

Sequenciamento para ambientes *job shop* usando de regras de despacho combinadas resolvido com algoritmo *branch and bound*.

Evandro Bittencourt (UNIVILLE) evandrobitt@gmail.com
Valdésio Benevenuti (UDESC) valdesio.benevenuti@udesc.br

Resumo:

Este artigo apresenta os resultados do sequenciamento em ambientes do tipo por processo (*job shop*) através do uso de regras de despacho combinadas. A utilização de regras de despacho é a maneira mais rápida de sequenciar tarefas num ambiente por processo, onde existe a combinação de diversas tarefas com diversas máquinas, ou centro de trabalhos. Apesar da facilidade, o resultado do sequenciamento usando regras de despacho deixa a desejar, em especial, quando o objetivo é realizar todas as tarefas em menor tempo, ou seja minimizar o *makespan*. Por outro lado, a enumeração de todas as alternativas demanda longo intervalo de tempo computacional, devido a razão combinatória dos problemas de sequenciamento. Dessa maneira, combinou-se o uso das regras de despacho, duas a duas numa heurística enumerativa, utilizando-se um algoritmo *Branch and Bound*. Para os testes foram utilizadas instâncias da literatura. A comparação dos resultados obtidos mostrou que a melhor situação foi usar a regra de despacho menor tempo de processamento combinada com a regra menor data final. Para a instância mt06 o resultado encontrado foi o ótimo, *makespan* igual a 55 unidades de tempo. Para a instância mt10 o resultado encontrado foi 954 unidades de tempo.

Palavras chave: Sequenciamento, Regras de despacho, Algoritmo *Branch and Bound*.

Sequencing for job shop environments using dispatching rules combined solved with the branch and bound algorithm.

Abstract

This article presents the results of sequencing the type of process (job shop) through the use of rules combined dispatch environments. The use of dispatching rules is the fastest way to sequence tasks in a process environment, where there is a combination of several tasks with different machines, or work center. Despite the ease, the result of sequencing using dispatching rules is weak, especially when the goal is to accomplish all the tasks in shorter time, ie minimize the *makespan*. Furthermore, enumerating all alternatives demand long period of computational time because of problems due to the combinatorial sequencing. Thus, the combined use of dispatching rules two by two in an enumerative heuristics using a Branch and Bound algorithm. For tests were repeated instances in the literature. The comparison of the results showed that the best situation was to use the rule of order less processing time rule combined with the lower end date. For instance mt06 the result found was great, *makespan* equal to 55 time units. For instance mt10 the result found was 954 time units.

Key-words: Scheduling, Dispatching rules, Branch and Bound algorithm.

1. Introdução

O sequenciamento de tarefas (ordens de produção) é uma das atividades diárias da gestão da produção. O objetivo de sequenciamento é conciliar as necessidades de produção de produtos ou serviços com os recursos disponíveis no sistema produtivo. No ambiente competitivo atual, a sobrevivência das empresas depende de sequenciamento efetivo das operações (PINEDO, 2012). 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 termos de imagem da empresa.

Manter a confiabilidade é fundamental para as empresas participarem do mercado atuante. Segundo Slack et al. (2009), a confiabilidade é um dos cinco objetivos da gestão da produção, sendo que o cumprimento das datas com os clientes internos e externos é fundamental para ser eficaz.

Devido a importância da sequência das atividades para o planejamento da produção o problema de sequenciamento é um dos mais estudados na engenharia de produção. A dificuldade do problema de sequenciamento depende do conjunto de variáveis envolvidas. Por um lado, a quantidade e condições das tarefas, por outro lado a quantidade e condições das máquinas (centros de trabalho), e ainda a maneira como tarefas e máquinas se relacionam. Essas correlações entre tarefas e máquinas fazem com que as soluções possíveis, para o problema de sequenciamento, sejam resultado de uma combinação. Segundo Pinedo (2012), os problemas de sequenciamento de uma maneira geral podem ser tipificados em determinísticos ou probabilísticos, com vários subtipos dependendo da combinação das tarefas versus máquinas.

