DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES PARA UMA LINHA DE PRODUÇÃO *FLOWSHOP*: MÉTODOS DE SOLUÇÃO

Márcio Antônio Ferreira Belo Filho
Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos - SP, Brasil
e-mail: marciobf@icmc.usp.br

Maristela Oliveira dos Santos
Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos - SP, Brasil
e-mail: mari@icmc.usp.br

Cláudio Nogueira de Meneses
Universidade Federal do ABC - Centro de Matemática, Computação e Cognição
Rua Santa Adélia, 166, 09210-170, Santo André - SP, Brasil
e-mail: claudio.meneses@ufabc.edu.br

RESUMO

Neste trabalho um problema de dimensionamento e sequenciamento de lotes em ambiente *flowshop* é considerado. O objetivo é determinar uma programação de produção visando minimizar os custos de produção e de estoque. Um modelo matemático da literatura é apresentado e heurísticas baseadas em Programação Inteira Mista (*MIP*) foram implementadas. Também propomos a aplicação da metaheurística Times Assíncronos (*A-Teams*) para resolução do problema. As metodologias desenvolvidas e os resultados obtidos são apresentados neste artigo.

PALAVRAS CHAVE. Problema de Dimensionamento e Sequenciamento de Lotes, *Flowshop*, Metaheurísticas, Programação Inteira Mista

ABSTRACT

In this work a lot sizing and sequencing problem at a flowshop environment is considered. The objective is to establish a production scheduling to minimize the production and inventory costs. A mathematical model of the literature is presented and heuristics based on Linear Mixed-Integer Programming (MIP) were implemented. We also propose the application of metaheuristic Asynchronous Teams (A-Teams) to solve the problem. The methodologies developed and the results are presented in this paper.

KEYWORDS. Lot Sizing and Scheduling Problem, Flowshop, Metaheuristics, Linear Mixed-Integer Programming

1 Introdução

As indústrias buscam melhorar o planejamento de sua produção para obter um melhor posicionamento na concorrência dos mercados, visando minimizar o custo de sua produção e aumentar a eficiência de sua cadeia produtiva. O planejamento de produção é o planejamento de aquisição de recursos e matérias-primas, assim como o planejamento das atividades relacionadas à transformação da matéria-prima em produtos satisfazendo à demanda de consumidores da maneira mais eficiente e econômica possível.

O planejamento de produção é dependente do ambiente de produção já que os tempos de cada unidade de um determinado lote dependem da sua rota através dos estágios de produção (máquinas), da disponibilidade desses estágios e dos tempos de outras unidades do mesmo lote ou mesmo de outros produtos. Em Pinedo (2008) os ambientes de produção são classificados e definidos de acordo com a configuração das máquinas e a rota dos produtos nas máquinas. O ambiente de produção *flowshop* é caracterizado por lotes de produtos sendo produzidos por uma quantidade de máquinas ligadas em série. Dessa forma, um fluxo unidirecional contínuo de lotes de produtos passa pela série de máquinas. Os problemas de *flowshop* encontram aplicações na indústria automotiva, química, eletrônica, alimentícia, farmacêutica, têxtil, e em qualquer linha de produção onde uma rota definida por estágios em série para toda produção é a mesma.

O problema investigado neste trabalho é baseado nos trabalhos de Mohammadi et al. (2009, 2010) e consiste na determinação de uma programação de produção em um ambiente de produção *flowshop*, com demanda dinâmica determinística em um horizonte de planejamento finito. São consideradas restrições de capacidade de produção, estoques de produtos intermediários e acabados, tempos de preparação de máquinas dependentes da sequência de produtos, preparações de máquinas mantidas entre períodos (*setup carryover*). Decisões de dimensionamento de lotes e sequenciamento dos produtos devem ser tomadas buscando minimizar o custo total de produção definido pelo custos de processamento de cada produto em cada máquina, o custo de preparação de máquinas e os custos relativos a estoques de produtos acabados e inacabados.

Métodos presentes na literatura foram implementados para o problema proposto. Nos trabalhos de Mohammadi et al. (2009, 2010) foi apresentado um modelo matemático para o problema. Um software que implementa métodos para resolver problemas de programação linear inteira foi utilizado, e os tempos computacionais para a resolucão exata de problemas-teste foram relativamente altos. Deste modo, heurísticas baseadas na estratégia de horizonte rolante e heurísticas do tipo *relax-and-fix* foram propostas pelos autores.

Neste artigo é proposta uma metaheurística baseada em Times Assíncronos, cuja arquitetura faz com que agentes heurísticos atuem de forma concomitante em memórias de soluções, unindo-se assincronamente por meio de diferentes estratégias e visando encontrar soluções factíveis de boa qualidade. Realçamos que é inédito o uso desta técnica para tentar resolver o problema em questão. Com o intuito de demonstrar, empiricamente, a qualidade das soluções geradas por esta técnica, estas soluções são comparadas com as encontradas por procedimentos propostos pela literatura.

2 Revisão da Literatura

O problema de programação da produção é muito estudado na literatura, devido a sua aplicação direta na indústria e trata o problema de alocar recursos para tarefas dentro de períodos de tempo, geralmente buscando minimizar custos e melhorar a eficiência da produção. Sua teoria pode ser vista em Pinedo (2008). O problema de programação da produção considerado trata decisões de dimensionamento e de sequenciamento de lotes.

