



DIMENSIONAMENTO DE LOTES E PROGRAMAÇÃO DO FORNO NUMA FUNDIÇÃO DE PEQUENO PORTE

Silvio Alexandre de Araujo

Departamento de Informática, Universidade Estadual de Maringá,
Avenida Colombo, 5790, CEP 87020-900, Maringá, PR,
e-mail: silvio@din.uem.br

Marcos Nereu Arenales

Departamento de Ciências de Computação e Estatística,
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, C.P. 668, CEP13560-970, São Carlos, SP,
e-mail: arenales@icmc.usp.br

Alistair Richard Clark

Faculty of Computing, Engineering and Mathematical Sciences,
University of the West of England, Bristol, BS16 1QY, England,
e-mail: alistair.clark@uwe.ac.uk

Recebido em 15/5/2003

Aceito em 25/3/2004

Resumo

Este trabalho trata de um problema prático de dimensionamento e seqüenciamento de lotes em uma fundição. Existem dois níveis de decisão inter-relacionados: programação do forno, em que um tipo de liga é escolhido para ser produzido em cada período, e planejamento das máquinas de moldagem, que especifica a quantidade de cada item deve ser produzida em cada período, isto é, dimensionamento dos lotes. Um modelo de programação inteira mista é proposto e um método de solução baseado em busca local é desenvolvido. O planejamento da produção é feito usando a técnica de horizonte rolante, em que somente a programação para os períodos imediatos é implementada. Os resultados dos testes computacionais são analisados e comparados com os resultados obtidos utilizando um pacote comercial.

Palavras-chave: programação inteira mista, planejamento e programação da produção, fundições.

1. Introdução

Atualmente, no Brasil, as fundições estão presentes em todas as regiões, produzindo desde peças simples de uso doméstico até as mais sofisticadas, como autopeças de veículos e peças e/ou partes de máquinas e equipamentos de uso industrial. Segundo o relatório da ABIFA-Associação Brasileira de Fundição (2002), o setor produziu em novembro de 2002 uma média de 8.735 toneladas/dia e gera cerca de 41.500 empregos diretos.

A grande maioria da produção brasileira de fundidos é fabricada por **fundições cativas**, que são departamentos de grandes empresas (tais como montadoras de automóveis) cuja produção é destinada, basicamente, para consumo próprio. Estas fundições cativas estão orientadas para a produção em série, automatizada e fabricam grandes quantidades de cada item. Com isso, a demanda de empresas menores que precisam do produto fundido em quantidades pequenas, como as empresas que fabricam implementos agrícolas e peças para manutenção de máquinas, é atendida por fundições de pequeno e médio porte chamadas **fundições de mercado**, que se dedicam exclusivamente às vendas para terceiros.

Nos últimos anos, as fundições de mercado vêm registrando um grande crescimento e a prática gerencial baseada apenas no bom senso e na experiência do administrador de processos não tem sido suficiente. Fernandes e Leite (2002) fizeram uma pesquisa em 30 fundições de mercado, pertencentes aos cinco principais pólos do interior do estado de São Paulo, e concluíram que 80% das fundições têm interesse em investir em sistemas automatizados para gestão da produção. Várias delas já possuem alguns sistemas informatizados, por exemplo, para emissão de ordens de compra e produção e para codificação de materiais, porém apenas uma possui sistema informatizado para a programação da produção.

Este trabalho enfoca problemas de planejamento e programação da produção que surgem numa fundição de mercado do interior do estado de São Paulo. Embora o trabalho seja desenvolvido para um caso particular, pode ser estendido para várias outras pequenas e médias fundições que necessitam de uma melhoria na estrutura organizacional, mas dispõem de recursos escassos para investir em pesquisas, além de não terem essa cultura.

Revisões bibliográficas de trabalhos que tratam de problemas de planejamento e programação da produção podem ser encontradas em Billington et al. (1983), Bahl et al. (1987), Maes e van Wassenhove (1988), Goyal e Gunasekaran (1990), Potts e van Wassenhove (1992), Kuik et al. (1994), Drexl e Kimmis (1997), Potts e Kovalyov (2000) e Karimia et al. (2003).

Entretanto, existem poucos trabalhos que tratam de problemas de planejamento e programação da produção voltados para o setor de fundições. Dentre os trabalhos encontrados estão alguns do nosso grupo de pesquisa: Gonçalves-Vianna e Arenales (1995), Silva (2001), Santos-Meza et al. (2002), Araujo e Arenales (2003) e Araujo e Arenales (2004).

Foram encontrados na literatura pesquisada alguns trabalhos realizados para grandes fundições que tratam, especificamente, do problema de seqüenciamento das tarefas nas máquinas. Sounderpandian e Balashanmugam (1991), Gravel et al. (2000). No entanto, a grande maioria dos trabalhos encontrados na literatura trata do problema de programação da produção em grandes siderúrgicas. Tang et al. (2001) fazem uma revisão bibliográfica de trabalhos de planejamento e programação da produção aplicados à indústria de fabricação de aço. Vários trabalhos tratam de problemas práticos encontrados em grandes siderúrgicas: Petersen et al. (1992), Hamada et al. (1995), Bowers et al. (1995), Hendry et al. (1996), Lee et al. (1996), Lopes et al. (1998), Tang et al. (2000a), Tang et al. (2000b).

2. O Processo produtivo

Um fluxograma do processo produtivo foi criado e é apresentado na Figura 1:

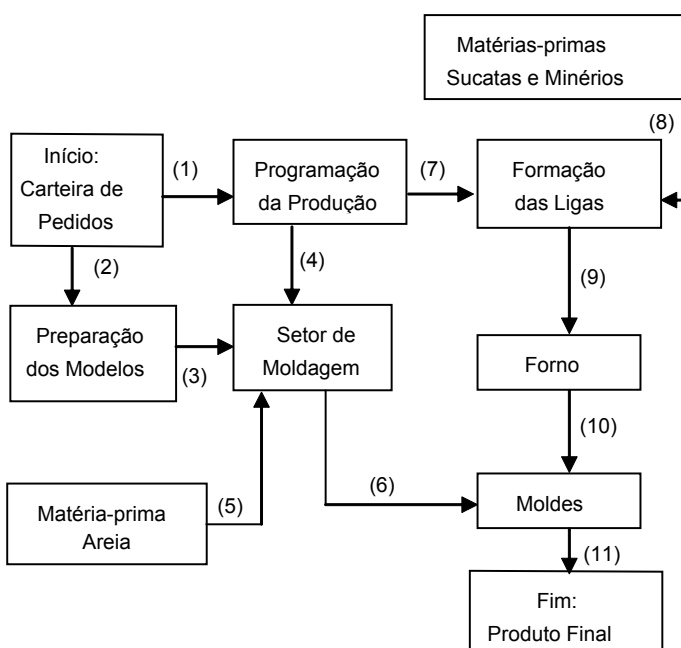


Figura 1. Processo produtivo da fundição.

A lista de itens demandados é organizada numa **carteira de pedidos** com todas as especificações, incluindo prazo de entrega. As informações dos pedidos dos clientes são passadas para o setor no qual é feita a **programação da produção (1)**. A programação é feita diariamente de acordo com a capacidade da empresa. O gerente de processos toma as decisões baseadas no seu bom senso, dando preferência a itens que estejam com o prazo de entrega bastante atrasado e itens pedidos por clientes “especiais” (grandes clientes ou clientes que estão pressionando). Além disso, alguns itens são fabricados de forma a tentar aproveitar a capacidade total do forno, ou seja, é feito um encaixe de itens que utilizam a mesma liga. A **carteira de pedidos** é passada também para o setor de **preparação dos modelos (2)**. Em geral, os modelos são fornecidos pelo cliente e são utilizados para fazer os moldes dos itens.

As decisões tomadas pela **programação da produção** são passadas para o **setor de moldagem (4)**, que utiliza a **areia como matéria prima (5)** e os **modelos (3)** para a fabricação dos **moldes (6)**. O **setor de programação da produção** também tem a função de definir quais as ligas e em que quantidades serão produzidas. As decisões são tomadas de acordo com os itens que deverão ser produzidos. Estas decisões são passadas para a **formação das ligas (7)**, onde é feita a mistura de **matérias-primas (sucatas e minérios de ferro) (8)** que deverá formar determinada liga. Esta mistura é então colocada no **forno** para a fusão **(9)**.