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

O objetivo deste trabalho foi aplicar a combinação das regras de despacho para solucionar o problema de sequenciamento no sistema por processo envolvendo n tarefas e n máquinas. Para isso, foi utilizado um algoritmo enumerativo baseado na heurística *branch and bound* (algoritmo B&B).

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

Continuando, a seção 3 apresenta as ferramentas utilizadas bem como o algoritmo B&B utilizado.

Os resultados são apresentados na seção 4. 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*.

As variáveis envolvidas no problema de sequenciamento caracterizam as máquinas e tarefas envolvidas e a forma que as máquinas se correlacionam com as tarefas. Entre as variáveis pode-se citar: roteiro de fabricação de cada tarefa, ou seja a sequência das operações de cada tarefa nas máquinas; o tempo de processamento de cada operação das tarefas em cada máquina; a data de liberação de cada operação das tarefas em cada máquina; a data de entrega

prometida de cada operação das tarefas em cada máquina e também o peso de cada operação das tarefas traduzindo a importância para o processo.

Os modelos principais: modelo de máquina única, modelo *flow shop* e modelo *job shop* serão detalhados nas subseções seguintes.

2.1 Modelo de máquina única

É o modelo mais simples, pois as possibilidades de sequenciamento ficam condicionadas apenas a sequência das tarefas. Apesar disso, segundo Pinedo (2012), os modelos de máquina única são importantes por várias razões: pode ser um caso especial de todos os outros modelos e pode auxiliar a resolução relaxada de outros modelos. A Tabela 1 é um exemplo de um problema de máquina única, onde as quatro tarefas são caracterizadas pelo tempo de processamento (p_j), pela data de liberação (r_j) e pela data de entrega prometida (d_j).

O objetivo da solução é definir a ordem de produção das tarefas minimizando a soma dos tempos de atraso, considerando como a diferença entre a data de término de processamento de cada tarefa menos a data de entrega prometida.

Tarefas	1	2	3	4
p_j	3(1)	1(3)	2(6)	4(7)
r_j	2(8)	3(5)	5(10)	6(10)
d_j	3(5)	4(4)	6(8)	1(9)

Fonte: Adaptado de Pinedo (2012)

Tabela 1 – Problema de sequenciamento no modelo de máquina única contendo 4 tarefas.

2.2 Modelo *flow shop*

O modelo *flow shop* se aplica ao sequenciamento nos sistemas por produto, onde as diversas tarefas possuem um roteiro de fabricação igual, ou seja, são realizadas numa linha de produção onde as máquinas são posicionadas numa ordem específica.

Segundo Slack et al.(2009), as linhas de produção foram introduzidas com o advento da produção em massa, com objetivo de acelerar o ritmo, diminuindo o tempo de execução de produtos e serviços. Em contra partida, o sistema por produto limita a quantidade de bens e serviços que podem ser processados.

2.3 Modelo *job shop*

O modelo *job shop* envolve tarefas que são processadas num roteiro de fabricação individualizado. Ou seja, cada tarefa possui o seu roteiro de fabricação. O sequenciamento das atividades no sistema de produção por processo geralmente é resolvido pelo modelo *job shop*.

O problema de sequenciamento pode ser representado matricialmente considerando as tarefas a serem executadas com o roteiro de fabricação de cada tarefa, especificando a máquina e o tempo de processamento. Dessa maneira, exemplificando um problema envolvendo três tarefas e três máquinas:

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

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} .

Conforme (1) o roteiro de fabricação da tarefa envolve o processamento na máquina 1 num tempo de necessário igual a 3, após o processamento na máquina 2 com tempo de 3, e termina o processamento na máquina 3 com um tempo de 2 unidades de tempo. Como já foi dito,

nesse modelo cada tarefa tem um roteiro diferente, configurando um sistema produtivo por processo, do tipo *job shop*. A complexidade do problema de sequenciamento para o modelo *job shop* aumenta com o número de tarefas e máquinas.