O problema de dimensionamento de lotes consiste em determinar a quantidade de lotes a serem produzidos e a data dessa produção. Essa produção está inserida num horizonte de planejamento e existe uma demanda de produtos a ser abastecida. O objetivo é geralmente de cunho econômico, simulando os custos de produção, que são tradicionalmente divididos em custos de estoque, custos de processamento de produtos e custos de preparação de máquinas. Na literatura muitos trabalhos abordam o problema, dentre eles Karimi et al. (2003) e Jans e Degraeve (2008).

O problema de sequenciamento de produtos consiste em estabelecer sequências de produtos a serem produzidos em cada máquina. O problema de sequenciamento é bem discutido em Pinedo (2008). Dentro do contexto de linha de produção *flowshop*, o primeiro problema de *flowshop scheduling*, resolvido polinomialmente por Johnson (1954), tratava de um problema de sequenciamento com duas máquinas com o objetivo de minimizar o tempo final de produção. Para um número de máquinas maior que três, o problema é NP-Difícil, de acordo com Garey et al. (1976). Algumas variações estão presentes na literatura, como, por exemplo, apenas permitir soluções permutacionais, existência de preparações de máquinas, diferentes políticas de tratamento do estoque intermediário, dentre outras. É prática industrial e da literatura considerar que todos os estágios de produção tenham a mesma sequência de produção, o que define o problema de *flowshop* permutacional.

A literatura de sequenciamento geralmente assume que as decisões de dimensionamento de lotes já estão decididas, enquanto os problemas de dimensionamento de lotes desconsideram ou aproximam os detalhes de sequenciamento. O problema integrado de programação de produção é abordado pelos trabalhos de Potts e Wassenhove (1992), Drexl e Kimms (1997) e Zhu e Wilhelm (2006). Todos os trabalhos abordam diversos ambientes de produção. Para linhas de produção do tipo *flowshop* as referências são escassas. No trabalho de Zhu e Wilhelm (2006) há apenas três referências para *flowshop*, todas elas utilizam abordagens heurísticas e nenhum modelo matemático é apresentado.

Recentemente, os trabalhos de Mohammadi et al. (2009, 2010) apresentaram um modelo matemático com decisões integradas de dimensionamento de lotes e de sequenciamento para linhas de produção *flowshop*, considerando preparações de máquinas dependentes da sequência, preservação de preparação e programações de produção não-permutacionais. Os autores desenvolveram limitantes inferiores, heurísticas do tipo *relax-and-fix* e heurísticas baseadas na estratégia de horizonte rolante. Algumas dessas heurísticas desconsideram soluções cuja programação de produção é não-permutacional. A permutabilidade permite que heurísticas sejam capazes de resolver maiores problemas-teste.

3 Definição do Problema

O problema considerado está situado numa linha de produção do tipo *flowshop*, onde máquinas dispostas em série transformam matérias-primas em diversos produtos acabados independentes entre si. Decisões de sequenciamento e dimensionamento de lotes devem ser feitas num horizonte de planejamento finito com o objetivo de uma produção eficiente e econômica. O problema considera que a sequência de produção em todas as máquinas é a mesma, ou seja, trata-se de um problema de *flowshop* permutacional. Assim, restringe-se o número de soluções com relação ao problema de *flowshop* puro. Há uma demanda externa para produtos acabados (processados pela última máquina) e esta é sempre atendida sem atrasos. Todos os recursos estão disponíveis no início do horizonte de planejamento.

Todas as máquinas possuem restrições de capacidade e devem ser preparadas para mudanças de produtos. O processamento dos produtos e as preparações de máquinas (*setup*) possuem custos e consomem tempo (capacidade) de produção. Os *setups* devem iniciar e

terminar no mesmo período. No início do horizonte de planejamento, as máquinas estão previamente preparadas para determinado produto, já os demais períodos mantêm a última preparação do período anterior, ou seja, as preparações de máquinas são preservadas entre períodos (*setup carryover*). As preparações entre produtos semelhantes são nulas e todas as outras obedecem à desigualdade triangular. Para cada período é permitido N (número de produtos) preparações de máquina, que significa que a sequência de produção de cada máquina é composta por N produtos não necessariamente distintos.

Entre as máquinas m e $m+1$ há um estoque intermediário irrestrito onde produtos processados pela máquina m esperam para serem processados por $m+1$. O tempo de transporte e de outros processos entre máquinas sucessivas é desprezível. O processo de um lote na máquina m só se inicia se todo o lote está disponível para produção, ou seja, todo lote já foi processado pela máquina $m-1$ e está no estoque intermediário. Para simular o tempo de espera da máquina para produção, utiliza-se a abordagem de Fandel e Stammen-Hegene (2006), denominada produto fantasma. O produto fantasma está associado à produção de determinado produto e possui o mesmo tempo de processamento nas máquinas. Não há demanda ou custos associados para produtos fantasmas. O modelo é descrito a seguir, juntamente com os seus índices, parâmetros e variáveis de decisão.

