

Sequenciamento para ambientes *job shop* usando a heurística *shifting bottleneck* com diferentes formas de otimização.

Evandro Bittencourt (UNIVILLE) evandrobitt@gmail.com
Valdésio Benevenuto (UDESC) valdesio.benevenuto@udesc.br

Resumo:

Este artigo apresenta os resultados do sequenciamento em ambientes do tipo por processo (*job shop*) através do uso da heurística *shifting bottleneck* comparando a otimização considerando como objetivo a minimização do atraso e a minimização das somas dos atrasos. A heurística *shifting bottleneck* é uma das formas mais rápidas e eficazes na minimização do *makespan* dos sequenciamentos nos ambientes *job shop*. Ele é base para outros processos heurísticos e metaheurísticos com vantagens. Durante a solução da heurística as restrições disjuntivas que definem a sequência das tarefas nas máquinas são otimizadas incluindo-se uma máquina por vez no sequenciamento resolvendo um problema relaxado para máquina única. A inclusão das máquinas é definida considerando-se o maior impacto no *makespan* de forma a minimizar o atraso. Neste artigo consideramos a minimização da soma dos atrasos como objetivo inicial, diferentemente do usado na heurística original. Também se combinou os dois tipos de otimização. Para os testes foram utilizadas instâncias da literatura. A comparação dos resultados obtidos mostrou que a melhor situação foi usar a soma dos atrasos para a instância mt06 o que resultou no *makespan* igual a 55 unidades de tempo, ótimo para esse exemplo. Para a instância mt10 o melhor resultado encontrado foi de 971 unidades de tempo combinando-se os dois tipos de otimização.

Palavras chave: Sequenciamento, Heurística *shifting bottleneck*, Ambiente *job shop*.

Scheduling for job shop environments using shifting bottleneck heuristic with different forms of optimization.

Abstract

This article presents the results of scheduling of type environments for process (*job shop*) through the use of shifting bottleneck heuristic optimization considering comparing the objective of minimizing the delay and the minimization of the sums of delays. The shifting bottleneck heuristic is one of the fastest and most effective ways to minimize the makespan of sequencing in job shop environments. It is the basis for other heuristic and metaheuristic processes with benefits. During the heuristic solution of the disjunctive constraints that define the sequence of tasks on the machines are optimized by including a sequencing machine at a time in solving a relaxed problem for single machine. The inclusion of the machines is defined considering the greatest impact on the makespan so as to minimize the delay. In this paper we consider the minimization of the sum of the delays as initial goal, unlike the heuristics used in the original. Also combined the two types of optimization. For tests were repeated instances in the literature. The comparison of the results showed that the best situation was to use the sum of delays for instance mt06 which resulted in makespan equal to

55 time units, great for this example. For instance mt10 the best result found was 971 time units combining two types of optimization.

Key-words: Scheduling, Shifting bottleneck heuristic, Job shop.

1. Introdução

A gestão da produção envolve diversos aspectos, desde conhecer os objetivos estratégicos da produção, desenvolver a estratégia da produção propondo novos produtos, serviços e processos de produção, planejar e controlar a produção e propor melhorias num ciclo virtuoso. O resultado conforme Slack et al. (2009), é o aumento da contribuição da função produção para a corporação.

Dentre as atividades diárias da produção, o sequenciamento das tarefas (ordens de produção) impacta diretamente na utilização efetiva da capacidade do sistema de produção. Os recursos produtivos são escassos e devem ser utilizados de maneira otimizada, evitando sobrecargas e subutilização.

Segundo Pinedo (2012), o sequenciamento efetivo das operações define a sobrevivência das empresas no ambiente competitivo atual. Ainda segundo o mesmo autor, as corporações devem ter confiabilidade, cumprindo as datas de entrega firmadas com os clientes, o fracasso pode resultar em gastos desnecessários e prejuízo em temas de imagem da empresa.

Segundo Slack et al. (2009), a confiabilidade é um dos cinco objetivos da gestão da produção, sendo que o cumprimento das datas prometidas com os clientes internos e/ou externos é fundamental para ser eficaz.

O sequenciamento das operações é um dos problemas mais estudados pela Engenharia de Produção. A combinação das variáveis do sistema produtivo, em especial, o número de operações, de tarefas, de máquinas e de mão-de-obra dificulta a otimização na busca da minimização dos gastos envolvidos.

