

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

Guilherme Vazquez Etcheverry

**PROGRAMAÇÃO DE TAREFAS EM
MÁQUINAS PARALELAS NÃO-
RELACIONADAS COM TEMPOS DE
SETUP DEPENDENTES DA SEQUÊNCIA**

Porto Alegre

2012

Guilherme Vazquez Etcheverry

**Programação de Tarefas em Máquinas Paralelas Não-relacionadas com Tempos de
setup Dependentes da Sequência**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul como requisito parcial à obtenção do título de Mestre em Engenharia de Produção, modalidade Profissional, na área de concentração em Sistemas de Produção.

Orientador: Prof. Michel José Anzanello, PhD.

Porto Alegre

2012

Guilherme Vazquez Etcheverry

**Programação de Tarefas em Máquinas Paralelas Não-relacionadas com Tempos de
setup Dependentes da Sequência**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Produção na modalidade Profissional e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul.

Prof. Michel José Anzanello, PhD

Orientador PPGEP / UFRGS

Prof^a. Carla S. Ten Caten, Dr.

Coordenador PPGEP / UFRGS

Banca Examinadora:

Prof. Flávio S. Fogliatto, PhD (PPGEP/UFRGS)

Prof^a. Giovana S. Pasa, Dra. (PPGEP/UFRGS)

Prof. Miguel Sellitto, Dr. (PPGEPS/UNISINOS)

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Michel José Anzanello, pela paciência, disponibilidade, dedicação e pelo conhecimento transmitido. Os inúmeros encontros e discussões foram muito proveitosos e de fundamental importância para a realização desta dissertação de mestrado.

Gostaria de expressar meus agradecimentos aos professores do Departamento de Engenharia de Produção e Transportes (DEPROT) pelos conteúdos ensinados e aos funcionários do DEPROT pela cordialidade e prestatividade sempre a mim dedicadas.

Aos colegas da EPCOS do Brasil Ltda., pelo apoio e incentivo ao longo de toda a jornada, desde o período das aulas até a elaboração dos artigos e desta dissertação.

Aos familiares e amigos pelo incessante incentivo e pela compreensão de que os momentos de minha ausência estavam sendo dedicados à conquista de um objetivo pessoal e profissional.

ETCHEVERRY, G. V. *Programação de tarefas em máquinas paralelas não-relacionadas com tempos de setup dependentes da sequência* 2012. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Rio Grande do Sul, Brasil.

RESUMO

A concorrência nos mercados mundiais impõe a necessidade de aumento da competitividade das empresas que desejam assumir posições de liderança nos segmentos em que atuam. Neste ínterim, a programação de tarefas contribui para que as empresas promovam a eficiente utilização dos recursos produtivos visando a realização de seus objetivos estratégicos. Esta dissertação enfoca a programação de tarefas em máquinas paralelas não-relacionadas e com tempos de *setup* dependentes da sequência de processamento. Primeiramente é abordado o objetivo de minimização do atraso total e do tempo total para a conclusão de um conjunto de tarefas, através de uma heurística de três etapas que (i) ordena as tarefas pelo *WSPT* (*Weighted Shortest Processing Time*), (ii) aloca as tarefas às máquinas e (iii) aprimora a solução proposta pela etapa (ii) através de *Tabu Search*. Quando aplicada em um ambiente de manufatura real composto por duas máquinas paralelas não-relacionadas no processo de metalização de filmes plásticos em alto vácuo, a heurística resulta em um desvio de 1,1% para o tempo total de processamento das tarefas e 4,6% para o atraso total, em comparação ao resultado ótimo obtido por enumeração. Na sequência, o objetivo passa a ser a minimização simultânea do atraso e do adiantamento das tarefas através de uma heurística de três etapas que (i) caracteriza o conjunto de tarefas por um conjunto de métricas, (ii) aloca as tarefas às máquinas através de uma versão modificada do *ATCS* (*Apparent Tardiness Cost with Setup*) de Lee e Pinedo (1997), e (iii) aprimora a solução final com *Tabu Search*. A aplicação em dados reais resulta em 14% de desvio em relação à solução ótima obtida por enumeração. Quando aplicada em cenários com data de entrega, tempos de processamento e *setup* simulados, a heurística resulta em desvio médio de 18% da solução ótima gerada por enumeração para pelo menos 70% das simulações.

Palavras-chave: programação de tarefas, máquinas paralelas não-relacionadas, tempos de *setup*, adiantamento, atraso

ETCHEVERRY, G. V. *Programação de tarefas em máquinas paralelas não-relacionadas com tempos de setup dependentes da sequência* 2012. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Rio Grande do Sul, Brasil.

ABSTRACT

The competition in worldwide markets lead the companies to increase the competitiveness in order to take leading positions in their industries. In this sense, scheduling plays an important role leading the companies to reach their strategic goals through efficient utilization of manufacturing resources. This dissertation focuses on the scheduling unrelated parallel machines with sequence dependent setup times. First goal is to minimize the completion time and total weighted tardiness, through a three phase heuristic which (i) sort the jobs with *WSPT*, (ii) allocate the jobs to the machines and (iii) improve final solution with Tabu Search. Once applied to a real manufacturing environment composed by two unrelated parallel machines, in high vacuum plastic films metallisation process, the heuristic results in 1.1% of deviation from total weighted completion time and 4.6% of deviation from weighted tardiness, in relation to the optimal solution obtained from total enumeration. Next goal is the simultaneous minimization of weighted earliness and tardiness, through a three phase heuristic which (i) characterize the jobs, (ii) allocate the jobs to the machines with a modified version of Lee and Pinedo's (1997) ATCS and (iii) improve final solution with Tabu Search. The application in real data results in 14% of deviation from the optimal solution obtained by enumeration. When applied to simulated scenarios of due date, processing and setup time, the heuristic results in average deviation of 18% from optimal solution obtained by enumeration to at least 70% of the simulations.

Key words: scheduling, unrelated parallel machines, setup times, earliness, tardiness

LISTA DE FIGURAS

Figura 2.1 – Iterações do TS para obtenção da solução ótima do exemplo.....	27
Figura 2.2 - Sequência de tarefas alocadas às máquinas 1 e 2.....	30
Figura 2.3 - Sequência de tarefas alocadas após a execução do <i>Tabu Search</i>	30
Figura 2.4 – Solução ótima obtida por enumeração e solução obtida pela heurística após <i>Tabu Search</i>	31
Figura 3.1 – Distribuição de tarefas alocadas por máquina.....	47
Figura 3.2 – Desvio da solução da etapa 2 da heurística em relação à enumeração.....	47
Figura 3.3 – Desvio da solução do TS em relação à enumeração.....	48

LISTA DE QUADROS

Quadro 2.1 – Passos da etapa 1 da heurística.....	24
Quadro 2.2 – Passos da etapa 2 da heurística.....	25
Quadro 2.3 - Passos da etapa 3 da heurística.....	28
Quadro 3.1 – Relações de influência entre os termos de folga e de preparação em função dos valores de k_1 e k_2	43
Quadro 3.2 – Passos da determinação do valor de k_1 e de k_2	44
Quadro 3.3 – Passos da etapa 2 da heurística.....	44
Quadro 3.4 – Passos da etapa 3 da heurística.....	45

LISTA DE TABELAS

Tabela 2.1 – Tempos de processamento e <i>setup</i> entre tarefas.....	25
Tabela 2.2 - Cálculos intermediários da etapa 2 da heurística.....	26
Tabela 2.3 - Tempos de processamento das tarefas.....	28
Tabela 2.4 – Tarefas ordenadas pelo WSPT.....	29
Tabela 2.5 – Tempos de <i>setup</i> entre tarefas.....	30
Tabela 2.6 – Desempenho das etapas da heurística em relação à enumeração total ($\sum C_j$) ..	31
Tabela 2.7 – Desempenho das etapas da heurística em relação à enumeração total ($\sum w_j T_j$).	32
Tabela 3.1 – Desempenho das etapas da heurística em relação à enumeração.....	46
Tabela 3.2 – Intervalos utilizados na geração das simulações.....	47
Tabela 3.3 – Desvios em relação à enumeração.....	48
Tabela 3.4 – Proporção de soluções ótimas por máquina obtida pelo TS.....	49

SUMÁRIO

1. INTRODUÇÃO	11
1.1 Considerações Iniciais	11
1.2 Justificativa do Tema.....	12
1.3 Objetivos.....	13
1.4 Procedimentos Metodológicos	13
1.5 Delimitações da Pesquisa	14
1.6 Estrutura da Dissertação	15
1.7 Referências	15
2. PRIMEIRO ARTIGO: HEURÍSTICA PARA PROGRAMAÇÃO DE TAREFAS EM MÁQUINAS PARALELAS NÃO-RELACIONADAS COM TEMPOS DE <i>SETUP</i> DEPENDENTES DAS TAREFAS.....	17
2.1 Introdução	18
2.2 Referencial Teórico	19
2.3 Método.....	23
2.4 Resultados e Discussão.....	28
2.5 Conclusão	32
2.6 Referências	33
3. SEGUNDO ARTIGO: MINIMIZAÇÃO SIMULTÂNEA DO ADIANTAMENTO E ATRASO EM MÁQUINAS PARALELAS NÃO-RELACIONADAS COM TEMPOS DE <i>SETUP</i> DEPENDENTES DA SEQUÊNCIA DAS TAREFAS.....	36
3.1 Introdução	37
3.2 Referencial teórico.....	38
3.3 Método de Trabalho.....	41
3.4 Resultados e Discussões	46
3.5 Conclusão	49
3.6 Referências	49
4. CONSIDERAÇÕES FINAIS.....	53
4.1 Conclusões.....	53
4.2 Sugestões para trabalhos futuros	54

1. Introdução

1.1 Considerações Iniciais

A programação de tarefas é um processo de decisão que desempenha relevante papel na maximização do desempenho operacional e financeiro das empresas. Pela sua grande aplicabilidade, a eficiente programação de tarefas se constitui em importante diferencial competitivo, razão pela qual tem sido amplamente pesquisada em diversos ambientes de manufatura.

A programação de tarefas se constitui no processo de organização e escolha do recurso mais adequado para a execução de uma tarefa, considerando restrições de tempo e de relacionamento entre tarefas, visando aprimorar um ou mais objetivos. A programação de tarefas usualmente visa minimizar o tempo total decorrido para completar a última tarefa de um conjunto de tarefas, o tempo total de conclusão de todas as tarefas, o atraso, a conclusão adiantada das tarefas, o número total de tarefas atrasadas ou uma combinação destes objetivos. Estes mesmos objetivos podem ser considerados em conjunção com um fator de priorização, que assume a forma de um fator de importância ou um fator representativo de custo da tarefa (MORTON & PENTICO, 1993; PINEDO, 2008). Em particular, a minimização da soma dos custos de estocagem decorrentes da conclusão adiantada de tarefas e das penalidades impostas pelos atrasos tem recebido crescente atenção por conta do seu alinhamento com os conceitos de produção *just-in-time* (SCHUTTEN & LEUSSINK, 1996; LEE & PINEDO, 1997; SIVRIKAYA-SERIFOGLU & ULUSOY, 1999; OMAR & TEO, 2006; JEONG & KIM, 2008; RIOS-SOLIS & SOURD, 2008; CHEN, 2009; RAGHAVAN & VENKATARAMANA, 2009; GACIAS et al., 2010; LEE et al., 2010).

O problema de programação de tarefas em um ambiente de máquinas paralelas não-relacionadas (ou seja, cada máquina possui uma velocidade de processamento não associada a outras máquinas), problema abordado nesta dissertação, consiste na escolha da máquina mais apropriada para execução de cada tarefa, consideradas as restrições envolvidas, e a determinação da melhor sequência de processamento das tarefas para otimizar os indicadores de desempenho desejados (LÓPEZ et al., 1995; MOKOTOFF, 2001; PFUND et al., 2004; PINEDO, 2008; CHANG & CHEN, 2011; FANJUL-PEYRO & RUIZ, 2011; LIN et al., 2011).

1.2 Justificativa do Tema

Os ambientes de manufatura são usualmente organizados em um ou mais estágios de máquinas simples, máquinas em linha, máquinas em paralelo ou uma combinação destes. Os ambientes constituídos por máquinas em paralelo, no entanto, são particularmente atrativos para novas pesquisas, tanto no aspecto teórico como prático.

No aspecto teórico, o tema aqui abordado se apoia na elevada complexidade matemática do arranjo em questão, o qual é muito superior aos tradicionais problemas de programação de tarefas. Por esse motivo, muitas abordagens sugeridas na literatura sofrem de excesso de abstração e carência de aplicação prática, fazendo com que o desenvolvimento de heurísticas simples e eficientes encontre respaldo em recentes pesquisas em programação de tarefas (GUPTA & STAFFORD Jr, 2006; MOKOTOFF, 2001; LIAW et al., 2003; PINEDO, 2008; LIN et al., 2011).

Dentre as diversas abordagens para o problema em questão, merecem destaque aquelas voltadas à minimização do tempo total para conclusão da sequência de tarefas, sujeita ou não a restrições relacionadas aos tempos de preparação (MOKOTOFF, 2001; PFUND et al., 2004). Entretanto, a pesquisa relacionada a máquinas paralelas não-relacionadas considerando tempos de preparação dependentes da sequência, com os objetivos de minimização do tempo de conclusão das tarefas e da soma do atraso e do adiantamento das tarefas, se apresenta pouco explorada, constituindo-se em oportunidade de pesquisas (PFUND et al., 2004).

No contexto prático, a pesquisa proposta se justifica pelos inúmeros cenários compostos por máquinas em paralelo não-relacionadas, normalmente oriundos da necessidade de renovação do maquinário existente ou do aumento da capacidade instalada (visto que máquinas de gerações tecnológicas distintas possuem velocidades de operação e tempos de preparação distintos).

As funções-objetivo tratadas neste estudo, especialmente a minimização do tempo de conclusão e da soma do adiantamento e atraso das tarefas, também encontram respaldo prático. A pontualidade de entrega dos produtos aos clientes é de vital importância para minimizar os custos de estocagem da produção finalizada antecipadamente e as despesas com multas contratuais e fretes expressos decorrentes da finalização da produção após a data de entrega prometida ao cliente. Da mesma forma, a minimização do tempo total de produção resulta na minimização dos custos operacionais e aumento da competitividade da empresa. Neste contexto, o ambiente de manufatura e o objetivo da programação de tarefas assumem

importância estratégica no gerenciamento das operações.