	Índices		**Parâmetros**
i, j, k	Produtos	N	Número de produtos diferentes
m	Máquinas	M	Número de Máquinas
t	Períodos	T	Número de Períodos
n, n'	Posição na Sequência	d_{jt}	Demanda externa de j em t
	Variáveis	C_{mt}	Capacidade de m em t
I_{jmt}	Estoque de j após processo em m,t	h_{jm}	Custo de estoque de j em m
y^n_{ijt}	Define se o n-ésimo *setup* de i a j	p_{jmt}	Custo de produção de j em m, t
	é feito em t	b_{jm}	Tempo de produção de j em m
x^n_{jmt}	Quantidade de j em m, t no	W_{ijm}	Custo de *setup* de i a j em m
	n-ésimo lote da sequência	S_{ijm}	Tempo de *setup* de i a j em m
q^n_{jmt}	Tempo de espera para x^n_{jmt}	$j0$	Configuração inicial das máquinas

Minimizar

$$\sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} W_{ijm}\cdot y^n_{ijt} + \sum_{n=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} p_{jmt}\cdot x^n_{jmt} + \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} h_{jm}\cdot I_{jmt} \tag{1}$$

Sujeito a:

$$d_{jt} = I_{jMt-1} + \sum_{n=1}^{N} x^n_{jMt} - I_{jMt} \qquad j = 1,\ldots,N; t = 1,\ldots,T \tag{2}$$

$$I_{jmt-1} + \sum_{n=1}^{N} x^n_{jmt} = I_{jmt} + \sum_{n=1}^{N} x^n_{jm+1t} \qquad j = 1,\ldots,N; m = 1,\ldots,M-1; t = 1,\ldots,T \tag{3}$$

$$\begin{gathered}\sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ijm}\cdot y^n_{ijt} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{jm}\cdot q^n_{jmt} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{jm}\cdot x^n_{jmt} \leq \\ \sum_{n=1}^{n'}\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ijm+1}\cdot y^n_{ijt} + \sum_{n=1}^{n'}\sum_{j=1}^{N} b_{jm+1}\cdot q^n_{jm+1t} + \sum_{n=1}^{n'-1}\sum_{j=1}^{N} b_{jm+1}\cdot x^n_{jm+1t} \\ n' = 1,\ldots,N; m = 1,\ldots,M-1; t = 1,\ldots,T\end{gathered} \tag{4}$$

$$\sum_{n=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ijm}\cdot y^n_{ijt} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{jm}\cdot q^n_{jmt} + \sum_{n=1}^{N}\sum_{j=1}^{N} b_{jm}\cdot x^n_{jmt} \leq C_{mt} \qquad m = 1,\ldots,M; t = 1,\ldots,T \tag{5}$$

$$x^n_{jmt} \leq (\frac{C_{mt}}{b_{jm}}) \sum_{i=1, i\neq j(n>1)}^{N} y^n_{ijt} \qquad n = 1, \ldots, N; j = 1, \ldots, N; m = 1, \ldots, M; t = 1, \ldots, T \tag{6}$$

$$q^n_{jmt} \leq (\frac{C_{mt}}{b_{jm}}) \sum_{i=1}^{N} y^n_{ijt} \qquad n = 1, \ldots, N; j = 1, \ldots, N; m = 1, \ldots, M; t = 1, \ldots, T \tag{7}$$

$$y^1_{ji1} = 0 \qquad j \neq j0; i = 1, \ldots, N \tag{8}$$

$$\sum_{i=1}^{N} y^1_{j0i1} = 1 \tag{9}$$

$$\sum_{j=1}^{N} y^n_{jit} = \sum_{k=1}^{N} y^{n+1}_{ikt} \qquad n = 1, \ldots, N-1; i = 1, \ldots, N; t = 1, \ldots, T \tag{10}$$

$$\sum_{j=1}^{N} y^N_{jit-1} = \sum_{k=1}^{N} y^1_{ikt} \qquad i = 1, \ldots, N; t = 2, \ldots, T \tag{11}$$

$$y^n_{ijt} \in \{0, 1\}, I_{jmt}, x^n_{jmt}, q^n_{jmt} \geq 0 \tag{12}$$

$$I_{jm0} = 0 \qquad j = 1, \ldots, N; m = 1, \ldots, M \tag{13}$$

A expressão (1) introduz a função objetivo do problema, que simula os custos de produção dessa linha de produção. Os custos envolvidos são de preparações de máquinas dependentes da sequência, de produção e de estoque. As equações (2) e (3) representam o fluxo de produção entre as máquinas, culminando na satisfação da demanda, sem atrasos.

A restrição (4) garante a interação vertical entre as máquinas, assim um produto só pode ser processado em determinada máquina se foi previamente processado na máquina anterior. O lado esquerdo da restrição (4) representa o tempo de término do processamento do n-ésimo produto na máquina m, enquanto o lado direito indica o tempo de início do processamento do mesmo lote em $m + 1$. A restrição (5) impõe uma capacidade máxima para cada máquina em cada período. As restrições (6) e (7) restringem o tamanho do lote de determinado produto e de seu produto fantasma à existência da preparação da máquina.

