

PROGRAMAÇÃO DE TAREFAS BASEADA EM CURVAS DE APRENDIZADO PARA LINHAS DE PRODUÇÃO CUSTOMIZADAS

LEARNING CURVE-BASED SCHEDULING IN CUSTOMIZED ASSEMBLY LINES

Michel J. Anzanello*E-mail: anzanello@producao.ufrgs.br

Flavio S. Fogliatto*Email: ffogliatto@producao.ufrgs.br

* Universidade Federal do Rio Grande do Sul, Porto Alegre, RS

Resumo: A manufatura de produtos personalizados, também conhecida como customização em massa, implica no aumento da variedade de modelos e demanda redução no tamanho dos lotes de produção. Tarefas que dependem da habilidade humana são especialmente afetadas nesse contexto, visto que os trabalhadores precisam se adaptar às características do novo modelo. Esse processo de adaptação dificulta a operacionalização de técnicas de programação de tarefas, uma vez que o tempo de processamento da tarefa é de difícil determinação. Este artigo integra informações oriundas da modelagem de curvas de aprendizado a heurísticas de programação de tarefas voltadas à minimização do tempo demandado para término de um conjunto de tarefas. O tempo de processamento de lotes (tarefas) com diferentes tamanhos e graus de complexidade é estimado através de curvas de aprendizado, sendo então utilizado em heurísticas para programação de tarefas em equipes de trabalhadores paralelos não relacionados. A heurística recomendada, identificada através de simulação, apresenta um desvio de 4,9% em relação ao tempo de término da sequência ótima de tarefas e conduz a um nível satisfatório de ocupação entre as equipes. A heurística é aplicada em um processo da indústria calçadista.

Palavras-Chave: Curvas de aprendizado. Programação de tarefas. Máquinas paralelas não relacionadas

Abstract: Mass customization implies in large variety of product models and small lot sizes. Human-based activities are highly affected by the learning process as a new model is put into production. That makes estimation of lot processing times difficult and jeopardizes the efficiency of scheduling techniques. The method proposed in this paper integrates information from learning curve modeling and scheduling heuristics, aiming at minimizing the completion time of jobs to be processed. Lot processing time is estimated by means of learning curve modeling and then used in scheduling heuristics for the unrelated parallel workers problem. The completion time generated by the recommended heuristic, determined through simulation, deviates 4.9% from the optimal sequence and yields good work balance among teams of workers. The heuristic is demonstrated in a shoe manufacturing process.

Keywords: Learning curves. Scheduling. Unrelated parallel machines

1 INTRODUÇÃO

Uma grande variedade de modelos de produtos com lotes de tamanho reduzido caracteriza ambientes voltados à manufatura de produtos personalizados (ambientes costumeiramente referidos como Customização em Massa - CM). Tais ambientes demandam flexibilidade dos recursos produtivos com vistas a sua rápida adequação aos requisitos de um novo modelo de produto (Da SILVEIRA *et al.*, 2001). A diversidade de modelos gera perdas em operações que envolvem procedimentos manuais, visto que a adaptação dos trabalhadores ao novo modelo em produção pode ocorrer de forma lenta e com geração de não-conformes. A programação de lotes de produção se torna especialmente difícil nessas circunstâncias, já que o tempo demandado para a conclusão de um lote sob o efeito do aprendizado dos trabalhadores é desconhecido. De tal forma, a combinação de modelagem por Curva de Aprendizado (CA) e técnicas de programação de tarefas pode auxiliar na solução de um problema de importância prática.

CAs são modelos de regressão não-linear que associam o desempenho de um trabalhador (ou equipe de trabalhadores) a características de uma tarefa. A idéia é descrever a melhoria no desempenho à medida que repetições são efetuadas, sendo o desempenho usualmente medido em termos do número de unidades produzidas em determinado intervalo de tempo de operação (UZUMERI e NEMBHARD, 1998).

Técnicas de programação de tarefas, por sua vez, são amplamente utilizadas em setores industriais e de serviços. O objetivo é alocar tarefas aos recursos produtivos de forma a otimizar algum indicador, geralmente vinculado à minimização do tempo de produção (PINEDO, 2008).

O entendimento do efeito do aprendizado dos trabalhadores sobre as técnicas de programação de tarefas é bastante incipiente. Estudos pioneiros de Biskup (1999) investigaram o efeito do aprendizado na posição da tarefa considerando apenas um trabalhador. Recentemente, Mosheiov e Sidney (2003) integraram CAs, com parâmetros específicos para cada tarefa, a formulações de programação linear. No trabalho desses autores, o objetivo é minimizar o tempo de duração total da programação em dois cenários: no primeiro, as tarefas são executadas por um único trabalhador; no segundo, as tarefas são executadas por diversos trabalhadores.

Os estudos de Mosheiov e Sidney (2003), no entanto, utilizam distribuições uniformes para gerar o tempo de processamento de uma tarefa. Esse procedimento não

é adequado para estimar tempos de processamento que seguem um padrão funcional (curvas), como em cenários caracterizados pelo aprendizado dos trabalhadores na execução de um novo modelo de produto. Nesses casos, faz-se necessária uma abordagem mais precisa para estimação dos tempos de processamento, visto que estes afetam significativamente a eficiência das técnicas de programação de tarefas.