1.3 Objetivos

Os objetivos desta dissertação são:

- a) Apresentar as pesquisas recentes envolvendo máquinas paralelas não-relacionadas.
- b) Propor heurísticas para solução de problemas de programação de tarefas envolvendo máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência de processamento das tarefas;
- c) Propor adaptações a sistemáticas de programação de tarefas já consolidadas na literatura, como o ATCS de Lee e Pinedo (1997);

Esta dissertação objetiva primeiramente apresentar uma heurística para minimizar o atraso das tarefas e o tempo total de conclusão de um conjunto de tarefas sujeitas a um fator de priorização. O segundo objetivo é apresentar uma heurística para minimização da soma do atraso e do encerramento adiantado de um conjunto de tarefas sujeitas a um fator de priorização e com tempos de *setup* dependentes da sequência de processamento. O desempenho das heurísticas propostas é comparado ao resultado ótimo gerado através de enumeração total. O ambiente de manufatura considerado consiste em duas máquinas paralelas não-relacionadas com tempos de preparação dependentes da sequência de processamento das tarefas, em um processo de manufatura de componentes eletrônicos.

1.4 Procedimentos Metodológicos

A pesquisa em questão se trata de uma pesquisa aplicada de abordagem quantitativa realizada em duas etapas: (i) pesquisa bibliográfica para levantamento das pesquisas recentes do tema em estudo e (ii) pesquisa experimental, na forma de modelagem e simulação, para elaboração de uma heurística de programação de tarefas que satisfaça aos objetivos de minimização considerados.

A pesquisa bibliográfica busca levantar as recentes pesquisas relacionadas ao tema em publicações de periódicos internacionais e em livros de autores consagrados em programação de tarefas, dentre os quais Pinedo (2008) e Morton & Pentico (1993). A pesquisa experimental se baseia em estudo de caso específico, o ambiente de manufatura de máquinas paralelas não-relacionadas, com distintas funções-objetivo a serem minimizadas em cada um

dos dois artigos que compõem a pesquisa.

Os dois artigos utilizam uma versão aprimorada do procedimento de solução dos problemas proposto por Bank e Werner (2001). Uma terceira etapa, para aprimoramento dos resultados, é adicionada ao procedimento original proposto pelos autores para solucionar problemas complexos, tais como os objetivos de minimização da pesquisa.

O primeiro artigo descreve a elaboração de uma heurística de programação de tarefas para o ambiente de manufatura composto por duas máquinas paralelas com tempos de *setup* dependentes da sequência, que satisfaça os objetivos de (i) minimização do atraso total e (ii) minimização do tempo total para a realização de todas as tarefas, sujeitos a um fator de priorização. Utilizando tempos de processamento e de *setup* determinísticos, a heurística para minimização de ambos os problemas ordena as tarefas pelo WSPT, aloca as tarefas às máquinas e aprimora o resultado da segunda etapa com *Tabu Search*.

O segundo artigo trata, para o mesmo ambiente de manufatura, da elaboração de uma heurística de programação de tarefas para satisfazer o objetivo de minimização simultânea do adiantamento e do atraso das tarefas, sujeitas a um fator de priorização. A heurística é aplicada a dados reais e a cenários simulados de tempos de processamento e *setup*, data de entrega e fator de priorização, onde se realizam as etapas de caracterização do conjunto de tarefas por um conjunto de métricas, alocação das tarefas às máquinas através de uma versão modificada do ATCS de Lee e Pinedo (1997) e aprimoramento da solução final com *Tabu Search*.

1.5 Delimitações da Pesquisa

A pesquisa em questão está limitada ao ambiente de manufatura de máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência. Contudo, a pesquisa pode ser facilmente adaptada para o ambiente de manufatura de máquinas paralelas ou idênticas, sem *setup*, com *setup* dependente ou independente da sequência, desde que as devidas modificações nos tempos de processamento e *setup* sejam realizadas.

Outra limitação das abordagens propostas está no tratamento de sistemas compostos por um único estágio (visto que arranjos multiestágios são comumente encontrados em aplicações práticas). Por fim, não são elencados aspectos de custos nas funções-objetivos abordadas.

1.6 Estrutura da Dissertação

A dissertação está estruturada em quatro capítulos. O primeiro capítulo introduz o tema, sua importância, a contribuição da pesquisa, os objetivos e suas justificativas, o método de trabalho, a estrutura e as limitações da pesquisa.

O segundo capítulo apresenta o primeiro artigo da dissertação, o qual desenvolve uma heurística de programação de tarefas que utiliza um procedimento de três etapas para minimizar (i) o atraso total e (ii) o tempo total para a realização de todas as tarefas. A heurística ordena as tarefas pelo WSPT, aloca as tarefas às máquinas e aprimora a solução final com *Tabu Search*. Os resultados aprimorados da heurística são comparados com a solução ótima obtida por enumeração.

O terceiro capítulo apresenta o segundo artigo da dissertação, o qual apresenta uma heurística de programação de tarefas que, de forma similar ao primeiro artigo, utiliza um procedimento de três etapas para minimizar simultaneamente o adiantamento e atraso de um conjunto de tarefas. A heurística do segundo artigo caracteriza o conjunto de tarefas, aloca as tarefas às máquinas utilizando uma versão modificada do ATCS de Lee e Pinedo (1997) e aprimora a solução final com *Tabu Search*. Os resultados aprimorados da heurística são comparados com a solução ótima dos problemas obtida por enumeração.

O quarto capítulo apresenta as conclusões e as limitações da pesquisa. Oportunidades de pesquisas futuras em continuidade à pesquisa atual também são comentadas.

1.7 Referências

- BANK, J. & WERNER, F.** *Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties.* Mathematical and Computer Modelling 33 (2001) 363-383
- CHANG, P. C. & CHEN, S. H.** *Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times.* Applied Soft Computing 11 (2011) 1263-1274
- CHEN, J. F.** *Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints.* International Journal of Advanced Manufacturing Technology 44 (2009) 1204-1212
- FANJUL-PEYRO, L. & RUIZ, R.** *Size-reduction heuristics for the unrelated parallel machines scheduling problems.* Computers & Operations Research 38 (2011) 301-309
- GACIAS, B.; ARTIGUES, C. & LOPEZ, P.** *Parallel machine scheduling with precedence constraints and setup times.* Computers & Operations Research 37 (2010) 2141-2151
- GUPTA, J. N. D. & STAFFORD Jr, E. F.** *Flowshop scheduling research after five decades.* European Journal of Operational Research 169 (2006) 699-711.

- JEONG, S. J. & KIM, K. S.** *Parallel machine scheduling with earliness-tardiness penalties and space limits.* International Journal of Advanced Manufacturing Technology 37 (2008) 793-802
- LEE, Y. H. & PINEDO, M.** *Scheduling jobs on parallel machines with sequence-dependent setup times.* European Journal of Operational Research 100 (1997) 464-474
- LEE, Z. J.; LIN, S. W. & YING, K. C.** *Scheduling jobs on dynamic parallel machines with sequence-dependent setup times.* International Journal of Advanced Manufacturing Technology 47 (2010) 773-781
- LIAW, C. F.; LIN, Y. K.; CHENG, C. Y. & CHEN, M.** *Scheduling unrelated parallel machines to minimize total weighted tardiness.* Computers & Operations Research 30 (2003) 1777-1789
- LIN, Y. K.; PFUND, M. E. & FOWLER, J. W.** *Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems.* Computers & Operations Research 38 (2011) 901-916
- LÓPEZ, O. C.; BARCIA, R. M. & EYADA, O.** *Problema de programação da produção: um esquema de classificação.* Revista Produção 5 (1995) 145-168
- MOKOTOFF, E.** *Parallel machine scheduling problems: a survey.* Asia-Pacific Journal of Operational Research 18 (2001) 193-242
- MORTON, T. E. & PENTICO, D. W.** *Heuristic scheduling systems: with applications to production systems and project management.* New York: John Wiley & Sons 1993.
- OMAR, M. K. & TEO, S. C.** *Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach.* Applied Mathematics and Computation 181 (2006) 1008-1017
- PFUND, M.; FOWLER, J. W. & GUPTA, J. N. D.** *A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems.* Journal of the Chinese Institute of Industrial Engineers, Vol. 21, No 3, (2004) 230-241
- PINEDO, M. L.** *Scheduling: Theory, Algorithms, and Systems.* New York: Springer 2008.
- RAGHAVAN, N. R. S. & VENKATARAMANA, M.** *Parallel processor scheduling for minimizing total weighted tardiness using ant colony optimization.* International Journal of Advanced Manufacturing Technology 41 (2009) 986-996
- RIOS-SOLIS, Y. A. & SOURD, F.** *Exponential neighborhood search for a parallel machine scheduling problem.* Computers & Operations Research 35 (2008) 1697-1712
- SCHUTTEN, J. M. J. & LEUSSINK, R. A. M.** *Parallel machine scheduling with release dates, due dates and family setup times.* International Journal of Production Economics 46-47 (1996) 119-125
- SIVRIKAYA-SERIFOGLU, F. & ULUSOY, G.** *Parallel machine scheduling with earliness and tardiness penalties.* Computers & Operations Research 26 (1999) 773-787

2 PRIMEIRO ARTIGO

HEURÍSTICA PARA PROGRAMAÇÃO DE TAREFAS EM MÁQUINAS PARALELAS NÃO-RELACIONADAS COM TEMPOS DE *SETUP* DEPENDENTES DAS TAREFAS

Resumo

Este artigo propõe uma heurística para programação de tarefas em um ambiente de manufatura composto por máquinas paralelas não-relacionadas com tempos de preparação (*setup*) dependentes da sequência de processamento das tarefas. As funções-objetivo consideradas consistem na minimização da soma do tempo ponderado de conclusão e do atraso ponderado das tarefas. Uma heurística de três etapas é proposta para (i) realizar uma ordenação inicial das tarefas, (ii) alocar as tarefas às máquinas, e (iii) aprimorar a solução final através da ferramenta de otimização *Tabu Search*. O desempenho da heurística é comparado à solução ótima obtida por enumeração total. Ao ser aplicada em um ambiente de manufatura de componentes eletrônicos composto por duas máquinas paralelas não-relacionadas com tempos de processamento e de preparação determinísticos, a heurística proposta resulta em um desvio de 1,1% em relação ao valor ótimo da soma do tempo ponderado de conclusão e 4,6% do mínimo atraso ponderado das tarefas.

Palavras-chaves: Sequenciamento da produção, máquinas paralelas não relacionadas, tempos de *setup*, *tabu search*

Abstract

This article proposes a scheduling heuristic for unrelated parallel machines with sequence dependent setup times. The minimization of total weighted completion time and the minimization of weighted tardiness are the objectives of interest. A three phase heuristic is proposed to (i) propose an initial order for allocation of jobs to machines, (ii) allocate the jobs to the machines, and (iii) improve final solution of both objectives with Tabu Search. The heuristic performance is evaluated comparing the quality of proposed solution with the optimal solution obtained from total enumeration. The

proposed model is applied in an industry of electronic components industry composed of two unrelated parallel machines with deterministic processing and setup times. We obtained a 1.1% deviation for the optimal total weighted completion time and 4.6% of deviation for the weighted tardiness.

Keywords: *job scheduling, unrelated parallel machines, setup times, tabu search*

2.1 Introdução

A programação de tarefas (*job scheduling*) é um processo gerencial de fundamental importância na maximização do desempenho operacional e financeiro das empresas. Pela sua grande aplicabilidade, a eficiente programação de tarefas se constitui em importante diferencial competitivo, razão pela qual tem sido amplamente pesquisada em diversos ambientes de manufatura.

Utilizada regularmente nas empresas manufatureiras e de serviços, a programação de tarefas se constitui no processo de ordenamento das tarefas e escolha do recurso mais adequado para a execução de uma tarefa, considerando restrições de tempo e de relacionamento entre tarefas (PINEDO, 2008). A programação de tarefas usualmente visa minimizar o tempo de conclusão de um conjunto de tarefas (*completion time*), o atraso (*tardiness*), a conclusão adiantada das tarefas (*earliness*), o número total de tarefas atrasadas (*tardy jobs*) ou uma combinação destes objetivos. Tais objetivos podem ser integrados a um fator de priorização (*weight*), que assume a forma de um fator de importância ou um fator representativo de custo da tarefa (MORTON & PENTICO, 1993; PINEDO, 2008).

Os ambientes de manufatura podem apresentar configurações diversas, de acordo com o nível de customização do produto em questão. Ambientes constituídos por máquinas não-relacionadas em paralelo (ou seja, cada máquina possui uma velocidade de processamento não associada a outras máquinas) são especialmente atrativos, tanto do ponto de vista teórico como prático. No aspecto teórico, ambientes de manufatura compostos por máquinas em paralelo não-relacionadas são tidos como uma das classes de mais difícil solução dentre os problemas de sequenciamento. No contexto prático, tais arranjos representam muitos ambientes de manufatura reais, gerados pela necessidade de renovação do maquinário existente, ou pelo aumento da capacidade instalada, visto que máquinas de gerações tecnológicas distintas possuem velocidades de operação e tempos de preparação distintos (MOKOTOFF, 2001; LIAW et al., 2003; PINEDO, 2008; LIN et al., 2011).

Tal relevância tem justificado o elevado número de abordagens focadas no arranjo em questão, principalmente com vistas à minimização do tempo total para conclusão da última tarefa (*makespan*) (MOKOTOFF, 2001; PFUND et al., 2004). Entretanto, a inserção do tempo de preparação de máquina dependente da sequência ainda se constitui em tópico pouco explorado na literatura (PFUND et al., 2004).

Este artigo apresenta uma heurística para minimizar duas funções-objetivo de interesse prático: o atraso ponderado das tarefas, e a soma ponderada do tempo de conclusão de um conjunto de tarefas, ambas com tempos de preparação dependentes da sequência das tarefas. A sistemática proposta é composta por três etapas: (i) ordenação prévia das tarefas de acordo com uma regra alinhada à função-objetivo em análise, (ii) alocação das tarefas às máquinas de maneira a assegurar balanço entre os tempos de processamento alocados e sequenciamento preliminar, e (iii) refino do sequenciamento das tarefas alocadas a cada máquina. O método proposto é aplicado em um processo de manufatura de componentes eletrônicos composto por duas máquinas paralelas não-relacionadas com tempos de preparação dependentes da sequência de tarefas. O desempenho da heurística proposta é comparado ao resultado ótimo gerado através de enumeração total.

O artigo está estruturado como segue: a seção 2.2 apresenta o referencial teórico sobre programação de tarefas em ambiente de manufatura de máquinas paralelas não-relacionadas. As seções 2.3 e 2.4 apresentam, respectivamente, o método proposto e discussão dos resultados. Por fim, a seção 2.5 apresenta a conclusão da pesquisa e direções futuras.

