



SISTEMA DE APOIO À DECISÃO PARA PROGRAMAÇÃO DA PRODUÇÃO EM FUNDIÇÕES DE MERCADO

Rodolfo Florence Teixeira Junior

Grupo de pesquisa Engenharia de Produção, Departamento de Informática,
Centro de Tecnologia, Universidade Estadual de Maringá,
Av. Colombo, 5790, CEP 87020-900 Maringá, PR,
e-mail: florenceteixeira@yahoo.com.br

**Flavio Cesar Faria Fernandes
Neocles Alves Pereira**

Grupo de pesquisa PLACOP, Departamento de Engenharia de Produção,
Programa de Pós-Graduação, Universidade Federal de São Carlos,
Rod. Washington Luís, Km 235, CEP 13565-905, São Carlos, SP,
e-mail: dfcf@power.ufscar.br, neocles@power.ufscar.br

Recebido em 10/2/2005
Aceito em 17/5/2006

Resumo

Este trabalho tem por objetivo propor um sistema de apoio à decisão (SAD) para elaboração da programação da produção em fundições que trabalham com produtos personalizados sob encomenda, também conhecidas como fundições de mercado. Para isto, o ambiente produtivo das fundições de mercado é estruturado por meio do sistema de controle da produção conhecido como PBC, sendo propostas, também, três modelagens decisórias viáveis para o processo de programar a produção: um modelo de programação inteira binária, um modelo heurístico clássico de busca em árvore, conhecido como beam search e um modelo meta-heurístico baseado em algoritmos genéticos. Este trabalho apresenta ainda os resultados experimentais obtidos com a aplicação dos modelos decisórios e a análise da adequação do SAD proposto em fundições de mercado representativas do interior do Estado de São Paulo.

Palavras-chave: sistema de apoio à decisão, programação da produção, fundições de mercado.

1. Introdução

Ao contrário das “fundições cativas”, as “fundições de mercado” apresentam número de clientes e variedade de produtos finais demasiadamente alta; possuem um ambiente produtivo por processo com características muito peculiares que influenciam e tornam complexa a programação da produção. Como boa parte dos produtos fabricados são inéditos, as mensurações dos tempos envolvidos com as operações produtivas envolvem incertezas e variabilidades difíceis de serem estimadas.

Apesar da diversidade de combinações, no geral, as fundições de mercado possuem as mesmas operações básicas de fabricação. Um esquema produtivo típico de uma empresa deste setor pode ser visualizado na Figura 1.

No geral, as operações produtivas das fundições de mercado podem ser agrupadas nas mesmas etapas básicas de fabricação, quais sejam:

- Moldagem: produzir moldes a partir de modelos fornecidos pelos clientes;

- Vazamento: Fundir ligas de metais e introduzi-las nos moldes; e
- Acabamento: Retirar o sólido resultante dos moldes e separar a peça resultante dos anexos do processo de fundição, trabalhar as rebarbas e defeitos de fundição, além de realizar uma série de tratamentos físicos e controle de qualidade, e também, em alguns casos, sofrer usinagem e/ou polimentos.

A complexidade do ambiente produtivo das fundições de mercado acarreta vários problemas e dificuldades para gerenciar sua produção como, por exemplo, dificuldade para fixar e cumprir prazos de entrega, excesso de movimentação dos itens em processo, causando ineficiências produtivas e problemas de qualidade do molde, altos níveis de estoques em processo que causam um impacto negativo sobre o tempo de resposta e a qualidade do produto fundido. Uma análise *in loco* do processo de