Este artigo apresenta uma abordagem cujo objetivo é minimizar o tempo demandado para término de tarefas caracterizadas pela necessidade de aprendizado dos trabalhadores em ambientes de CM. O método sugerido utiliza a curva de aprendizado hiperbólica para caracterização quantitativa do aprendizado de trabalhadores submetidos a tarefas de complexidades distintas, como proposto em Anzanello e Fogliatto (2007a). As curvas obtidas na modelagem são então utilizadas para estimar o tempo de processamento de novas tarefas. De acordo com a terminologia adotada em técnicas de programação de tarefas, cada trabalhador é visto como uma máquina paralela não relacionada, já que a velocidade de execução das tarefas difere entre os trabalhadores de forma independente. Neste artigo, trabalhador e equipe de trabalhadores assumem o mesmo significado; tarefas e lotes de produção são também sinônimos. Utilizando-se a representação padrão, tem-se um problema do tipo $R_m \parallel \sum C$, onde R_m denota m equipes de trabalhadores em paralelo não relacionados, e $\sum C$ é o somatório do tempo demandado para término de um conjunto de tarefas.

Na etapa seguinte do método aqui proposto, geram-se quatro heurísticas simplificadas através da combinação de heurísticas disponíveis na literatura, voltadas à programação de tarefas em máquinas paralelas não relacionadas. As quatro heurísticas são operacionalizadas em três estágios: (i) definição da ordem para distribuição das tarefas – 2 regras são testadas neste estágio; (ii) distribuição das tarefas a serem executadas pelas l equipes de forma a garantir o equilíbrio no tempo de ocupação das equipes, evitando ociosidade – 2 regras são testadas neste estágio; (iii) programação das tarefas alocadas a cada equipe com vistas à minimização do tempo demandado para término do conjunto de tarefas, levando a m problemas do tipo $1 \parallel \sum C$.

As heurísticas sugeridas são testadas através de simulação em cenários nos quais os lotes de produção (tarefas) apresentam diferentes tamanhos e complexidades. Os resultados são comparados à sequência ideal, obtida através de enumeração total. A heurística recomendada apresenta desvio de 4,9% quando comparada ao tempo ótimo

demandado para término do conjunto de tarefas, além de conduzir a um nível satisfatório de equilíbrio na ocupação das equipes de trabalhadores. A heurística é então aplicada a um processo de fabricação de calçados composto por 3 equipes de trabalhadores e 90 lotes de tamanhos e complexidades distintos.

O presente artigo traz duas contribuições originais: (i) propõe a utilização de CAs para estimação do tempo de processamento demandado por equipes de trabalhadores na execução de tarefas com diferentes complexidades; e (ii) desenvolve e testa heurísticas para programação de tarefas em equipes de trabalhadores paralelos não relacionados.

O método proposto captura o efeito do aprendizado através dos tempos de processamento, os quais permitem programar as tarefas de forma mais precisa em ambientes caracterizados por CM.

O restante do artigo é organizado como segue. A seção 2 apresenta uma breve introdução as CAs e a técnicas de programação de tarefas em máquinas paralelas não relacionadas. Na seção 3 é apresentado o método para programar tarefas caracterizadas pelo aprendizado dos trabalhadores. A seção 4 apresenta os resultados obtidos por simulação e posterior aplicação da heurística recomendada em um estudo de caso na indústria calçadista. A seção 5 traz a conclusão do artigo.

2 REFERENCIAL TEÓRICO

Nesta seção, revisam-se os dois principais assuntos abordados nesse artigo: CAs e programação de tarefas em máquinas paralelas não relacionadas.

2.1 Curvas de aprendizado

CAs são representações matemáticas do desempenho de um trabalhador quando submetido a uma tarefa manual repetitiva. Trabalhadores demandam menos tempo para executar a tarefa à medida que repetições são efetuadas, seja pela familiaridade adquirida com os meios de produção, adaptação às ferramentas utilizadas ou pela descoberta de atalhos para realização da tarefa (WRIGHT, 1936; TEPLITZ, 1991, ANZANELLO, FOGLIATTO, 2007b). Modelos para CAs vêm sendo propostos na

literatura ao longo dos anos, com destaque para modelos de potência, como o de Wright, e para modelos hiperbólicos.

O modelo de Wright é a função matemática de aprendizado mais conhecida na literatura, devido a sua simplicidade e eficiência na representação de dados empíricos. A expressão matemática do modelo é:

$$t = U_1 z^b, \quad (1)$$

onde z representa o número de unidades produzidas, t designa o tempo (ou custo) médio acumulado para produção de z unidades, U_1 é o tempo (ou custo) para produzir a primeira unidade, e b corresponde à inclinação da curva ($-1 \leq b \leq 0$).

O modelo hiperbólico de aprendizado permite uma descrição mais precisa do processo de aprendizado, se comparado ao modelo de Wright. O modelo hiperbólico de três parâmetros, reportado por Mazur e Hastie (1978), é dado por:

$$y = k(x + p/x + p + r), \quad (2)$$

tal que $p + r > 0$. Na eq. (2), y descreve o desempenho do trabalhador em termos de unidades produzidas após x unidades de tempo de prática acumulada ($y \geq 0$ e $x \geq 0$), k corresponde ao limite superior de y ($k \geq 0$), p designa a experiência prévia na tarefa em unidades de tempo ($p \geq 0$), e r é o tempo de operação demandado para atingir um nível $k/2$, isto é, metade do desempenho máximo.

O modelo hiperbólico de aprendizado oferece uma melhor caracterização do perfil de aprendizado, permitindo otimizar a alocação de tarefas a trabalhadores (UZUMERI; NEMBHARD, 1998). Em Anzanello e Fogliatto (2007a), tarefas são alocadas a equipes de acordo com os parâmetros estimados para o modelo hiperbólico de curva de aprendizado. Assim, equipes mais produtivas recebem tarefas mais longas, ao passo que equipes caracterizadas pelo rápido aprendizado recebem tarefas com tamanhos menores de lote. O referido estudo, entretanto, não abordou o problema de programação de tarefas alocadas às equipes.