Outra forma de representar o modelo *job shop* é graficamente na forma de uma rede de precedência conforme a Figura 1, que representa o problema (1). Cada operação das tarefas é um evento na rede de precedência, além disso, tem-se o evento inicial 0, e o evento final *. Os arcos conjuntivos são as restrições impostas pelo roteiro de fabricação de cada tarefa. Por outro lado, os arcos disjuntivos geram as possibilidades combinatórias de sequenciamento numa determinada máquina. A definição do direcionamento dos arcos disjuntivos resulta na sequência das operações numa determinada máquina. Para melhorar a visualização da Figura 1, foi utilizada a cor amarela para representar a máquina 1, as operações de cada tarefa nessas máquinas e os arcos disjuntivos. Da mesma maneira, a cor verde foi utilizada para representar a máquina 2, e a cor azul a máquina 3.

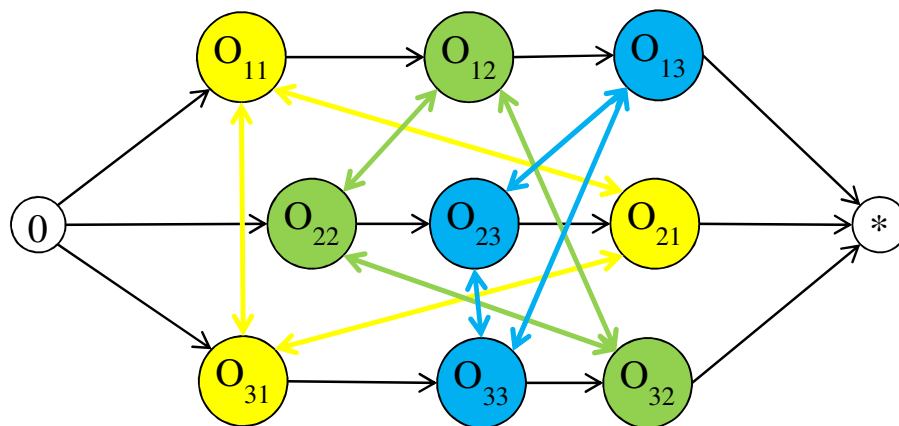


Figura 1 – Representação gráfica do modelo job shop com arcos conjuntivos e arcos disjuntivos

O resultado do sequenciamento é uma matriz que indica a sequência das atividades nas máquinas envolvidas. O objetivo do sequenciamento pode ser, entre outros, o do mínimo tempo para executar todas as tarefas, ou seja minimizar o *makespan*.

Um resultado ótimo para o problema (1) representado pelo gráfico da Figura 1 pode ser o seguinte:

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

A matriz solução S representa o sequenciamento nas máquinas, dessa maneira s_{ij} é a tarefa que é processada na máquina i na sequência j .

A solução do sequenciamento pode também ser representada graficamente com um gráfico de Gantt. Para a solução (2) do problema (1) a representação gráfica pode ser visualizada na Figura 2.

A representação gráfica da solução com um gráfico permite a visualização caracterizada das restrições impostas num problema de sequenciamento *job shop*. Por exemplo, apesar da máquina 2 estar disponível após o processamento da tarefa 2, a próxima tarefa, a tarefa 1, é feita somente no tempo 3, pois essa é a data de término do processamento da tarefa 1 na máquina anterior, máquina 1.

A solução (2) representada na Figura 2 para o problema (1) não é a única solução ótima

(*makespan* = 9) para o exemplo. Geralmente a combinação de tarefas e máquinas gera vários sequenciamentos ótimos.

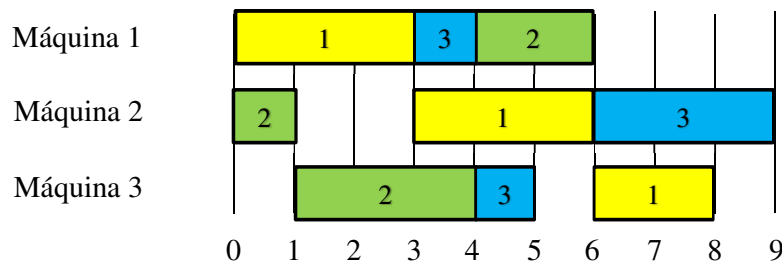


Figura 2 – Representação gráfica da solução (2) do problema (1).