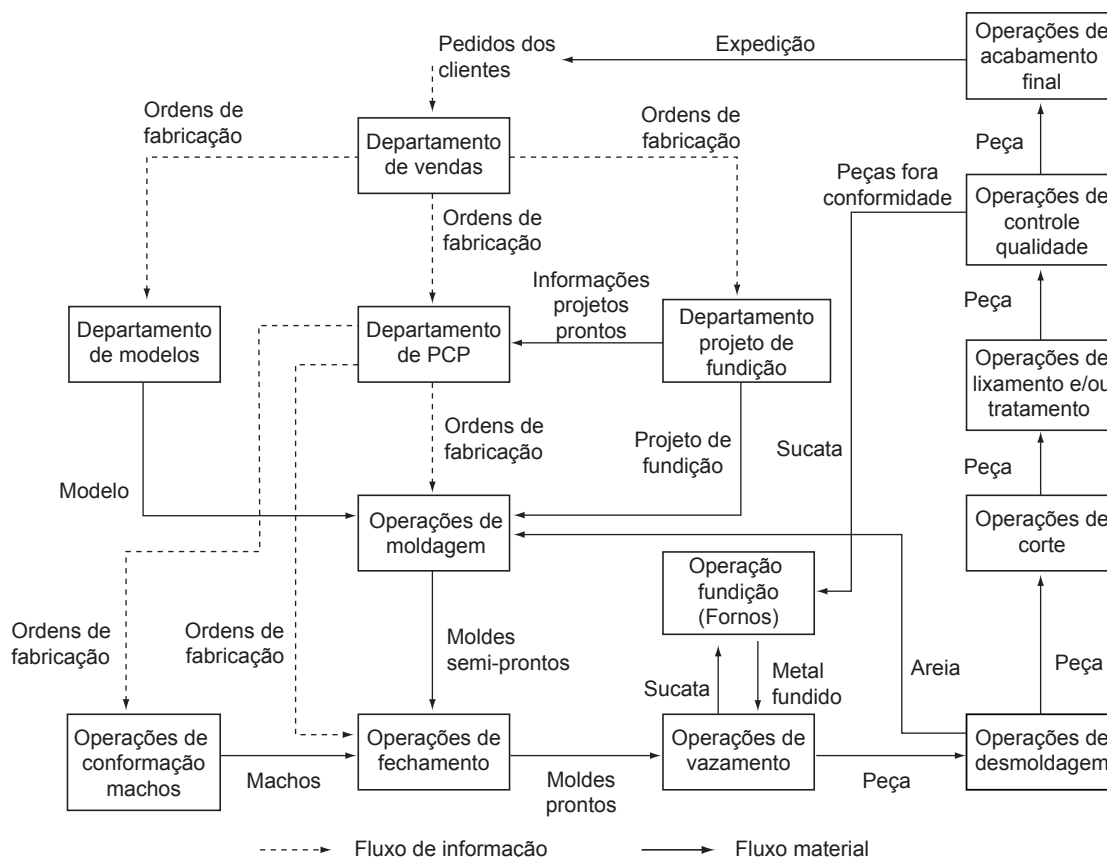


Figura 1. Visão geral do processo de produção em uma fundição de mercado.

produção (seção 9) deste setor industrial mostra que grande parte dos problemas citados é decorrente dos procedimentos atualmente empregados para elaborar a sua programação de produção e que as melhorias almejadas podem ser perfeitamente contempladas pelo sistema desenvolvido neste trabalho. Assim, este trabalho tem por objetivo apresentar uma solução viável para o processo de elaboração da programação da produção em fundições de mercado. Para isto, propõe-se que o controle da produção seja estruturado segundo a lógica do sistema *PBC* e que um SAD coordene as decisões que serão apoiadas por uma base de modelos consistente.

2. Programação da produção em fundições de mercado

O processo de programar a produção em fundições de mercado é pouco tratado na literatura da área. Dentre os trabalhos disponíveis, relativos apenas ao processo produtivo das fundições, pode-se citar Shivappa e Babu (1997). Especificamente a programação da produção em fundições cativas (baixa variedade e alto volume: geralmente as peças são próprias ou de pouquíssimos clientes e raramente são inéditas) foi tratada por Vianna e Arenales (1995), Araujo e Arenales (2003) e Pacciarelli e Pranzo (2004).

Quanto aos trabalhos dirigidos especificamente a fundições de mercado, pode-se citar o *survey* realizado por Fernandes e Leite (2002) sobre as fundições de mercado do interior do Estado de São Paulo, apontando os principais problemas do setor e as prioridades de investimentos em pesquisas. Dentre os resultados obtidos pelos autores, 30% das fundições de mercado avaliadas apontaram que é prioritário melhorar o planejamento e controle da produção. Especificamente sobre a programação da produção em fundições de mercado, Silva (2001) propôs um modelo heurístico para programação das corridas de fornos, entretanto não realiza a consideração das demais operações produtivas. Já Voorhis et al. (2001) propuseram uma formulação com base em um modelo de programação inteira, considerando a maior parte das restrições existentes na programação da produção em fundições de mercado, mas sem garantia de interação com o usuário.

3. Visão geral do esquema de programação da produção proposto

O processo de elaboração da programação da produção proposto tomará por base o sistema de controle da produção conhecido como sistema *PBC* (Burbidge, 1996). Tomando por base o processo produtivo típico de uma fundição de mercado, suas operações produtivas

poderão ser convenientemente divididas nos seguintes estágios:

- Estágio de acúmulo de pedidos: Neste estágio é realizado um acúmulo das ordens de produção de acordo com os pedidos dos clientes e também outras necessidades produtivas provenientes do planejamento da produção da empresa;
- Estágio de moldagem: Neste estágio todas as operações necessárias para concluir os moldes dos itens de pedidos programados para vazamento são realizadas, envolvendo o projeto do fundido, moldagem, confecção dos machos, fechamento do molde, entre outras;
- Estágio de vazamento: Neste estágio, de posse dos moldes prontos para vazar, é realizada a fundição das ligas metálicas nos fornos para que sejam, então, inseridas nos moldes; e
- Estágios de acabamento: Por fim, nestes estágios, realizam-se todas as operações necessárias para terminar os itens de pedidos, envolvendo desde a desmoldagem, corte, rebarba, lixamentos, tratamentos térmicos e físicos, controle de qualidade e acabamentos finais.

Estes estágios farão parte de cada ciclo do *PBC* (Figura 2).

A programação da produção terá por objetivo decidir quais itens de pedido disponíveis na carteira de pedidos a programar, provenientes do estágio de acúmulo de pedidos, do ciclo atual e também de ciclos anteriores, que ainda não tenham sido programados, serão programados para o ciclo. Para isto, todas as restrições produtivas deverão ser satisfeitas, pois todos os itens programados deverão ser completados em seu próprio ciclo de programação.

Para que o modelo possa considerar os principais fatores que influenciam a decisão de programar ou não um item de pedido, serão considerados como prêmios ou penalidades mensurados ou estimados sob a forma financeira para cada item de pedido que pertence à carteira de

pedidos em aberto, além de prêmios e penalidades decorrentes do impacto do programa de produção gerado no processo de produção. Estes dados poderão ser, por fim, considerados concomitantemente com a lucratividade estimada para cada item de pedido, permitindo uma análise do retorno financeiro esperado em certa programação da produção para um ciclo. O SAD proposto possibilita que os valores estimados para estes prêmios ou penalidades subjetivos possam ser avaliados, quanto ao impacto na programação gerada, de forma interativa, por meio do ciclo decisório do sistema. Visando facilitar a forma na qual o tomador de decisões irá lidar com todos os prêmios e penalidades envolvidos com um item a programar e também visando maior flexibilidade de aplicação dos modelos propostos, este trabalho propõe que estes fatores sejam todos condensados em uma única variável que apresentará o “valor” de se programar um item de pedido.