2.2 Programação de Tarefas em Máquinas Paralelas não Relacionadas

A programação de tarefas em máquinas paralelas tem recebido crescente atenção na literatura, conforme reportado por Chen e Sin (1990) e Pinedo (2008). Uma subcategoria desse problema contempla máquinas paralelas não relacionadas; nela, o tempo de processamento de uma tarefa depende exclusivamente da máquina na qual a mesma é processada, não existindo associação entre as máquinas (PINEDO, 2008). Segundo Yu *et al.* (2002), problemas determinísticos de programação de tarefas envolvendo máquinas paralelas não relacionadas estão entre os de mais difícil solução, o que justifica a proposição de abordagens alternativas para solucionar o problema. Na verdade, a maioria desses problemas enquadra-se na categoria *NP-hard*, demandando tempo exponencial de processamento para serem resolvidos.

Na sequência são apresentadas algumas abordagens para resolver problemas de programação de tarefas em máquinas paralelas não relacionadas. Mokotoff e Jimeno (2002) apresentam diversas heurísticas utilizando enumeração parcial para resolver o problema da minimização do tempo total da programação ou *makespan* (tempo transcorrido entre o início e fim de uma tarefa ou conjunto de tarefas). Chen e Wu (2006) propõem uma heurística para minimizar o atraso total contemplando restrições secundárias, tais como tempos de preparação, limitações na disponibilidade de recursos e restrições associadas aos processos. Similarmente, Kim *et al.* (2009) apresentam uma heurística para minimizar o tempo demandado para término de um conjunto de tarefas em cenários caracterizados por restrições de precedência. Nestes cenários, uma tarefa deve ser iniciada imediatamente após o início da tarefa precedente. Outras abordagens para a minimização do atraso total considerando máquinas paralelas não relacionadas são reportadas por Suresh e Chaudhuri (1994) e Randhawa e Kuo (1997). Por fim, uma comparação de algoritmos para programação de tarefas em máquinas paralelas não relacionadas é apresentada em Jungwattanakit *et al.* (2009). Tais algoritmos apóiam-se em algoritmo genético, *tabu search* e *simulated annealing*.

Pesquisas sobre o impacto do processo de aprendizagem em problemas de programação de tarefas são ainda incipientes. Biskup (1999) avaliou a influência do efeito do aprendizado (ganho de performance após sucessivas repetições) na programação das tarefas executadas em uma única máquina. O objetivo era minimizar o

makespan e o tempo ponderado para término de um conjunto de tarefas, considerando uma data de entrega comum a todas as tarefas. Mosheiov (2001a, b) ampliou os resultados de Biskup (1999) para o caso de múltiplas máquinas, bem como máquinas paralelas idênticas, utilizando um modelo de CA com parâmetros idênticos para todas as tarefas. Em trabalho mais recente, Mosheiov e Sidney (2003) incorporaram CAs com diferentes parâmetros conforme a tarefa para avaliar o impacto de padrões distintos de aprendizado sobre a programação de tarefas. Os parâmetros das CAs eram utilizados em formulações de programação inteira cujo objetivo era a minimização de funções objetivo envolvendo indicadores não-compostos, como o tempo de fluxo e o *makespan*, bem como o tempo de fluxo em máquinas paralelas não relacionadas.

3 MÉTODO

O método proposto visa programar tarefas que demandam aprendizado dos trabalhadores em ambientes de CM. O tempo de processamento das tarefas depende da adequação dos trabalhadores aos modelos em produção e, por consequência, impacta no escalonamento das tarefas. O método é composto por dois passos principais. No Passo 1, CAs são utilizadas para estimar o tempo de processamento de lotes de produção com tamanhos e complexidades distintas. No Passo 2, heurísticas voltadas à minimização do somatório do tempo demandado para término de um conjunto de tarefas em máquinas paralelas não relacionadas utilizam os tempos de processamento do Passo 1 para programar as tarefas nas diversas equipes de trabalhadores. Esses passos são detalhados na sequência.

As heurísticas sugeridas apóiam-se nos seguintes pressupostos: (i) todos os lotes estão disponíveis para processamento no instante de tempo zero; (ii) a divisão dos lotes não é permitida, ou seja, um lote deve ser inteiramente processado pela equipe a que foi alocado; e (iii) equipes de trabalhadores não processam dois ou mais lotes simultaneamente.

3.1 Passo 1

Este passo inicia com a seleção das equipes de trabalhadores para coleta de dados de desempenho na execução das tarefas. As equipes devem ser constituídas por

trabalhadores familiarizados com as tarefas, e devem apresentar baixa rotatividade. Equipes são representadas por $i = 1, \dots, l$.

Na sequência, selecionam-se os modelos de produtos a serem analisados. Produtos com elevada demanda por personalização são candidatos naturais nesta etapa. Os produtos selecionados são descritos em termos de suas características relevantes, incluindo aspectos dos seus componentes ou complexidade de manufatura, e podem ser avaliados de forma objetiva ou subjetiva. Um procedimento de formação de grupos (clusterização) é então executado utilizando tais características como variáveis.

Objetiva-se criar famílias de modelos de produtos dos quais dados de desempenho possam ser coletados. O procedimento de formação de grupos confere significância estatística às famílias, tal que peculiaridades de um modelo possam ser generalizadas aos demais modelos inseridos na mesma família (JOBSON, 1992; HAIR *et al.*, 1995). As famílias de modelos são representadas por $f = 1, \dots, F$.