As restrições (8) e (9) impõem que o primeiro *setup* do primeiro período seja do produto no qual a máquina já está preparada para outro produto, e que qualquer outra configuração de preparação é nula. As equações (10) e (11) estão relacionadas ao sequenciamento dos produtos por meio da limitação entre *setups* sucessivos, inclusive no caso da preservação da preparação de máquina. A restrição (12) indica o tipo das variáveis de decisão. Por último, não há estoque no início do horizonte de planejamento, como define (13).

4 Procedimentos de Solução

Nos trabalhos de Mohammadi et al. (2009, 2010) foi apresentado um modelo matemático para o problema proposto, porém, para problemas-teste de médio e grande porte o tempo necessário para a resolução exata do problema é impraticável. Dessa forma, os autores propõem heurísticas para encontrar boas soluções em tempo computacional razoável. Essas heurísticas são divididas naquelas baseadas na estratégia de horizonte rolante e heurísticas do tipo *relax-and-fix*. Os autores também propõem limitantes inferiores, assim, para problemas-teste maiores os métodos podem ser comparados. Os limitantes e algumas destas heurísticas foram desenvolvidos para o caso não-permutacional, deste modo, se fez necessário algumas adaptações para que fosse considerado apenas soluções permutacionais. Em ambos os trabalhos, os autores propõem como pesquisa futura o uso de metaheurísticas, devido

ao fato do problema ser de difícil resolução. Desta forma, propomos a utilização da metaheurística Times Assíncronos (*A-Teams*), uma abordagem inédita ao problema. Todos os procedimentos implementados são apresentados nas subseções seguintes.

4.1 Limitantes Inferiores

Métodos de solução exatos para a formulação matemática proposta são, na prática, ineficientes para exemplos de médio a grande porte, pois necessitam de bastante tempo para encontrar a solução ótima. O uso de limitantes inferiores é uma boa maneira de lidar com essa situação, porém, bons limitantes são difíceis de encontrar. No trabalho de Mohammadi et al. (2010) dois limitantes são apresentados. O primeiro limitante inferior (*L1*) é obtido relaxando todas as variáveis binárias do modelo matemático (1)-(13). O segundo limitante inferior (*L2*) é derivado de *L1* e adiciona a equação (14).

$$\sum_{i=1}^{N} y_{ijt}^{1} + \sum_{i=1, i\neq j}^{N} \sum_{n=2}^{N} y_{ijt}^{n} = a_{jt} \qquad j = 1, \ldots, N, t = 1, \ldots, T \tag{14}$$

As variáveis a_{jt} são binárias. A equação (14) implica que se, no período t, há o processo do produto j, este só pode ser processado em uma única posição da sequência de produção, exceto no caso onde posições adjacentes na sequência de produção pertencem ao mesmo produto j (por isso o $i \neq j$ no segundo somatório). A equação (14) só pode ser agregada à relaxação quando os tempos e custos de preparação de máquina obedecem à desigualdade triangular, pois, do contrário, tal equação acrescentaria uma restrição ao conjunto de soluções do modelo proposto, e dessa forma não seria mais um limitante. O limitante *L2* é um problema inteiro misto, com um número de variáveis inteiras reduzido, se comparado ao problema original. Para problemas-teste maiores, o limitante falha em encontrar uma solução ótima em tempo computacional razoável.

4.2 Heurísticas baseadas em Horizonte Rolante

A estratégia de horizonte rolante para problemas de dimensionamento de lotes supõe que decisões de determinados períodos são pouco influentes para os demais períodos. O procedimento iterativo decompõe o horizonte de planejamento em três seções: 1) A seção inicial é composta por períodos cujas decisões já foram tomadas por iterações anteriores e estão parcialmente ou totalmente fixas (congeladas); 2) A seção central possui o conjunto de períodos a serem considerados para se tomar decisões; 3) A seção final inclui os últimos períodos e suas decisões são desconsideradas até o momento. Algumas simplificações podem ser utilizadas para tornar o procedimento heurístico mais eficiente. No início de uma heurística com horizonte rolante, geralmente não há variáveis congeladas, deixando a seção inicial vazia. A cada iteração um número de períodos é adicionado à seção inicial, enquanto períodos são subtraídos da seção final, tendo de passar pela seção central.

A primeira heurística de horizonte rolante (*RH1*), de Mohammadi et al. (2010) tem como seção central um único período, e toma decisões relativas a este período. Todas as variáveis de decisão (binárias e reais) dos períodos anteriores (seção inicial) estão fixas. Todas as variáveis inteiras são relaxadas na seção final. A Figura 1 apresenta a divisão do procedimento heurístico em T iterações. Como um conjunto de variáveis são relaxadas, a heurística resolve a cada iteração problemas menores, tornando o método capaz de obter soluções de maneira mais eficiente. A heurística foi modificada para considerar apenas soluções permutacionais.