Atualmente, a empresa utiliza apenas dois fornos, com capacidade de 120 kg e 360 kg e tempo de fusão de 45 minutos e 1 hora e 55 minutos, respectivamente. Devido à limitação da energia contratada, não é possível a utilização dos dois simultaneamente. Além disso, não é aconselhável que seja feita a troca de forno durante um mesmo dia de trabalho, devido ao alto tempo de preparação. No entanto, essas trocas vêm ocorrendo na prática.

Um forno pode ser ligado com uma carga abaixo de sua capacidade total (deve-se ter pelo menos 1/3 da capacidade total). Apesar da subutilização do forno não ser ideal do ponto de vista de programação, é preferível subutilizar o forno de 360 kg (se já estiver ligado) com apenas 120 kg, do que desligar o forno de 360 kg e ligar o de 120 kg. Em muitos casos a subutilização do forno ocorre devido a falhas na programação, sendo, então, necessário fazer um lote de itens que não utilize toda a capacidade do forno. Outro fato que ocorre, devido ao mau planejamento, é a sobra de liga no forno, sendo necessário despejar o líquido no chão para formar lingotes, que retornarão para o estoque de matéria-prima. Com o atual nível de produção, é comum ocorrerem filas de moldes esperando pela liga que deve ser fundida.

Após a fusão, a liga é vazada nos **moldes (10)** previamente preparados. O tempo de solidificação, para a maioria das ligas, é de apenas 10 minutos. Após o vazamento é feito um choque térmico, colocando a peça na água. Os canais de vazamento e massalotes são retirados e voltam a ser **maté-**

ria-prima (sucata). Finalmente, a peça é levada para um pequeno estoque de **produtos finais (11)**.

3. Modelagem matemática

Como ocorre na maioria dos modelos matemáticos que representam problemas práticos, algumas simplificações foram consideradas no modelo apresentado nesta seção. Tais simplificações viabilizam a resolução do modelo e não comprometem muito a representação prática do problema.

Na modelagem matemática é considerado apenas o forno maior (360 kg). Cabe lembrar que não é possível a utilização dos dois fornos ao mesmo tempo, no entanto, em alguns casos particulares, pode acontecer que a soma dos pesos das demandas de todos os itens que devem ser produzidos numa determinada liga seja menor que 120 kg, ou seja, abaixo de 1/3 da capacidade do forno de 360 kg e, portanto, não é possível utilizá-lo. Neste caso, tais demandas devem ser retiradas da carteira de pedidos e acumuladas numa carteira separada até completar uma quantidade de cargas suficiente, para que sejam feitas num mesmo dia de trabalho, utilizando, neste dia, somente o forno menor (120 kg). No caso do vencimento do prazo de entrega dos itens, que estão na carteira separada, ocorrer antes que se tenha uma quantidade suficiente para um dia de trabalho, existe a possibilidade de se trabalhar num sábado pela manhã, em hora extra, com o forno menor.

Antes de apresentar o modelo matemático, serão feitas algumas observações com respeito às características do problema prático em questão. A carteira de pedidos usada nos testes computacionais tem 26 diferentes ligas, as quais são utilizadas para produzir 403 diferentes itens demandados. Cada item deve ser feito por uma única liga, os itens são demandados em diferentes quantidades (1 a 400 unidades) e têm diferentes pesos (0,1 Kg a 300 Kg). A capacidade do forno é 360 Kg por carga e cada carga demora em torno de 2 horas, sendo possível fazer no máximo 10 cargas do forno por dia. Um horizonte de planejamento de 5 dias é suficiente, segundo o proprietário da empresa, totalizando 50 períodos, se cada carga do forno for planejada separadamente. Por fim, a minimização do atraso no atendimento à demanda aparece como o principal objetivo da empresa.

No modelo de programação inteira mista proposto a seguir, o forno é considerado o gargalo do processo produtivo e os seguintes dados são utilizados:

Índices:

- $k = 1, \dots, K$ tipos de ligas;
- $i = 1, \dots, N$ tipos de itens; e
- $t = 1, \dots, T$ períodos de tempo.

Dados do problema:

- Cap - capacidade do forno por período;
- L - número máximo de cargas do forno por dia;
- ρ_i - peso bruto do item i ;
- α_i - dias de atraso do item i quando $t = 0$, isto é, no início da programação (para pedidos que não estão em atraso, α_i recebe valores não positivos, dependendo da quantidade de dias

entre o início da programação e o dia de entrega do pedido e $\alpha_i = 0$ significa que o prazo de entrega do item vence no dia em que está sendo feita a programação);

β_i - nível de prioridade do item i (por meio deste fator é possível “forçar” que determinado item seja produzido);

d_{it}^- - demanda da peça i no período t (se $t = L, 2L, 3L, \dots$ então $d_{it}^- \geq 0$ senão $d_{it}^- = 0$);

$S(k)$ - conjunto de itens i que usam a liga k (cada item utiliza um, e somente um, tipo de liga, ou seja: $\{1, \dots, N\} = S(1) \cup \dots \cup S(k), S(h) \cap S(j) = \emptyset, \forall h \neq j$);

H_{it}^- - penalidade pelo atraso na entrega de uma unidade do item i no período t (se $t = L, 2L, 3L, \dots$ então $H_{it}^- = \rho_i \alpha_i \beta_i$ senão $H_{it}^- = 0$);

H_{it}^+ - penalidade por antecipação de uma unidade do item i no período t (se $t = L, 2L, 3L, \dots$ então $H_{it}^+ = \rho_i$ senão $H_{it}^+ = 0$); e

S_k - penalidade por preparação da liga k .

Variáveis do problema:

X_{it} - quantidade produzida do item i no período t ;

I_{it}^+ - quantidade estocada do item i no período t ;

I_{it}^- - quantidade atrasada do item i no período t ;

Y_t^k - variável binária ($Y_t^k = 1$ indica que o forno é preparado para produzir a liga k no período t , caso contrário, $Y_t^k = 0$); e

Z_t^k - variável que indica se é cobrado o custo de preparação para a liga k no período t : $Z_t^k = 0$ se $Y_{t-1}^k \geq Y_t^k$ e $Z_t^k = 1$ se $Y_{t-1}^k < Y_t^k$. Embora esta variável seja binária em essência, pode ser modelada como contínua.

Modelo:

$$\text{Minimizar } \sum_{i=1}^N \sum_{t=1}^T (H_{it}^- I_{it}^- + H_{it}^+ I_{it}^+) + \sum_{k=1}^K \sum_{t=1}^T (S_k Z_t^k) \quad (1)$$

Sujeito a:

$$I_{i,t-1}^+ - I_{i,t-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, T \end{matrix} \quad (2)$$

$$\sum_{i \in S(k)} \rho_i X_{it} \leq Cap Y_t^k \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, T \end{matrix} \quad (3)$$

$$Z_t^k \geq Y_t^k - Y_{t-1}^k \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, T \end{matrix} \quad (4)$$

$$\sum_{k=1}^K Y_t^k \leq 1 \quad t = 1, \dots, T \quad (5)$$

$$Y_t^k \in \{0,1\} \quad (Y_0^k = 0) \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, T \end{matrix} \quad (6)$$

$$Z_t^k \geq 0 \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, T \end{matrix} \quad (7)$$

$$X_{it} \geq 0 \text{ inteira} \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, T \end{matrix} \quad (8)$$

$$I_{it}^+ \text{ e } I_{it}^- \geq 0 \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, T \end{matrix} \quad (9)$$

Antes de explicar a formulação (1)-(9) as seguintes observações devem ser feitas:

- i - a demanda (d_{it}) assim como as penalidades por atraso e antecipação (H_{it}^- e H_{it}^+) são positivas somente no último período de cada dia. Com isso, esses valores são diários na prática, o que faz mais sentido do que considerar que esses valores sejam positivos a cada carga do forno, que representa algumas horas;
- ii - se um mesmo item i é demandado em diferentes períodos, serão considerados como diferentes itens, um para cada período. Esta consideração teve que ser feita para retratar o principal objetivo da fundição, que consiste na minimização do número máximo de dias de atraso para determinado pedido. Para retratar esse objetivo, foram associadas penalidades a cada item (H_{it}^-) que crescem de acordo com o número de dias em atraso (α_i). Assim, se, no início da programação, determinado item tem uma demanda em atraso e tem também uma demanda para algum dia posterior ao dia do planejamento, serão considerados diferentes valores α_i e, conseqüentemente, penalidades diferentes serão associadas a este item, levando-nos à consideração de dois itens diferentes. Embora este seja um fator que pode aumentar bastante o tamanho do problema, isso não ocorre no caso prático em questão, devido às características da carteira de pedidos, na qual, raramente, se tem um pedido para o mesmo item em dois dias diferentes; e
- iii - o fato de considerar itens diferentes quando existe apenas um, com prazos de entrega distintos, não causa falsas preparações, pois as preparações são das ligas, as quais produzem determinados conjuntos de peças e, os itens diferentes (que na verdade representam apenas um) seguramente são fabricados com a mesma liga. Os modelos clássicos, que associam preparações aos itens, criariam falsas preparações ao distinguirem itens com prazos distintos.