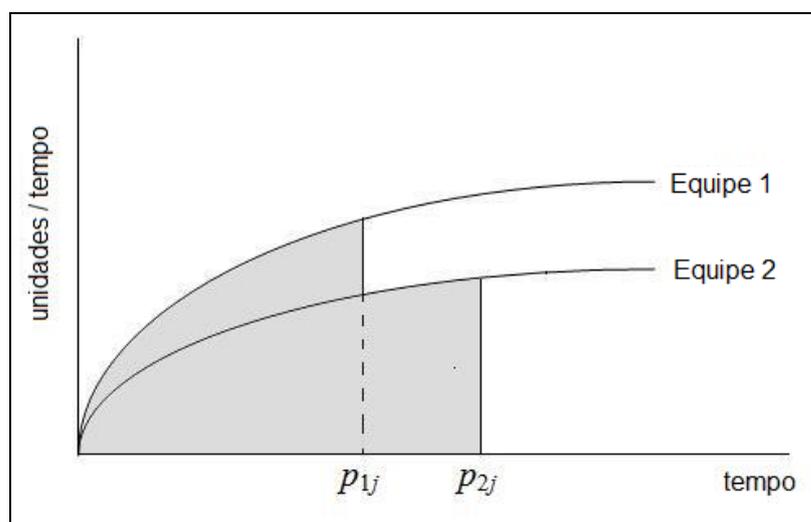
Dados de desempenho devem ser coletados de postos de trabalho tidos como gargalos de produção. Gargalos são definidos como operações complexas que requerem maior esforço do trabalhador em termos de tempo de aprendizado e destreza. Todas as combinações de i e f devem ser amostradas quanto ao desempenho, e repetições são recomendadas. Os dados de desempenho para cada combinação devem ser coletados no início da operação, sendo armazenados até que não sejam percebidas maiores alterações no padrão de desempenho. A coleta pode ser efetivada através da contagem do número de itens produzidos em determinado intervalo de operação.

Os dados de desempenho são analisados através da curva de aprendizado hiperbólica apresentada na eq. (2). O modelo hiperbólico é utilizado devido ao seu bom desempenho em estudos empíricos (ver NEMBARD e UZUMERI, 2000, e ANZANELLO e FOGLIATTO, 2007a), mas outros modelos podem ser testados. Os parâmetros da curva hiperbólica podem ser estimados através de rotinas de regressão não-linear disponíveis em aplicativos de análise estatística. Dados de desempenho são modelados como uma variável y (número de unidades produzidas), a qual é dependente do tempo de processamento x . Assim, cada família f é caracterizada por um conjunto de parâmetros k_{if} , p_{if} e r_{if} estimados a partir dos dados de desempenho obtidos para a equipe i . Por conveniência, parâmetros médios \bar{k}_{if} , \bar{p}_{if} e \bar{r}_{if} são calculados a partir de mensurações repetidas de desempenho de uma mesma família e equipe, e o parâmetro

\bar{k}_{ij} é então ajustado para quantificar o número de unidades produzidas por minuto, para facilitar a modelagem das CAs. Na sequência, f conjuntos de curvas contendo i curvas em cada conjunto são modeladas. Tais curvas representam o perfil de desempenho de cada equipe ao processar produtos de determinada família.

As curvas são então utilizadas para estimar o tempo de processamento p_{ij} , representando o tempo demandado pela equipe i para completar a tarefa j . A área sob cada curva quantifica o número de unidades produzidas em um dado intervalo de tempo, como ilustrado na área sombreada da Figura 1. Dessa forma, o tempo de processamento p demandado para concluir um lote de produção com Q unidades pode ser estimado através da integração da curva de 0 a p , até atingir-se uma área equivalente a Q . Esse procedimento é repetido para cada equipe. Observe que p_{ij} denota o tempo para processar a totalidade de unidades no lote (Q unidades), e não uma única unidade.

Figura 1- Tempo de processamento para as equipes 1 e 2



Fonte: Autores

3.2 Passo 2

As heurísticas propostas são sistematizadas em três estágios: (i) definição de uma ordem para distribuição dos lotes; (ii) alocação dos lotes às equipes de trabalhadores buscando equilíbrio na ocupação dos times; e (iii) programação dos lotes dentro de cada equipe com vistas à minimização do somatório do tempo demandado para término de um conjunto de tarefas. Esses estágios são detalhados na sequência.

Estágio 1 – Definição da ordem de distribuição dos lotes

N lotes devem ser ordenados de acordo com uma regra de priorização; duas regras são testadas:

- (i) Regra da Diferença Decrescente no Tempo de Processamento dos Lotes – Esta regra foi proposta por Adamopoulos e Pappis (1998). Para cada lote j , defina as duas equipes (A e B) com o menor tempo de processamento (p_{ij} , $i = A, B$) e calcule a diferença absoluta entre esses tempos, conforme eq. (3). As tarefas são então ordenadas em ordem decrescente de D_j e alocadas às equipes.

$$D_j = | p_{Aj} - p_{Bj} | \quad (3)$$

- (ii) Regra da Diferença Crescente no Tempo de Processamento dos Lotes – Similar à regra anterior, porém com tarefas alocadas às equipes em ordem crescente de D_j .