A programação da produção deve considerar também as restrições produtivas de cada estágio do ciclo de produção, assim como as restrições de coordenação entre os estágios e entre os ciclos, relativas ao seqüenciamento e economia (quando são fundidas em seqüência ligas metálicas que agilizam o processo de fundição) das ligas a serem fundidas, como também as restrições de capacidade dos fornos, com o objetivo de maximizar a somatória dos valores dos itens de pedido desta programação da produção.

4. Formulação do modelo de programação matemática

Em Goldberg e Luna (2000), Sherali e Driscoll (2000) e Toth (2000), pode-se ter uma visão geral dos avanços e dificuldades existentes para a solução de modelos de programação matemática. A seguir, o modelo proposto (este modelo será tratado como *PLI*):

- $i = 1..I$ é o índice do conjunto S_i de itens de pedidos em aberto;
- $l = 1..L$ é o índice do conjunto S_L de ligas de metais que podem ser usadas para fundir os itens de pedidos;
- $r = 1..R$ é o índice do conjunto S_r de reatores;
- $f = 1..F$ é o índice do conjunto S_f de fornos;
- $c = 1..C$ é o índice do conjunto S_c de corridas de cada forno;
- $om = 1..OM$ é o índice do conjunto S_{OM} de operações do processo produtivo do estágio de moldagem;
- $ea = 1..EA$ é o índice do conjunto S_{EA} de estágios de acabamento; e
- $oa = 1..OA$ é o índice do conjunto S_{OA} de operações do processo produtivo de cada estágio de acabamento.

Considere ainda os índices $m = 1..L$ e $n = 1..L$ como analogias, para efeito de modelagem, ao índice l ;

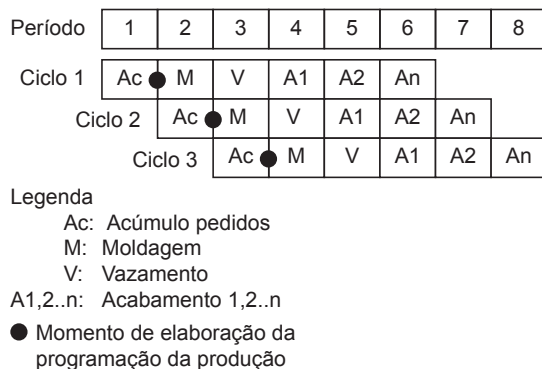


Figura 2. Sistema de controle da produção *PBC* aplicado às fundições de mercado.

A solução do modelo envolve decidir os valores das seguintes variáveis de decisão binárias:

$$a_{i,c,f,r} = \begin{cases} 1 & \text{se o item } i \text{ for vazado na corrida } c \\ & \text{do forno } f \text{ do reator } r \\ 0 & \text{caso contrário.} \end{cases};$$

$$b_{l,c,f,r} = \begin{cases} 1 & \text{se a liga } l \text{ for fundida na corrida } c \\ & \text{do forno } f \text{ do reator } r \\ 0 & \text{caso contrário.} \end{cases}; e$$

$$e_{l,c,f,r} = \begin{cases} 1 & \text{se for econômico fundir a liga } l \text{ na} \\ & \text{corrida } c \text{ do forno } f \text{ do reator } r \\ 0 & \text{caso contrário.} \end{cases}.$$

Os dados do modelo são definidos em letras gregas como segue: ω_r é o número máximo de corridas possível para o reator de índice r no horizonte de tempo considerado; π_i é o peso do item de pedido de índice i ; ρ_i é o valor do item de pedido de índice i ; $\lambda_{i,l}$ será 1, se o item i usar a liga l e 0, caso contrário; $\varphi_{f,r}$ é a capacidade (em peso) do forno de índice f do reator de índice r ; $\kappa_{f,r}$ é o custo de uma corrida do forno de índice f do reator de índice r ; $\xi_{f,r}$ é o valor da economia conseguida, caso o forno de índice f do reator de índice r realize duas corridas em seguida de ligas que agilizam o processo de fundição (ligas econômicas). $\eta_{m,n}$ será 1 se, em um forno, for possível fundir a liga m após ter fundido a liga n , já $\gamma_{m,n}$ será 1, caso, em um forno, for econômico fundir a liga m após ter fundido a liga n ; $\varepsilon_{l,f,r}$ será 1, caso a liga l tenha sido a última liga usada pelo forno de índice f do reator de índice r no ciclo anterior; Ψ_{om} é a capacidade (em minutos) máxima da operação de índice om do estágio de moldagem; $\tau_{ea,oa}$ é a capacidade máxima da operação de índice oa do estágio de acabamento de índice ea ; δ_{om} é a capacidade cambiável máxima da operação de índice om do estágio de moldagem; $\mu_{ea,oa}$ é a capacidade cambiável máxima da operação de índice oa do estágio de acabamento de índice ea ; $\alpha_{i,om}$ é o consumo (em minutos) total pelo item de índice i da operação de índice om do estágio de moldagem; $\beta_{i,ea,oa}$ é o consumo total pelo item de índice i da operação de índice oa do estágio de acabamento de índice ea ;