Os métodos usados para a solução dos problemas de sequenciamento depende da dificuldade envolvida. Os métodos ótimos são utilizados geralmente em problemas de baixa e média complexidade, enquanto os métodos heurísticos e metaheurísticos são utilizados, em especial, para problemas de alta complexidade (YAMADA, 2003).

O objetivo deste trabalho foi aplicar a heurística *shifting bottleneck* com variação no tipo da otimização, combinando a maior atraso como na versão original e somando-se os atrasos.

A seção 2 apresenta o problema de sequenciamento, suas representações e formas de solução.

Continuando, a seção 3 apresenta a heurística *shifting bottleneck*. Na seção 4 os materiais e métodos utilizados.

Os resultados são apresentados na seção 5. Terminando temos a conclusão do artigo e as referências.

2. O Problema do sequenciamento

Os problemas de sequenciamento podem ser tipificados considerando a relação entre máquinas e tarefas. Segundo Pinedo (2012), os modelos principais são: modelo de máquina única; modelo de máquina única avançado; modelo de máquinas paralelas; modelo *flow shop*; modelo *flow shop* flexível; modelo *job shop* e modelo *open shop*. Na subseção seguinte detalhamos o modelo *job shop*, que é o modelo utilizado neste trabalho.

2.1 Modelo *job shop*

No modelo *job shop* cada tarefa é processada num roteiro de fabricação particular. Ou seja, cada tarefa possui o seu próprio roteiro de fabricação. O modelo *job shop* é o problema que aparece quando o sistema produtivo é organizado por atividades.

O modelo *job shop* pode ser representado na forma matricial:

$$R = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 1 & 3 & 2 \end{bmatrix} \quad T = \begin{bmatrix} 3 & 3 & 2 \\ 1 & 3 & 2 \\ 1 & 1 & 3 \end{bmatrix} \quad (1)$$

Onde r_{ij} é o roteiro de fabricação, determina o número da máquina na sequência de fabricação de cada tarefa i na operação j ; e o t_{ij} é o tempo de processamento da tarefa i na máquina r_{ij} .

A representação pode ser feita também graficamente com um rede de precedência, onde as operações são os nodes e os arcos são, em primeiro lugar, as restrições conjuntivas, roteiro de fabricação das tarefas, e em segundo lugar as restrições disjuntivas, sequência de fabricação das tarefas nas máquinas (YAMADA, 2003).

O problema pode ser representado também matematicamente conforme Balas e Vazacopoulos (1998):

$$\text{Minimizar } t^* \quad (2)$$

Sujeito A:

$$t_j - t_i \geq p_i \quad (i, j) \in A, \quad (3)$$

$$t_i \geq 0, \quad i \in N, \quad (4)$$

$$t_j - t_i \geq p_i \vee t_i - t_j \geq p_j, \quad (i, j) \in E_k, \quad k \in M. \quad (5)$$

Onde:

(2) é a função objetiva;

(3) restrições conjuntivas;

(4) restrições de não negatividade;

(5) restrições disjuntivas;

$N = \{0, 1, \dots, *\}$ é o conjunto de todas as operações, 0 é a operação inicial e $*$ é a operação final;

M é o conjunto das máquinas;

A é o conjunto de restrições conjuntivas;

E_k é os pares de restrições disjuntivas de cada máquina k ;

p_j são os tempos de processamento da operação j , maior que zero;

t_j é a data de início da operação j , maior que zero.

A solução da modelagem matemática usa programação inteira para resolver as restrições disjuntivas. Conforme Seda (2007), a solução usando solvers é condicionada a quantidade de variáveis geradas na modelagem. Por exemplo, para um problema *job shop* com 3 tarefas e 3 máquinas o desenvolvimento do modelo de programação inteira exige a utilização de 20 variáveis de decisão.

O objetivo do sequenciamento é geralmente minimizar o tempo de execução de todas as operações das tarefas, ou seja minimizar o *makespan*. Quando existe a condição de datas de

entrega, o objetivo pode ser minimizar o atraso, considerando a tarefa mais atrasada ou a soma dos atrasos de todas as tarefas. Pode também aparecer pesos diferentes para as diferentes operações ou tarefas. Assim, dessa maneira a função objetivo vai ser representada com diferentes coeficientes, para cada variável de decisão.