Estágio 2 – Alocação dos lotes às equipes de trabalhadores

Neste estágio decide-se a equipe que processará cada lote seguindo a ordem estabelecida no Estágio 1. Este estágio deve garantir uma distribuição de carga (tempo de operação) equilibrada entre as I equipes, evitando ociosidade ou sobrecarga. Note que os parâmetros da curva de aprendizado não são utilizados neste procedimento de alocação, visto que os tempos de processamento já capturaram sua influência. Duas regras são testadas neste estágio:

- (i) Regra do Tempo de Processamento Acumulado – Objetiva-se balancear o tempo de processamento em cada equipe i . Um lote l é testado em cada equipe de acordo com a eq. (4), e alocado à equipe com o menor T_i . Z_i é o número de lotes já alocados à equipe i . Esse procedimento é repetido até que todos os lotes tenham sido alocados. Essa regra é uma modificação da sistemática de distribuição sugerida por Bank e Werner (2001).

$$T_i = \sum_{j=1}^{Z_i} p_{ij} + p_{iI} \quad i=1, \dots, I, \quad (4)$$

- (ii) Regra do Tempo de Processamento e Número de Lotes Acumulados – Esta regra foi sugerida por Adamopoulos e Pappis (1998). A regra aloca lotes às equipes monitorando tanto o tempo de processamento como o número de lotes já alocados a cada equipe. A regra é sistematizada em duas fases: alocação regular (primária) e alocação secundária.

Inicialmente determina-se $H = \lfloor N/I \rfloor$, onde H é o limite superior no número de lotes a serem alocados às equipes na fase primária da regra. O primeiro lote é destinado à equipe com o menor p_{ij} . Esse procedimento é repetido para todos os lotes e monitorado pelo limite superior H . No caso do número de lotes já alocados à equipe i exceder H , esse lote é temporariamente alocado a uma equipe fictícia (*dummy team*). Após completar-se a fase primária de alocação, inicia-se a fase secundária alocando-se os lotes da equipe fictícia às equipes reais com base no tempo acumulado de processo.

O tempo de processamento p_{ij} do primeiro lote remanescente é adicionado aos tempos acumulados de todas as equipes candidatas e destinado em definitivo à equipe com o menor tempo acumulado de processo. Esse procedimento é repetido até esgotarem-se os lotes da equipe fictícia.

Estágio 3 – Programação dos lotes dentro de cada equipe

Os m problemas do tipo $1||\sum C$ são programados seguindo a regra do Menor Tempo de Processamento (*Shortest Processing Time*). Dentro de cada equipe, os tempos de processamento são arranjados em ordem crescente ($p_1 \leq p_2 \leq \dots \leq p_n$), onde p_j é o tempo de processamento do lote j (PINEDO, 2008). Observe que um único sub-índice é utilizado para identificar o lote, uma vez que cada equipe é considerada de forma separada neste estágio.

As 4 heurísticas geradas através da combinação dos estágios acima descritos são apresentadas na Tabela 1, e identificadas por H_u , onde $u = 1, \dots, 4$.

Tabela 1 - Heurísticas propostas

HEURÍSTICA	ESTÁGIO 1 - Definição da ordem de distribuição dos lotes	ESTÁGIO 2 - Alocação dos lotes às equipes	ESTÁGIO 3 - Sequenciamento dos lotes dentro de cada equipe
H1	Diferença decrescente no tempo de processamento dos lotes	Tempo de processamento acumulado	Minimização do somatório das datas de término das tarefas utilizando a regra do Menor Tempo de Processamento
H2	Diferença decrescente no tempo de processamento dos lotes	Tempo de processamento e número de lotes acumulados	
H3	Diferença crescente no tempo de processamento dos lotes	Tempo de processamento acumulado	
H4	Diferença crescente no tempo de processamento dos lotes	Tempo de processamento e número de lotes acumulados	

Fonte: Autores

A comparação entre as heurísticas propostas e o escalonamento ótimo gerado pela enumeração completa é efetivada através de dois critérios:

- (i) Desvio relativo entre os somatórios dos tempos demandados para término de um conjunto de tarefas gerados pela heurística (ΣC_{heuris}) e pela sequência ótima ($\Sigma C_{ótimo}$), conforme a eq. (5). ΣC_{heuris} refere-se ao somatório do tempo demandado pela equipe com maior carga de trabalho apontada pela heurística, ao passo que $\Sigma C_{ótimo}$ refere-se ao menor somatório de tempo de término da equipe com a maior carga dentre todas as combinações de equipes e lotes; e
- (ii) Desbalanço de carga entre as equipes, o qual é calculado da seguinte maneira. Suponha que o tempo de processamento acumulado (carga) da equipe A é 450 minutos e da equipe B é 500 minutos. Neste caso, existe um desbalanço de 10% [=1-(450/500)]. Valores pequenos de desbalanço são desejados, indicando equilíbrio na carga dos lotes alocados às equipes. É importante enfatizar que critérios de desbalanço não são avaliados no procedimento de enumeração, o qual restringe-se à busca do menor tempo de conclusão. Contudo, o menor tempo de conclusão está usualmente associado a uma sequência que promove balanço de carga entre as equipes.

$$desvio = \frac{\sum C_{heuris} - \sum C_{ótimo}}{\sum C_{ótimo}} \quad (5)$$

4 EXEMPLO NUMÉRICO

O método proposto é aplicado em um processo da indústria calçadista. Esse segmento tem enfrentado uma crescente tendência de CM evidenciada pela redução significativa no tamanho dos lotes de produção e aumento na variedade de modelos. Independente do modelo em produção, a etapa de costura do calçado constitui-se no gargalo do processo produtivo, fazendo com que a produção seja dependente da habilidade manual dos trabalhadores.

Vinte modelos de calçados são caracterizados em termos de sua complexidade através das seguintes variáveis para formação de grupos: complexidade geral, complexidade das partes (desdobrada em quatro categorias), número de componentes no calçado e tipo de calçado. As seis primeiras variáveis são subjetivamente avaliadas por especialistas usando uma escala de 3 pontos, onde 3 denota a mais alta complexidade. A variável ‘tipo de calçado’ apresenta dois níveis: 1 para sapatos e sandálias (mais simples em termos de montagem) e 2 para botas. Um procedimento de formação de grupos *k*-means aplicado aos vinte modelos conduziu à formação de três famílias de complexidades distintas: *Fácil*, *Média* e *Difícil*.