A segunda heurística de horizonte rolante (*RH2*), de Mohammadi et al. (2010), é análoga à heurística *RH1* exceto que as variáveis de decisão fixas na seção inicial são apenas as

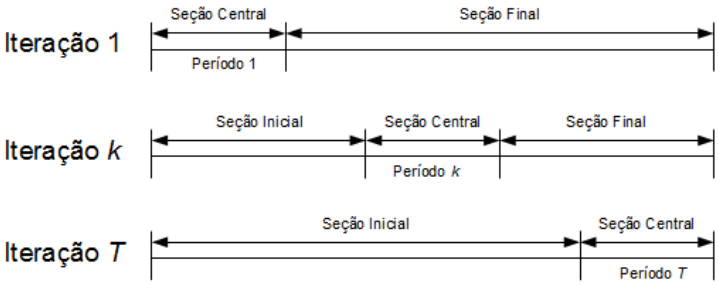

Figura 1: Estratégia de horizonte rolante utilizada

binárias. Ou seja, apenas as variáveis que definem o sequenciamento nas máquinas são congeladas, enquanto as decisões de dimensionamento de lotes, tamanho do estoque e quantidade de produtos fantasmas estão livres, permitindo que na tomada de decisão de outros períodos essas variáveis assumam valores diferentes.

A terceira heurística de horizonte rolante (*RH3*), de Mohammadi et al. (2009), é uma adaptação da heurística desenvolvida por Nawaz et al. (1983) (denominada *NEH*), bastante utilizada na literatura do problema de *flowshop*, por ser uma heurística simples, flexível e eficiente. A heurística *NEH* é um procedimento construtivo de inserção. A ordem de inserção da heurística *RH3* é a ordem decrescente dos valores de W_j, onde $W_j = \sum_{m=1}^{M} \sum_{n=1}^{N} W_{ijm}$, para todo produto j. Para cada período a ordem de inserção pode ser modificada retirando-se aqueles produtos cujo estoque de períodos anteriores é suficiente para atender sua demanda. Dada uma ordem de inserção de produtos, gera-se uma sequência de produção inserindo-se, um a um, os produtos na melhor posição da sequência atual, tal que a soma dos custos de preparação é menor. Como alguns produtos foram retirados da sequência de inserção e como é necessário N posições na sequência de produção, completa-se as posições finais com o último produto produzido até então. Na seção inicial da heurística *RH3* as variáveis binárias (y) dos períodos anteriores a seção central são congeladas. Na seção final adota-se todas as restrições (2)-(13), porém todas as variáveis são relaxadas. A seção central é formada apenas pelo período t. Para cada iteração, a partir da ordem de inserção, gera-se a sequência de produção para o período t, define as variáveis de sequenciamento y de acordo com a sequência de produção e resolve-se o problema linear resultante.

4.3 Heurística *Relax-and-Fix*

O métodos heurísticos do tipo *relax-and-fix* são bem conhecidos na literatura para problemas inteiros misto (*MIP*). O método relaxa todas as variáveis inteiras, e, a cada iteração do algoritmo, determina que um conjunto dessas variáveis sejam novamente inteiras. A partir desse momento resolve-se o *MIP* com um número menor de variáveis inteiras, e parte dessas variáveis serão fixadas para a iteração seguinte. A heurística *Relax-and-Fix* (*RF1*) de Mohammadi et al. (2010), define, a cada iteração, o produto a ser produzido na posição n, no período t. A heurística *RF1* faz N iterações a cada período, definindo ao final de cada período a sua sequência de produção. Consequentemente o número total de iterações é dado por $T \cdot N$. A cada iteração um par (n, t) é atribuído. As variáveis que tornam a ser inteiras na iteração (n, t) são dadas por y_{ijt}^{n}, e, ao final da resolução do *MIP*, são todas congeladas para as iterações posteriores.

4.4 Times Assíncronos

A metaheurística Times Assíncronos (A-Teams) foi proposta por de Souza e Talukdar (1993). Um A-Teams é composto por agentes e memórias compartilhadas, e pode ser caracterizado predominantemente por: 1) Os agentes são autônomos e tomam decisões de seleção de soluções, estratégia de uso das mesmas e controle de alocação de memória; 2) As comunicações são assíncronas, assim os agentes podem ler e escrever informação nas memórias compartilhadas sem qualquer sincronização entre eles; 3) Os agentes tomam, modificam e salvam informações continuamente nas memórias compartilhadas. A-Teams são efetivos na resolução de problemas combinatórios de difícil resolução, onde há vários algoritmos implementados, mas que nenhum deles é totalmente satisfatório. Chega-se a um meio termo combinando-os para obter melhores respostas num tempo reduzido. Para uma leitura mais avançada no tema recomenda-se Talukdar et al. (2003).

Os agentes são normalmente heurísticas simples, mas metaheurísticas (busca tabu, *simulated annealing*) também podem ser usadas. Os agentes são divididos em agentes construtores, destruidores, de melhoria e de migração. Os agentes construtores são responsáveis pela criação e inclusão de soluções nas memórias compartilhadas. Os agentes de melhoria lêem soluções das memórias compartilhadas e, após processamento, retornam soluções para as memórias, que são inseridas juntamente com a eliminação de soluções da memória feita pelos agentes destruidores. Os agentes de migração podem mudar as soluções de memória a memória, podendo atuar nessas soluções, seja factibilizando-as para outra memória (as memórias podem possuir restrições diferentes, podendo ser relaxadas), seja aumentando a diversidade de soluções entre as memórias. Qualquer agente pode ser excluído/incluído do/no conjunto de agentes facilmente. Assim, a heurística é bastante flexível, podendo os agentes ser adicionados ou retirados, inclusive de maneira adaptativa, melhorando o desempenho da heurística.