2.2 Referencial Teórico

A programação de tarefas em máquinas paralelas consiste na alocação de n tarefas a m máquinas candidatas, com o objetivo de otimizar uma função-objetivo associada a uma medida de desempenho (MOKOTOFF, 2001). Ambientes de manufatura constituídos por máquinas em paralelo podem ser categorizados em três classes: (i) máquinas paralelas idênticas (*identical parallel machines*) – tempo de processamento das tarefas é o mesmo em qualquer uma das máquinas, (ii) máquinas paralelas uniformes (*uniform parallel machines*) – tempo de processamento da tarefa é dependente da velocidade da máquina que irá processar a tarefa, e (iii) máquinas paralelas não-relacionadas (*unrelated parallel machines*) - tempo de processamento da tarefa é dependente, simultaneamente, da velocidade da máquina e de características da tarefa (CHEN & POWELL, 1999; MOKOTOFF, 2001; CHAO et al., 2008; PINEDO, 2008; LI & YANG, 2009; WU & JI, 2009; FANJUL-PEYRO & RUIZ, 2011; LIN

et al., 2011).

O problema de programação de tarefas em máquinas paralelas não-relacionadas consiste, inicialmente, na escolha da máquina mais apropriada para cada tarefa, e então na determinação da melhor sequência de processamento das tarefas alocadas a cada máquina (LÓPEZ et al., 1995; MOKOTOFF, 2001; PFUND et al., 2004; PINEDO, 2008; CHANG & CHEN, 2011; FANJUL-PEYRO & RUIZ, 2011; LIN et al., 2011). Esse arranjo de máquinas é detalhado nas seções que seguem.

2.2.1 Máquinas paralelas não-relacionadas

Duas funções-objetivo associadas a máquinas paralelas não-relacionadas têm recebido crescente atenção: (i) minimização do tempo total ponderado de conclusão, e (ii) minimização do atraso ponderado de entrega. Tais problemas são classificados como *NP-hard*, ou seja, requerem um tempo de processamento não polinomial (exponencial) para obtenção da solução ótima à medida que o número de máquinas ou tarefas é aumentado. Por conta disto, diversos algoritmos e heurísticas têm sido propostos para gerar soluções satisfatórias em tempos reduzidos de processamento computacional (LÓPEZ et al., 1995; PFUND et al., 2004; PINEDO, 2008).

Ao utilizar-se o sistema de representação proposto por Graham et al. (1979), os problemas (i) e (ii) são descritos, respectivamente, como $Rm / S_{jk} / \sum w_j C_j$ e $Rm / S_{jk} / \sum w_j T_j$. Nesta representação, o primeiro campo identifica o ambiente das máquinas, assumindo a representação R para máquinas paralelas não-relacionadas; o índice m representa o número de máquinas em paralelo. O segundo campo identifica as restrições do sistema: (i) dependente da sequência (S_{jk}) (*sequence dependent*) – tempo de preparação é variável em função da tarefa atual e da tarefa subsequente, ou (ii) independente da sequência (*sequence independent*) – tempo de preparação é fixo, independentemente da sequência de processamento das tarefas (ALLAHVERDI et al., 2008; PINEDO, 2008). Por fim, o terceiro campo especifica a função-objetivo considerada: (i) $\sum w_j C_j$ – soma do tempo ponderado para conclusão de um conjunto de tarefas; e (ii) $\sum w_j T_j$ – soma ponderada do atraso. A presença de um fator de priorização (*weight*) é identificada através da letra w antes do critério de otimização. Na sequência são descritas abordagens recentes acerca das funções-objetivo acima descritas.

2.2.1.1 Minimização do tempo ponderado de conclusão em máquinas paralelas não-relacionadas

Heurísticas com vistas à minimização do tempo total de conclusão da última tarefa

envolvendo máquinas paralelas não-relacionadas têm sido amplamente sugeridas na literatura. Grande parte das abordagens, no entanto, não considera tempos de preparação dependentes das tarefas.

Dentre os autores que não assumem a presença de tempos de preparação, $Rm/\sum w_j C_j$, Fanjul-Peyro & Ruiz (2011; 2010) utilizaram metaheurísticas de busca iterativa local e redução de tamanho do problema original para conjuntos de tarefas com tempos de processamento distribuídos uniformemente. As heurísticas propostas alcançaram notável desempenho frente a outras proposições, gerando resultados próximos à solução ótima. Lin et al. (2011) propuseram uma sistemática de dois estágios baseada em relaxação de programação linear e procedimentos de redistribuição e intercâmbio. Tais procedimentos geram resultados ótimos para pequeno número de tarefas e aproximações satisfatórias para conjuntos com elevado número de tarefas.

Para a mesma função-objetivo, Gairing et al. (2005) propuseram uma abordagem combinatória de enumeração parcial baseada em programação inteira com *Branch-and-bound* (B&B) e planos de corte, formulando o problema como um caso geral de fluxo máximo indivisível. Grigoriev et al. (2005) desenvolveram um algoritmo em duas etapas, realizando uma relaxação da programação linear inteira seguida de um procedimento de arredondamento para provar a proximidade do resultado gerado com a solução ótima. Por sua vez, Srivastava (1998) utilizou heurísticas baseadas em *Tabu Search* (TS) para determinar o desvio percentual de ambas em relação à solução ótima.

Valendo-se de ferramentas de otimização, Wu e Ji (2009) propuseram um algoritmo genético para minimizar o tempo total decorrido para completar a última tarefa em uma linha de montagem de placas de circuitos impressos. A utilização de programação quadrática integrada ao algoritmo B&B foi aplicada por Plateau e Rios-Solis (2010) para resolver o problema $Rm/\sum w_j C_j$. Os autores alegam que os resultados obtidos se constituem no estado-da-arte para o problema em questão. Focados na mesma função-objetivo, Chen e Powell (1999) propuseram um método utilizando a decomposição de Dantzig-Wolfe, partindo da partição da formulação inicial e resolvendo os nós do algoritmo B&B como um subproblema de máquina simples através de uma sistemática de geração de colunas (*column generation* - CG).

Li e Yang (2009) apresentam uma classificação para modelos baseados em variáveis de atribuição e tempo de liberação das tarefas, relaxação linear e de programação quadrática convexa, algoritmos exatos e heurísticas para os problemas de máquinas paralelas com vistas

à minimização do tempo total de conclusão das tarefas, com e sem fator de priorização ($R_m // \sum w_j C_j$ e $R_m // \sum C_j$). Para o problema $R_m // \sum w_j C_j$, Shim (2009) abordou um esquema de ramificação baseado no algoritmo B&B com o objetivo de reduzir o tempo computacional para obtenção da solução ótima. Por fim, Correa et al. (2009) também estudaram a função-objetivo de minimização do tempo total decorrido para completar a última tarefa, sem considerar os tempos de preparação e permitindo interrupções no processamento das mesmas, ou seja, uma tarefa pode ser interrompida e retornar ao processamento posteriormente (*preemption*).

Ao incluir-se o tempo de preparação em arranjos de máquinas paralelas não-relacionadas com vistas à minimização do tempo de conclusão, $R_m / S_{jk} / \sum C_{max}$, destaca-se o estudo de Weng et al (2001), o qual compara sete heurísticas para resolução do problema; a melhor heurística utiliza uma lógica de alocação de tarefas com base na menor relação entre o tempo de processamento somado ao de preparação e ao seu fator de priorização. Com propósitos similares, Hsu et al. (2010) propuseram uma solução em tempo polinomial considerando o tempo de preparação proporcional ao tempo de processamento total até a tarefa imediatamente anterior.

Abordando o mesmo problema, um algoritmo genético (AG) utilizando intercâmbio adjacente e não adjacente de tarefas foi desenvolvido por Chang e Chen (2011) para aumento da velocidade de convergência da solução, enquanto que Ying et al. (2010) utilizaram um procedimento de *Simulated Annealing* (SA) restrito para eliminar movimentos de busca ineficientes. Rabadi et al. (2006) compararam uma metaheurística de busca prioritária aleatória (MHBPA) com heurística de particionamento (HP). Os três trabalhos obtiveram soluções ótimas para conjuntos compostos por poucas tarefas e superaram os resultados gerados por AG, SA e HP para conjuntos com maior número de tarefas.

Por sua vez, Arnaout et al. (2008; 2010) utilizaram um algoritmo baseado em *Ant Colony Optimisation* (ACO) de dois estágios em comparação com MHBPA, HP e TS para o problema em questão, concluindo que ACO apresentou desempenho superior às demais heurísticas. Por fim, Tavakkoli-Moghaddam et al. (2009) consideraram o problema $R_m / S_{jk} / \sum C_j$ com datas distintas de liberação e de conclusão entre as tarefas, obtendo resultados satisfatórios através da otimização por AG.

2.2.1.2 Minimização do atraso ponderado das tarefas em máquinas paralelas não-relacionadas

Contrariamente ao vasto número de estudos que tratam do problema $Rm // \sum w_j C_j$, percebe-se uma quantidade reduzida de abordagens com vistas à minimização da soma do atraso ponderado em máquinas paralelas não-relacionadas ($R_m // \sum w_j T_j$). Lin et al. (2011) e Liaw et al. (2003) abordam o problema propondo, respectivamente, heurísticas de dois estágios baseadas em *apparent tardiness cost* (ATC) e B&B. A minimização do atraso total das tarefas com tempos de preparação dependentes das tarefas sem fator de priorização ($R_m / S_{jk} / \sum T_j$) foi estudada por Chen e Chen (2009), que utilizaram uma heurística baseada no *apparent tardiness cost with setup* (ATCS) e SA. A mesma função-objetivo foi analisada por Kim et al. (2002) através de uma metaheurística baseada em SA, e por Silva e Magalhães (2006), que adaptaram heurísticas existentes para o problema de uma empresa produtora de fibras acrílicas.

A minimização do atraso total das tarefas com tempos de preparação dependentes das tarefas com fator de priorização, $R_m / S_{jk} / \sum w_j T_j$, foi abordada por Paula et al. (2010) através de um algoritmo *relax-and-cut* baseado em relaxação Lagrangeana, e por Logendran et al. (2007), que propuseram seis algoritmos baseados em TS. Por sua vez, Kim et al. (2003) acrescentam datas de entrega ao problema em questão, comparando o desempenho de heurísticas baseadas em regras de ordenação e SA.

2.3 Método

Considere o problema de programação de n tarefas em m máquinas paralelas não-relacionadas com tempos de preparação (*setups*) dependentes da sequência de processamento das tarefas. Todas as tarefas possuem datas de liberação e conclusão comuns, e seu processamento, uma vez iniciado, não pode ser interrompido.

Os objetivos considerados consistem em (i) minimizar o tempo total ponderado de conclusão ($Rm / S_{jk} / \sum w_j C_j$) e (ii) minimizar o atraso ponderado de um conjunto de tarefas ($Rm / S_{jk} / \sum w_j T_j$). Em função da complexidade de tais arranjos, Bank e Werner (2001) sugerem um procedimento de duas etapas para a solução do problema: na primeira etapa, as tarefas devem ser ordenadas para, na segunda etapa, serem alocadas às máquinas. O método proposto neste artigo adiciona uma terceira etapa ao procedimento de Bank e Werner (2001): (1) ordenação inicial das tarefas, (2) alocação das tarefas às máquinas, e (3) refino da ordenação das tarefas alocadas a cada máquina através de uma versão simplificada do algoritmo *Tabu Search*. As

duas primeiras etapas são comuns para ambos os objetivos; na terceira etapa, funções-objetivo distintas promovem ordenamentos específicos às tarefas. Essas etapas são detalhadas na sequência.

Etapa 1 – Ordenação inicial das tarefas

A primeira etapa da heurística realiza a ordenação inicial das tarefas para posterior alocação às máquinas, conforme apresentado no Quadro 2.1. As tarefas são ordenadas pela regra *WSPT (Weighted Shortest Processing Time)*, a qual ordena as tarefas de forma decrescente de w_n / p_{nm} , onde w_n , o fator de priorização da tarefa n , representa a importância de uma tarefa dentro do conjunto de tarefas ou o valor agregado à tarefa (PINEDO, 2008); p_{nm} representa o tempo de processo da tarefa n na máquina m .

Passos	Atividade
Passo 1	Calcula $\overline{p_{nm}}$ e $w_n / \overline{p_{nm}}$ de cada tarefa
Passo 2	Aplica <i>WSPT</i>

Quadro 2.1 – Passos da etapa 1 da heurística

Como o tempo de processamento difere de uma máquina para outra, a ordenação é realizada de forma decrescente de $w_n / \overline{p_n}$, fazendo com que sejam priorizadas tarefas com elevado fator de priorização e baixo tempo médio de processamento das tarefas nas máquinas consideradas ($\overline{p_n}$).

Etapa 2 – Alocação das tarefas às máquinas

Nesta etapa as tarefas são alocadas às máquinas com base no tempo total de processamento acumulado em cada máquina, T_m , calculado através da equação (1). T_m é calculado para cada uma das k tarefas alocadas à máquina m através do somatório do tempo de processamento da j -ésima tarefa (p_j) com o tempo de *setup* entre as tarefas $j-1$ e j , $S_{(j-1)j}$. Repare que o tempo de *setup* é considerado nulo na alocação da primeira tarefa a cada máquina, e que os tempos de processamento e de *setup* são determinísticos.

$$T_m = \sum_{j=1}^k (p_j + S_{(j-1)j}) \quad (1)$$

As primeiras m tarefas ordenadas no passo 1 são alocadas às máquinas em ordem crescente do valor de T_m (ou seja, por se tratar de um problema de *setup* dependente da sequência de tarefas, o tempo de *setup* somente será considerado quando se conhece a primeira tarefa

alocada a cada máquina). A próxima tarefa candidata à alocação tem seu tempo de processamento somado ao tempo de *setup*, de acordo com sua tarefa antecessora, e alocada à máquina com o menor T_m . O ciclo se repete até que todas as tarefas tenham sido alocadas às máquinas. Esses passos são resumidos no Quadro 2.2, e asseguram uma distribuição equilibrada de tarefas entre as máquinas, de maneira a minimizar ociosidade dos recursos.

Passo	Atividade
$i = 1$ a n (índice de ordenação da tarefa na etapa 1)	
Passo 1	Calcule T_m
Passo 2	Aloque a tarefa ordenada na etapa 1 à máquina com menor T_m . (Se $T_m = T_{m+1}$ aloque a tarefa à máquina de menor índice).
Passo 3	$i = i + 1$
Passo 4	Se $i \leq n$, execute o passo 1, senão passe para a Etapa 3 (próxima etapa da heurística).