O resultado pode ser representado por uma matriz, que é a solução do sequenciamento nas máquinas, como para o problema (1) representado por (6):

$$S = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \end{bmatrix} \quad (6)$$

Onde s_{ij} é a tarefa que é processada na máquina i na sequência j .

Graficamente a solução do sequenciamento pode ser representada num gráfico de Gantt, onde as operações aparecem sequenciadas em cada máquina.

Existem diversos métodos para resolver um problema de sequenciamento. Pode-se usar programação matemática, métodos de enumeração como o *branch and bound*, heurísticas como a *shifting bottleneck* e metaheurísticas. Também é possível encontrar soluções combinadas como a procura local e metaheurísticas como em Balas e Vazacopoulos (1998).

3. Heurística *shifting bottleneck*

Neste trabalho consideramos o nome da heurística na língua inglesa, evitando tradução, já que não existe consenso dentro da literatura portuguesa.

O heurística *shifting bottleneck* (hSB), foi proposta inicialmente por Adams, Balas e e Zawack (1988). Segundo Pinedo (2012), a hSB é um dos procedimentos para sequenciamento no sistema *job shop* mais vantajoso em termos de minimização do *makespan*.

A hSB é um processo iterativo, o sequenciamento é feito adicionando as restrições disjuntivas de cada máquina, uma por vez, escolhendo a máquina que mais afeta o *makespan* da sequência parcial. No início o sequenciamento é realizado considerando-se somente as restrições conjuntivas, baseadas no roteiro de fabricação de cada tarefa. Então, as máquinas são adicionadas considerando-se aquela que proporcina o maior impacto, ou seja, a máquina gargalo.

Após a adição de cada máquina é realizado um processo de reotimização considerando as máquinas já adicionadas. Retirando e reposicionado cada máquina se houver melhora no *makespan*.

A sequência das atividades da máquina gargalo é definida resolvendo o problema de forma relaxada, considerando o ambiente de máquina única.

Na hSB original o objetivo no problema de máquina única é a minimização do atraso, que é igual a minimização do impacto sobre o *makespan* da sequência parcial.

Como exemplo de resolução usando-se a hSB na Figura 1 tem-se o problema (1) resolvido somente com as restrições conjuntivas.

As datas são obtidas resolvendo-se a rede de precedência, assim para a operação O_{23} a primeira data de início é 1, primeira data de término igual a 4, últimas data de início igual a 3 e última data de término igual a 6. Observa-se que o *makespan* é igual a 8.

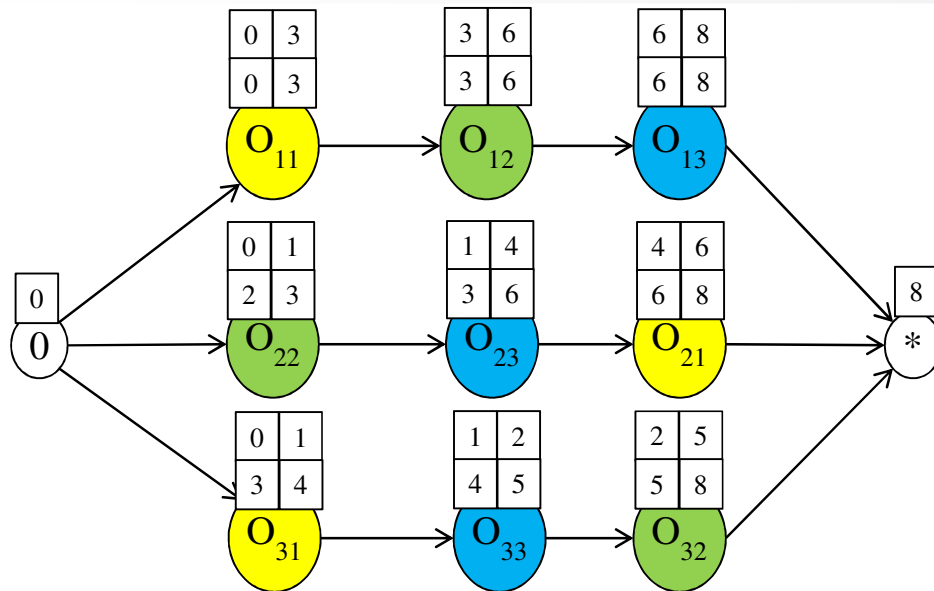


Figura 1 – Resolução do problema 1 considerando as restrições conjuntivas com a determinação das datas: primeira data de início; primeira data de término; última data de início e última data de término.