A metaheurística Times Assíncronos proposta possui um agente construtor, um único agente destruidor e sete agentes de melhoria. A cada nova solução, o processo de cálculo da função objetivo (1) e das restrições de capacidade (5) deve ser realizado. Esse processo é definido como uma iteração da metaheurística. Um agente pode consumir uma ou mais iterações. A solução gerada pelos agentes pode ser infactível. A infactibilidade é medida nas restrições de capacidade (5). As outras restrições são sempre obedecidas, para qualquer solução criada ou melhorada. Na memória compartilhada as soluções são ordenadas, da melhor para a pior, a partir de dois critérios: o menor valor da função objetivo para as soluções factíveis e a menor capacidade adicional utilizada (para soluções infactíveis).

O agente construtor dimensiona o número de lotes de cada período, tomando os valores que satisfazem a demanda do mesmo (política lote-por-lote). A sequência dos produtos é dada aleatoriamente. Dessa forma, estoques não são realizados nas soluções geradas. O agente de destruição considera a qualidade da solução, de acordo com os dois critérios de ordenação da memória compartilhada. Se a solução gerada por um agente pertencer a memória compartilhada ou ser pior que as soluções da memória compartilhada, então a memória compartilhada fica inalterada. Caso contrário, a solução gerada é inserida na memória compartilhada de forma ordenada, respeitando a capacidade da memória.

Todos os agentes de melhoria escolhem uma solução da memória compartilhada por meio de seleção por torneio de tamanho 2 (*Tournament Selection*). Assim, são escolhidas, aleatoriamente, duas soluções da memória compartilhada e a melhor delas é utilizada pelo agente de melhoria. Sempre que um agente atuar na solução provocando transferências de produção de um período (período fonte)para outro período (período destino), caso não haja a produção de determinado produto específico no período destino, a produção transferida será inserida numa posição aleatória.

O primeiro agente adota a vizinhança definida por troca de dois produtos quaisquer na sequência de produção. Um período aleatório é escolhido e o agente analisa soluções vizinhas à solução inicial, alterando a sua sequência de produção, mantendo as decisões relacionadas ao dimensionamento de lotes. O segundo agente é análogo ao primeiro agente, entretanto a vizinhança definida é de inserção simples, onde se escolhe um produto da sequência e o insere em outra posição. O terceiro agente busca melhorar uma solução alterando a sequência de produção de dois períodos consecutivos. Utilizando inserção simples, o último produto de determinado período e a primeira posição da sequência do período seguinte são modificadas para que os produtos que ocupam essas posições sejam os mesmos. Esse agente aposta que a preservação da preparação das máquinas entre períodos gera melhores soluções.

O quarto agente de melhoria adianta toda a produção de um produto de um período para outro imediatamente anterior. O objetivo desse agente é eliminar custos de preparação de máquinas e diminuir o consumo de capacidade do processo de produção. Em contrapartida haverá custos relacionados ao estoque de produtos. O agente consome apenas uma iteração. O quinto agente é análogo ao agente anterior, porém analisa a capacidade de cada máquina do período anterior, permitindo a transferência parcial de produção de um produto de um período para o período adjacente anterior. O sexto agente, ao contrário dos dois anteriores, posterga a produção excedente. Seleciona aleatoriamente um período e um dos produtos que possui produção excedente nesse período e posterga toda a produção excedente para o período imediatamente posterior, eliminando parte do estoque desse período.

O sétimo agente é dividido em duas fases e mescla os dois últimos agentes. Na primeira fase posterga-se a produção de um produto excedente de um período para o período posterior. Na segunda fase, dados os mesmos períodos, adianta-se parte da produção de um produto distinto do escolhido anteriormente. O agente busca possibilidades de melhorias no custo de produção, principalmente pela troca de produtos que estarão em estoque em determinado período, além de possíveis reduções no consumo da capacidade.

5 Experimentos Computacionais

Todos os testes foram realizados usando um computador com processador Intel Core 2 Duo E7500 com velocidade de $2.93GHz$, memória L2 cache de $3MB$ e $3,2GB$ de RAM sob o sistema operacional Windows 7. A metaheurística Times Assíncronos foi implementada em C e os tempos de CPU foram computados usando a função clock(). A resolução exata do modelo matemático e as heurísticas baseadas em *MIP* foram implementadas utilizando bibliotecas do CPLEX 12.1. Todos os métodos foram programados para utilizar apenas um processador (programação sequencial).

Os problemas-teste foram gerados de acordo com os experimentos propostos no artigo de Mohammadi et al. (2010). Considerando as combinações do número de produtos, máquinas e períodos (N, M, T), os problemas-teste possuem distintas configurações, desde $(3, 3, 3)$ a $(7, 7, 7)$. Os parâmetros de produção são gerados de maneira uniforme, de acordo com intervalos e valores propostos em Mohammadi et al. (2010). Dessa forma, seja $U(min; max; passo)$ a escolha aleatória no intervalo fechado entre min e max dado um $passo$, que restringe o número de elementos disponíveis no intervalo ao conjunto $\{min; min+passo; \ldots; max - passo; max\}$, considerando que o $max - min$ é divisível por $passo$.