Quadro 2.2 – Passos da etapa 2 da heurística

Um exemplo numérico desta etapa é agora apresentado para quatro tarefas, ordenadas de 1 a 4 pela etapa 1, a serem alocadas em duas máquinas paralelas não-relacionadas com vistas à minimização de $\sum w_j C_j$. Os tempos de processamento das tarefas em cada máquina e o tempo de *setup* são mostrados na Tabela 2.1.

i	w_n	p_{n1}	p_{n2}	S_{jk}	1	2	3	4
1	3	30	20	1	-	0	5	0
2	4	26	30	2	5	-	0	8
3	7	28	32	3	10	4	-	0
4	1	28	36	4	0	3	6	-

Tabela 2.1 – Tempos de processamento e *setup* entre tarefas

O passo inicial consiste em calcular T_1 e T_2 para as tarefas 1 e 2. A heurística aloca a tarefa 2 à máquina 1 ($T_1 = 26$) e a tarefa 1 à máquina 2 ($T_2 = 20$). Para determinar as máquinas que receberão as tarefas 3 e 4, calculam-se novamente os valores de T_1 e T_2 . Assim, a tarefa 3 é alocada à máquina 1 ($T_1 = 54$) e a tarefa 4 à máquina 2 ($T_2 = 56$).

i	T_1	T_2
3	$26 + (28 + 0) = 54$	$20 + (32 + 5) = 57$
4	$26 + (28 + 8) = 62$	$20 + (36 + 0) = 56$

Tabela 2.2 – Cálculos intermediários da etapa 2 da heurística

Etapa 3 – Ordenação das tarefas alocadas a cada máquina via *Tabu Search*

Nesta etapa, as tarefas alocadas a cada máquina têm seu ordenamento refinado através do algoritmo *Tabu Search* (TS) simples de Glover (1989) com vistas à minimização das funções-objetivo $\sum w_j C_j$ e $\sum w_j T_j$. Por fim, a heurística proposta compara a solução refinada com a solução ótima obtida por enumeração total.

O TS é um procedimento para solução de problemas combinatoriais baseado na busca da solução ótima global na vizinhança da solução atual, através do intercâmbio de duas tarefas adjacentes. Partindo de uma solução inicial, o TS realiza o intercâmbio de tarefas adjacentes gerando $n-1$ vizinhanças adjacentes à solução inicial. Para cada uma das vizinhanças geradas, calcula-se o valor da função-objetivo do problema. A vizinhança com menor (maior) valor, no caso de problemas de minimização (maximização), é escolhida para prosseguir na busca da solução ótima na próxima iteração do algoritmo (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008).

A cada iteração, o par de tarefas no qual o intercâmbio gerou a vizinhança com menor (maior) valor da função-objetivo é armazenado em uma lista de movimentos proibidos (lista *tabu*), os quais não podem ser realizados nas iterações seguintes. Evita-se, de tal modo, que a busca retorne a vizinhanças anteriormente visitadas e cujo valor da função-objetivo é superior (inferior) ao atual. A lista *tabu* possui um número fixo de entradas e, a cada novo par de tarefas incluído na lista, o par mais antigo é excluído e seu intercâmbio pode ser realizado novamente nas próximas iterações (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008).

A sistemática de operacionalização do TS é agora ilustrada para a minimização de $\sum w_j C_j$. Suponha que as quatro tarefas do exemplo anterior foram alocadas à máquina 1. Partindo de uma solução inicial (3, 2, 1, 4) com função objetivo igual a 449, a primeira iteração do TS realiza os intercâmbios adjacentes das tarefas, gerando as sequências (2, 3, 1, 4), (3, 1, 2, 4), (3, 2, 4, 1). O valor da função objetivo das três sequências é calculado pela equação (2), cujos resultados são, respectivamente, 448, 456 e 442.

$$\sum_{j=1}^n w_j \cdot (p_j + S_{jk}) \quad (2)$$

A sequência (3, 2, 4, 1) substitui a sequência inicial (3, 2, 1, 4) como a solução ótima atual e o par (1, 4), intercambiado na iteração, assume a posição superior da lista tabu. Na segunda iteração, a sequência (3, 2, 4, 1) tem três novos intercâmbios adjacentes, gerando as sequências (2, 3, 4, 1), (3, 4, 2, 1) e (3, 2, 1, 4). A sequência (3, 2, 1, 4) realiza o intercâmbio proibido entre as tarefas 1 e 4, sendo, portanto, desconsiderada. O valor da função-objetivo das sequências (2, 3, 4, 1) e (3, 4, 2, 1) são, respectivamente, 418 e 445. Assim, a sequência (2, 3, 4, 1) assume o lugar da sequência (3, 2, 4, 1) como a solução ótima atual e o par (3,2) ocupa o primeiro lugar da lista tabu, empurrando o par (1,4), intercambiado na iteração anterior, para a segunda posição da lista.

Por fim, o único intercâmbio permitido na sequência (2, 3, 4, 1) é entre as tarefas 3 e 4, gerando a sequência (2, 4, 3, 1), cuja função objetivo é 498. Desta forma, a sequência (2, 3, 4, 1), cujo valor da função objetivo é 418, resulta na solução ótima do problema. A Figura 2.1 mostra a sequência de iterações realizada pelo TS para obter a solução ótima do exemplo.

Sequência de tarefas (valor da função objetivo)				
Seq. Inicial	3214 (449)			
Iteração 1	2314 (448)	3124 (456)	3241 (442)	
Iteração 2			2341 (418)	3421 (445) 3214 (tabu)
Iteração 3		3241 (tabu)	2431 (498)	2314 (tabu)
Iteração 4	3214 (tabu)	2134 (461)	2341 (tabu)	

Figura 2.1 – Iterações do TS para obtenção da solução ótima do exemplo

O valor da função objetivo gerada pelo TS é então comparado com a solução ótima obtida através de enumeração total. O Quadro 2.3 sumariza os passos da terceira etapa da heurística para a minimização do tempo total de execução das tarefas.

Sequência	Atividade
Passo 1	Aplicar <i>Tabu Search</i> para obtenção da solução ótima
Passo 2	Comparar a solução obtida com a solução ótima obtida por enumeração

Quadro 2.3 – Passos da etapa 3 da heurística

Por fim, a minimização da segunda função-objetivo ($\sum w_j T_j$) com uma data de entrega comum para todas as tarefas segue os passos descritos no Quadro 3, valendo-se da equação (3).

$$\sum w_j T_j = \max(C_j - d, 0) \cdot w_j \quad (3)$$

2.4 Resultados e Discussão

A heurística proposta foi aplicada em um ambiente de manufatura composto por 14 tarefas e duas máquinas paralelas não-relacionadas, no processo de metalização de filmes plásticos em alto vácuo. O processo consiste na deposição de uma camada de metal sobre o filme plástico com a finalidade de obter determinadas características elétricas ou de aparência. A Tabela 2.3 mostra o tempo de processamento das tarefas em cada máquina (p_1 e p_2), o tempo médio de processamento (\bar{p}_n) e o fator de priorização w , utilizados na etapa 1 da heurística. O tempo médio de processamento é oriundo da média aritmética dos tempos de processamento p_1 e p_2 . Os tempos de processamento e de *setup*, quando existente, são de natureza determinística e correspondem a tempos reais do processo de metalização.

Tarefa	p_1	p_2
1	56	52
2	65	60
3	89	90
4	93	93
5	82	90
6	85	93
7	75	82
8	75	82
9	88	108
10	79	97
11	71	86
12	62	75
13	53	64
14	44	52

Tabela 2.3 - Tempos de processamento das tarefas

O resultado da etapa 1 da heurística, apresentado na Tabela 2.4, corresponde à aplicação da ordenação WSPT, tomando por base o quociente entre o fator de priorização e o tempo médio de processamento das tarefas (w / \bar{p}_{nm}).

Tarefas	\bar{p}_n	w	w / \bar{p}_{nm}
1	54	212,5	3,94
2	62,5	244,6	3,91
4	93	146,3	1,57
3	89,5	131,4	1,47
6	89	94,4	1,06
5	86	82,1	0,95
7	78,5	60,8	0,78
8	78,5	52,2	0,67
14	48	22,5	0,47
9	98	35,9	0,37
10	88	27,8	0,32
13	58,5	18,4	0,31
12	68,5	20,5	0,30
11	78,5	22,4	0,29

Tabela 2.4 – Tarefas ordenadas pelo WSPT

Na sequência, a etapa 2 da heurística aloca as tarefas às máquinas com menor tempo total acumulado. Os tempos expressos nas Figuras 2.2 a 2.4 são apresentados em termos de $\sum C_j$ para facilitar a visualização dos efeitos das alterações das posições das tarefas nas máquinas 1 e 2. O ordenamento é idêntico quando considerado o termo de priorização, gerando $\sum w_j C_j$. Concluída a alocação da primeira tarefa à máquina, soma-se o tempo total acumulado ao tempo de *setup* (ver apêndice) entre a tarefa anteriormente alocada e a tarefa candidata à alocação, como critério de decisão para alocação da próxima tarefa. A Tabela 2.5 mostra os tempos de *setup* entre as tarefas.

Tarefa	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	-	0	0	15	15	0	15	15	20	20	20	20	20	20
2	0	-	0	15	15	0	15	15	20	20	20	20	20	20
3	0	0	-	15	15	0	15	15	20	20	20	20	20	20
4	15	15	15	-	15	15	0	15	20	20	20	20	20	20
5	15	15	15	15	-	15	15	0	20	20	20	20	20	20
6	0	0	0	15	15	-	15	15	20	20	20	20	20	20
7	15	15	15	0	15	15	-	15	20	20	20	20	20	20
8	15	15	15	15	0	15	15	-	20	20	20	20	20	20
9	20	20	20	20	20	20	20	20	-	15	15	15	15	15
10	20	20	20	20	20	20	20	20	15	-	15	15	15	15
11	20	20	20	20	20	20	20	20	15	15	-	0	15	0
12	20	20	20	20	20	20	20	20	15	15	0	-	15	0
13	20	20	20	20	20	20	20	20	15	15	15	15	-	15
14	20	20	20	20	20	20	20	20	15	15	0	0	15	-

Tabela 2.5 – Tempos de *setup* entre tarefas

A Figura 2.2 apresenta o resultado da alocação das tarefas às máquinas (o número da tarefa aparece no centro de cada barra do gráfico de Gantt), resultante da etapa 2 da heurística, com $\Sigma C_j = 635$ na máquina 1 e $\Sigma C_j = 554$ na máquina 2. Esta distribuição de tarefas às máquinas não é ótima, pois a solução obtida por enumeração resulta no mesmo número de tarefas por máquina, porém com $\Sigma C_j=558$ na máquina 1 e $\Sigma C_j=613$ na máquina 2.

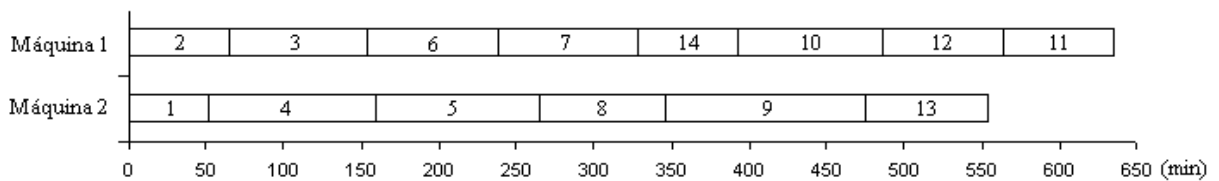
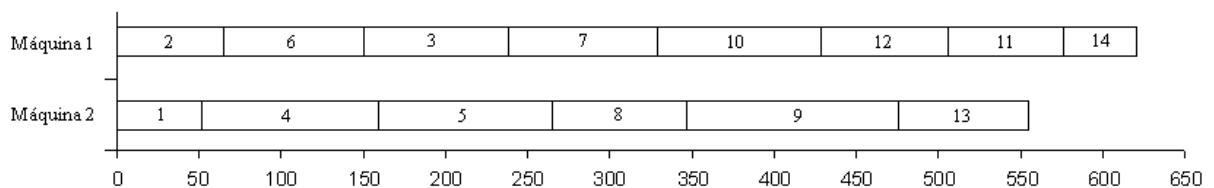


Figura 2.2 – Sequência de tarefas alocadas às máquinas 1 e 2 pela etapa 2

A etapa 3 da heurística aplica o TS em cada máquina, refinando a solução fornecida pela etapa anterior. A Figura 2.3 apresenta o novo sequenciamento, gerando $\Sigma C_j = 620$ na máquina 1 e $\Sigma C_j = 554$ na máquina 2.

Figura 2.3 – Sequência de tarefas alocadas após a execução do *Tabu Search*

A aplicação do TS reduz o ΣC_j da máquina 1 de 635 para 620, o qual representa a melhor

solução para o conjunto de tarefas alocadas àquela máquina. Tal melhora decorre do reposicionamento das tarefas 6, 3, 10 e 14, conforme ilustrado nas Figuras 2.2 e 2.3. O tempo total da máquina 2, $\sum C_j = 554$, não foi aprimorado pelo TS, pois o resultado fornecido pela etapa 2 da heurística já se constitui na melhor solução daquele conjunto.

Por fim, o resultado final da heurística é comparado com a sequência proposta pela enumeração total, na Figura 2.4. A heurística e a enumeração alocam o mesmo número de tarefas por máquina, contudo, o tempo total entre ambas difere, pois a enumeração considera apenas os fatores quantitativos utilizados na primeira etapa da heurística.

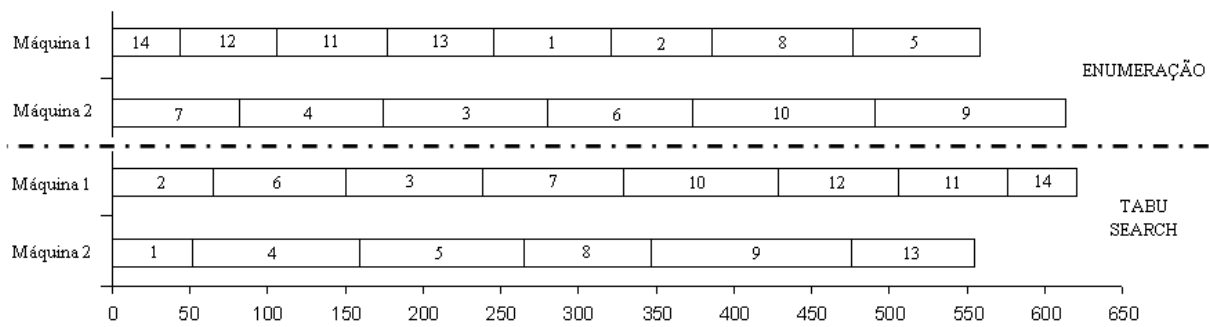


Figura 2.4 – Solução ótima obtida por enumeração e solução obtida pela heurística após *Tabu Search*

Por outro lado, a heurística considera um parâmetro qualitativo (w), além dos parâmetros quantitativos, desta forma limitando a solução obtida pelo TS à combinação das tarefas alocadas à máquina na etapa 2 que resulte no menor tempo total.