Na formulação (1)-(9), a função objetivo (1) procura retratar a maior preocupação da empresa que consiste em minimizar os atrasos na entrega dos pedidos. Para tanto, uma penalidade é associada às variáveis I_{it}^- que representam estoques negativos, ou seja, atraso no atendimento da demanda. Além disso, a função objetivo minimiza também as penalidades por antecipação associadas às variáveis de estoque I_{it}^+ . No entanto, as penalidades por atrasos são muito maiores que as penalidades por antecipação.

Obviamente, o desejo de não ter atrasos poderia ser modelado com restrições, o que, no entanto, poderia introduzir infactibilidades desnecessárias, uma vez que é admissível o atraso no atendimento da demanda, ou seja, estoques negativos. A variável I_{it}^- funciona como uma variável artificial usada para obter uma solução factível. Se $I_{it}^- > 0$ para algum i , significa que não foi possível atender à demanda no horizonte de planejamento estabelecido com a capacidade disponível nesse horizonte, pois existe atraso no último período para algum item. No entanto, se $I_{it}^- > 0$, para algum i com $t \neq T$ e

$I_{it}^- = 0 \forall i$, então existe capacidade suficiente para produzir a demanda dentro do horizonte de planejamento estabelecido. Vale observar que, os estoques negativos correspondem a atrasos e o gerente de programação pode, calibrando os parâmetros da função objetivo (1), proibir ou não atrasos de certos itens. Além disso, os estoques negativos fornecem uma ferramenta para a avaliação dos prazos de entrega. Por exemplo, se $d_{i1} = 30$ e $I_{i1}^- = 10$, $I_{i2}^- = 5$, $I_{i3}^- = 0$, significa que a demanda para o item i será totalmente atendida com 3 dias de atraso. Desta forma, é possível alertar o cliente e, se necessário, recalibrar os parâmetros relativos a esta peça.

Ainda na função objetivo (1), existe uma terceira parcela que, assim como as penalidades por antecipações (H_{it}^+), tem um peso pequeno em relação às penalidades por atraso (H_{it}^-). Esta parcela consiste nas penalidades por preparação que é um valor cobrado de forma a evitar as trocas de ligas entre períodos. Estes custos de preparação foram acrescentados ao modelo para representar o desejo da empresa em tentar minimizar tais trocas de ligas. Além disso, em visitas a outras fundições foi possível observar que tais custos podem ser bastante expressivos. Portanto, o modelo com penalidades por preparação é mais geral e facilita a extensão para outros casos práticos. No entanto, para o estudo em questão, as decisões com respeito a que itens produzir em cada período são tomadas, principalmente, de acordo com as penalidades por atrasos.

As restrições (2) são de balanceamento de estoque e consideram tanto estoques positivos como negativos. As restrições (3) são devidas à limitação de capacidade do forno, em que o peso total da liga produzida no período é limitado pela capacidade do forno. As restrições (3) asseguram também que itens, que utilizam determinada liga, só poderão ser produzidos em determinado período, se a liga for fundida neste período ($Y_t^k = 1$ significa que a liga k é produzida no período t). Mudanças de ligas são representadas pelas restrições (4). As restrições (5) garantem que apenas uma liga pode ser produzida em cada carga do forno. Para representar um período ocioso (onde nenhuma peça deve ser produzida), a mesma liga que estava sendo feita no período anterior é mantida, com isso, não é cobrado nenhum custo de preparação.

Devido às características da empresa, na qual existem alguns pedidos para a fabricação de apenas 1 ou 2 itens pesados, a integralidade das variáveis de produção X_{it} não podem ser relaxadas como na maioria dos modelos de dimensionamento de lotes. Seria possível relaxar tais variáveis para alguns itens leves que têm quantidades demandadas relativamente grandes. No entanto, como os itens têm de ser fabricados a partir de fornadas individuais (não é possível deixar um molde com apenas 60% de liga e na próxima fornada completar o molde), tais itens são bastante representativos em cada fornada. Dessa forma, as variáveis de produção foram mantidas inteiras, representando uma classe de problemas de dimensionamento de lotes em que tais não podem ser relaxadas. As variáveis Y_t^k são binárias, sendo que, $Y_0^k = 0 \ k = 1, \dots, K$. Embora as variáveis Z_t^k sejam contínuas,

a função objetivo em conjunto com as restrições (4) fazem com que assumam, na otimalidade, apenas valores 0 ou 1.

O modelo (1)-(9) tem algumas semelhanças com os modelos de dimensionamento e seqüenciamento de lotes apresentados em Drexl e Kimms (1997). Observe que as decisões de seqüenciamento já estão embutidas, quando se determina a seqüência de cargas do forno pelas variáveis Y_t^k . Entretanto, de forma diferente dos modelos de Drexl e Kimms (1997), nos quais a preparação para o item i num determinado período permite a produção deste único item no período, aqui a preparação da liga k permite a produção de um conjunto de itens que utilizam a liga k . Assim, a idéia de *small bucket* está presente no modelo (1)-(9) de outra maneira, ou seja, permite a produção de um único conjunto de itens por período, e não de apenas um único item. Portanto, o modelo (1)-(9) pode ser considerado um modelo de dimensionamento (variáveis X_{it}) e seqüenciamento (variáveis Y_t^k) de lotes.

4. Método de solução

O modelo matemático (1)-(9), o qual consiste num modelo de programação inteira mista, com os dados reais ($N = 403$, $K = 26$ e um horizonte de planejamento de 5 dias), tem: 1.300 variáveis binárias Y_t^k , 20.150 variáveis inteiras X_{it} , 1.300 variáveis contínuas Z_t^k e 41.106 variáveis contínuas I_{it}^1 e I_{it}^- . Ao tentar resolver este problema num Pentium III 500 MHz com 512 MB de memória RAM, utilizando a linguagem de modelagem AMPL (Fourer et al., 1993) com o solver Cplex 7.1 (ILOG CPLEX 7.1, 2001), não foi possível encontrar uma solução de boa qualidade (ver Tabela 1).

Por outro lado, como as penalidades por atraso são subjetivamente estimadas e a carteira de pedidos é atualizada freqüentemente dentro do horizonte de planejamento, na prática, a solução ótima deste modelo requer ajustes periódicos. Conseqüentemente, é possível resolver o problema

de forma satisfatória, usando modelos menores que incluem variáveis inteiras somente para os períodos imediatos (para o problema prático em questão, serão considerados períodos imediatos apenas os períodos relativos ao primeiro dia de trabalho, ou seja, $L = 10$ cargas do forno). Posteriormente, o horizonte é rolado progressivamente para uma nova aplicação do modelo (Clark e Clark, 2000).

Portanto, na próxima seção, um modelo de horizonte rolante é proposto, no qual somente as decisões relativas aos períodos imediatos são implementadas, posteriormente o horizonte é rolado e o modelo é aplicado novamente com as informações atualizadas de demanda, estoque e capacidade.

4.1 Modelo de Horizonte Rolante

Para a apresentação do modelo de horizonte rolante são necessárias as seguintes modificações no modelo (1)-(9):

- Novo dado: l_t número máximo de cargas do forno no período t , $t = 1, \dots, L, \dots, T$;
- Redefinição de Y_t^k como: Y_t^k número de cargas do forno com a liga k no período t ;
- Substituição das restrições (5) por: $\sum_{k=1}^K Y_t^k \leq l_t$
 $t = 1, \dots, T.$ (5')

Desta forma, um período de tempo t pode ter diferentes tamanhos, dependendo do valor de l_t , ou seja, pode corresponder a uma única carga do forno ($l_t = 1$) ou a um conjunto de cargas ($l_t > 1$), por exemplo, um dia tem $l_t = 10$.