O modelo *PLI* possui o seguinte objetivo:

$$\max \left(\sum_{i=1}^I \sum_{f=1}^F \sum_{r=1}^R (a_{i,c,f,r} \cdot \rho_i) - \sum_{l=1}^L \sum_{c=1}^C \sum_{f=1}^F \sum_{r=1}^R (b_{l,c,f,r} \cdot \kappa_{f,r} - e_{l,c,f,r} \cdot \xi_{f,r}) \right) \quad (1)$$

Sujeito às restrições:

$$\sum_{f=1}^F \sum_{c=1}^C \sum_{l=1}^L b_{l,c,f,r} \leq \omega_r \quad \forall r \quad (2)$$

$$\sum_{i=1}^I (a_{i,c,f,r} \cdot \pi_i) \leq \varphi_{f,r} \quad \forall c, f, r \quad (3)$$

$$\sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C a_{i,c,f,r} \leq 1 \quad \forall i \quad (4)$$

$$a_{i,c,f,r} \cdot \lambda_{i,l} \leq b_{l,c,f,r} \quad \forall i, l, c, f, r \quad (5)$$

$$\text{se } c > 1 \text{ então } \sum_{l=1}^L b_{l,c,f,r} \leq \sum_{l=1}^L b_{l,c-1,f,r} \quad \forall c, f, r \quad (6)$$

$$\text{se } (c > 1 \text{ e } \eta_{m,n} = 0) \text{ então } b_{m,c-1,f,r} + b_{n,c,f,r} \leq 1 \quad \forall m, n, c, f, r \quad (7)$$

$$\text{se } (c = 1 \text{ e } \eta_{m,n} = 0) \text{ então } \varepsilon_{m,f,r} + b_{n,c,f,r} \leq 1 \quad \forall m, n, c, f, r \quad (8)$$

$$\sum_{l=1}^L b_{l,c,f,r} \leq 1 \quad \forall c, f, r \quad (9)$$

$$\sum_{i=1}^I \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (a_{i,c,f,r} \cdot \alpha_{i,om}) \leq \Psi_{om} + \delta_{om} \quad \forall om \quad (10)$$

$$\sum_{i=1}^I \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (a_{i,c,f,r} \cdot \beta_{i,ea,oa}) \leq \tau_{ea,oa} + \mu_{ea,oa} \quad \forall ea, oa \quad (11)$$

$$\sum_{om=1}^{OM} \sum_{i=1}^I \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (\text{se } \delta_{om} > 0 \text{ então } a_{i,c,f,r} \cdot \alpha_{i,om}) \leq \sum_{om=1}^{OM} (\text{se } \delta_{om} > 0 \text{ então } \Psi_{om}) \quad (12)$$

$$\sum_{oa=1}^{OA} \sum_{i=1}^I \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (\text{se } \mu_{ea,oa} > 0 \text{ então } a_{i,c,f,r} \cdot \beta_{i,ea,oa}) \leq \sum_{oa=1}^{OA} (\text{se } \mu_{ea,oa} > 0 \text{ então } \tau_{ea,oa}) \quad \forall ea \quad (13)$$

$$\text{se } c = 1 \text{ então } e_{l,c,f,r} = 0 \quad \forall l, c, f, r \quad (14)$$

$$e_{l,c,f,r} \leq b_{l,c,f,r} \quad \forall l, c, f, r \quad (15)$$

$$\text{se } (c > 1 \text{ e } \gamma_{m,n} = 0) \text{ então } b_{m,c-1,f,r} + e_{n,c,f,r} \leq 1 \quad \forall m, n, c, f, r \quad (16)$$

A formulação do modelo proposto visa maximizar a função objetivo (1), em que sua primeira parcela consiste na soma dos valores de todos os itens de pedido programados e sua segunda parcela representa o custo total de todas as corridas de fornos necessárias para compor o programa. Quanto às restrições do modelo, a restrição (2) cuida para que o número de corridas de fornos total em um ciclo não seja maior que a capacidade dos reatores destes fornos, enquanto a integridade da capacidade máxima do forno utilizado em cada corrida é garantida pela restrição (3). A restrição (4) cuida para que um item de pedido não seja programado em mais de uma corrida de forno, enquanto a restrição (5) garante que um item de pedido só seja programado em uma corrida de forno que irá fundir a liga metálica na qual é composto. Cabe observar

também que a restrição (5) garante que um item de pedido não seja incluído em uma corrida que não irá acontecer.

As restrições (6), (7) e (8) são responsáveis por ordenar as corridas nos fornos. A restrição (6) cuida para que as corridas aconteçam em ordem. As restrições (7) e (8) garantem que uma seqüência de ligas inviáveis não ocorra, a restrição (7) garante as corridas que acontecem durante o ciclo de produção e a restrição (8) garante que a primeira corrida de um forno, em um ciclo, seja viável com a última corrida deste forno no ciclo de produção anterior. Já a restrição (9) cuida para que em uma corrida de forno só seja fundido um único tipo de liga metálica. Quanto às operações produtivas do estágio de moldagem e dos estágios de acabamento, as Equações 10 e 11 garantem que a soma da capacidade máxima e do adicional de capacidade que uma operação poderá contar, cambiando com a capacidade de uma outra operação para as operações para cada estágio de produção, tanto para o estágio de moldagem como para os estágios de acabamento, respectivamente, sejam respeitadas pelos itens de pedidos programados. Por fim, as restrições (12) e (13) tratam especificamente da possibilidade de se cambiar as capacidades das operações, garantindo que, para as operações que possibilitam cambiar sua capacidade, a capacidade total destas operações no estágio de moldagem e nos estágios de acabamento, respectivamente, sejam respeitadas. A restrição (14) tem por objetivo garantir que as primeiras corridas de cada forno em um ciclo não sejam consideradas econômicas. A restrição (15) garante que uma corrida só poderá ser considerada econômica caso aconteça. Por fim, a restrição (16) garante que uma corrida que foi precedida por uma liga que não agilize o processo de fundição não seja considerada econômica.