	2ª Etapa Distribuição de tarefas	3ª Etapa <i>Tabu Search</i>	Enumeração total
Tempo total	635	620	613
Desvio em relação à enumeração total	3,6%	1,1%	--

Tabela 2.6 – Desempenho das etapas da heurística em relação à enumeração total ($\sum C_j$)

A Tabela 2.6 mostra que os desvios do $\sum C_j$ gerados pela segunda e terceira etapas da heurística, em relação à solução ótima, foram de 3,6% e 1,1%, respectivamente. Tal resultado demonstra a importância da inclusão de uma terceira etapa na abordagem proposta por Bank e Werner (2001) com vistas ao refino da sequência obtida na etapa 2. Observe ainda que as tarefas alocadas às máquinas pelo TS e pela enumeração diferem em 50% na máquina 1 (quatro entre oito tarefas) e 75% na máquina 2 (quatro entre seis tarefas). A função-objetivo de minimização do tempo total na presença de um fator de priorização ($\sum w_j C_j$) é de 130.567 para a máquina 1 e 101.543 para a máquina 2.

A função-objetivo de minimização do atraso ponderado das tarefas ($\sum w_j T_j$) é calculada separadamente em cada máquina após a obtenção da sequência de processamento gerada pela etapa 2 da heurística. Considerando uma data de entrega comum de 400, a sequência de tarefas gerada para as máquinas 1 e 2 apresenta, respectivamente, o valor de 11.045 e 5.526. Na sequência, aplica-se o TS para aprimorar a solução obtida pela distribuição de tarefas da etapa 2, o qual reduz o valor de $\sum w_j T_j$ da máquina 1 para 10.534. A Tabela 2.7 apresenta o desvio de $\sum w_j T_j$ gerado pelas etapas 2 e 3 da heurística em relação à enumeração total (9,7% e 4,6%, respectivamente).

	2ª Etapa	3ª Etapa	Enumeração
	Distribuição de tarefas	<i>Tabu Search</i>	total
Atraso total	11.045	10.534	10.066
Desvio em relação à enumeração total	9,7%	4,6%	--

Tabela 2.7 – Desempenho das etapas da heurística em relação à enumeração total ($\sum w_j T_j$)

2.5 Conclusão

A programação de tarefas é um processo de decisão com impacto direto no desempenho operacional e financeiro das empresas, oportunizando àquelas que a utilizam diferenciais competitivos no mercado que atuam. O presente artigo visou minimizar o tempo total ponderado de conclusão ($\sum w_j C_j$) e o atraso ponderado ($\sum w_j T_j$) de um conjunto de tarefas em um ambiente de manufatura de duas máquinas paralelas não-relacionadas com tempos determinísticos de processamento e de *setup*, este último dependente da sequência de processamento das tarefas.

A heurística proposta se apoia em três etapas: (i) ordenar as tarefas para alocação com base em uma regra semelhante ao WSPT, (ii) alocar as tarefas às máquinas, e (iii) aprimorar a solução obtida na etapa (ii) através do algoritmo *Tabu Search* (TS) simples de Glover (1989), realizando o intercâmbio adjacente de tarefas de uma mesma máquina. A heurística proposta engloba fatores qualitativos e quantitativos: o fator qualitativo de priorização (w), representando a agregação financeira por unidade de processamento, é determinante na ordenação das tarefas na etapa 1. As etapas 2 e 3 da heurística utilizam apenas fatores quantitativos do tempo de processamento e tempo de *setup*, quando existente, para obtenção das soluções.

Quando comparada à solução ótima obtida por enumeração total, a segunda etapa da heurística apresenta desvios de 3,6% e 9,7% para as funções $\sum w_j C_j$ e $\sum w_j T_j$, respectivamente.

Com a aplicação do TS, o desvio para o tempo de conclusão total é reduzido para 1,1% e o desvio da solução ótima para o atraso total é reduzido para 4,6%. Estes resultados expressam a influência do fator de priorização da etapa 1 no desvio das soluções obtidas pela heurística.

Contudo, a heurística está limitada ao ambiente de manufatura de máquinas paralelas não-relacionadas com tempos determinísticos de processamento e de *setup*. O mesmo problema, porém com tempos estocásticos de *setup* dependentes da sequência de processamento, passa a ser de interesse para futuras pesquisas. A utilização de funções-objetivo mais complexas, como a minimização simultânea do atraso e do adiantamento de um conjunto de tarefas sujeitas a um fator de priorização, também constitui-se em tema de interesse para o arranjo de máquinas paralelas não-relacionadas com tempos de *setup* determinísticos e dependentes da sequência de processamento.

2.6 Referências

- ALLAHVERDI, A.; NG, C. T.; CHENG, T. C. E. & KOVALYOV, M. Y.** *A survey of scheduling problems with setup times or costs.* European Journal of Operational Research 187 (2008) 985-1032
- ARNAOUT, J. P.; RABADI, G. & MUSA, R.** *A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times.* Journal of Intelligent Manufacturing 21 (2010) 693-701
- ARNAOUT, J. P.; MUSA, R. & RABADI, G.** *Ant colony optimization algorithm to parallel machine scheduling problem with setups.* 4th IEEE Conference on Automation Science and Engineering (2008) 578-582
- BANK, J. & WERNER, F.** *Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties.* Mathematical and Computer Modelling 33 (2001) 363-383
- BRUCKER, P.** *Scheduling algorithms.* Springer 2007.
- BRUCKER, P. & KNUST, S.** *Complex Scheduling.* Springer 2006.
- CHAO, C. W.; CHUANG, M. C. & LIAO, C. J.** *A heuristic for parallel machine scheduling with machine preference for the electroetching of aluminium foil.* Proceedings of the IEEE IEEM (2008) 345-348
- CHANG, P. C. & CHEN, S. H.** *Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times.* Applied Soft Computing 11 (2011) 1263-1274
- CHEN, C. & CHEN, C.** *Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times.* International Journal of Advanced Manufacturing Technology 43 (2009) 161-169
- CHEN, Z. & POWELL, W. B.** *Solving parallel machine scheduling problems by column generation.* INFORMS Journal on Computing 11 (1999) 78-94
- CORREA, J. R.; SKUTELLA, M. & VERSCHAE, J.** *The power of preemption on unrelated machines and*

applications to scheduling orders. LNCS 5687 (2009) 84-97

FANJUL-PEYRO, L. & RUIZ, R. *Iterated greedy local search methods for unrelated parallel machine scheduling*. European Journal of Operational Research 207 (2010) 55-69

_____. *Size-reduction heuristics for the unrelated parallel machines scheduling problems*. Computers & Operations Research 38 (2011) 301-309

GAIRING, M.; MONIEN, B. & WOCLAW, A. *A faster combinatorial approximation algorithm for scheduling unrelated parallel machines*. ICALP 2005, LNCS 3580 (2005) 828-839

GLOVER, F. *Tabu Search – Part I*. Journal of Computing v.1 no.3 (1989) 190-206

_____. *Tabu Search – Part II*. Journal of Computing v.2 no.1 (1990a) 4-32

_____. *Tabu Search: a tutorial*. Interfaces 20 (1990b) 74-94

GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K. & RINNOOY KAN, A. H. G. *Optimization and approximation in deterministic sequencing and scheduling: a survey*. Annals of Discrete Mathematics 5 (1979) 287-326

GRIGORIEV, A.; SVIRIDENKO, M. & UETZ, M. *Unrelated parallel machine scheduling with resource dependent processing times*. IPCO 2005, LNCS 3509 (2005) 182-195

HSU, C.; KUO, W. & YANG, D. *Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects*. Applied Mathematical Modelling 35 (2011) 1492-1496

KIM, D.; KIM, K.; JANG, W. & CHEN, F. F. *Unrelated parallel machine scheduling with setup times using simulated annealing*. Robotics and Computer Integrated Manufacturing 18 (2002) 223-231

KIM, D.; NA, D. & CHEN, F. F. *Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective*. Robotics and Computer Integrated Manufacturing 19 (2003) 173-181

LI, K. & YANG, S. *Non-identical parallel-machine scheduling research with minimizing total weighted completion times: Models, relaxation and algorithms*. Applied Mathematical Modelling 33 (2009) 2145-2158

LIAW, C. F.; LIN, Y. K.; CHENG, C. Y. & CHEN, M. *Scheduling unrelated parallel machines to minimize total weighted tardiness*. Computers & Operations Research 30 (2003) 1777-1789

LIN, Y. K.; PFUND, M. E. & FOWLER, J. W. *Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems*. Computers & Operations Research 38 (2011) 901-916

LOGENDRAN, R.; McDONELL, B. & SMUCKER, B. *Scheduling unrelated parallel machines with sequence-dependent setups*. Computers & Operations Research 34 (2007) 3420-3438

LÓPEZ, O. C.; BARCIA, R. M. & EYADA, O. *Problema de programação da produção: um esquema de classificação*. Revista Produção 5 (1995) 145-168

MOKOTOFF, E. *Parallel machine scheduling problems: a survey*. Asia-Pacific Journal of Operational Research 18 (2001) 193-242

MORTON, T. E. & PENTICO, D. W. *Heuristic scheduling systems: with applications to production systems*

and project management. New York: John Wiley & Sons 1993.

PAULA, M. R.; MATEUS, G. R. & RAVETTI, M. G. *A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times*. *Computers & Operations Research* 37 (2010) 938-949

PFUND, M.; FOWLER, J. W. & GUPTA, J. N. D. *A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems*. *Journal of the Chinese Institute of Industrial Engineers*, Vol. 21, No 3, (2004) 230-241

PINEDO, M. L. *Scheduling: Theory, Algorithms, and Systems*. New York: Springer 2008.

PLATEAU, M. C. & RIOS-SOLIS, Y. A. *Optimal solutions for unrelated parallel machines scheduling problems using convex quadratic reformulations*. *European Journal of Operational Research* 201 (2010) 729-736

RABADI, G.; MORAGA, R. J. & AL-SALEM, A. *Heuristics for the unrelated parallel machine scheduling problem with setup times*. *Journal of Intelligent Manufacturing* 17 (2006) 85-97

SHIM, S. *Generating subproblems in branch and bound algorithms for parallel machines scheduling problems*. *Computers & Industrial Engineering* 57 (2009) 1150-1153

SILVA, C. & MAGALHÃES, J. M. *Heuristic lot size scheduling on unrelated parallel machines with applications on the textile industry*. *Computers & Industrial Engineering* 50 (2006) 76-89

SRIVASTAVA, B. *An effective heuristic for minimising makespan on unrelated parallel machines*. *Journal of the Operational Research Society* 49 (1998) 886-894

WENG, M. X.; LU, J. & REN, H. *Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective*. *International Journal of Production Economics* 70 (2001) 215-226

WU, Y. & JI, P. *A scheduling problem for PCB assembly: a case with multiple lines*. *International Journal of Manufacturing Technology* 43 (2009) 1189-1201

YING, K. C.; LEE, Z. J. & LIN, S. W. *Makespan minimization for scheduling unrelated parallel machines with setup times*. *Journal of Intelligent Manufacturing* (2010) Publicação online

3 SEGUNDO ARTIGO

MINIMIZAÇÃO SIMULTÂNEA DO ADIANTAMENTO E ATRASO DE TAREFAS EM MÁQUINAS PARALELAS NÃO-RELACIONADAS COM TEMPOS DE *SETUP* DEPENDENTES DA SEQUÊNCIA DE TAREFAS

Resumo

Este artigo propõe uma heurística para programação de tarefas baseada na regra ATCS, de Lee e Pinedo (1997), para a minimização da soma ponderada do adiantamento e atraso de tarefas com tempos de preparação dependentes da sequência de processamento em máquinas paralelas não-relacionadas. A heurística é operacionalizada em três etapas: (i) caracterização do conjunto de tarefas, (ii) alocação das tarefas às máquinas, e (iii) refino da solução por *Tabu Search*. A solução gerada pela heurística é comparada à solução ótima obtida por enumeração em dados reais e em cenários simulados com distintos tempos de processamento e preparação, data de entrega e fator de priorização. A heurística gerou um desvio médio de 14% em relação à solução ótima para dados reais de um processo de metalização de filmes plásticos em alto vácuo e 18% para as simulações, onde em 70% dos casos a heurística obteve a solução ótima do problema em pelo menos uma máquina.

Palavras-chaves: programação de tarefas, máquinas paralelas não-relacionadas, tempos de setup, adiantamento, atraso

Abstract

This article proposes a heuristic based on a modified version of Lee and Pinedo's (1997) ATCS to minimize the sum of weighted earliness and tardiness of unrelated parallel machines with sequence dependent setup times. A three step heuristic relies on the following steps: (i) characterisation of the jobs, (ii) allocation of jobs to machines, and (iii) solution improvement with Tabu Search. Solutions yielded by the heuristic are compared to the optimal solution obtained from enumeration for real and simulated data in different scenarios of processing and setup times, due dates and weight. Results

show 14% of deviation from optimal solution for real data and 18% for the simulations, where the optimal solution was obtained in at least one machine in 70% of the simulations.

Keywords: *job scheduling, unrelated parallel machines, setup times, earliness, tardiness*

3.1 Introdução

A programação de tarefas (sequenciamento) é um processo de fundamental importância na maximização do desempenho operacional e financeiro das empresas. Pela sua grande aplicabilidade, a eficiente programação de tarefas se constitui em importante diferencial competitivo, razão pela qual tem sido amplamente pesquisada em diversos ambientes de manufatura.

Utilizada regularmente nas empresas manufatureiras e de serviços, a programação de tarefas se constitui no processo de organização e escolha do recurso mais adequado para a execução de uma tarefa, considerando restrições de tempo e de relacionamento entre tarefas, visando aprimorar um ou mais objetivos. Dentre tais objetivos, a (i) minimização dos custos de estocagem decorrentes da conclusão adiantada de tarefas, e (ii) a minimização das penalidades impostas por atrasos são objetivos recorrentes em empresas que aplicam os conceitos de produção *just-in-time* (SCHUTTEN & LEUSSINK, 1996; LEE & PINEDO, 1997; SIVRIKAYA-SERIFOGLU & ULUSOY, 1999; OMAR & TEO, 2006; JEONG & KIM, 2008; RIOS-SOLIS & SOURD, 2008; CHEN, 2009; RAGHAVAN & VENKATARAMANA, 2009; GACIAS et al., 2010; LEE et al., 2010).

Dentre os diversos sistemas que visam minimizar a soma do atraso e adiantamento de um conjunto de tarefas, destacam-se os ambientes constituídos por máquinas em paralelo. Tais arranjos são especialmente atrativos tanto do ponto de vista teórico como do ponto de vista prático. No aspecto teórico, os ambientes de máquinas em paralelo constituem-se num caso geral do ambiente de máquina simples e, na visão prática, representam muitos ambientes de manufatura reais (MOKOTOFF, 2001; LIAW et al., 2003; PINEDO, 2008; LIN et al., 2011). Problemas associados a arranjos de máquinas não-relacionadas em paralelo são gerados pela necessidade de renovação do maquinário existente ou aumento da capacidade instalada, visto que máquinas de gerações tecnológicas distintas possuem velocidades de operação e tempos de preparação peculiares.