O modelo de horizonte rolante pode ser aplicado ao problema prático em questão fazendo-se ($l_t = 1$, $t = 1, \dots, 10$) e programando assim, o primeiro dia de planejamento de forma detalhada, considerando períodos curtos, nos quais cada carga é programada separadamente. Para os outros 4 dias de planejamento, basta considerar períodos grandes que agregam várias cargas do forno ($l_{11} = 10$, $l_{12} = 10$, $l_{13} = 10$ e $l_{14} = 10$). O planejamento para estes 4 dias futuros é feito

Tabela 1. Resultado do Cplex aplicado ao modelo (1)-(9).

Tempo de Busca	Melhor Solução Atual (MS)	Melhor Limitante Inferior (MLI)	GAP (%) (MS-MLI)/MS	B&B Nós Pesquisados
30 minutos	Não Factível	-----	-----	-----
52 minutos	631.441	546.339	13,50	1.522
5 horas	631.441	546.430	13,46	3.500
10 horas	631.441	546.587	13,44	4.500
20 horas	631.441	546.799	13,40	8.000

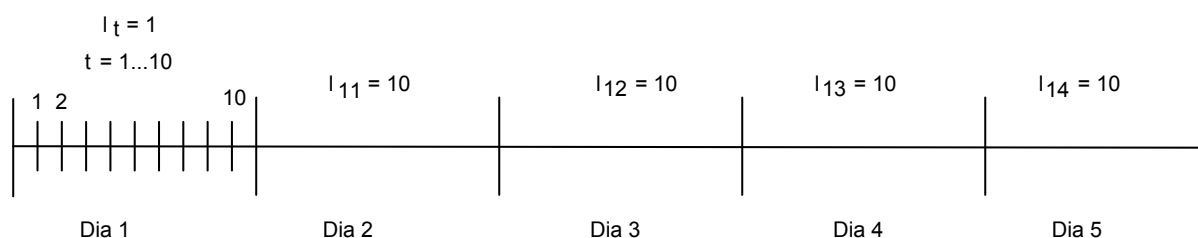


Figura 2. Descrição da aplicação do Método de Horizonte Rolante.

apenas para a avaliação da capacidade. Portanto, é possível fazer um planeamento para 5 dias de trabalho, utilizando apenas 14 períodos, como é ilustrado na Figura 2.

As decisões relativas aos dez primeiros períodos de tempo (primeiro dia) são realmente implementadas, enquanto que as decisões para os outros quatro dias são utilizadas apenas para que se tenha uma avaliação da capacidade disponível, dando também uma idéia do que deverá ser produzido e de possíveis atrasos. Posteriormente, o horizonte de planeamento é avançado em $L = 10$ períodos e o modelo é aplicado novamente com a carteira atualizada.

Cabe observar que, as variáveis Y_t^k são binárias para $t = 1, \dots, 10$, isto é, para o primeiro dia de programação, pois cada carga é programada separadamente. Além disso, as variáveis Z_t^k fazem sentido apenas para o primeiro dia de programação, sendo substituídas as restrições (4) por (4').

$$Z_t^k \geq Y_t^k - Y_{t-1}^k \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, L \end{matrix} \quad (4')$$

Para reduzir o tamanho do problema e o tempo computacional, as condições de integralidade das variáveis X_{it} são relaxadas para os períodos $t = L + 1, \dots, T$. O mesmo poderia ser feito para as variáveis Y_t^k , mas para se ter uma boa avaliação da capacidade, tais variáveis foram mantidas inteiras para os períodos $t = L + 1, \dots, T$. Portanto, as restrições (6)-(8) devem ser substituídas por (6')-(8''), dadas no modelo a seguir.

Substituindo no modelo (1)-(9) todas as restrições propostas anteriormente nessa seção, surge um novo modelo, o qual chamaremos de modelo de **Horizonte Rolante (HR)**:

Modelo HR:

$$\text{Minimizar} \sum_{i=1}^N \sum_{t=1}^T (H_{it}^- I_{it}^- + H_{it}^+ I_{it}^+) + \sum_{k=1}^K \sum_{t=1}^T (s_k Z_t^k) \quad (1)$$

Sujeito a:

$$I_{i,t-1}^+ - I_{i,t-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, T \end{matrix} \quad (2)$$

$$\sum_{i \in S(k)} \rho_i X_{it} \leq \text{Cap} Y_t^k \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, T \end{matrix} \quad (3)$$

$$Z_t^k \geq Y_t^k - Y_{t-1}^k \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, L \end{matrix} \quad (4')$$

$$\sum_{k=1}^K Y_t^k \leq I_t \quad t = 1, \dots, T \quad (5')$$

$$Y_t^k \in \{0,1\} \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, L \end{matrix} \quad (6')$$

$$Y_t^k \geq 0 \text{ inteira} \quad \begin{matrix} k = 1, \dots, K \\ t = L+1, \dots, T \end{matrix} \quad (6'')$$

$$Z_t^k \geq 0 \quad \begin{matrix} k = 1, \dots, K \\ t = 1, \dots, L \end{matrix} \quad (7')$$

$$X_{it} \geq 0 \text{ inteira} \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, L \end{matrix} \quad (8')$$

$$X_{it} \geq 0 \quad \begin{matrix} i = 1, \dots, N \\ t = L+1, \dots, T \end{matrix} \quad (8'')$$

$$I_{it}^+ \text{ e } I_{it}^- \geq 0 \quad \begin{matrix} i = 1, \dots, N \\ t = 1, \dots, T \end{matrix} \quad (9)$$

A demanda é distribuída para os períodos de planeamento, de acordo com as datas de entrega, da seguinte forma: as demandas das peças com vencimento no dia da programação foram colocadas para a última carga do primeiro dia de programação, que equivale ao período $t = L = 10$; as demandas com datas de entrega marcadas para os quatro dias posteriores, foram colocadas nos períodos posteriores, de acordo com o horizonte de planeamento, ou seja, demandas para o dia posterior (segundo dia), recaem no período $t = L + 1 = 11$, para o terceiro dia no período $t = L + 2 = 12$, e assim por diante.

O modelo HR mantém semelhanças com os modelos de dimensionamento e seqüenciamento de lotes descritos em Drexl e Kimms (1997). As idéias presentes no modelo *General Lot-Sizing and Scheduling Problem* (Fleischmann e Meyr, 1997 e Drexl e Kimms, 1997) aparecem no modelo HR de maneira um pouco diferente. Observe que podem ser considerados cinco períodos grandes (5 dias), sendo que o primeiro período está subdividido em dez sub-períodos, enquanto que os outros quatro períodos, não estão subdivididos.

O modelo HR aplicado a um horizonte de planeamento de 5 dias com os dados reais ($N = 403$, $K = 26$ e $T = 14$) tem 260 variáveis binárias Y_t^k ($t = 1, \dots, L$), 104 variáveis inteiras Y_t^k ($t = L + 1, \dots, T$), 4.030 variáveis inteiras X_{it} ($t = 1, \dots, L$), 1.612 variáveis contínuas X_{it} ($t = L + 1, \dots, T$), 260 variáveis contínuas Z_t^k ($t = 1, \dots, L$) e 13.962 variáveis contínuas I_{it}^+ e I_{it}^- .

Cabe observar que a solução do modelo HR não produz uma solução para o modelo (1)-(9), pois apenas dez cargas do forno (um dia) são programadas separadamente e os outros dias são programados de forma aproximada. Entretanto, ao aplicar o modelo HR por cinco vezes, atualizando os dados a cada aplicação, chega-se a uma solução factível para o modelo (1)-(9), tornando possível a comparação dos resultados.

Apesar do modelo HR ser menor que o modelo (1)-(9), ainda não é pequeno o suficiente para ser resolvido, usando o Cplex 7.1, em tempo computacional razoável. A melhor solução encontrada depois de dez horas apresenta um *gap* de 5,19% (veja Tabela 2).