Visando obter maior viabilidade de solução da modelagem proposta, pretende-se abrir mão de uma possível solução exata, sem que se perca a noção de proximidade da solução obtida com esta possível solução exata. Para isto, serão propostos dois modelos, com base no modelo *PLI*, que forneçam, respectivamente, um limite superior e inferior para a solução exata. O primeiro modelo é uma relaxação do modelo *PLI* e toma por base, para isto, a equação de restrição (3) realizando a relaxação desta equação como segue.

$$\sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C \sum_{i=1}^I (a_{i,c,f,r} \cdot \pi_i \cdot \lambda_{i,l}) \leq \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (b_{i,c,f,r} \cdot \varphi_{f,r}) \quad \forall l \quad (17)$$

A equação de restrição (17) garante que o peso total dos itens de pedido fundidos de certa liga seja menor ou igual à somatória das capacidades de todas as corridas de fornos que irão fundir a referida liga. Porém, não garante que, em certa corrida de forno, este terá capacidade para

vazar todos os itens de pedido programados para a referida corrida. Apesar da substituição da equação de restrição (3) pela Equação 17 não garantir uma solução factível, é trivial que esta substituição causa uma relaxação da solução do modelo *PLI*, uma vez que toda a solução para *PLI*, considerando a restrição (3), será uma solução válida para *PLI* considerando a restrição alternativa (17), apesar do contrário não ser verdadeiro.

Desta forma, considere a'_i como uma variável de decisão binária em que:

$$a'_i \begin{cases} 1 & \text{se o item de pedido } i \text{ for vazado na} \\ & \text{programação do ciclo} \\ 0 & \text{caso contrário} \end{cases}$$

Considerando a nova variável de decisão, a Equação 17 poderá ser reescrita, sem perda de funcionalidade, da seguinte forma:

$$\sum_{i=1}^I (a'_i \cdot \pi_i \cdot \lambda_{i,l}) \leq \sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (b_{i,c,f,r} \cdot \varphi_{f,r}) \quad \forall l \quad (18)$$

A substituição da variável $a_{i,c,f,r}$ por a'_i permite simplificar a Equação 17. A análise da substituição da variável $a_{i,c,f,r}$ por a'_i e da restrição (3) pela Equação 18 no modelo *PLI* permite avaliar que não altera sua estrutura principal de forma a garantir que a solução exata com esta alteração continue fornecendo um limite superior para *PLI*, ou seja, que estas substituições continuem representando uma relaxação de *PLI*. Com a substituição da variável $a_{i,c,f,r}$ por a'_i :

a função objetivo (14) se transforma em:

$$\max \left(\sum_{i=1}^I (a'_i \cdot \rho_i) - \sum_{l=1}^L \sum_{c=1}^C \sum_{f=1}^F \sum_{r=1}^R (b_{i,c,f,r} \cdot \kappa_{f,r} - e_{i,c,f,r} \cdot \xi_{f,r}) \right) \quad (19)$$

Quanto às equações de restrição que serão afetadas, pode-se perceber que a equação de restrição (4) deixará de ser necessária, pois a substituição da nova variável a reduziria a $a'_i \leq 1 \quad \forall_i$. Por sua vez, a equação de restrição (5) também não será mais necessária, pois cuida para que os itens de pedido sejam programados em uma corrida na qual seja fundido a liga metálica desejada para este item. Entretanto, a equação de restrição (17) já desobrigava este relacionamento entre os itens de pedidos e as corridas de forno particulares. Conseqüentemente, a Equação 18 também atende a esta funcionalidade de forma que a equação de restrição (5) se torna desnecessária. Por fim, as equações de restrições (10), (11), (12) e (13) não são influenciadas pela falta de informação da corrida, na qual os itens de pedido serão vazados. Desta forma, estas equações poderão ser substituídas, sem perda de generalidade, pelas seguintes equações de restrição, respectivamente:

$$\sum_{i=1}^I (a'_i \cdot \alpha_{i,om}) \leq \psi_{om} + \delta_{om} \quad \forall om \quad (20)$$

$$\sum_{i=1}^I (a'_i \cdot \beta_{i,ea,oa}) \leq \tau_{ea,oa} + \mu_{ea,oa} \quad \forall ea, oa \quad (21)$$

$$\sum_{om=1}^{OM} \sum_{i=1}^I (se \delta_{om} > 0 \text{ então } a'_i \cdot \alpha_{i,om}) \leq \sum_{om=1}^{OM} (se \delta_{om} > 0 \text{ então } \psi_{om}) \quad (22)$$

$$\sum_{oa=1}^{OA} \sum_{i=1}^I (se \mu_{ea,oa} > 0 \text{ então } a'_i \cdot \beta_{i,ea,oa}) \leq \sum_{oa=1}^{OA} (se \mu_{ea,oa} > 0 \text{ então } \tau_{ea,oa}) \quad \forall ea \quad (23)$$

A solução fornecida pelo modelo *PLIRX* pode ser usada como dado de entrada em um segundo modelo que forneça uma solução factível. Este modelo tem a mesma estrutura que o modelo *PLI*, com as restrições fornecidas pela solução do modelo *PLIRX*, ou seja, é um modelo restrito de *PLI*, sendo convenientemente tratado por *PLIRT*. O modelo *PLIRT* considera além dos dados do modelo *PLI* os seguintes dados: $h_{l,c,f,r}$ será 1, caso a liga l seja fundida na corrida c do forno f do reator r , e 0, caso contrário. Já $k_{l,c,f,r}$ será 1, caso seja econômico fundir a liga l na corrida c do forno f do reator r , e 0, caso contrário. Estes dados serão iniciados com os valores fornecidos pela solução de *PLIRX*.