A pesquisa relacionada a máquinas paralelas não-relacionadas sujeitas a tempos de preparação dependentes da sequência apresenta diversas lacunas por conta de sua complexidade matemática, especialmente quando funções-objetivo mais complexas são consideradas. Uma dessas funções refere-se à minimização simultânea do atraso e do adiantamento das tarefas, a qual se apresenta pouco explorada e se constitui em oportunidade de pesquisas (PFUND et al., 2004).

Este artigo apresenta uma nova heurística para minimização da soma ponderada do atraso e adiantamento de um conjunto de tarefas com tempos de *setup* dependentes da sequência de processamento. Para tanto, modifica-se a heurística ATCS de Lee e Pinedo (1997). Diferentemente do ATCS original, que refina a solução final com *Simulated Annealing* (SA) para minimizar a soma ponderada do atraso de máquinas paralelas idênticas, a heurística proposta refina a solução final através do algoritmo *Tabu Search* (TS). De tal forma, este artigo apoia-se em uma ferramenta de otimização mais simples que o SA, porém investe em um arranjo de máquina mais complexo: um ambiente composto por máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência de processamento, além de abordar uma função-objetivo mais complexa que a originalmente estudada por Lee e Pinedo (1997).

A heurística proposta é aplicada em dados reais e em cenários simulados de um processo de manufatura de componentes eletrônicos composto por máquinas paralelas não-relacionadas com tempos de preparação dependentes da sequência de tarefas. Nas duas situações o desempenho da heurística proposta é comparado ao resultado ótimo obtido através de enumeração total.

O artigo está estruturado como segue: a seção 3.2 apresenta o referencial teórico sobre programação de tarefas em ambiente de manufatura de máquinas paralelas não-relacionadas. As seções 3.3 e 3.4 tratam, respectivamente, do método de trabalho e da discussão dos resultados. Por fim, a seção 3.5 apresenta a conclusão da pesquisa e direções futuras.

3.2 Referencial Teórico

Em programação de tarefas, usualmente se utiliza a denominação “problema” para definir genericamente a necessidade de otimizar uma função-objetivo relacionada a um ambiente de manufatura (PINEDO, 2008). O problema de programação de tarefas em máquinas paralelas consiste na atribuição de n tarefas (cada qual com seu tempo de processamento) a m máquinas, com o objetivo de otimizar uma medida de desempenho (MOKOTOFF, 2001).

Ambientes de manufatura compostos por máquinas paralelas podem ser categorizados em três classes: (i) máquinas paralelas idênticas – tempo de processamento das tarefas é o mesmo em qualquer uma das máquinas, (ii) máquinas paralelas uniformes – tempo de processamento da tarefa é dependente da velocidade da máquina que irá processar a tarefa, e (iii) máquinas paralelas não-relacionadas - tempo de processamento da tarefa é dependente, simultaneamente, da velocidade da máquina e da natureza da tarefa a ser executada (CHEN & POWELL, 1999; MOKOTOFF, 2001; CHAO et al., 2008; PINEDO, 2008; LI & YANG, 2009; WU & JI, 2009; FANJUL-PEYRO & RUIZ, 2011; LIN et al., 2011). A representação de tais arranjos segue a classificação de três campos proposta por Graham et al. (1979). Nesta representação, o primeiro campo identifica o ambiente das máquinas, podendo assumir a representação P (máquinas paralelas idênticas), Q (máquinas paralelas uniformes) ou R (máquinas paralelas não-relacionadas). O segundo campo identifica restrições associadas às tarefas, as quais podem incluir o tempo de preparação (*setup*) dependente da sequência (S_{jk}), a possibilidade de interrupção do processamento de uma tarefa (*preemption*), e a presença de um recurso restritivo, entre outros (GRAHAM et al., 1979). O terceiro campo especifica o critério de otimização escolhido, normalmente com o objetivo de minimização, podendo incluir o atraso de entrega de uma tarefa j , T_j (*tardiness*), o adiantamento de conclusão de uma tarefa j , E_j (*earliness*), ou uma combinação destes (GRAHAM et al., 1979). Um fator de importância ou de priorização (*weight*) é identificado através da letra w , posicionado antes do critério a ser otimizado.

Abordagens de sequenciamento em ambientes compostos por máquinas paralelas não-relacionadas têm endereçado diversas funções-objetivo. Chen e Powell (1999) propuseram uma sistemática para minimização do número total de tarefas atrasadas sem restrições e com fator de priorização ($R_m // \sum w_j U_j$), enquanto que Chen e Chen (2009) estenderam a mesma heurística para cenários afetados por tempos de preparação dependentes das tarefas ($R_m / S_{jk} / \sum w_j U_j$), propondo metaheurísticas baseadas em vizinhança variável (VV) e *Tabu Search* (TS). Complementarmente, Tavakkoli-Moghaddam et al. (2009) abordaram o problema ($R_m / r_j ; d_j ; S_{jk} / \sum U_j$) utilizando algoritmo genético (AG).

Outra função-objetivo amplamente abordada consiste na minimização do atraso de entrega das tarefas. Lin et al. (2011) e Liaw et al. (2003) propuseram heurísticas de dois estágios baseadas em *apparent tardiness cost* (ATC) e Branch & Bound (B&B) para o problema $R_m // \sum w_j T_j$. A minimização da soma do atraso total das tarefas com tempos de preparação dependentes das tarefas sem fator de priorização ($R_m / S_{jk} / \sum T_j$) foi estudada por Chen

(2009), que utilizou uma heurística baseada no *apparent tardiness cost with setup* (ATCS) e *Simulated Annealing* (SA), por Kim et al. (2002), que utilizaram uma metaheurística baseada em SA, e por Silva e Magalhães (2006), que adaptaram diversas heurísticas para a minimização do atraso em uma empresa de fibras acrílicas. Já a minimização da soma do atraso total ponderado das tarefas com tempos de preparação dependentes das tarefas ($R_m / S_{jk} / \sum w_j T_j$) foi abordada por Paula et al. (2010), os quais propuseram um algoritmo baseado em relaxação Lagrangeana, e por Logendran et al. (2007), que desenvolveram seis algoritmos baseados em TS. Por sua vez, Kim et al. (2003) incluíram a restrição de datas de entrega na formulação ($R_m / S_{jk} ; d_j / \sum w_j T_j$), comparando o desempenho de quatro heurísticas de busca baseadas em regras de ordenação.

Focados na mesma função-objetivo e utilizando regras de despacho de tarefas, Lee et al. (1997) propuseram a heurística ATCS para máquinas simples como aprimoramento da regra ATC originalmente proposta por Vepsalainen e Morton (1987) e Ow e Morton (1989). Tal problema consistia na minimização da soma do atraso ponderado total e com tempos de preparação dependentes da sequência ($1 / S_{jk} / \sum w_j T_j$). Posteriormente, Lee e Pinedo (1997) propuseram o ATCS para máquinas paralelas idênticas com o mesmo objetivo ($P_m / S_{jk} / \sum w_j T_j$). Tal heurística é executada em três fases: pré-processamento, sequenciamento e pós-processamento. Na fase de pré-processamento, as tarefas são caracterizadas por métricas e o tempo total de conclusão das tarefas é estimado. A segunda fase sequencia as tarefas nas máquinas de acordo com o cálculo do índice de priorização de cada tarefa e, por fim, a fase de pós-processamento utiliza SA para aprimorar a solução gerada na segunda fase (LEE & PINEDO, 1997).

Recentemente, Pfund et al. (2008) propuseram o *apparent tardiness cost with setups and ready times index* (ATCSR) para o problema $P_m / r_j ; S_{jk} / \sum w_j T_j$, o qual adiciona ao índice de priorização calculado pelo ATCS um termo referente ao momento de disponibilização de uma tarefa para processamento. Utilizando projeto de experimentos, os autores comprovaram o desempenho superior do ATCSR em comparação ao ATCS e a outras heurísticas disponíveis, dentre as quais o *batch apparent tardiness cost with setups and ready times* (BATCS), o BATCS modificado de Mason et al. (2002), o *X-dispatch* (X-RM) de Rachamadugu e Morton (1982) e o X-RM modificado de Morton e Pentico (1993).

Consolidando a sequência de desdobramentos oriundos do ATC para máquinas paralelas não-relacionadas, Driessel e Mönch (2010) propuseram uma busca em VV utilizando a solução gerada pelo ATCSR com solução inicial. Similarmente, Lamothe et al. (2010) propuseram o

apparent tardiness cost with setups and overall equipment effectiveness (ATCSOEE) e o *apparent tardiness cost with setups and overall equipment effectiveness flexibility* (ATCSOEEF), os quais consideram no índice de priorização calculado o valor agregado à máquina com a alocação de uma tarefa e a flexibilidade do equipamento para seu processamento.

Uma das funções-objetivo associadas a máquinas paralelas não-relacionadas que tem recebido crescente atenção da literatura (e abordada neste artigo) é a minimização da soma ponderada do adiantamento e atraso de um conjunto de tarefas, com tempos de *setup* dependentes da sequência de processamento $[R_m / S_{jk} / \sum (w_j E_j + w_j T_j)]$. Tal problema é classificado como *NP-hard*, ou seja, requer um tempo de processamento não-polinomial (exponencial) para obtenção da solução ótima. Dentre as abordagens propostas para esse problema, destaca-se a de Javadi e Tabrizi (2010), a qual utiliza uma metaheurística híbrida baseada em SA e AG; os resultados superaram os resultados da aplicação isolada do AG. Com propósitos semelhantes, Plateau e Rios-Solis (2010) incluíram fatores de priorização e datas distintas de liberação para as tarefas $(R_m / r_j / \sum w_j (E_j + T_j))$, integrando programação quadrática ao algoritmo B&B. Não foram encontradas na literatura, no entanto, abordagens que utilizem adaptações da regra ATCS para a minimização do sistema de interesse deste artigo, $R_m / S_{jk} / \sum (w_j E_j + w_j T_j)$.

3.3 Método de trabalho

Considere o problema de programação de n tarefas em m máquinas paralelas não-relacionadas com tempos de preparação dependentes da sequência de processamento. As tarefas possuem uma data comum de liberação e seu processamento, uma vez iniciado, não pode ser interrompido. O objetivo considerado é a minimização da soma ponderada do atraso e do adiantamento de um conjunto de tarefas $(R_m / S_{jk} / \sum w_j (E_j + T_j))$, em um ambiente de máquinas paralelas não-relacionadas.

A heurística proposta é executada em três etapas. A primeira etapa é semelhante à heurística ATCS de Lee e Pinedo (1997), ou seja, são calculadas métricas para caracterização das tarefas e uma estimativa do tempo total de conclusão das mesmas. A segunda etapa aloca as tarefas às máquinas utilizando o índice de priorização calculado a partir do tempo de processamento p e tempo médio de processamento \bar{p} da máquina disponível, priorizando a alocação da tarefa com o maior índice. Na terceira etapa, as tarefas alocadas a cada máquina têm seu ordenamento refinado através do TS. As etapas da heurística proposta são detalhadas na sequência.

3.3.1 Etapa 1 – Caracterização das tarefas

A primeira etapa da heurística caracteriza as tarefas através de métricas associadas ao prazo de entrega, ao tempo de preparação (*setup*) e à quantidade de tarefas por máquina. As métricas associadas ao prazo de entrega são a amplitude R e o fator de estrangulamento τ , calculados pelas equações (1) e (2). Os termos d_{max} , d_{min} , \bar{d} e C_{max} se referem, respectivamente, ao maior prazo de entrega, menor prazo de entrega, prazo de entrega médio e ao tempo de conclusão de todas as tarefas (LEE & PINEDO, 1997; LEE et al., 1997).

$$R = \frac{(d_{max} - d_{min})}{C_{max}} \quad (1)$$

$$\tau = 1 - \left(\frac{\bar{d}}{C_{max}} \right) \quad (2)$$

A amplitude R é uma medida do espalhamento das datas de entrega em relação ao tempo de conclusão das tarefas, enquanto que o fator τ indica o nível de estrangulamento com relação ao cumprimento das datas de entrega; quanto mais próximo de 1, mais estrangulada está a tarefa. Com relação ao tempo de preparação, o fator de severidade η , calculado pela equação (3), expressa a proporção do tempo médio de preparação \bar{s} em relação ao tempo médio de processamento \bar{p} das tarefas (LEE & PINEDO, 1997; LEE et al., 1997). No caso de máquinas paralelas não-relacionadas, o fator de severidade η considera a média dos tempos médios de processamento \bar{p} de cada máquina.

$$\eta = \frac{\bar{s}}{\bar{p}} \quad (3)$$

O fator tarefa-máquina μ , calculado através da equação (4), indica a densidade média esperada de tarefas por máquina. O tempo de conclusão de todas as tarefas (C_{max}), calculado pela equação (5), é estimado por uma proporção entre os tempos médios de preparação e da média dos tempos médios de processamento das tarefas. O fator β , obtido através da equação (6), considera a influência do tempo de preparação sobre a estimativa de C_{max} para $\mu \geq 5$, critério este determinado em Lee & Pinedo através de experimentos empíricos.

$$\mu = \frac{n}{m} \quad (4)$$

$$C_{\max} = \left(\beta \cdot \bar{s} + \bar{p} \right) \cdot \mu \quad (5)$$

$$\beta = 0,4 + \frac{10}{2} \cdot \frac{\eta}{\mu} \quad (6)$$

3.3.2 Etapa 2 – Alocação das tarefas às máquinas

Nesta etapa a heurística aloca as tarefas às máquinas, com base no índice de priorização do ATCS, calculado pela equação (7). O cálculo do índice de priorização considera a influência de três termos: o termo WSPT (*Weighted Shortest Processing Time*), o termo de folga e o termo de preparação. O termo WSPT busca priorizar tarefas de elevado fator de priorização e reduzido tempo de processamento. O termo de folga privilegia as tarefas em atraso, ou com menor tempo de folga, e o termo de preparação prioriza as tarefas com menor tempo de preparação. O índice $I_k(t, j)$ representa o índice de priorização da tarefa k no tempo t dado que a tarefa j foi a última a ser processada na máquina em questão (PFUND et al., 2008).

$$I_k(t, j) = \underbrace{\frac{w_k}{p_k}}_{\text{Termo WSPT}} \cdot e^{\underbrace{\left(\frac{\max(d_k - p_k - t, 0)}{k_1 \cdot \bar{p}} \right)}_{\text{Termo de folga}}} \cdot e^{\underbrace{\left(-\frac{s_{jk}}{k_2 \cdot \bar{s}} \right)}_{\text{Termo de preparação}}} \quad (7)$$

Os termos de folga e de preparação são dependentes dos parâmetros de escala k_1 e k_2 . Estes, por sua vez, são dependentes dos valores de τ , R , η e μ . Os parâmetros de escala k_1 e k_2 regulam a influência do termo de folga e do termo de preparação no cálculo do índice de priorização, como mostra o Quadro 3.1.

k_1	k_2	Influência do termo de folga em relação ao termo de preparação	Influência do termo de preparação em relação ao termo de folga
>1	<1	maior	menor
<1	>1	menor	maior
1	1	igual	igual

Quadro 3.1 – Relações de influência entre os termos de folga e de preparação em função dos valores de k_1 e k_2

Lee e Pinedo (1997), através de experimentos, determinaram relações empíricas para a estimativa de k_1 e k_2 , expressos pelas equações (8) e (9). A determinação dos valores de k_1 e k_2 é obtida seguindo as etapas do Quadro 3.2

$$k_1 = 1,2 \cdot \ln(\mu) - R \quad (8)$$

$$k_2 = \frac{\tau}{A_2 \cdot \sqrt{\eta}} \quad (9)$$

Passos	Determinação de k_1	Determinação de k_2
Passo 1	Aplique a equação (8).	Se $\tau < 0,8$, $A_2 = 1,8$. Caso contrário $A_2 = 2$.
Passo 2	Subtraia 0,5 de k_1 se $\tau < 0,5$ ou se $\eta < 0,5$ e $\mu > 5$.	Aplique a equação (9)

Quadro 3.2 – Passos da determinação do valor de k_1 e de k_2

Na sequência, procede-se à primeira alocação de tarefa. Para tanto, o $I_k(t, j)$ de todas as tarefas, considerando todas as máquinas, é calculado e a tarefa com o maior índice é alocada à máquina com maior valor de $I_k(t, j)$. Repare que o tempo de preparação é considerado nulo na alocação da primeira tarefa de cada máquina.