Entretanto, é possível melhorar o desempenho da técnica de horizonte rolante resolvendo o problema em duas fases, utilizando um método chamado **relaxe-e-fixe** (Wolsey, 1998). A primeira fase consiste em fixar as variáveis Y_t^k ($t = 1, \dots, L$), ou seja, fixar as ligas que deverão ser feitas nas $L = 10$ cargas do primeiro dia de programação, as quais são as decisões mais importantes no método de horizonte rolante. Para tanto, numa primeira fase, o modelo é resolvido considerando apenas as 260 variáveis Y_t^k ($t = 1, \dots, L$) como binárias e todas as outras variáveis com as condições de integralidade relaxadas. Portanto, o

Tabela 2. Resultados do Cplex aplicado ao modelo HR (apenas uma vez).

Tempo de Busca	Melhor Solução Atual (MS)	Melhor Limitante Inferior (MLI)	GAP (%) (MS-MLI)/MS	B&B Nós Pesquisados
5 minutos	622.004	512.482	17,61	250
1 horas	622.004	537.269	13,62	6.200
5 horas	622.004	540.933	13,03	25.500
6,5 horas	571.128	541.152	5,25	32.450
10 horas	571.128	541.497	5,19	44.900

problema fica apenas com 260 variáveis binárias e 19.968 variáveis contínuas e pode ser resolvido utilizando o pacote Cplex. Após fixar as variáveis binárias para o primeiro dia, na segunda fase do método a condição de integralidade sobre as 104 variáveis Y_t^k ($t = L + 1, \dots, T$) e as 4.030 variáveis X_{ii} ($t = 1, \dots, L$) são reconsideradas e o Cplex é utilizado novamente por alguns minutos para obter uma solução inteira factível.

Os resultados computacionais do método de horizonte rolante, aplicado em conjunto com o método relaxe-e-fixe (apresentados na Seção 5), foram muito melhores quando comparados com os resultados do método de horizonte rolante, aplicado com todas variáveis inteiras, como descrito anteriormente. Observe pela restrição (3) que, para os primeiros $L = 10$ períodos, ao fixar as variáveis Y_t^k , todas variáveis X_{ii} que não pertencem à liga fixada podem ser eliminadas, facilitando a resolução do modelo, ou seja, uma vez que a variável binária Y_t^k é fixada em 1, segue das restrições (3) que $X_{ii} = 0$, para todo $i \notin S(k)$.

4.2 Busca local

Como é possível observar na Seção 4.1, o método relaxe-e-fixe envolve a resolução de dois problemas de otimização inteira. O primeiro, e mais importante (pois decide as variáveis que serão realmente implementadas na solução do problema prático), consiste em fixar as variáveis inteiras para o primeiro dia. O segundo problema consiste em determinar valores inteiros para as 104 variáveis Y_t^k ($t = L + 1, \dots, T$) e para as 4030 variáveis X_{ii} ($t = 1, \dots, L$).

Voltando ao primeiro problema, um método de busca local será usado para fixar as variáveis binárias Y_t^k ($t = 1, \dots, L$). Os algoritmos de busca local (ou busca em vizinhança) (Reeves, 1993; Pirlot, 1996 e Aarts e Lenstra, 1997) caracterizam-se por partirem de uma solução factível inicial (que pode ser obtida por uma outra heurística) e alterá-la, iterativamente, para soluções vizinhas também factíveis, mas com melhor valor da função objetivo, até que um critério de parada seja satisfeito.

Considere um problema: $\min F(s)$ sujeito a $s \in S$, em que S é um conjunto de soluções factíveis. Neste ponto, é necessária a definição de vizinhança. Para cada $s \in S$, é possível associar um conjunto $N(s) \subseteq S$ chamado vizinhança de s , em que $N(s)$ é o conjunto de soluções, às quais é possível chegar a partir de s , com apenas uma operação elementar chamada movimento, que deve ser definida para cada problema específico. Um método de busca local começa com uma solução inicial $s_0 \in S$ e, a cada passo n , faz uma transição para uma nova solução s_{n+1} pertencente à vizinhança $N(s_n)$ da solução atual s_n . Se a solução vizinha s_{n+1} tem o

valor da função objetivo melhor que a solução s_n , ou seja, se $F(s_{n+1}) < F(s_n)$, então, s_{n+1} é adotada como a nova solução. Senão, novas soluções vizinhas podem ser testadas na tentativa de melhorar a solução atual. Se nenhuma solução melhor for encontrada, a busca é interrompida e a solução atual é considerada a melhor solução encontrada.

Explicações mais detalhadas de métodos de busca local e de outros métodos mais elaborados chamados meta-heurísticas podem ser encontradas em Goldberg (1989), Reeves (1993), Laguna (1995), Díaz et al. (1996), Gen e Cheng (1997) e Glover e Laguna (1997).

Voltando ao caso prático em estudo, para aplicar a busca local na fixação das 260 variáveis binárias Y_t^k ($t = 1, \dots, L$), o procedimento começa com uma solução inicial, que pode ser obtida aleatoriamente (o fato de considerar estoques negativos permite a obtenção de uma solução factível independentemente dos valores de Y_t^k ($t = 1, \dots, L$)). Com os valores das variáveis Y_t^k ($t = 1, \dots, L$) fixos, todas as outras variáveis são relaxadas e o problema linear restante é então resolvido. Na iteração seguinte, as variáveis Y_t^k ($t = 1, \dots, L$) são modificadas (representado um movimento da busca local) e o problema linear é resolvido novamente. A cada iteração, a melhor solução é mantida como solução atual. Quando o critério de parada é satisfeito, as variáveis Y_t^k ($t = 1, \dots, L$) estarão fixadas na melhor solução encontrada.

Devido às características do problema estudado (as quais serão comentadas com mais detalhes na Seção 5), não foi preciso muito esforço para se chegar a uma combinação de parâmetros que obtivesse boas soluções para este exemplo prático particular. A seguir são dadas algumas explicações sobre esses parâmetros.

A representação da solução consiste num vetor com $L = 10$ coordenadas, e cada coordenada representa a liga feita na correspondente carga do forno no primeiro dia de programação. Por exemplo, a solução representada pelo vetor (2, 2, 20, 1, 4, 4, 8, 2, 10, 3) indica que a liga 2 é produzida nas duas primeiras cargas do forno, a terceira carga é feita com a liga 20, e assim por diante. A solução inicial é construída aleatoriamente, ou seja, cada carga do forno é associada a uma liga de forma aleatória. O critério de parada consiste em 1.000 iterações da busca local.

Um movimento bastante simples foi considerado para mudar de uma solução atual para uma solução vizinha. Neste movimento uma posição do vetor é escolhida aleatoriamente e a liga que está sendo produzida é modificada para outra liga. Por exemplo, se a sexta posição do vetor é escolhida aleatoriamente

te para ser modificada e a liga 24 é escolhida, então, a nova solução será (2, 2, 20, 1, 4, 24, 8, 2, 10, 3). A nova liga pode ser escolhida aleatoriamente ou usando uma heurística gulosa, com maior probabilidade de escolha para as ligas que têm mais itens associados ($|S(k)|$). A seguir é apresentado um algoritmo para facilitar o entendimento do método:

Algoritmo 1 (Busca Local aplicada ao problema prático)

- Determine aleatoriamente uma solução inicial $\{ Y_t^k \text{ binárias, } t = 1, \dots, L \}$;
- Relaxe as condições de integralidade das variáveis X_{it} ($t = 1, \dots, T$) e Y_t^k ($t = L + 1, \dots, T$);
- Resolva o modelo linear restante e mantenha a solução como a solução atual;
- Faça IT iterações e, em cada iteração:
 - Gere uma solução vizinha: selecione aleatoriamente um período t ($1 \leq t \leq L$) e, então, escolha qual a nova liga k que deverá ser produzida neste período. Em outras palavras, fixe o novo valor das variáveis Y_t^k ; e
 - Resolva o modelo linear restante e mova para a solução vizinha, se esta for melhor que a solução atual (neste caso, a solução vizinha passa a ser a solução atual).
- Pare e imprima a solução atual como a melhor solução (para o problema relaxado) encontrada pela busca; e
- Fixe os valores das variáveis Y_t^k na melhor solução encontrada no passo 4. Restaure a integralidade das variáveis X_{it} ($t = 1, \dots, L$) e Y_t^k ($t = L + 1, \dots, T$) (Para tanto, utilize uma heurística de arredondamento ou um pacote de otimização inteira).