Agora é feita a introdução de cada máquina verificando-se o efeito sobre o *makespan*. Para a máquina 2 tem-se o problema de sequenciamento de máquina única conforme a Tabela 1.

Tarefas	1	2	3
r_i (cabeça)	3	0	2
p_i (tempo de processamento)	3	1	3
q_i (cauda)	2	5	0

Tabela 1 – Problema de sequenciamento de máquina única para M_2 , considerando os tempos de cabeça, processamento e cauda para cada tarefa.

A cabeça r é a primeira data de início da operação (de cada tarefa) obtida na resolução da rede de precedência (Figura 1), que pode ser definida como a data da liberação do pedido. A cauda q é calculada como a diferença entre o *makespan* e a última data de término obtida na resolução da rede de precedência (Figura 1). A data de entrega de cada operação pode ser definida como a soma dos tempos $r_i + p_i + q_i$, sendo que o atraso é a maior violação na data de entrega.

A solução encontrada para o problema da Tabela 1 é a sequência mostrada na Tabela 2.

Tarefas	2	1	3
r_i (cabeça)	0	3	2
p_i (tempo de processamento)	1	3	3
q_i (cauda)	5	2	0
s_i (início)	0	3	6
d_i (entrega)	6	8	9
a_i (atraso)	0	0	1

Tabela 2 – Melhor solução para sequenciamento considerando o menor atraso.

A data de início s_i é determinada considerando-se o término da operação anterior ou o tempo da cabeça r_i . A data de entrega, por sua vez, é determinada somando-se a data de início o tempo de processamento p_i e a cauda q_i . O atraso a_i é a diferença positiva em relação ao *makespan* anterior.

Nesse exemplo, a máquina 2 é a que causa maior impacto no *makespan*, a máquina gargalo, sendo incluída no sequenciamento conforme a Figura 2.

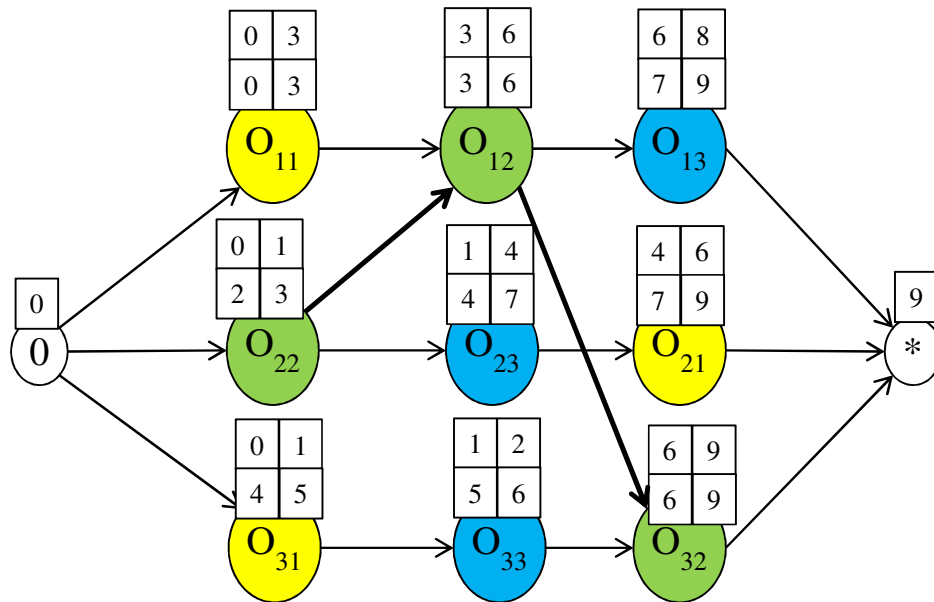


Figura 2 – Sequenciamento retultante da inclusão da máquina 2, a máquina gargalo, que gerou um impacto no *makespan* em uma unidade de tempo.

Nota-se na Figura 2 o acréscimo no *makespan*, passando para 9 (nove) unidades de tempo. Dessa maneira, cada máquina é adicionada ao sequenciamento num processo iterativo. A resolução do problema de máquina única é feito por enumeração geralmente usando-se um algoritmo *branch and bound*.

