Constraints	: 203
# Variables	: 192 (156 integer)
# Nonzeros	: 823
Model Type	: MIP
Direction	: minimize
SOLVER	: CPLEX 12.3
Phase	: Postsolving
Iterations	: 82023656
Nodes	: 8246068 (Left: 3591840)
Best LP Bound	: 8885,614382 (Gap: 59.07%)
Best Solution	: 21707 (Post: na)
Solving Time	: 10976.34 sec (Peak Mem: 1207.8 Mb)
Program Status	: Integer solution
Solver Status	: Terminated by solver
Total Time	: 10977.66 sec
Memory Used	: 790.6 Mb
Memory Free	: 1416.4 Mb

Figura 7 - Resumo para resolução do problema com 11 trabalhos

Fonte: Autor

## 5. Conclusão

O modelo apresentado na Seção 3.2 mostrou-se bastante eficaz, porém limitado pela complexidade do problema à medida que o número de trabalhos aumenta. A teoria apresentada para os modelos E/T trata, em sua maioria, os problemas onde há uma data prometida comum a todos os trabalhos, não havendo a inserção de tempos ociosos entre os trabalhos. Mesmo em alguns trabalhos com datas prometidas distintas, ainda não são levados

em consideração os tempos ociosos entre os trabalhos e até nesses trabalhos já é percebida a dificuldade da busca pela solução ótima para um número de trabalhos maior que 20.

Dessa forma, o presente trabalho ilustra uma situação em que os resultados foram considerados bastante satisfatórios, atingindo-se um valor mínimo para as somas das defasagens em relação às datas prometidas, devidamente penalizados, para os 19 trabalhos, sequenciando de forma ótima todos os trabalhos.

Como mostrado em 4.1, a sequência ótima do Modelo E/T é dada pela sequência: 19 → 7 → 6 → 4 → 3 → 16 → 17 → 5 → 18 → 15 → 14 → 13 → 12 → 8 → 10 → 11 → 9 → 1 → 2

Esse sequenciamento, que leva em consideração as datas prometidas e as penalidades de defasagem, tem um tempo de conclusão máximo de 748,43 minutos e um custo total encontrado para a minimização dos desvios 25.205,1.

O problema resolvido na Seção 4.1, que é achar a sequência ótima de produção através da minimização dos desvios devidamente ponderados em relação às datas prometidas distintas para 19 trabalhos com tempos de preparação de máquina entre os trabalhos, leva às seguintes conclusões:

- A inserção de tempos ociosos entre os trabalhos é fundamental, principalmente quando as datas prometidas são demasiadamente distantes umas das outras;
- O problema pode ser subdividido em vários subproblemas de acordo com as datas prometidas, ou seja, os trabalhos orbitam em torno de datas prometidas comuns ou próximas. No entanto, deve-se observar se não ocorrerá interferência entre os trabalhos agrupados pelas datas prometidas e caso isso aconteça, um estudo mais apurado deve ser feito;
- No caso de trabalhos com datas prometidas próximas ou iguais, a sequência ótima dependerá de todos os parâmetros do sistema (tempo de processamento, tempo de *setup* e das penalidades por antecipação e atraso).
- Cabe reconhecer que em ambientes produtivos com elevada sobrecarga, os custos de oportunidade de melhorar a sequência as custas de ver a máquina ociosa pode ter suas limitações.

## 6. Referências Bibliográficas

- [1] ABDUL-RAZAQ, T. e POTTS, C. (1988), Dynamic Programming State-Space Relaxation for Single-Machine Scheduling. J. Opnl. Res. Soc. 39, 141-152.
- [2] BAKER, K. R. e SCUDDER, G. D. (1990), Sequencing with Earliness and Tardiness Penalties: A Review. Operations Research, v. 38, p. 22-36.
- [3] BUSTAMANTE, L. M. (2006), Minimização do Custo de Antecipação e Atraso para o Problema de Sequenciamento de uma Máquina com Tempo de Preparação Dependente da Sequência: Aplicação em uma Usina Siderúrgica. Dissertação de Mestrado, Programa de Pós Graduação em Engenharia de Produção, UFMG, Belo Horizonte.
- [4] GAREY, M., TARJAN, R. e WILFONG, G. (1988), One- Processor Scheduling With Symmetric Earliness and Tardiness Penalties. Math. Opns. Res. 13, 330-348.
- [5] MANNE, A. S., (1960), On the Job-shop Scheduling Problem. Operations Research, v. 8, p. 219-223.
- [6] PINEDO, M. L., (2008) Scheduling: Theory, Algorithms, and Systems. Ed. Springer, 3<sup>rd</sup> Ed.
- [7] SOUZA, M. J., (2011), Inteligência Computacional para Otimização. Notas de aula 2011/1 do Departamento de Computação, UFOP, <http://www.decom.ufop.br/prof/marcone/ Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>. acesso em 07 Out, 2011
- [8] WAGNER, H. M., (1959), An Integer Programming Model for Machine Scheduling. Naval Research Logistics Quarterly, v. 6, p. 131-140.