Por fim, o modelo *PLIRT* utiliza todas as equações de restrição do modelo *PLI* e também as novas restrições:

$$se (h_{l,c,f,r} = 1) \text{ então } b_{l,c,f,r} = 1 \quad \forall l, c, f, r \quad (24)$$

$$se (k_{l,c,f,r} = 1) \text{ então } e_{l,c,f,r} = 1 \quad \forall l, c, f, r \quad (25)$$

As duas novas equações de restrição (24) e (25) têm por objetivo garantir que toda a corrida de forno que for definida para acontecer por meio do processo de solução do modelo *PLIRX*, também terá de, obrigatoriamente, acontecer na solução do modelo *PLIRT*. Desta forma, pode-se afirmar que *PLIRT* é um modelo restrito de *PLI* e que toda a solução fornecida por *PLIRT* também será solução de *PLI*, representando um limite inferior à solução exata de *PLI*.

Apesar de o modelo *PLIRX* ser potencialmente viável de solução, será proposta uma nova abordagem ao modo de compor as corridas de fornos. Nesta nova abordagem, a partir do peso total de cada tipo de liga metálica a ser fundida de acordo com a carteira de pedidos em aberto, buscam-se quais corridas de forno melhor possam realizar esta fundição. Este modelo será tratado por *PLIA*. Para a formulação do modelo *PLIA*, é necessário introduzir os seguintes dados: θ_l é o peso total da liga de índice l ; e σ_l é o valor total da liga de índice l .

A solução do modelo relaxado *PLIA* possui o seguinte objetivo:

$$\max \left(\sum_{l=1}^L \sum_{c=1}^C \sum_{f=1}^F \sum_{r=1}^R \left(b_{l,c,f,r} \cdot \left(\frac{\sigma_l}{\theta_l} \cdot \varphi_{f,r} - \kappa_{f,r} \right) + e_{l,c,f,r} \cdot \xi_{f,r} \right) \right) \quad (26)$$

O modelo *PLIA* estará sujeito às restrições (2), (6), (7), (8), (9), (14), (15) e (16). E também à nova restrição:

$$\sum_{r=1}^R \sum_{f=1}^F \sum_{c=1}^C (b_{l,c,f,r} \cdot \varphi_{f,r}) \leq \theta_l \quad \forall l \quad (27)$$

A formulação do modelo *PLIA* visa maximizar o retorno das corridas de fornos de um ciclo de produção. Para isto, a função objetivo (26) maximiza a relação entre o valor total de cada liga metálica e seu peso total multiplicado pela capacidade dos fornos, nos quais a referida liga metálica será fundida. A nova restrição (27) definida no modelo *PLIA* cuida para que as corridas de fornos se limitem ao peso total disponível de cada liga metálica, não permitindo que uma corrida de forno aconteça caso não exista liga suficiente para completar o forno.

5. Formulação do modelo heurístico

O modelo aproximativo clássico proposto toma por base o método heurístico de busca conhecido como *beam search*. Este método heurístico realiza um processo de busca em árvore e foi proposto originariamente na área de inteligência artificial, procurando avaliar certo número de soluções em paralelo. O número de soluções examinadas será o diâmetro ou largura da busca. Este método foi inicialmente empregado em sistemas de reconhecimento de voz e reconhecimento de imagem, tendo sido também comumente aplicado em problemas na área de programação de operações em ambientes *flow-shop*, *job-shop*, *open-shop*, em sistemas de manufatura flexível, entre outros, proporcionando resultados satisfatórios (veja, por exemplo: Colin (1998), Sabuncuoglu e Karabuk (1998), Ghirardi e Potts (2003), McMullen e Tarasewich (2003), Croce et al. (2004) e Pacciarelli e Pranzo (2004)).

Para a aplicação deste método heurístico, primeiramente é necessário realizar uma estruturação conveniente do problema a ser resolvido segundo uma árvore decisória multiestágio. Para isto, é proposta uma árvore decisória de tal forma que cada vértice $v_y \in V$, em que $V = \{v_1 \dots v_n\}$ é conjunto de vértices da árvore, representa uma solução parcial para o problema. Neste modelo, cada arco $a_x \in A$ representa uma decisão isolada, em que $A = \{a_1 \dots a_m\}$ é o conjunto de arcos ou conexões entre as soluções parciais do problema. Seja ainda, para cada arco $a_x \in A$ o seu vértice sucessor e antecessor representado, respectivamente, por $suc_{a_x} = \{v_y \setminus v_y \in V\}$ e $ant_{a_x} = \{v_y \setminus v_y \in V\}$.