5. Resultados computacionais

Para comparar o desempenho dos métodos implementados, dados reais foram utilizados. O primeiro método, denotado

por CPLEX, consiste num método geral embutido no pacote Cplex e baseado na técnica *branch-and-bound*, ou seja, consiste na aplicação pura e simples do pacote Cplex ao modelo original (1)-(9), considerando um horizonte de planejamento de 5 dias. Todas as cargas do forno foram planejadas separadamente e numa única rodada, como mostra a Figura 3, em que as linhas pontilhadas denotam que as cargas do dia são programadas separadamente e são realmente implementadas. O segundo método, chamado CPLEX-HR, consiste em aplicar o pacote Cplex no modelo de horizonte rolante (modelo HR) usando o método relaxe-e-fixe descrito na Seção 4.1. Finalmente, o terceiro método, chamado BL-HR, aplica o método de busca local ao modelo HR (Seção 4.2).

Somente 5 dias de planejamento foram considerados para comparar os resultados dos métodos aplicados no modelo HR com o método CPLEX aplicado ao modelo (1)-(9), ou seja, o horizonte de planejamento dos métodos aplicados ao modelo HR, não foi rolado além de 5 dias, pois, caso contrário, não seria possível a comparação com o método aplicado ao modelo (1)-(9), no qual somente 5 dias são considerados. Desta forma, nos métodos CPLEX-HR e BL-HR aplicados ao modelo HR, na rodada 1 são considerados 5 dias de planejamento, mas somente as variáveis relativas ao primeiro dia são realmente implementadas. Os dados são, então, atualizados (isto é, os itens que foram produzidos no primeiro dia da rodada 1 são subtraídos na carteira de pedidos) e o método é rodado novamente (rodada 2) programando do dia 2 até o dia 5. No entanto, somente as variáveis relativas ao segundo dia de planejamento são realmente implementadas. Após cinco rodadas, os 5 dias são detalhadamente programados.

Deve-se observar que, na prática, os 5 dias de planejamento devem ser mantidos em todas as rodadas. Além disso, no dia-a-dia da programação, os dados relativos aos 5 dias de planejamento, considerados inicialmente, são comumente alterados (por exemplo, com novas demandas) antes que se

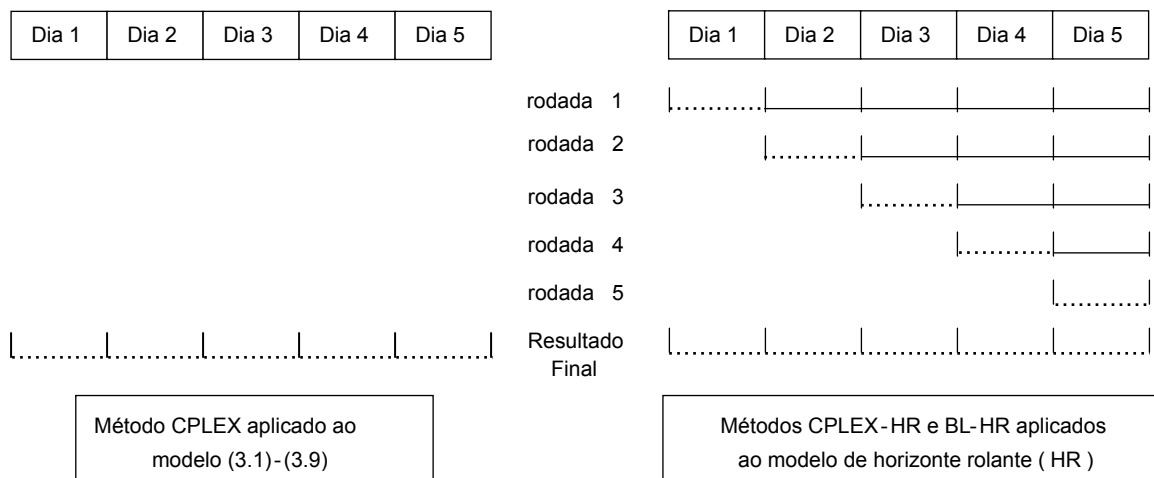


Figura 3. Desenho ilustrativo: como cada método foi aplicado para obter os resultados. (as linhas pontilhadas correspondem às programações detalhadas a serem implementadas).

conclua este horizonte de planejamento. A Figura 4 explica como o método de horizonte rolante é aplicado na prática.

Para obter os resultados computacionais, primeiramente o método CPLEX é rodado por 10 horas (36.000 segundos) para resolver o modelo (1)-(9) com 5 dias de planejamento, programando cada carga separadamente. Para comparar os outros dois métodos, inicialmente o método BL-HR é rodado 25 vezes para cada um dos 5 dias de planejamento e, em cada uma destas 25 rodadas, o método é reiniciado com uma nova solução aleatória e as iterações são feitas até que o critério de parada seja satisfeito. Desta forma, a cada rodada, os resultados podem ser diferentes devido à aleatoriedade da busca local (observe que os dados são os mesmos para as 25 rodadas). Uma média do tempo computacional (sobre essas 25 rodadas) é calculada e então o método CPLEX-HR é rodado por um tempo limitado à média calculada.

A Tabela 3 mostra o tempo computacional de cada rodada do método de horizonte rolante. Observe que o tempo dos métodos CPLEX-HR e BL-HR são iguais devido à limitação imposta ao método CPLEX-HR. Eventualmente, uma solução ótima poderia ter sido obtida em tempo inferior, mas isso não aconteceu. Como o número de variáveis diminui a cada rodada, o tempo computacional também decresce. Observe que os tempos computacionais totais, para os métodos baseados na estratégia de horizonte rolante (CPLEX-HR e BL-HR), são muito menores do que os do método CPLEX que não utiliza esta estratégia.

A Tabela 4 compara a qualidade da solução (valor médio da função objetivo sobre 25 rodadas no caso do método BL-HR) dos três métodos, considerando cada dia separadamente e também a solução total para os 5 dias.

Observe que, quando se aplica o método CPLEX para o

modelo completo com 5 dias, programando cada carga separadamente, mesmo rodando por 10 horas, a qualidade da solução é muito pior do que a qualidade das soluções obtidas pelos métodos que utilizam a estratégia de horizonte rolante (métodos CPLEX-HR e BL-HR). Comparando apenas os dois métodos que utilizam horizonte rolante é possível observar que o método de busca local (BL-HR) foi ligeiramente melhor que o método relaxe-e-fixe (CPLEX-HR).

Ao rodar o método CPLEX, limitantes inferiores para o valor da solução dos 5 dias são gerados. A Figura 5 mostra o melhor limitante inferior encontrado (depois de 20 horas rodando o método CPLEX) e o *gap* entre este limitante inferior e o valor das soluções encontradas por cada um dos métodos. Observe que, o *gap* entre o método BL-HR e o limitante inferior é de apenas 2,97%, ou seja, a solução encontrada pelo método BL-HR está relativamente próxima da solução ótima.

Várias estratégias foram testadas com o intuito de melhorar o desempenho da busca local tais como, diferentes tipos de movimentos mais elaborados, técnicas de redução de vizinhança (Clark, 2002) e estratégias meta-heurísticas, como, por exemplo, *simulated annealing*. No entanto, estes métodos não melhoraram os resultados. Provavelmente isso ocorreu pelo fato das penalidades por atraso serem muito maiores do que as penalidades por antecipação e por preparação e, no exemplo prático utilizado, por existir uma grande quantidade de peças atrasadas. Assim, a alta penalidade por atrasos faz com que não exista um *trade-off* entre preparação e atraso ou entre preparação e antecipação, ou seja, sempre será melhor minimizar os atrasos, mesmo que para isso sejam necessárias várias preparações, do que atrasar mais, para diminuir o número de preparações. Conseqüentemente, existem poucos ótimos locais e a busca local simples obtém bons resultados

Tabela 3. Tempo computacional (em segundos) de cada método.

Métodos	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	5 Dias
CPLEX	----	----	----	----	----	36000
CPLEX-HR	960	852	564	444	276	3096
BL-HR	960	852	564	444	276	3096

Tabela 4. Valor da função objetivo para cada método.