4. Materiais e métodos

O sequenciamento neste artigo foi feito utilizando-se a heurística *Shifting Bottleneck* (hSB), proposta inicialmente por Adams *et al.* (1988). Na proposta original o objetivo no problema de máquina única é a minimização do atraso, que é igual a minimização do impacto sobre o *makespan* da seqüência parcial.

Neste trabalho, o problema de máquina única foi resolvido considerando a minimização da soma dos atrasos (soma dos valores da última linha da Tabela 2, por exemplo) combinada com o procedimento original.

Como exemplos de problemas de sequenciamento foram utilizadas instâncias conhecidas na literatura. Esses problemas foram testados por diversos pesquisadores, o que facilita a comparação dos resultados obtidos.

Na Tabela 3 o problema mt06 de Muth e Thompson (1963), com *makespan* ótimo igual a 55 unidades de tempo, considerando medianamente difícil (YAMADA, 2003).

Na Tabela 4 o problema mt10 de Muth e Thompson (1963), com *makespan* ótimo de 930 unidades de tempo. Problema extremamente difícil, se manteve sem solução por 3 décadas (YAMADA, 2003).

Tarefas	Operações					
	1	2	3	4	5	6
1	3(1)	1(3)	2(6)	4(7)	6(3)	5(6)
2	2(8)	3(5)	5(10)	6(10)	1(10)	4(4)
3	3(5)	4(4)	6(8)	1(9)	2(1)	5(7)
4	2(5)	1(5)	3(5)	4(3)	5(8)	6(9)
5	3(9)	2(3)	5(5)	6(4)	1(3)	4(1)
6	2(3)	4(3)	6(9)	1(10)	5(4)	3(1)

Fonte: Adaptado de Muth e Thompson (1963)

Tabela 3 – Problema de sequenciamento mt06, 6 tarefas *versus* 6 máquinas, para cada tarefa a sequência das máquinas e tempo de processamento entre parênteses.

Tarefas	Operações									
	1	2	3	4	5	6	7	8	9	10
1	1(29)	2(78)	3(9)	4(36)	5(49)	6(11)	7(62)	8(56)	9(44)	10(21)
2	1(43)	3(90)	5(75)	10(11)	4(69)	2(28)	7(46)	6(46)	8(72)	9(30)
3	2(91)	1(85)	4(39)	3(74)	9(90)	6(10)	8(12)	7(89)	10(45)	5(33)
4	2(81)	3(95)	1(71)	5(99)	7(9)	9(52)	8(85)	4(98)	10(22)	6(43)
5	3(14)	1(6)	2(22)	6(61)	4(26)	5(69)	9(21)	8(49)	10(72)	7(53)
6	3(84)	2(2)	6(52)	4(95)	9(48)	10(72)	1(47)	7(65)	5(6)	8(25)
7	2(46)	1(37)	4(61)	3(13)	7(32)	6(21)	10(32)	9(89)	8(30)	5(55)
8	3(31)	1(86)	2(46)	6(74)	5(32)	7(88)	9(19)	10(48)	8(36)	4(79)
9	1(76)	2(69)	4(76)	6(51)	3(85)	10(11)	7(40)	8(89)	5(26)	9(74)
10	2(85)	1(13)	3(61)	7(7)	9(64)	10(76)	6(47)	4(52)	5(90)	8(45)

Fonte: Adaptado de Muth e Thompson (1963)

Tabela 4 – Problema de sequenciamento mt10, 10 tarefas *versus* 10 máquinas, para cada tarefa a sequência das máquinas e tempo de processamento entre parênteses.

Para solução do problema de máquina única, subproblema da resolução da hSB, no caso da minimização do atraso foi utilizada um algoritmo *branch and bound* proposta por Carlier e Pinson (1989) conforme Yamada (2003). Para a consideração da minimização da soma dos atrasos foi desenvolvida um algoritmo *branch and bound* próprio. Os algoritmos para sequenciamento foram desenvolvidos em linguagem C++ compilados no programa Dev-C++ 5.4.2 com compilador TDM-GCC 4.7.1 versão 32-bit. O microcomputador usado para rodar o programa foi um Notebook Acer com Intel Core i5.

5. Resultados obtidos

Os resultados dos sequenciamentos realizados são apresentados na Tabela 5 para o problema mt06 e na Tabela 6 para o problema mt10.