Dados de desempenho foram coletados de três equipes de trabalhadores (*Equipes 1, 2 e 3*), sendo cada equipe formada por aproximadamente 40 trabalhadores organizados em uma linha de produção. Modelos pertencentes às três famílias foram distribuídos às equipes de forma balanceada. Dados de desempenho foram coletados na forma de número de itens produzidos no intervalo de 10 minutos, e ajustados à curva de aprendizado hiperbólica utilizando um aplicativo estatístico. Coletas repetidas de dados dentro de uma mesma família deram origem aos parâmetros médios \bar{k}_{if} , \bar{p}_{if} e \bar{r}_{if} . O parâmetro \bar{k}_{if} foi manipulado para quantificar a produção em unidades por minuto com vistas ao processo de integração. A Tabela 2 traz os parâmetros ajustados da curva hiperbólica.

Tabela 2 - Parâmetros ajustados da curva de aprendizado hiperbólica

	<i>Fácil</i>			<i>Média</i>			<i>Difícil</i>		
	<i>Equipe 1</i>	<i>Equipe 2</i>	<i>Equipe 3</i>	<i>Equipe 1</i>	<i>Equipe 2</i>	<i>Equipe 3</i>	<i>Equipe 1</i>	<i>Equipe 2</i>	<i>Equipe 3</i>
\bar{k} (unidades/min)	1.19	1.30	1.26	1.62	1.34	2.66	0.94	1.11	1.57
\bar{p} (min)	80.3	62.9	51.5	15.9	14.4	16.1	77.9	21.1	34.1
\bar{r} (min)	145.9	122.5	66.6	46.9	69.8	38.0	68.7	50.9	97.3

Fonte: Autores

Os parâmetros \bar{k}_{ij} , \bar{p}_{ij} e \bar{r}_{ij} são então utilizados na construção das curvas de aprendizado agrupadas de acordo com a família de complexidade. Três gráficos correspondendo às famílias *Fácil*, *Média* e *Difícil* foram gerados, sendo cada gráfico composto por três curvas referentes às três equipes analisadas. As áreas sob as curvas permitiram estimar o tempo de processamento de cada equipe, os quais são então utilizados nas heurísticas de programação de tarefas.

O desempenho das heurísticas propostas foi comparado à sequência ótima, obtida através da enumeração completa das possíveis combinações dos lotes em cada equipe. Tendo-se em vista o elevado número de potenciais combinações, foram considerados apenas 10 lotes e 2 equipes. Assumiu-se que o tamanho dos lotes (em unidades) segue uma distribuição Normal com parâmetros μ e σ^2 ; i.e. $N \sim (\mu, \sigma^2)$. Foram testadas três distribuições de tamanho de lote com base na opinião de especialistas de processo: $N \sim (500,100)$, $N \sim (300,75)$ e $N \sim (150,25)$. À cada lote foi associado um nível de complexidade através de uma distribuição uniforme no intervalo [1,3], onde 1 representa a família *Fácil*. A Tabela 3 exemplifica os tempos de processamento demandados pelas *Equipes* 1 e 2 para concluir cada um dos 10 lotes gerados pela distribuição $N \sim (500,100)$.

Tabela 3 - Tempo de processamento estimado pelas CAs

Família	Tamanho do lote (unidades)	Tempo de processamento na Equipe 1 (horas)	Tempo de processamento na Equipe 2 (horas)
<i>Difícil</i>	457	8,7	7,2
<i>Difícil</i>	333	6,6	5,6
<i>Fácil</i>	513	9,4	8,7
<i>Difícil</i>	529	9,9	8,1
<i>Média</i>	385	6,8	3,4
<i>Fácil</i>	619	11,0	10,2
<i>Fácil</i>	550	9,3	9,1
<i>Média</i>	496	8,4	4,2
<i>Difícil</i>	533	9,9	8,2
<i>Difícil</i>	517	9,7	8,0

Fonte: Autores

Cada heurística foi testada através de 200 repetições para cada distribuição de tamanho de lote. A primeira coluna numérica da Tabela 4 apresenta o desvio médio, estimado de acordo com a eq. (5). A segunda e terceira colunas numéricas apresentam o desbalanço médio de carga para as heurísticas e sequência ótima, respectivamente. Esses resultados representam a média das três distribuições de tamanho de lote testadas.

Os desvios gerados pelas heurísticas concentram-se no intervalo de 4,9% a 9,6%, os quais são considerados satisfatórios por conta da simplicidade das heurísticas testadas. Recomenda-se a utilização da heurística H1 por conciliar o menor desvio ao menor nível de desbalanço de carga.

Na sequência, avaliaram-se os desvios em função do tamanho dos lotes de produção, conforme apresentado na Tabela 5. Não são verificadas diferenças significativas ou tendências no perfil dos desvios. A heurística H1 apresenta desempenho superior em todas as distribuições testadas.