Métodos	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	5 Dias
CPLEX	248.498	168.665	113.539	68.504	32.235	631.441
CPLEX-HR	236.057	159.903	103.358	56.267	18.153	573.739
BL-HR	236.082	156.883	100.192	54.211	18.211	565.580

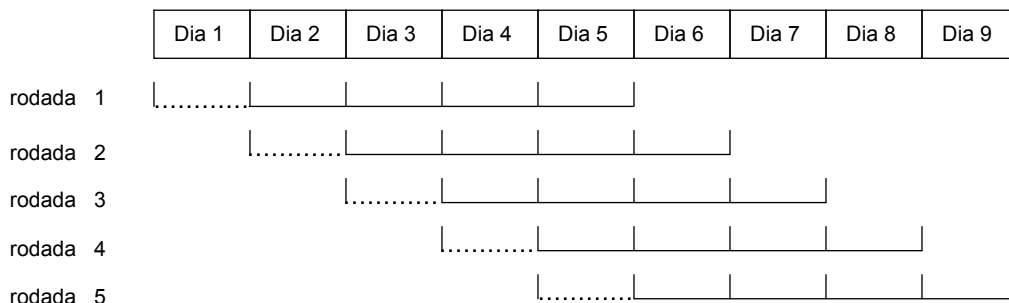


Figura 4. Desenho ilustrativo: como método de horizonte rolante é aplicado na prática. (as linhas pontilhadas correspondem às programações detalhadas a serem implementadas)

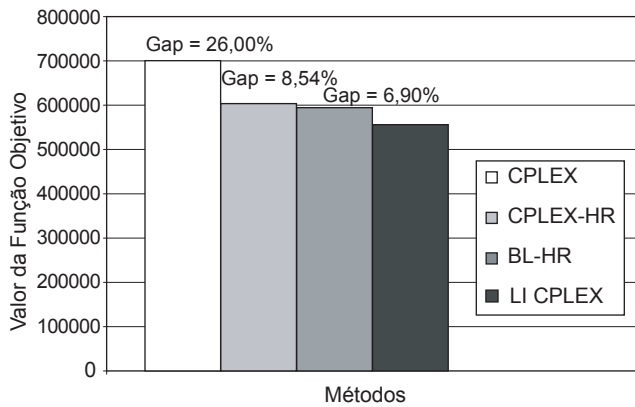


Figura 5. Resultados dos 3 métodos considerando cada dia separadamente.

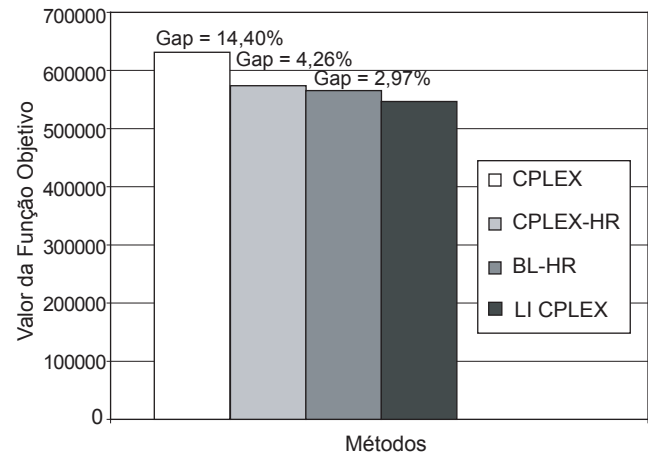


Figura 6. Gaps das soluções de cada método considerando os 5 dias de planejamento.

Tabela 5. Qualidade da solução para cada método considerando perda de capacidade por preparação.

Métodos	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	5 Dias
CPLEX	252.797	181.831	129.360	86.357	50.239	700.584
CPLEX-HR	240.789	165.440	109.255	64.197	23.835	603.516
BL-HR	240.994	161.827	105.857	61.585	24.099	594.363

sem a necessidade de estratégias meta-heurísticas. Entretanto, este é um exemplo particular de um caso prático. Este *trade-off* pode surgir quando os custos de preparação são incrementados (mas este não é o principal objetivo da fundição em estudo, na qual se deseja, principalmente, minimizar atrasos) ou quando o tempo de preparação é incluído nas restrições de capacidade (também não é o caso do problema prático em questão). Depois de visitar outras fundições, foi possível observar que, em alguns processos, o tempo de preparação deve ser considerado. Desta forma, para dar algumas direções para futuros trabalhos, o método foi rodado novamente substituindo as restrições (3) por:

$$\sum_{i \in S(k)} \rho_i X_{it} + st_k Z_t^k \leq Cap Y_t^k \quad k = 1, \dots, K$$

$$t = 1, \dots, T \quad (3')$$

em que: st_k representa a perda de capacidade devido à preparação da liga k .

O mesmo conjunto de exemplos foi rodado considerando esta perda de capacidade por preparação e os resultados computacionais são apresentados na Tabela 5 e na Figura 6.

Os resultados computacionais, com a inclusão de perda de capacidade por preparação, mostram que a qualidade das soluções obtidas pelos métodos que utilizam a estratégia de horizonte rolante (CPLEX-HR e BL-HR) continuaram melhores que os resultados obtidos pelo método CPLEX. Além disso, os resultados do método de busca local BL-HR também se mantiveram ligeiramente melhores do que os resultados do método relaxe-e-fixe (CPLEX-HR).

O *gap* entre o limitante inferior e o método BL-HR cresceu de 2,97% para 6,9%, sugerindo ser possível melhorar os resul-

tados da busca local com estratégias meta-heurísticas. Desta forma, um novo trabalho está sendo desenvolvido no qual um novo modelo é proposto considerando tempos e custos de preparação dependentes da seqüência Araujo et al. (2004).

6. Conclusões e trabalhos futuros

Neste trabalho tratamos de um processo produtivo que surge numa fundição de pequeno porte do interior paulista, que, embora seja um único caso, representa uma parcela significativa do setor. O problema foi modelado matematicamente como um problema de programação inteira mista, cujo principal objetivo foi a minimização dos atrasos no atendimento à demanda. Para resolver este modelo, foi proposto um método de busca local em que, a cada iteração, um conjunto de variáveis é fixado e o problema é resolvido com todas as outras variáveis relaxadas. No final da busca, um pacote de otimização inteira é utilizado para “arredondar” as variáveis relaxadas para valores inteiros. O planejamento da produção é feito usando a técnica de horizonte rolante, no qual somente a programação feita para períodos imediatos é realmente implementada, depois disso o horizonte é rolando para frente e o modelo é aplicado novamente com os dados atualizados.

Os resultados computacionais mostraram que a abordagem por horizonte rolante foi eficiente para um exemplo prático específico. Tentativas de resolver o modelo, utilizando o método *branch-and-bound* embutido no pacote Cplex, levaram um tempo computacional inviável para ser utilizado na prática. Além disso, a qualidade da solução encontrada pelo pacote foi pior do que as soluções encontradas pelas abordagens por horizonte rolante, em particular a heurística BL-HR. Isso mostra a importância do desenvolvimento de métodos de solução baseados em características específicas do problema.