Existem diversos métodos para resolver um problema de sequenciamento. Os métodos exatos utilizando-se Programação Mista ou algoritmos de enumeração como o *branch and bound*. As heurísticas aproximadas usando regras de despacho ou o algoritmo de transferência de gargalo. Existem ainda os métodos baseados em metaheurísticas, como por exemplo Algoritmo Genético, Busca Tabu, Recozimento Simulado, Colônia de Formigas, Colônia Artificial de Abelhas, entre outros. Muitas vezes existem aplicações híbridas, com métodos metaheurísticos e de refinamento baseado em procura local com enumeração.

Nos eventos brasileiros de engenharia de produção recentes pode-se destacar os seguintes trabalhos: Rocha et al. (2012) que usou programação mista para minimizar o *makespan* em um sistema por batelada; Rotta e Coelho (2012) que usaram programação linear para minimizar o *makespan* de projetos; Andrade et al. (2012), que usaram GRASP para ambiente de máquinas paralelas idênticas minimizando o *makespan*; Silva Neto et al. (2012), que usaram a heurística de Johnson para minimizar o *makespan* em ambiente *flow shop*; Ribeiro et al. (2012), que usaram algoritmo genético para minimizar o *makespan* em ambiente *job shop*; Ferreira et al. (2013), que usaram algoritmo genético para minimizar o atraso em ambiente de máquina única; Etcheverry e Anzanello (2013), que usaram busca tabu para minimizar o atraso ponderado em ambiente de duas máquinas; Frascitati e Tavares Neto (2013), que usaram a metaheurística colônia de formigas para minimizar o *makespan* em ambiente de máquina única e Pereira e Barbosa (2013), que usaram algoritmo genético mais busca local para minimizar o *makespan* de ambiente *flow shop*.

3. Materiais e métodos

O sequenciamento neste artigo foi feito utilizando-se o algoritmo proposto por Giffler e Thompson (1960) (aqui denominado como algoritmo GT), que produz sequências ativas. O sequenciamento ativo é aquele sem atraso direto, ou seja, sem o atraso que possa ser eliminado com uma inversão simples das operações no sequenciamento das máquinas. Conforme Yamada (2003), a heurística GT é base para outras heurísticas e metaheurísticas de sequenciamento no sistema *job shop*.

A Figura 3 representa o momento da escolha da próxima operação a ser sequenciada usando-se o algoritmo GT. A primeira decisão é escolher a operação com menor tempo de término, no exemplo a operação O_{11} , isso define a máquina 1, agora escolhe-se aleatoriamente entre as operações sequenciáveis que são produzidas na máquina 1 e que possuem tempo de início menor que o tempo de término da operação O_{11} . Após a escolha da operação, no exemplo O_{31} , os tempos das demais operações na máquina 1 são revistos, no exemplo para O_{11} e O_{51} , e a próxima operação da tarefa 3, O_{32} no exemplo, entra na escolha da próxima iteração.

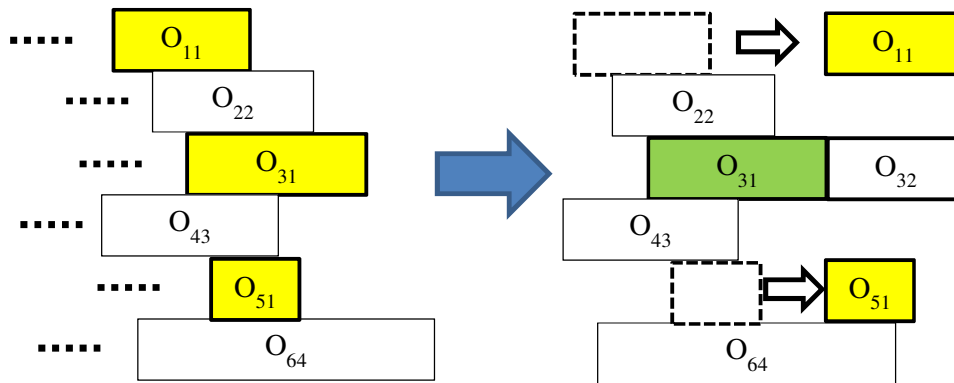


Figura 3 – Exemplo de sequenciamento através do algoritmo Giffler e Thompson, as operações O_{11} , O_{22} , O_{31} , O_{43} , O_{51} , e O_{64} ($O_{\text{Tarefa Máquina}}$) são as próximas sequenciáveis das 6 tarefas. A tarefa O_{11} define a máquina 1, e a escolha aleatória recai sobre a operação O_{31} .