Tabela 4 - Desempenho das heurísticas propostas com base no desvio e desbalanço de carga (média das três distribuições de tamanho de lote)

Heurística	Desvio	Desbalanço de carga Heurística	Desbalanço de carga Sequência Ótima
H1	4,9	9,1	2,8
H2	9,6	25,5	2,9
H3	6,8	12,2	2,9
H4	7,9	18,8	3,4

Fonte: Autores

Tabela 5 - Desvios sob distribuições distintas de tamanho de lote

Heurística	Distribuição do tamanho do lote (em unidades)		
	$N \sim (500, 100)$	$N \sim (300, 75)$	$N \sim (150, 25)$
H1	4,8	6,6	3,5
H2	8,3	11,6	8,8
H3	8,2	7,0	5,4
H4	8,4	8,5	6,9

Fonte: Autores

A heurística H1 é então aplicada em um caso real da indústria calçadista. São considerados 90 lotes de produção e três equipes de trabalhadores. O apêndice traz o número de itens e a complexidade de cada lote. A Tabela 6 apresenta a sequência recomendada para produção dos lotes, bem como o % de ocupação de cada equipe. Optou-se por apresentar o desbalanço de carga em termos do % de tempo que cada equipe permaneceu ocupada, tendo-se em vista que essa medida é mais intuitiva do que o % de desbalanço quando mais de duas equipes são consideradas. Um elevado número de lotes foi alocado à *Equipe 3* por conta de seu desempenho final superior e maior taxa de aprendizado, os quais são expressos pelos parâmetros k e r , respectivamente (ver Tabela 2). A heurística conduz à razoável equilíbrio no tempo de ocupação entre as equipes. Por fim, a sequência gerada pelo método foi comparada a 100 cenários simulados onde as tarefas foram randomicamente alocadas às 3 equipes: o tempo médio para conclusão do conjunto de tarefas gerado pelo método proposto é 16% menor, em média, do que o tempo demandado pelas sequências randômicas, e conduz aos melhores índices de ocupação entre as equipes.

5 CONCLUSÃO

Este artigo apresentou um método para programação de tarefas em cenários de elevada CM caracterizados pela necessidade de aprendizado dos trabalhadores. O método integra curvas de aprendizado a heurísticas de programação de tarefas. As curvas são utilizadas para estimar o tempo de processamento demandado por equipes

de trabalhadores submetidas a tarefas de diferentes tamanhos e complexidades. Os tempos estimados foram utilizados em quatro opções de heurísticas oriundas de modificações em heurísticas disponíveis na literatura para o problema de máquinas paralelas não relacionadas.

Tabela 6 - Sequência de produção recomendada e tempo de ocupação de cada equipe

Equipe	Sequência dos lotes																			Ocupação (%)	
<i>Equipe 1</i>	28	34	75	22	6	25	57	65	30	7	59	66	72	3	56	19	13	87	81	41	75
	18	47	73	63	58	84															
<i>Equipe 2</i>	32	67	8	21	14	76	43	82	78	10	15	70	11	2	64	71	12	24	74	53	82
	27	85	68	45	9	77	55														
<i>Equipe 3</i>	52	89	33	42	62	39	23	29	50	16	1	79	4	36	83	69	80	31	44	54	100
	86	48	90	88	60	5	49	37	46	17	35	61	20	38	26	51	40				

Fonte: Autores

A heurística recomendada conduz a desvios de 4,9% em relação à sequência ótima determinada por meio de enumeração total, além de conduzir a um equilíbrio satisfatório de carga entre as equipes. A heurística é então aplicada em um caso real da indústria calçadista, priorizando a alocação dos lotes à equipe detentora da melhor taxa de aprendizado.

Trabalhos futuros incluem a análise de cenários mais complexos de programação de tarefas de tarefas afetadas pelo processo de aprendizado, como nos arranjos *job shop*. Outro desdobramento visa analisar a influência da imprecisão da curva de aprendizado em estimar os tempos de processamento, e avaliar seu impacto sobre o escalonamento de tarefas. Pretende-se desenvolver indicadores de aderência da curva aos dados de desempenho e utilizar tais indicadores no ajuste dos tempos de processamento estimados.

REFERÊNCIAS

ADAMOPOULOS, G.; PAPPIS, C. Scheduling under a common due-date on parallel unrelated machines. **European Journal of Operational Research**, v. 105, p. 494-501, 1998.

- ANZANELLO, M.; FOGLIATTO, F.S. Learning curve modeling of work assignment in mass customized assembly lines. **International Journal of Production Research**, v. 45, p. 2919-2938, 2007a.
- ANZANELLO, M.; FOGLIATTO, F.S. Curvas de aprendizado: estado da arte e perspectivas de pesquisa. **Gestão & Produção**, v. 14, p.109-123, 2007b.
- 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**, v. 33, p. 363-383, 2001.
- BISKUP, D. Single-machine scheduling with learning considerations. **European Journal of Operational Research**, v. 115, p. 173-178, 1999.
- CHEN, T.; SIN, C. A state-of-the-art review of parallel-machine scheduling research. **European Journal of Operational Research**, v. 47, no. 3, p. 271-292. 1990.
- CHEN, J.; WU, T. Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. **Omega**, v. 34, p. 81-89, 2006.
- Da SILVEIRA, G.; BORESTEIN, D.; FOGLIATTO, F.S. Mass Customization: Literature Review and Research Direction. **International Journal of Production Economics**, v. 72, p. 1-13, 2001.
- HAIR, J. F.; ANDERSON, R. E.; TATHAM, R. L.; BLACK, W. C. **Multivariate Data Analysis with Readings**. Prentice-Hall Inc: New Jersey, 1995.
- JUNGWATTANAKIT, J.; REODECHA, M.; CHAOVALITWONGSE, P.; WERNER, F. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. **Computers & Operations Research**, v. 36, p. 358-378, 2009.
- KIM, E.; SUNG, C.; LEE, I. Scheduling of parallel machines to minimize total completion time subject to s-precedence constraints. **Computers & Operations Research**, v. 36, p. 698-710, 2009.
- MAZUR, J. E.; HASTIE, R. Learning as Accumulation: a Reexamination of the Learning Curve. **Psychological Bulletin**, v. 85, p. 1256-1274, 1978.
- MOKOTOFF, E.; JIMENO, J. Heuristics based on partial enumeration for the unrelated parallel processor scheduling problem. **Annals of Operations Research**, v. 117, p. 133-150, 2002.
- MOSHEIOV, G. Scheduling problems with learning effect. **European Journal of Operational Research**, v. 132, p. 687-693, 2001a.
- MOSHEIOV, G. Parallel machine scheduling with learning effect. **Journal of the Operational Research Society**, v. 52, p. 391-399, 2001b.