A Tabelas (5 e 6) apresentam os resultados do *makespan* obtido com o uso da otimização considerando o maior atraso (Tipo 1), a otimização considerando a soma dos atrasos (Tipo 2) e ainda a combinação dos dois tipos considerando início com Tipo 2 e a partir da inclusão de determinada máquina o Tipo 1. A reotimização foi feita considerando o Tipo 1 como objetivo, ou seja, minimização do *makespan*.

Os resultados indicaram que o objetivo Tipo 2 minimiza o *makespan* atingindo a solução ótima para a instância mt06. Também da mesma forma combinado-se Tipo 2 e Tipo 1 na inclusão da sexta máquina.

Testes	Tipo	Tipo 1	Tipo	MakeSpan
	Otimização	A partir da máquina	Reotimização	
1	1	-	1	57
2	2	-	1	55
3	2	2	1	58
4	2	3	1	61
5	2	4	1	55
6	2	5	1	58
7	2	6	1	55

Tabela 5 – Resultado do *makespan* no sequenciamento do problema mt06, considerando objetivo Tipo 1 – minimização do atraso e Tipo 2 minimização da soma dos atrasos no problema de máquina única.

A Tabela 6 apresenta os resultados do *makespan* obtido com as combinações do uso do objetivo Tipo 1 e Tipo 2, para a instância mt10.

Testes	Tipo	Tipo 1	Tipo	MakeSpan
	Otimização	A partir da máquina	Reotimização	
1	1	-	1	1092
2	2	-	1	988
3	2	2	1	1113
4	2	3	1	1113
5	2	4	1	1044
6	2	5	1	971
7	2	6	1	977
8	2	7	1	988
9	2	8	1	988
10	2	9	1	988
11	2	10	1	988

Tabela 6 – Resultado do *makespan* no sequenciamento do problema mt10, considerando objetivo Tipo 1 – minimização do atraso e Tipo 2 minimização da soma dos atrasos no problema de máquina única.

Os resultados indicaram que combinar os objetivos é a melhor maneira de se realizar o sequenciamento, fazendo Tipo 2 nas 4 primeiras máquina e Tipo 1 nas demais inclusões, atingindo um *makespan* igual a 971, acima do ótimo que é 930, mas aceitável visto que o tempo de processamento é menor que 1 minuto.

5. Conclusão

Este trabalho estudou a aplicação da heurística *shifting bottleneck* no ambiente *job shop*. A proposta foi baseada na variação do objetivo da otimização, considerando Tipo 1 como minimização do atraso e Tipo 2 como minimização da soma dos atrasos. Os resultados indicaram que o uso da otimização Tipo 2 resultou no *makespan* ótimo para a instância mt06, 6 tarefas e 6 máquinas. Sendo que para o problema mt10, o melhor resultado foi alcançado com a combinação da otimização Tipo 2 no início do processo, e o Tipo 1 a partir da inclusão da quinta máquina, atingindo um *makespan* igual a 971 unidades de tempo.

Todas as combinações, ou utilizações puras, do Tipo 1 e Tipo 2 na otimização consumiram um tempo computacional menor que 1 minuto, mostrando a viabilidade do uso da heurística *shifting bottleneck*.

Referências

ADAMS, J.; BALAS E.; ZAWACK, D. *The Shifting Bottleneck Procedure for Job Shop Scheduling*. Management Science 3, 391–401. 1988

BALAS, E.; VAZACOPOULOS, A. *Guided Local Search With Shifting Bottleneck for Job Shop Scheduling.* Management Science, Vol. 44. No.2, pp. 262-275, 1998.

CARLIER, J.; PINSON, E. *An algorithm for solving the job-shop problem.* Management Science, Vol. 35, No. 2, pp. 164–176, 1989.

MUTH, J. F.; THOMPSON, G. L. *Industrial Scheduling.* Englewood Cliffs. New Jersey: Prentice-Hall, 1963.

SEDA, M. *Mathematical Models of Flow Shop and Job Shop Scheduling Problems.* International Journal of Applied Mathematics and Computer Sciences, Vol. 4, No.4, pp 241-246, 2008.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção.* 3. Ed. São Paulo: Atlas, 2009.

PINEDO, M. P. *Scheduling: Theory, Algorithms and Systems.* 4. Ed. New York: Springer, 2012.

YAMADA, T. *Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems.* PhD. Thesis. Kyoto University. Kyoto, Japan, 2003.