A demanda (d_{jt}) é dada no intervalo $U(0; 180; 10)$ enquanto os custos de estoque (h_{jm}) são dados por $U(0,2; 0,4; 0,1)$. Tanto o tempo quanto os custos de processamento (b_{jm} e p_{jmt}, respectivamente) são gerados a partir do conjunto $U(1,5; 2,0; 0,1)$. Da mesma maneira, os tempos e custos de preparação de máquinas (S_{ijm} e W_{ijm}) são definidos por $U(35; 70; 5)$. No início da programação de produção todas as máquinas estarão configuradas

para determinado produto. O último parâmetro, a capacidade das máquinas, é dado por $\beta \cdot Cap_{min}$, sendo β um fator que pode ser 60%, 70% e 80%, enquanto Cap_{min} é a capacidade mínima para o caso extremo (caso onde todos os parâmetros possuem valor máximo). Para cada combinação (N, M, T) quinze problemas-teste distintos foram gerados, considerando que são cinco problemas-teste com valores diferentes de capacidade devido ao fator β.

Para avaliar a qualidade das heurísticas baseadas em *MIP*, o tempo foi limitado de acordo com o número de iterações necessárias para se encontrar uma solução. Assim, as heurísticas baseadas em estratégia de horizonte rolante (*RH1*, *RH2* e *RH3*) possuem tempo de solução limitado em $(7200/T)$ segundos para cada *MIP*. A heurística do tipo *relax-and-fix* (*RF1*) possui cada iteração (resolução de um *MIP*) limitada em $(7200/(T \cdot N))$ segundos. A resolução exata e o limitante inferior são limitados em 7200 segundos.

A metaheurística *A-Teams* proposta possui uma única memória compartilhada, com uma população de dez soluções. A metaheurística está limitada a cem mil iterações. A vizinhança de busca para alguns agentes está limitada a dez soluções.

Os resultados são dados pela Tabela 1. A primeira coluna denota as configurações (N, M, T) das classes de problemas-teste. A coluna "Original" corresponde à resolução do *MIP* original (1-13) e seus resultados são dados pela média dos resultados dos problemas-teste. Note que uma solução ótima, nesse caso, pode não ser encontrada, sendo então utilizada a melhor solução factível. As outras colunas correspondem aos métodos e seus resultados, dados de maneira relativa à solução gerada no problema inicial. Os tempos médios são dados entre parentêses abaixo da linha de resultados, para cada classe de problemas-teste.

	Original	LB2	RH1	RH2	RH3	RF1	A-Teams
$(3,3,3)$	$5379,028$	$-1,421\%$	$2,530$	$0,808\%$	$1,753\%$	$1,565\%$	$0,382\%$
	$(0,427)$	$(0,155)$	$(0,438)$	$(0,502)$	$(0,085)$	$(0,592)$	$(0,781)$
$(5,3,3)$	$7880,492$	$-1,621\%$	$4,163\%$	$0,938\%$	$4,338\%$	$5,999\%$	$0,623\%$
	$(16,379)$	$(0,631)$	$(2,044)$	$(2,257)$	$(0,139)$	$(1,719)$	$(0,970)$
$(3,5,3)$	$6654,179$	$-1,098\%$	$6,035\%$	$0,389\%$	$6,385\%$	$6,437\%$	$0,816\%$
	$(0,405)$	$(0,249)$	$(0,517)$	$(0,505)$	$(0,092)$	$(0,588)$	$(0,981)$
$(3,3,5)$	$7582,302$	$-0,310\%$	$3,986\%$	$0,779\%$	$4,999\%$	$4,733\%$	$1,155\%$
	$(0,72)$	$(0,395)$	$(0,596)$	$(0,730)$	$(0,133)$	$(1,054)$	$(1,114)$
$(5,5,5)$	$22468,817$	$-1,872\%$	$5,321\%$	$1,515\%$	$5,922\%$	$6,606\%$	$1,711\%$
	$(4605,694)$	$(25,621)$	$(24,476)$	$(24,607)$	$(0,508)$	$(5,307)$	$(2,193)$
$(7,5,5)$	$30164,791$	$-2,557\%$	$7,236\%$	$1,034\%$	$8,049\%$	$8,934\%$	$1,704\%$
	(7200)	$(274,215)$	$(357,426)$	$(356,123)$	$(1,178)$	$(25,592)$	$(2,523)$
$(5,7,5)$	$30353,459$	$-2,685\%$	$6,673\%$	$1,766\%$	$7,485\%$	$8,604\%$	$2,356\%$
	$(2757,594)$	$(42,364)$	$(54,407)$	$(55,222)$	$(0,626)$	$(6,670)$	$(2,795)$
$(5,5,7)$	$30563,221$	$-2,495\%$	$7,500\%$	$1,296\%$	$7,917\%$	$7,946\%$	$3,124\%$
	(7200)	$(250,965)$	$(40,338)$	$(39,798)$	$(1,004)$	$(12,660)$	$(2,885)$
$(7,7,7)$	$58605,555$	$-3,171\%$	$4,823\%$	$-1,865\%$	$6,043\%$	$6,472\%$	$-0,595\%$
	$(6750,834)$	$(6700,481)$	$(1638,318)$	$(1798,801)$	$(4,820)$	$(101,006)$	$(4,598)$
Média	$20122,205$	$-1,768\%$	$5,414\%$	$0,846\%$	$5,797\%$	$6,302\%$	$1,331\%$
Geral	$(3170,250)$	$(810,564)$	$(235,395)$	$(253,172)$	$(0,954)$	$(17,243)$	$(2,093)$