MOSHEIOV, G.; SIDNEY, J. Scheduling with general job-dependent learning curves. **European Journal of Operational Research**, v. 147, p. 665-670, 2003.

NEMBHARD, D. A.; UZUMERI, M. V. An Individual-Based Description of Learning within an Organization. **IEEE Transactions on Engineering Management**, v. 47, p. 370-378, 2000.

PINEDO, M. Scheduling, Theory, Algorithms and Systems. Springer: New York, 2008.
SURESH, V.; CHAUDHURI, D. Minimizing maximum tardiness for unrelated parallel machines. **International Journal of Production Economics**, v. 34, p. 223-229, 1994.

TEPLITZ, C. J. The Learning Curve Deskbook: A reference guide to theory, calculations and applications. Quorum Books: New York, 1991.

UZUMERI, M.; NEMBHARD, D. A Population of Learners: A New Way to Measure Organizational Learning. **Journal of Operations Management**, v. 16, p. 515-528, 1998.

WRIGHT, T. Factors affecting the cost of airplanes. **Journal of the Aeronautical Sciences**, v. 3, p. 122-128, 1936.

YU, L; SHIH, H.; PFUND, M.; CARLYLE, W.; FOWLER, J. Scheduling of unrelated parallel machines: an application to PWB manufacturing. **IIE Transactions**, v. 34, p. 921-931, 2002.



Artigo recebido em 10/03/2011 e aceito para publicação em 11/06/2011.

APÊNDICE: Tamanho e complexidade dos lotes de calçados analisados

Lote	Família	Tamanho do lote (unidades)	Lote	Família	Tamanho do lote (unidades)
1	<i>Média</i>	460	46	<i>Fácil</i>	525
2	<i>Difícil</i>	345	47	<i>Fácil</i>	450
3	<i>Difícil</i>	390	48	<i>Difícil</i>	430
4	<i>Difícil</i>	640	49	<i>Fácil</i>	590
5	<i>Fácil</i>	630	50	<i>Média</i>	485
6	<i>Média</i>	500	51	<i>Fácil</i>	360
7	<i>Difícil</i>	545	52	<i>Média</i>	730
8	<i>Média</i>	530	53	<i>Fácil</i>	500
9	<i>Fácil</i>	380	54	<i>Difícil</i>	465
10	<i>Difícil</i>	505	55	<i>Fácil</i>	130
11	<i>Difícil</i>	425	56	<i>Difícil</i>	335
12	<i>Fácil</i>	585	57	<i>Difícil</i>	620
13	<i>Fácil</i>	610	58	<i>Fácil</i>	360
14	<i>Média</i>	440	59	<i>Difícil</i>	510
15	<i>Difícil</i>	475	60	<i>Fácil</i>	760
16	<i>Média</i>	485	61	<i>Fácil</i>	455
17	<i>Fácil</i>	500	62	<i>Média</i>	545
18	<i>Fácil</i>	480	63	<i>Fácil</i>	420
19	<i>Fácil</i>	700	64	<i>Fácil</i>	820
20	<i>Fácil</i>	445	65	<i>Difícil</i>	580
21	<i>Difícil</i>	720	66	<i>Difícil</i>	475
22	<i>Média</i>	535	67	<i>Difícil</i>	855
23	<i>Média</i>	530	68	<i>Fácil</i>	420
24	<i>Fácil</i>	580	69	<i>Difícil</i>	560
25	<i>Difícil</i>	715	70	<i>Difícil</i>	445
26	<i>Fácil</i>	400	71	<i>Fácil</i>	620
27	<i>Fácil</i>	460	72	<i>Difícil</i>	440
28	<i>Média</i>	720	73	<i>Fácil</i>	440
29	<i>Média</i>	495	74	<i>Fácil</i>	520
30	<i>Média</i>	345	75	<i>Média</i>	565
31	<i>Difícil</i>	535	76	<i>Difícil</i>	605
32	<i>Média</i>	715	77	<i>Fácil</i>	350
33	<i>Média</i>	615	78	<i>Difícil</i>	545
34	<i>Média</i>	580	79	<i>Média</i>	460
35	<i>Fácil</i>	490	80	<i>Difícil</i>	545
36	<i>Difícil</i>	600	81	<i>Fácil</i>	540
37	<i>Fácil</i>	580	82	<i>Difícil</i>	550
38	<i>Fácil</i>	425	83	<i>Média</i>	365
39	<i>Média</i>	540	84	<i>Fácil</i>	275
40	<i>Fácil</i>	330	85	<i>Fácil</i>	445
41	<i>Fácil</i>	505	86	<i>Difícil</i>	465
42	<i>Média</i>	575	87	<i>Fácil</i>	580
43	<i>Difícil</i>	585	88	<i>Difícil</i>	300
44	<i>Difícil</i>	500	89	<i>Média</i>	620
45	<i>Fácil</i>	410	90	<i>Difícil</i>	390

Fonte: Autores