Após a alocação da primeira tarefa a cada máquina, as próximas tarefas candidatas à alocação têm seu $I_k(t, j)$ atualizado, considerando o instante t no qual uma das máquinas estará disponível. A partir deste instante, o $I_k(t, j)$ leva em conta o tempo de preparação dependente da tarefa anterior e os tempos médios de preparação e processamento das tarefas restantes. O ciclo se repete até que todas as tarefas tenham sido alocadas às máquinas. Esses passos são resumidos no Quadro 3.3.

Sequência	Atividade
Passo 1	Com as tarefas não alocadas calcule \bar{p} , \bar{s} , k_1 e k_2
Passo 2	Calcule o índice de priorização $I_k(t, j)$ de cada tarefa
Passo 3	Aloque a tarefa com maior $I_k(t, j)$ à primeira máquina disponível em t
Passo 4	Retorne ao passo 1 até que todas as tarefas tenham sido alocadas às máquinas

Quadro 3.3 – Passos da etapa 2 da heurística

Após a alocação de todas as tarefas às máquinas, o valor de $\sum w_j(E_j + T_j)$ de cada máquina é calculado pela equação (10). A solução gerada nesta etapa será refinada na etapa seguinte.

$$\sum_{j=1}^n w_j \cdot (E_j + T_j) = \sum_{j=1}^n w_j \cdot (\max(d_j - C_j, 0) + \max(C_j - d_j, 0)) \quad (10)$$

3.3.3 Etapa 3 – Refino na ordenação das tarefas alocadas às máquinas

Nesta etapa, a solução da etapa 2 é refinada pela reordenação das tarefas alocadas a cada máquina. O refinamento é executado pelo *Tabu Search* (TS) simples de Glover (1989), o qual testa a vizinhança da solução gerada pela etapa 2. A função-objetivo $\sum w_j (E_j + T_j)$ de cada vizinhança é calculada e assume como solução inicial do TS a vizinhança com a menor função-objetivo. A sistemática de operacionalização do TS é agora detalhada.

O TS é um procedimento para solução de problemas combinatoriais baseado na busca da solução ótima global na vizinhança da solução atual, através do intercâmbio de duas tarefas adjacentes. Partindo de uma solução inicial qualquer o TS realiza o intercâmbio de tarefas adjacentes, gerando $n-1$ vizinhanças adjacentes à solução inicial. Para cada uma das vizinhanças geradas, calcula-se a função-objetivo. A vizinhança com menor (maior) valor, no caso de problemas de minimização (maximização), é escolhida para prosseguir na busca da solução ótima na próxima iteração do algoritmo (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008).

A cada iteração, o par de tarefas no qual o intercâmbio gerou a vizinhança com menor (maior) valor da função-objetivo é armazenado em uma lista de movimentos proibidos (lista tabu) de serem realizados nas iterações seguintes, de modo a evitar que a busca retorne a vizinhanças anteriormente visitadas e cujo valor da função-objetivo é superior (inferior) ao atual. A lista tabu possui um número fixo de entradas e a cada novo par de tarefas entrante na lista, o par mais antigo é excluído e seu intercâmbio pode ser realizado novamente nas próximas iterações (GLOVER, 1989; 1990a; 1990b; MORTON & PENTICO, 1993; BRUCKER & KNUST, 2006; BRUCKER, 2007; PINEDO, 2008). O TS realiza um determinado número de iterações e retorna a solução de menor valor da função objetivo encontrada.

Ao final, os resultados gerados pelas etapas 2 e 3 da heurística são comparados à solução ótima obtida por enumeração total. O Quadro 3.4 sumariza os passos da terceira etapa da heurística.

Sequência	Atividade
Passo 1	Selecionada a solução inicial do TS da vizinhança com menor valor da função-objetivo
Passo 2	Aplica <i>Tabu Search</i> para refinamento da solução inicial
Passo 3	Compara a solução final com a solução ótima obtida por enumeração

Quadro 3.4 – Passos da etapa 3 da heurística

3.4 Resultados e Discussão

A heurística proposta foi inicialmente aplicada em dados reais de um ambiente de manufatura composto por duas máquinas paralelas não-relacionadas, no processo de metalização de filmes plásticos em alto vácuo. O processo de metalização consiste na deposição de uma camada de metal sobre o filme plástico com a finalidade de obter determinadas características elétricas ou de aparência.

Na sequência, a heurística foi aplicada em cenários simulados utilizando distribuições uniformes para gerar tempos de processamento e de *setup*, fator de priorização e data de entrega das tarefas. As distribuições uniformes, segundo Fanjul-Peyro e Ruiz (2010), são as distribuições mais recomendadas para geração de instâncias de teste que permitem o estabelecimento de um padrão de comparação entre diferentes heurísticas.

3.4.1 Aplicação em Dados Reais

A heurística foi aplicada a tempos reais de processamento, fator de priorização, data de entrega e tempos de preparação de 14 tarefas e duas máquinas paralelas não-relacionadas (ver Apêndice). A Tabela 3.1 apresenta os valores da função-objetivo resultantes das etapas 2 (ATCS) e 3 (*Tabu Search*) da heurística proposta, os quais são comparados com o resultado ótimo obtido por enumeração total. Após o refino através do *Tabu Search*, obtém-se um desvio de 14,4% em relação à solução ótima.

$\sum w_j(E_j + T_j)$	ATCS	<i>Tabu Search</i>	Enumeração
Máquina 1	69.346	36.305	31.740
Máquina 2	70.926	50.708	19.610
Menor desvio em relação à enumeração	118,5%	14,4%	--

Tabela 3.1 – Desempenho das etapas da heurística em relação à enumeração

3.4.2 Aplicação em Cenários Simulados

A aplicação da heurística em cenários simulados utilizou distribuições uniformes para geração dos tempos de processamento e *setup*, fator de priorização e data de entrega. Os parâmetros superior e inferior das distribuições, apresentados na Tabela 3.2, foram utilizados na geração de 400 cenários distintos de simulação.

Fatores simulados	Intervalo da distribuição uniforme
Tempo de processamento	40 a 120
Tempo de <i>setup</i>	1 a 20
Fator de priorização	20 a 250
Data de entrega	60 a 400

Tabela 3.2 – Intervalos utilizados na geração das simulações

A distribuição do número de tarefas alocadas por máquina, resultante da execução da segunda etapa da heurística (com base no índice de priorização do ATCS), é apresentada na Figura 3.1.

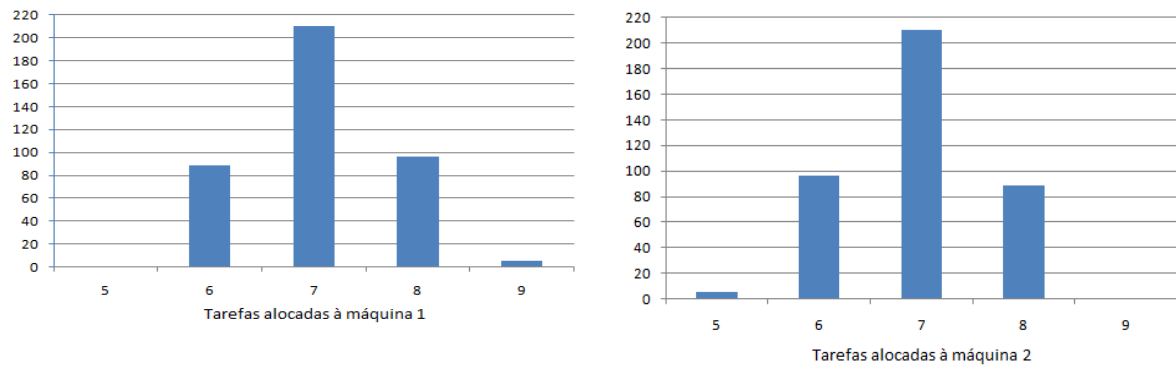


Figura 3.1 – Distribuição de tarefas alocadas por máquina

A ordenação proposta pela regra ATCS não gerou solução ótima em nenhuma das replicações. Os valores da função-objetivo apresentam desvio médio de 120,9% em relação à solução ótima apontada pela enumeração. A Figura 3.2 mostra a frequência dos desvios da função-objetivo da etapa 2 da heurística em relação à enumeração.

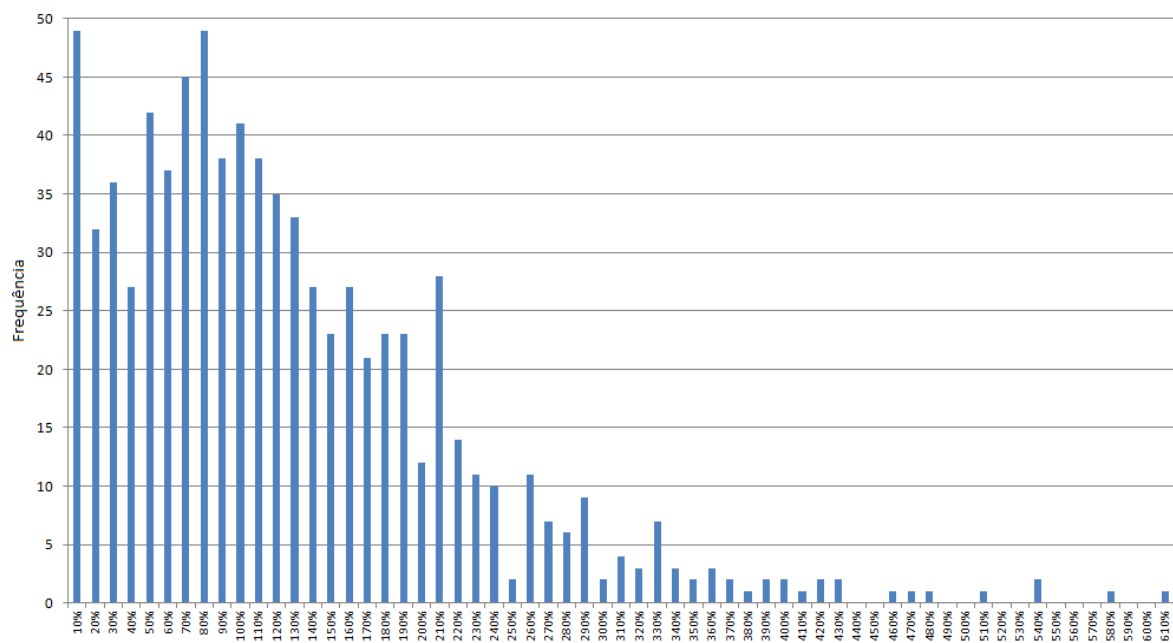


Figura 3.2 – Desvio da solução da etapa 2 da heurística em relação à enumeração

Na sequência, o TS é aplicado para aprimorar a solução obtida na etapa 2. O desvio médio em relação à enumeração é reduzido para 17,9%. A Figura 3.3 mostra a distribuição dos desvios das soluções aprimoradas pelo TS.

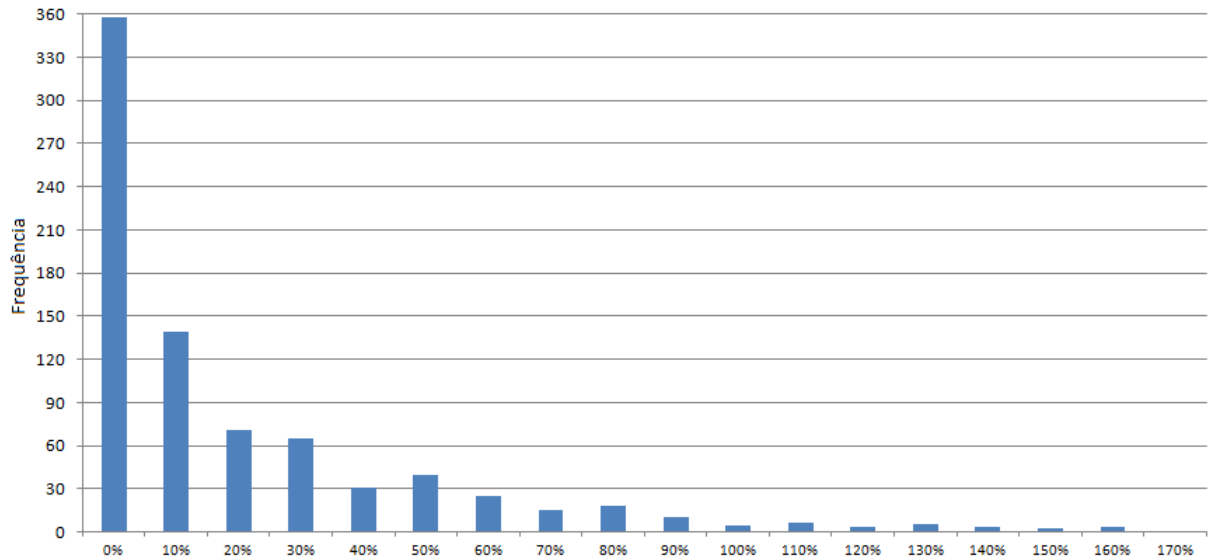


Figura 3.3 – Desvio da solução do TS em relação à enumeração

A Tabela 3.3 mostra o desvio médio das soluções geradas pelo ATCS e pelo TS em relação à enumeração e seus respectivos desvios padrões.