Seja ainda, $SA_{v_y} \subset A$ o conjunto composto pelo arco $a_x \in A$ de entrada de $v_y \in V$ e também pelo arco de entrada de seu vértice vizinho antecessor e assim recursivamente, até o vértice de origem, constituindo um caminho único da origem da árvore ao vértice $v_y \in V$. Considere também $SU_{v_y} \subset A$ o conjunto de todos os arcos decisórios $a_x \in A$ sucessores ao vértice $v_y \in V$. Considere ρa_{a_x} o valor de cada arco decisório $a_x \in A$, ou seja, o valor relacionado com a decisão. Seja $\rho v_{v_y} = \sum_{SA_{v_y}} \rho a_{a_x}$ o valor da

solução parcial do problema em cada vértice $v_y \in V$, de forma que, $\rho v_{v_y} = 0$ para o vértice de origem. Considere ainda D_e como o estágio decisório.

Para que o modelo proposto possa ser estruturado como um modelo de programação da produção em uma fundição de mercado é necessário realizar as seguintes considerações. Considere que cada arco decisório a_x , de entrada de um vértice v_y , represente uma possível corrida de forno, de forma que, $SA_{v_y} \subset A$, ou seja, o caminho de arcos decisórios que ligam v_y ao vértice de origem represente o seqüenciamento destas corridas no programa de produção gerado, em que, para cada vértice solução v_y deste caminho, sejam \bar{l}_{v_y} a liga, \bar{r}_{v_y} o reator e \bar{f}_{v_y} o forno da corrida do seu arco de entrada a_x .

Considere também $\bar{\psi}_{v_y, om} \forall om$ e $\bar{\tau}_{v_y, ea, oa} \forall ea, oa$ os valores das capacidades restantes das operações de moldagem e das operações de todos os estágios de acabamento em cada vértice solução v_y e $\bar{\delta}_{v_y, om} \forall om$ e $\bar{\mu}_{v_y, ea, oa} \forall ea, oa$ os valores das capacidades cambiáveis restantes de moldagem e das operações de todos os estágios de acabamento em cada vértice v_y . Considere ainda $SIA_{a_x} \subset S_I$ o conjunto de itens de pedidos programados na corrida de forno do arco a_x . Seja $SIV_{v_y} = \bigcup_{a_x \in SA_{v_y}} SIA_{a_x}$ o conjunto dos itens

de pedidos programados, em todos os arcos $a_x \subset A$ do caminho SA_{v_y} .

Cabe lembrar que os primeiros vértices da árvore, ou os vértices de origem, representam as últimas corridas do ciclo anterior, de forma que $\rho v_{v_y} = 0$, $\bar{\psi}_{v_y, om} = 0 \forall om$ e $\bar{\tau}_{v_y, ea, oa} = 0 \forall ea, oa$ para estes vértices, mas a liga \bar{l}_{v_y} , o reator \bar{r}_{v_y} e o forno \bar{f}_{v_y} estão caracterizados de acordo com a última corrida do ciclo anterior, devendo-se obedecer às relações de ordenamento das ligas nos fornos. Este modelo heurístico proposto será convenientemente tratado por *HE*.

A solução do modelo *HE* consiste em encontrar o vértice $v_y \in V$ de maior valor e apresentará em cada vértice solução as corridas de fornos programadas em seus arcos de entrada $a_x \subset A$, ou seja, $v_y = suc_{a_x}$, de tal forma que \bar{l}_{v_y} será a liga, \bar{r}_{v_y} será o reator e \bar{f}_{v_y} será o forno da referida corrida. E apresentará também em SIA_{a_x} , os itens de pedidos programados para cada corrida do vértice $v_y = suc_{a_x}$. O valor total da solução será o valor do último vértice da solução.

De acordo com Croce et al. (2004), o processo de avaliação das soluções parciais em cada estágio da árvore de busca será a “chave” para a aplicação eficiente do método de busca *beam search* em um problema específico. O método utilizado para esta escolha foi inspirado no procedimento atual de programação da produção das fundições de mercado. Nestas empresas, o processo empregado para compor as corridas dos fornos, toma por base uma listagem ordenada de forma decrescente das corridas de fornos mais interessantes. Uma vez que o número de corridas de fornos possível em um período de programação típico de uma fundição de mercado é relativamente pequeno (entre 8 e 32), buscou-se avaliar, em cada vértice possível de um estágio decisório, qual seria a seqüência de corridas de fornos resultante da aplicação de uma estratégia “gulosa” de intensificação. Ou seja, em cada estágio subsequente da árvore de decisão, de acordo com o vértice de solução $v_y \in V$ considerado, escolhe-se o arco decisório $a_x \in SU_{v_y}$ de maior valor ρa_{a_x} até se chegar à extremidade da árvore. Esta estratégia pode ser considerada como uma estratégia “gulosa” e irá caminhar pelos vértices conexos sucessivos mais valorados. Uma vez que o caminho entre a raiz e o extremo da árvore de decisão proposta será relativamente curto, é de se esperar que a aplicação desta estratégia gulosa de seleção seja computacionalmente viável.