Tabela 1: Comparação entre as heurísticas

De acordo com os experimentos realizados, percebemos que o limitante inferior utilizado é bastante próximo do valor de uma solução ótima. A heurística *RH2* obtém melhores resultados que a análoga *RH1*, num tempo similar. Isso mostra que fixar as variáveis de dimensionamento de lotes pode ser uma má idéia numa estratégia de horizonte rolante, para esse problema. A heurística *RH3* e a metaheurística *A-Teams* possuem os menores tempos computacionais. As melhores soluções são dadas pelo procedimento *RH2* e pelo

método Times Assíncronos, porém, para problemas-teste maiores, com relação ao tempo computacional gasto, a metaheurística obtém desempenho superior à heurística *RH2*.

6 Conclusões

Consideramos o problema de planejamento de produção envolvendo o dimensionamento e sequenciamento de lotes em um ambiente de produção *flowshop*, com tempos de preparação de máquinas dependentes da sequência, *setup carryover* e estoques intermediários conforme proposto por Mohammadi et al. (2009, 2010).

Para resolver o problema implementamos procedimentos exatos e heurísticas já encontradas na literatura e uma metaheurística do tipo Times Assíncronos foi proposta. De acordo com Mohammadi et al. (2010) a aplicação de metaheurísticas para a solução do problema é uma área promissora de pesquisa. Dessa forma, até onde sabemos, o uso da metaheurística proposta é inédito para o problema.

Os resultados obtidos sobre um conjunto de problemas-teste foram comparados e analisados. Os resultados finais do *A-Teams* proposto são bastante promissores. Um fato importante é que *A-Teams* são bastante flexíveis, podendo os agentes ser adicionados ou retirados de maneira adaptativa. Uma área de pesquisa futura é utilizar metaheurísticas híbridas, baseadas em *MIP*, que, por meio da resolução de problemas menores pode chegar a soluções melhores. Para metaheurísticas pode ser interessante também o uso de estratégias de paralelização. O método Times Assíncronos, por exemplo, se encaixa muito bem nessa perspectiva futura, já que seus agentes são assíncronos e independentes.

7 Agradecimentos

Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro para o desenvolvimento deste trabalho (Processo: 133235/2009-2).

Referências

de Souza, P. S. e Talukdar, S. N. (1993), Asynchronous organizations for multialgorithm problems, *ACM Symposium on Applied Computing*, Indianapolis.

Drexl, A. e Kimms, A. (1997), Lot sizing and scheduling: Survey and extensions, *European Journal of Operational Research*, 99, 221–235.

Fandel, G. e Stammen-Hegene, C. (2006), Simultaneous lot sizing and scheduling for multi-product multi-level production, *International Journal of Production Economics*, 104, 308–316.

Garey, M. R., Johnson, D. S. e Sethi, R. (1976), The complexity of flowshop and jobshop scheduling, *Mathematics of Operations Research*, 1, 117–129.

Jans, R. e Degraeve, Z. (2008), Modeling industrial lot sizing problems: a review, *International Journal of Production Research*, 46, 1619–1643.

Johnson, S. M. (1954), Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, 1, 61–68.

Karimi, B., Fatemi Ghomi, S. M. T. e Wilson, J. M. (2003), The capacitated lot sizing problem: a review of models and algorithms, *Omega*, 31, 365–378.

Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B., e Torabi, S.A. (2009), A new algorithmic approach for capacitated lot-sizing problem in flow shops with sequence-dependent setups, *The International Journal of Advanced Manufacturing Technology.*

Mohammadi, M., Fatemi Ghomi, S.M.T., Karimi, B., e Torabi, S.A. (2010), Mip-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups, *International Journal of Production Research*, 48, 2957–2973.

Nawaz, M., Enscore Jr., E.E. e Ham, I. (1983), A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, *Omega - The International Journal of Management Science*, 1, 91–95.

Pinedo, M. *Scheduling: Theory, Algorithms and Systems*, Springer, 3rd edition, 2008.

Potts, C.N. e Van Wassenhove, L.N. (1992), Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity, *Journal of Operational Research Society*, 43, 395–406.

Talukdar, S., Murthy, S. e Akkiraju, R., Asynchronous Teams, *Handbook of Meta-heuristics*, Springer, 537–556, 2003.

Zhu, X. e Wilhelm, W. E. (2006), Scheduling and lot sizing with sequence-dependent setup: A literature review, *IIE Transactions*, 38, 987–1007.