Neste trabalho, a escolha aleatória do algoritmo Giffler e Thompson foi alterada usando-se as seguintes regras de despacho:

- menor operação (no exemplo da Figura 1 operação O_{51});
- maior operação (no exemplo da Figura 1 operação O_{31});
- menor data inicial (no exemplo da Figura 1 operação O_{11});
- maior data inicial (no exemplo da Figura 1 operação O_{51});
- menor data final (no exemplo da Figura 1 operação O_{11});
- maior data final (no exemplo da Figura 1 operação O_{31}).

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 2 o problema mt06 de Muth e Thompson (1963), com *makespan* ótimo igual a 55 unidades de tempo, considerando medianamente difícil (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 2 – Problema de sequenciamento mt06, 6 tarefas *versus* 6 máquinas, para cada tarefa a sequência das máquinas e tempo de processamento entre parênteses.

Na Tabela 3 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).

Para testar as combinações das regras de despacho num processo enumerativo foi feito um algoritmo *branch and bound* baseado na heurística GT. Durante a formação do

sequenciamento, o *makespan* da opção, foi sendo avaliado considerando a data final da última operação sequenciada mais o tempo somado das demais operações ainda não sequenciadas naquela máquina. Essa avaliação parcial do *makespan* permitiu a poda das alternativas durante o algoritmo B&B.

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

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.

4. Resultados obtidos

Os resultados dos sequenciamentos realizados são apresentados na Tabela 3 para o problema mt06 e na Tabela 4 para o problema mt10. A diagonal principal das tabelas (valores entre colchetes) indica o *makespan* para as regras de despacho usadas isoladamente. Para algumas combinações de regras de despacho a enumeração de todas as possibilidades através de algoritmo *branch and bound* consumiu mais tempo computacional. Neste trabalho o objetivo foi o levantamento do resultado em termos de *makespan*, não sendo avaliado o tempo de processamento.

A Tabela 4 apresenta os resultados do *makespan* obtido com as combinações das regras de despacho, duas a duas, para a instância mt06.

Regras	Regras					
	1	2	3	4	5	6
1	[85]	55	58	60	55	85
2	55	[88]	55	61	78	59
3	58	55	[63]	57	58	62
4	60	61	57	[68]	57	66
5	55	78	58	57	[84]	55
6	85	59	62	66	55	[88]

Fonte: Primária (2014)

Tabela 4 – Resultado do *makespan* no sequenciamento do problema mt06, considerando o uso das 6 regras de despacho isoladas e em combinação duas a duas.

Os melhores resultados da Tabela 4 (em negrito) foram obtidos com a combinação das regras de despacho 1+2; 2+3; 1+5 e 5+6, que atingiram o *makespan* ótimo igual a 55 unidades de tempo. O tempo de processamento para todas as combinações das regras de sequenciamento foi menor que 1 segundo.

A Tabela 5 apresenta os resultados do *makespan* obtido com as combinações das regras de

despacho, duas a duas, para a instância mt10.

Regras	1	2	Regras	3	4	5	6
1	[1424]	1011	987	1110	954	1271	
2	1101	[1429]	999	1117	1112	1101	
3	987	999	[1308]	1018	984	1003	
4	1110	1117	1018	[1534]	1018	1076	
5	954	1112	984	1018	[1124]	1028	
6	1271	1101	1103	1076	1028	[1451]	

Fonte: Primária (2014)

Tabela 5 – Resultado do *makespan* no sequenciamento do problema mt10, considerando o uso das 6 regras de despacho isoladas e em combinação duas a duas.