Desvio em relação à enumeração	ATCS	TS
Média	120,9%	17,9%
Desvio Padrão	0,95	0,30

Tabela 3.3 – Desvios em relação à enumeração

O TS reduziu sensivelmente o desvio médio em relação à enumeração, tornando-o comparável em ordem de grandeza ao desvio obtido com a aplicação da heurística aos dados reais. Da mesma forma, o desvio padrão das soluções foi reduzido, diminuindo a dispersão das soluções obtidas anteriormente pelo ATCS.

Contudo, o TS não obteve a solução ótima para a totalidade de casos simulados. A Tabela 3.4 mostra que a solução ótima foi obtida em pelo menos uma máquina em 70,75% dos casos. Nas demais simulações, o TS aprimorou a solução anteriormente obtida pela segunda etapa da heurística, porém sem obter a solução ótima.

Número de máquinas com solução ótima	Nenhuma máquina	1 máquina	2 máquinas
% de ocorrência	29,25%	52,00%	18,75%

Tabela 3.4 – Proporção de soluções ótimas por máquina obtida pelo TS

3.5 Conclusão

A programação de tarefas é um processo de decisão estratégico de impacto direto no desempenho operacional e financeiro das empresas, oportunizando àquelas que a utilizam a diferenciais competitivos no mercado que atuam. Neste sentido, o trabalho visou minimizar a função-objetivo do adiantamento e do atraso total ($\sum w_j(E_j + T_j)$) de um conjunto de tarefas sujeitas a um fator de priorização, em um ambiente de manufatura de duas máquinas paralelas não relacionadas com tempos de *setup* dependentes da sequência de processamento das tarefas.

A heurística proposta consiste em três etapas: (i) caracterização do conjunto de tarefas, (ii) alocação das tarefas às máquinas utilizando uma versão modificada da heurística ATCS de Lee e Pinedo (1997), e (iii) aprimoramento da solução obtida na etapa 2 com *Tabu Search* (TS) simples de Glover (1989), realizando o intercâmbio adjacente de tarefas de uma mesma máquina.

A heurística foi aplicada a dados reais e em cenários simulados de tempos de preparação e de *setup*, fator de priorização e data de entrega. Quando comparada com a solução ótima obtida por enumeração, a heurística apresenta desvio de 14,4% para dados reais compostos por um conjunto de quatorze tarefas duas máquinas, e desvio médio de 17,9% para os cenários simulados. A solução ótima em pelo menos uma máquina foi obtida em 70,75% dos casos.

Desdobramentos futuros incluem a aplicação da heurística proposta em arranjos multiestágio e sujeitos a tempos de *setup* estocásticos.

3.6 Referências

- ALLAHVERDI, A.; NG, C. T.; CHENG, T. C. E. & KOVALYOV, M. Y. *A survey of scheduling problems with setup times or costs*. European Journal of Operational Research 187 (2008) 985-1032
- BRUCKER, P. *Scheduling algorithms*. Springer 2007.
- BRUCKER, P. & KNUST, S. *Complex Scheduling*. Springer 2006.
- CHAO, C. W.; CHUANG, M. C. & LIAO, C. J. *A heuristic for parallel machine scheduling with machine preference for the electroetching of aluminium foil*. Proceedings of the IEEE IEEM (2008) 345-348

- CHEN, C. & CHEN, C.** *Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times.* International Journal of Advanced Manufacturing Technology 43 (2009) 161-169
- CHEN, J. F.** *Scheduling on unrelated parallel machines with sequence- and machine-dependent setup times and due-date constraints.* International Journal of Advanced Manufacturing Technology 44 (2009) 1204-1212
- CHEN, Z. & POWELL, W. B.** *Solving parallel machine scheduling problems by column generation.* INFORMS Journal on Computing 11 (1999) 78-94
- DRIESSEL, R. & MÖNCH, L.** *Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times.* Computers & Industrial Engineering (2010) Article in press
- FANJUL-PEYRO, L. & RUIZ, R.** *Iterated greedy local search methods for unrelated parallel machine scheduling.* European Journal of Operational Research 207 (2010) 55-69
- _____. *Size-reduction heuristics for the unrelated parallel machines scheduling problems.* Computers & Operations Research 38 (2011) 301-309
- GACIAS, B.; ARTIGUES, C. & LOPEZ, P.** *Parallel machine scheduling with precedence constraints and setup times.* Computers & Operations Research 37 (2010) 2141-2151
- GLOVER, F.** *Tabu Search – Part I.* Journal of Computing v.1 no.3 (1989) 190-206
- _____. *Tabu Search – Part II.* Journal of Computing v.2 no.1 (1990a) 4-32
- _____. *Tabu Search: a tutorial.* Interfaces 20 (1990b) 74-94
- GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K. & RINNOOY KAN, A. H. G.** *Optimization and approximation in deterministic sequencing and scheduling: a survey.* Annals of Discrete Mathematics 5 (1979) 287-326
- JAVADI, M. M. & TABRIZI, B. B.** *A hybrid metaheuristic method to minimize cost on non-identical parallel machines.* 2nd International Conference on Software Technology and Engineering (ICSTE) (2010) 287-291
- JEONG, S. J. & KIM, K. S.** *Parallel machine scheduling with earliness-tardiness penalties and space limits.* International Journal of Advanced Manufacturing Technology 37 (2008) 793-802
- KIM, D.; NA, D. & CHEN, F. F.** *Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective.* Robotics and Computer Integrated Manufacturing 19 (2003) 173-181
- LAMOTHE, J.; MARMIER, F.; DUPUY, M.; GABORIT, P. & DUPONT, L.** *Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints.* Computers & Operations Research (2010) Article in press
- LEE, Y. H. & PINEDO, M.** *Scheduling jobs on parallel machines with sequence-dependent setup times.* European Journal of Operational Research 100 (1997) 464-474
- LEE, Z. J.; LIN, S. W. & YING, K. C.** *Scheduling jobs on dynamic parallel machines with sequence-dependent setup times.* International Journal of Advanced Manufacturing Technology 47 (2010) 773-781
- LI, K. & YANG, S.** *Non-identical parallel-machine scheduling research with minimizing total weighted*

- completion times: Models, relaxation and algorithms.* Applied Mathematical Modelling 33 (2009) 2145-2158
- LIAW, C. F.; LIN, Y. K.; CHENG, C. Y. & CHEN, M.** *Scheduling unrelated parallel machines to minimize total weighted tardiness.* Computers & Operations Research 30 (2003) 1777-1789
- LIN, Y. K.; PFUND, M. E. & FOWLER, J. W.** *Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems.* Computers & Operations Research 38 (2011) 901-916
- LOGENDRAN, R.; McDONELL, B. & SMUCKER, B.** *Scheduling unrelated parallel machines with sequence-dependent setups.* Computers & Operations Research 34 (2007) 3420-3438
- LÓPEZ, O. C.; BARCIA, R. M. & EYADA, O.** *Problema de programação da produção: um esquema de classificação.* Revista Produção 5 (1995) 145-168
- MASON, S. J.; FOWLER, J. W. & CARLYLE, W. M.** *A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops.* Journal of Scheduling 5 (3) (2002) 247-262
- MOKOTOFF, E.** *Parallel machine scheduling problems: a survey.* Asia-Pacific Journal of Operational Research 18 (2001) 193-242
- MORTON, T. E. & PENTICO, D. W.** *Heuristic scheduling systems: with applications to production systems and project management.* New York: John Wiley & Sons 1993.
- OMAR, M. K. & TEO, S. C.** *Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach.* Applied Mathematics and Computation 181 (2006) 1008-1017
- OW, P. S. & MORTON, T. E.** *The single machine early/tardy problems.* Management Science 35 (1989) 177-191
- PAULA, M. R.; MATEUS, G. R. & RAVETTI, M. G.** *A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times.* Computers & Operations Research 37 (2010) 938-949
- PFUND, M.; FOWLER, J. W.; GADKARI, A. & CHEN, Y.** *Scheduling jobs on parallel machines with setup times and ready times.* Computers & Industrial Engineering 54 (2008) 764-782
- PFUND, M.; FOWLER, J. W. & GUPTA, J. N. D.** *A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems.* Journal of the Chinese Institute of Industrial Engineers, Vol. 21, No 3, (2004) 230-241
- PINEDO, M. L.** *Scheduling: Theory, Algorithms, and Systems.* New York: Springer 2008.
- PLATEAU, M. C. & RIOS-SOLIS, Y. A.** *Optimal solutions for unrelated parallel machines scheduling problems using convex quadratic reformulations.* European Journal of Operational Research 201 (2010) 729-736
- RACHAMADUGU, R. V. & MORTON, T. E.** *Myopic heuristics for the single machine weighted tardiness problem.* Working Paper 30 82-83 GSIA. Carnegie Mellon University
- RAGHAVAN, N. R. S. & VENKATARAMANA, M.** *Parallel processor scheduling for minimizing total weighted tardiness using ant colony optimization.* International Journal of Advanced Manufacturing Technology

41 (2009) 986-996

RIOS-SOLIS, Y. A. & SOURD, F. *Exponential neighborhood search for a parallel machine scheduling problem.* Computers & Operations Research 35 (2008) 1697-1712

SCHUTTEN, J. M. J. & LEUSSINK, R. A. M. *Parallel machine scheduling with release dates, due dates and family setup times.* International Journal of Production Economics 46-47 (1996) 119-125

SILVA, C. & MAGALHÃES, J. M. *Heuristic lot size scheduling on unrelated parallel machines with applications on the textile industry.* Computers & Industrial Engineering 50 (2006) 76-89

SIVRIKAYA-SERIFOGLU, F. & ULUSOY, G. *Parallel machine scheduling with earliness and tardiness penalties.* Computers & Operations Research 26 (1999) 773-787

TAVAKKOLI-MOGHADDAM, R.; TAHERI, F.; BAZZAZI, M.; IZADI, M. & SASSANI, F. *Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints.* Computers & Operations Research 36 (2009) 3224-3230

VEPSALAINEN, A. & MORTON, T. *Priority rules for job shops with weighted tardiness costs.* Management Science 33 (1987) 1035-1047

WU, Y. & JI, P. *A scheduling problem for PCB assembly: a case with multiple lines.* International Journal of Manufacturing Technology 43 (2009) 1189-1201

APÊNDICE - Tempos de processamento, fator de priorização e data de entrega das tarefas

Tarefas	p₁	p₂	w	d
1	56	52	212,5	337
2	65	60	244,6	368
3	89	90	146,3	103
4	93	93	131,4	371
5	82	90	94,4	275
6	85	93	82,1	93
7	75	82	60,8	154
8	75	82	52,2	246
9	88	108	22,5	386
10	79	97	35,9	389
11	71	86	27,8	113
12	62	75	18,4	390
13	53	64	20,5	386
14	44	52	22,4	225

4 CONSIDERAÇÕES FINAIS

A programação de tarefas é um processo de decisão estratégico de impacto direto no desempenho operacional e financeiro das empresas, oportunizando àquelas que a utilizam diferenciais competitivos no mercado que atuam. Neste sentido, os artigos que compõem essa dissertação abordaram funções-objetivo que desdobram os objetivos estratégicos das empresas em objetivos operacionais para os processos de manufatura.

O primeiro artigo buscou a minimização da função-objetivo da soma ponderada do tempo total de conclusão ($\sum w_j C_j$) e da soma ponderada do atraso total ($\sum w_j T_j$) de um conjunto de tarefas em um ambiente de manufatura de duas máquinas paralelas não-relacionadas com tempos determinísticos de processamento e de *setup*, este último dependente da sequência de processamento das tarefas. A nova heurística proposta se apoia em três etapas: (i) ordenar as tarefas pelo *WSPT*, (ii) alocar as tarefas às máquinas para minimizar o tempo total, e (iii) aprimorar a solução obtida na etapa 2 com *Tabu Search* (TS) simples de Glover (1989), realizando o intercâmbio adjacente de tarefas de uma mesma máquina.

Quando comparada com a solução ótima obtida por enumeração, a heurística do primeiro artigo apresenta desvios da solução ótima das funções-objetivo $\sum w_j C_j$ e $\sum w_j T_j$, de 3,6% e 9,7%, respectivamente, após a alocação das tarefas na etapa 2. Com a aplicação do TS, o desvio para o tempo de conclusão total é reduzido para 1,1% e o desvio da solução ótima para o atraso total é reduzido para 4,6%.

O segundo artigo abordou a minimização da função-objetivo do adiantamento e do atraso total ponderado ($\sum w_j (E_j + T_j)$) de um conjunto de tarefas sujeitas a um fator de priorização, em um ambiente de manufatura de duas máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência de processamento das tarefas. A heurística é operacionalizada em três etapas: (i) caracterizar o conjunto de tarefas, (ii) alocar as tarefas às máquinas utilizando uma versão modificada da heurística ATCS de Lee e Pinedo (1997), e (iii) aprimorar a solução obtida na etapa 2 com *Tabu Search* (TS) simples de Glover (1989), realizando o intercâmbio adjacente de tarefas de uma mesma máquina.

A heurística foi aplicada a dados reais e em cenários simulados de tempos de preparação e de *setup*, fator de priorização e data de entrega. Quando comparada com a solução ótima obtida por enumeração, a heurística apresenta desvio de 14,4% para dados reais

e desvio médio de 17,9% para os cenários simulados. As simulações resultam na solução ótima em pelo menos uma máquina em 70,75% dos casos, resultado considerado satisfatório pela empresa cujo ambiente de produção serviu de base para a pesquisa.

As heurísticas propostas nesta dissertação podem ser extendidas em pesquisas futuras que envolvam ambientes de manufatura com arranjos de máquinas mais complexos e funções-objetivo que considerem restrições adicionais àquelas aqui consideradas. Neste contexto, o ambiente base da pesquisa busca se aproximar cada vez mais dos ambientes de manufatura reais e das restrições atualmente encontradas nos parques fabris das indústrias brasileiras.

Os arranjos de máquinas paralelas não-relacionadas em múltiplos estágios, conhecidos como *flow shop* híbrido, se constituem em ambientes de manufatura mais complexos e representativos dos processos de manufatura da atualidade. Na sequência, a pesquisa envolvendo tais ambientes prossegue, considerando, por exemplo, as restrições de datas de liberação distintas entre tarefas, relações de precedência entre tarefas, restrições de disponibilidade de equipamentos para processamento da tarefa.

Outra frente de pesquisa promissora consiste na inserção de variáveis estocásticas e fatores de qualidade nas heurísticas aqui propostas. Tais parâmetros demandarão adaptações no mecanismo de alocação de tarefas às máquinas e na sistemática de ordenamento das tarefas pós-alocação.