7. Agradecimentos

Os autores agradecem o apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências Bibliográficas

- AATTS, E. H. L.; LENSTRA, J. K. **Local Search in Combinatorial Optimization**, Chichester: John Wiley and Sons, 1997.
- ABIFA - Associação Brasileira de Fundição **Relatório anual do setor de fundição**. Disponível em: <<http://www.abifa.org.br>>. Acesso em: 2002.
- ARAUJO, S. A.; ARENALES, M. N. Dimensionamento de Lotes e Programação do Forno numa Fundição de Porte Médio. **Pesquisa Operacional**, v. 23, n. 3, p. 403-420, 2003.
- ARAUJO, S. A.; ARENALES, M. N.; CLARK, A. R. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. **XII Congresso Latino-Americano de Investigación Operacional (CLAIO)**. Havana, 2004
- ARAUJO, S. A.; ARENALES, M. N. Planejamento e Programação da Produção numa Fundição Cativa Automatizada de Grande Porte. **Investigação Operacional**, v. 24, n. 2, 2004. 12 p.
- BAHL, H. C.; RITZMAN, L. P.; GUPTA, J. N. D. Determining Lot Sizes and Resource Requirements: A Review. **Operations Research**, v. 35, p. 329-345, 1987.
- BILLINGTON, P. J.; MCCLAIN, J. O.; THOMAS, L. J. Mathematical Programming Approaches to Capacity MRP Systems: Review. **Formulation and Problem Reduction**, Management Science, v. 29, n. 10, p. 1126-1141.
- BOWERS, M. R.; KAPLAN, L. A.; HOOKER, T. L. A Two-Phase Model for Planning the Production of Aluminum Ingot. **European Journal of Operational Research**, v. 81, p.105-114, 1995.
- CLARK, A. R.; CLARK, S. J. Rolling-Horizon Lot-Sizing When Setup Times are Sequence-Dependent. **International Journal of Production Research**, v. 38, n. 10, p. 2287-2307, 2000.
- CLARK A. R. **Hybrid Heuristics for Planning Lot Sizes and Setups**, Internal Research Report MS-2002-3, Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England. Disponível em: <<http://www.uwe.ac.uk/cems/ms/MS-2002-3-Clark.pdf>>. Acesso em: 2002.
- DÍAZ, A.; GLOVER, F.; GHAZIRI, H. M.; GONZÁLEZ, J. L., LAGUNA, M.; MOSCATO, P.; TSENG, F. T. Optimización Heurística y Redes Neuronales. **Editorial Paraninfo**, Espanha, 1996.
- DREXL, A.; KIMMS, A. Lot Sizing and Scheduling - Survey and Extensions. **European Journal of Operational Research**, v. 99, p. 221-235, 1997.
- FERNANDES, F. C. F.; LEITE, R. B. Automação Industrial e Sistemas Informatizados de Gestão da Produção em Fundições de Mercado. **Gestão e Produção** v. 9, n. 3, p. 313-344, 2002.
- FLEISCHMANN, B.; MEYER, H. The General Lotsizing and Scheduling Problem, **Operational Research Spektrum**, v. 19, p. 11-21, 1997.
- FOURER, R.; GAY, D. M.; KERNIGHAN B. W. AMPL- A Modeling Language for Mathematical Programming. **Boyd and Frase**, Danvers, Massachusetts. Disponível em: <<http://www.ampl.com>>. Acesso em: 1993.
- GEN, M.; CHENG, R. Genetic Algorithms & Engineering Design, **John Wiley & Sons**, 1997.
- GLOVER, F.; LAGUNA, M. Tabu Search, **Kluwer Academic Publishers**, Norwell, Massachusetts, 1997.
- GOLDBERG, D. E. Genetic Algorithms in Search, **Optimization and Machine Learning**, Addison Wesley, Massachusetts, 1989.
- GONÇALVES-VIANNA; A. C. ARENALES, M. Problema de Programação da Produção numa Fundição Automatizada. **Gestão & Produção**, v. 2, p. 244-263, 1995.
- GRAVEL, M.; PRICE, W. L.; GAGNÉ, C. Scheduling Jobs in an Alcan Aluminium Foundry Using a Genetic Algorithm. **International Journal of Production Research**, v. 38, n. 13, p. 3031-3041, 2000.
- GOYAL, S. K.; GUNASEKARAN, A. Multi-Stage Production-Inventory Systems, **European Journal of Operational Research**, v. 46, p. 1-20, 1990.
- HAMADA, K.; BABA, T.; SATO, K.; YUFU, M. Hybridizing a Genetic Algorithm with Rule-based Reasoning for production Planning. **IEEE Expert**, v. 10, p. 60-67, 1995.
- HENDRY, L. C.; FOK, K. K.; SHEK, K. W. A Cutting Stock and Scheduling Problem in the Copper Industry. **Journal of Operational Research Society**, v. 47, n. 1, p. 38-47, 1996.
- ILOG S.A. **ILOG CPLEX 7.1 User's Manual**. BP 85, 9 Rue de Verdun, 94253 Gentilly Cedex, France. Disponível em: <<http://www.ilog.com>>. Acesso em: 2001.
- KARIMIA B; FATEMI GHOMIA S. M. T.; WILSON J. M. The capacitated lot sizing problem: A review of models and algorithms. **Omega**, v. 31, p. 365-378, 2003.
- KUIK, R., SALOMOM, M.; VAN WASSENHOSE, L. N. Batching Decisions: Structure and Models. **European Journal of Operational Research**, v. 75, p. 243-263, 1994.
- LAGUNA, M. Tabu Search Tutorial, **II Escola de Verano Latino-Americana de Investigación Operativa**, Rio de Janeiro, 1995.
- LAGUNA, M. A Heuristic for Production Scheduling and Inventory Control in the Presence of Sequence-Dependent Setup Times, **IIE Transactions** ,v. 31, p. 125-134, 1999.
- LEE, H. S., MURTHY, S. S.; HAIDER, S. W.; MORSE, D. V. Primary Production Scheduling at Steelmaking Industries. **IBM Journal of Research and Development**,

- v. 40, n. 2, p. 231-252, 1996.
- LOPES, L.; CARTER, M. W.; GENDREAU, M. The Hot Strip Mill Production Scheduling Problem: a tabu Search Approach. **European Journal of Operational Research**, 106, p. 317-335, 1998.
- MAES J.; VAN WASSENHOVE, L. N. Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review, **Journal of Operational Research Society**, v. 39, n. 11, p. 991-1004, 1988.
- PETERSEN, C. M.; SORENSEN, K. L.; VIDAL, R. V. V. Inter-process Synchronization in the Steel Production. **International Journal of production Research**, v. 30, n. 6, p. 1415-1425, 1992.
- PIRLOT, M. General Local Search Methods, **European Journal of Operational Research**, v. 92, p. 493-511, 1996.
- POTTS, C. N.; VAN WASSENHOVE, L. N. Integrating Scheduling with Batching and Lot-Sizing: a Review of Algorithms and Complexity. **Journal of Operational Research Society**, v. 43, n. 5, p. 395-406, 1992.
- POTTS, C. N.; KOVALYOV, M. Y. Scheduling with Batching: a review. **European Journal of Operational Research**, v. 120, p. 228-249, 2000.
- REEVES, C.R. **Modern Heuristic Techniques for Combinatorial Problems**. Blackwell, 1993.
- SANTOS-MEZA, E.; SANTOS, M. O.; ARENALES, M. N. A Lot-Sizing Problem in an Automated Foundry. **European Journal of Operational Research**, v. 139, n. 2, p. 490-500, 2002.
- SILVA, R. J.; MORÁBITO, R. Otimização da Programação de Cargas de Forno em uma Fábrica de Fundição em Aço-Inox. **Gestão & Produção**, v. 11, n. 1, p. 135-151, 2004.
- SOUNDERPANDIAN, J.; BALASHANMUGAM, B. Multi-product, Multifacility scheduling Using the Transportation Model: a Case Study. **Production and Inventory Management Journal**, v. 32 n. 4, p. 69-73, 1991.
- TANG, L.; LIU, J.; RONG, A.; YANG, Z. A Mathematical Programming Model for Scheduling Steelmaking-Continuous Casting Production. **European Journal of Operational Research**, v. 120, p. 423-435, 2000a.
- _____. A Multiple Traveling Salesman Problem Model for Hot Rolling Scheduling in Shanghai Baoshan Iron & Steel Complex. **European Journal of Operational Research**, v. 124, p. 267-282, 2000b.
- _____. A Review of Planning and Scheduling Systems and Methods for Integrated Steel Production. **European Journal of Operational Research**, v. 133, p. 1v-20, 2001.
- WOLSEY, L. A. **Integer Programming**, Wiley, New York, 1998.

LOT-SIZING AND FURNACE SCHEDULING IN A SMALL FOUNDRY

Abstract

This article discusses a practical lot-sizing and scheduling problem in a foundry, involving two interrelated decision levels: furnace scheduling – in which a type of alloy is selected for production in each period, and molding machine planning – which specifies the quantity of each item to be manufactured in each period, i.e., lot-sizing. A mixed integer programming (MIP) model is proposed for this problem and a solution method is developed based on local search. The production is planned using the rolling horizon technique, whereby only immediate-period schedules are implemented. The results of the computational tests are analyzed and compared with those obtained using a commercial package.

Keywords: *mixed integer programming, lot-sizing and scheduling, foundries.*