Utilizando as regras de despacho individuais, o melhor resultado para a instância mt10 foi a regra número 5, ou seja, despachar sempre a operação com data de término menor. O melhor resultado, considerando a combinação das regras de despacho da Tabela 5 (em negrito), foi a combinação 1 e 5 que resultou num *makespan* igual a 954. Comparando com o tempo ótimo, mínimo *makespan* igual a 930, o resultado ficou 24 unidades de tempo maior. Comparando-se com outras técnicas o resultado é melhor. Por exemplo, a solução encontrada por Yamada (2003) foi de 965 com o uso de metaheurística algoritmo genético simples.

5. Conclusão

Este trabalho estudou a aplicação das regras de despacho em sequenciamentos no ambiente *job shop*. A utilização das regras de despacho foi feita isoladamente e combinando duas a duas. Os testes foram realizados em problemas da literatura. A enumeração de todas as possibilidades de sequenciamento na combinação das regras de despacho foi feita através de algoritmo enumerativo tipo *branch and bound*.

Os resultados indicaram que a combinação das regras de despacho foi suficiente para conseguir sequenciamentos ótimos no caso do problema mt06, 6 tarefas e 6 máquinas. Para o problema mt10, o melhor resultado foi alcançado com a combinação das regras 1 e 5, ou seja despachando as operações com menor tempo e menor data final.

Os processos enumerativos, consideravelmente, consomem mais tempo computacional, mas, com o aumento da velocidade de processamento dos microcomputadores de vários núcleos, esses métodos podem ser uma alternativa na solução dos problemas de sequenciamento.

Referências

ANDRADE, G. P.; SIQUEIRA, E. C.; SOUZA, S. R.; FRANCA FILHO, M. F.; VITOR, J. F. A. *Um Estudo da Solução do Problema de Agendamento de Cirurgias Eletivas Utilizando Metaheurísticas*. ENEGEP 2012 – Encontro Nacional de Engenharia de Produção, 2012.

ETCHEVERRY, G. V.; ANZANELLO, M. *Sequenciamento em Máquinas Paralelas com Tempos de Setup Dependentes da Sequência*. ENEGEP 2013 – Encontro Nacional de Engenharia de Produção, 2013.

FERREIRA, R. J. P.; SOUZA, J. M. S.; FREJ, E. A. *Algoritmo Genético Multi-Objetivo para o Sequenciamento Integrado de Atividades de Produção e Manutenção*. ENEGEP 2013 – Encontro Nacional de Engenharia de Produção, 2013.

FRASCATI, G.; TAVARES NETO, R. F. *Aplicação da Meta-heurística Ant Colony Optimization para Eliminar Atrasos na Programação de Tarefas em Ambientes de Máquina Única*. SIMPEP 2013 – Simpósio de Engenharia de Produção, 2013.

GIFFLER, B.; THOMPSON, G. L. *Algorithms for solving production scheduling problems*. Operations Research, Vol. 8, pp. 487–503, 1960.

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

RIBEIRO, S. A. ; AHONEN, H. T.; ALVARENGA, A. G. *Sistema imune artificial para o problema de escalonamento job shop.* SIMPEP 2012 – Simpósio de Engenharia de Produção, 2012.

ROTTA I. S.; COELHO, W. D. P. *Um estudo do seqüenciamento de uma produção por encomenda em uma empresa do setor metal mecânico.* CONBREPPO 2012 – Congresso Brasileiro de Engenharia de Produção, 2012.

ROCHA R. P.; ROCHA, F. W.; CAMPOS, R. M.; JORGE, L. M. M.; PARAÍSO, P. R. *Um Estudo Comparativo do Excel/VBA e do GAMS para Minimização do Makespan em um Sistema de Produção por Batelada (SPPB).* CONBREPPO 2012 – Congresso Brasileiro de Engenharia de Produção, 2012.

SILVA NETO, J. P. ; LIMA, M. F.; RAVETTI, M. G. *Montagem de Cargas e Sequenciamento de Caminhões em um Centro de Distribuição.* SIMPEP 2012 – Simpósio de Engenharia de Produção, 2012.

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

PEREIRA, A. A. S.; BARBOSA, H. J. C. *Algoritmo Genético com Busca Local para Programação de Produção em Sistema Flow Shop Flexível com Tempos de Preparação Dependentes.* SIMPEP 2013 – Simpósio de Engenharia de Produção, 2013.

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.