Desta forma, considere $ASG_{v_y} \subset A$ ao conjunto de arcos do caminho resultante da aplicação desta estratégia, a partir de qualquer vértice solução $v_y \in V$ e $\rho g_{v_y} = \rho v_{v_y} + \sum_{ASG_{v_y}} \rho a_{a_x}$, o valor deste caminho para o vértice

solução $v_y \in V$. Considere também Θ o diâmetro de busca do método e $VSE \in V$ o conjunto dos vértices solução em um estágio decisório, de forma que, em cada novo estágio decisório, o método de solução irá criar um novo conjunto $VSE \in V$ compostos por Θ vértices solução $v_y \in V$ de maior valor ρg_{v_y} , em que cada $v_y \in V$ deste novo conjunto será um vértice filho dos vértices do conjunto $VSE \in V$ do estágio decisório anterior, até que o conjunto $VSE \in V$ de um novo estágio seja igual ao conjunto $VSE \in V$ do estágio anterior, representando o momento de parada do método de solução. Por fim, considerando que o processo de solução resulte no vértice $v_y \in V$ de acordo com o objetivo do modelo proposto, a solução do problema será representada por $S = (VS, AS)$, em que $AS = SA_{v_y}$ e $VS = \{suc_{a_x} \setminus a_x \in AS\}$.

A seguir, será apresentada uma visão geral do esquema de funcionamento da proposta de implementação do método *beam search* para a solução da árvore de decisão proposta por este trabalho. Para fins apenas de entendimento do funcionamento do método, será usada uma árvore decisória simplificada sem considerar os detalhes do ambiente de produção de uma fundição (a extensão

deste exemplo para o problema completo de uma fundição de mercado poderá ser realizado convenientemente). Para isto, será tomado por base o exemplo da Figura 3 sendo considerado o diâmetro de busca Θ igual a 3.

Na Figura 3, D_e é o estágio decisório, de forma que, em cada um destes estágios decisórios o processo de solução do método “caminha” para o próximo estágio por meio da escolha de um ou mais arcos decisórios $a_x \in SU_{v_y}$, em que $v_y \in V$ é o vértice solução do estágio atual.

Fazendo a primeira análise considerando o vértice de origem $v_1 \in V$ e analisando todos os seus arcos de saída $a_x \in SU_{v_1}$, resultará da aplicação da estratégia gulosa para escolha dos vértices no estágio $D_e = 1$ os valores $\rho g_{v_2} = 2050$, $\rho g_{v_3} = 2000$ e $\rho g_{v_4} = 1650$, incluindo então os vértices solução v_2, v_3 e $v_4 \in V$ em VSE . Para o estágio $D_e = 2$, tomando por base agora os vértices solução em VSE e analisando para cada vértice $v_y \in VSE$ todos os seus arcos de saída $a_x \in SU_{v_y}$, conseguem-se os valores para escolha $\rho g_{v_5} = 2050$, $\rho g_{v_6} = 1750$, $\rho g_{v_{11}} = 2000$, $\rho g_{v_{15}} = 1650$ e $\rho g_{v_{16}} = 1850$, escolhendo-se então os vértices solução v_5, v_{11} e $v_{16} \in V$ para compor o novo conjunto solução VSE . Para o estágio $D_e = 3$, considerando agora os vértices $v_y \in VSE$ e analisando, para cada um destes vértices em VSE todos os seus arcos de saída $a_x \in SU_{v_y}$, é possível chegar aos valores, para os novos vértices analisados, $\rho g_{v_7} = 1850$, $\rho g_{v_8} = 2050$, $\rho g_{v_{12}} = 2150$, $\rho g_{v_{13}} = 2000$, $\rho g_{v_{19}} = 1600$ e $\rho g_{v_{20}} = 1850$, escolhendo-se então os vértices solução $v_8,$

v_{12} e $v_{13} \in V$ para compor o novo conjunto solução VSE . Seguindo o mesmo raciocínio para o estágio decisório $D_e = 4$, cada vértice solução em VSE possui apenas um único arco de saída, forçando a se chegar aos vértices solução v_9, v_{22} e $v_{14} \in V$ com os valores $\rho g_{v_9} = 2050$, $\rho g_{v_{22}} = 2150$ e $\rho g_{v_{14}} = 2000$, compondo o novo conjunto VSE . Para o estágio $D_e = 5$, é possível escolher entre o valor do vértice solução $\rho g_{v_{14}} = 2000$, que não possui arco de saída, ou analisando os arcos de saída $a_x \in SU_{v_{14}}$ dos outros dois vértices, que compõem VSE , chega-se aos valores para escolha $\rho g_{v_{10}} = 2050$, $\rho g_{v_{23}} = 2200$ e $\rho g_{v_{24}} = 2150$ para os vértices sucessores dos vértices em VSE , escolhendo-se, então, os vértices v_{10}, v_{23} e $v_{24} \in V$ para compor o novo conjunto VSE . A decisão do último estágio $D_e = 6$ envolve escolher entre os valores dos vértices $\rho g_{v_{10}} = 2050$ e $\rho g_{v_{24}} = 2150$, que não possuem arcos de saída, ou o valor do vértice sucessor ao vértice $v_{23} \in V$ que será $\rho v_{27} = \rho g_{v_{27}} = 2200$, escolhendo-se então os vértices v_{10}, v_{24} e $v_{27} \in V$ para compor o novo conjunto VSE . Perceba que para um novo estágio decisório $D_e = 7$ nenhum dos vértices em VSE apresentaria sucessores, escolhendo-se, então, como resultado da solução o valor do vértice em VSE que possua o maior valor ρv_y em que $v_y \in VSE$, o que, no caso considerado, geraria como solução o caminho que leva a origem da árvore ao vértice $v_{27} \in V$, ou seja, $AS = SA_{v_{27}} \subset A$ e $VS = \{suc_{a_x} \setminus a_x \in SA_{v_{27}}\}$.

No exemplo tratado, o uso de um diâmetro de busca $\Theta = 3$ levou ao ótimo global. Infelizmente, não se pode

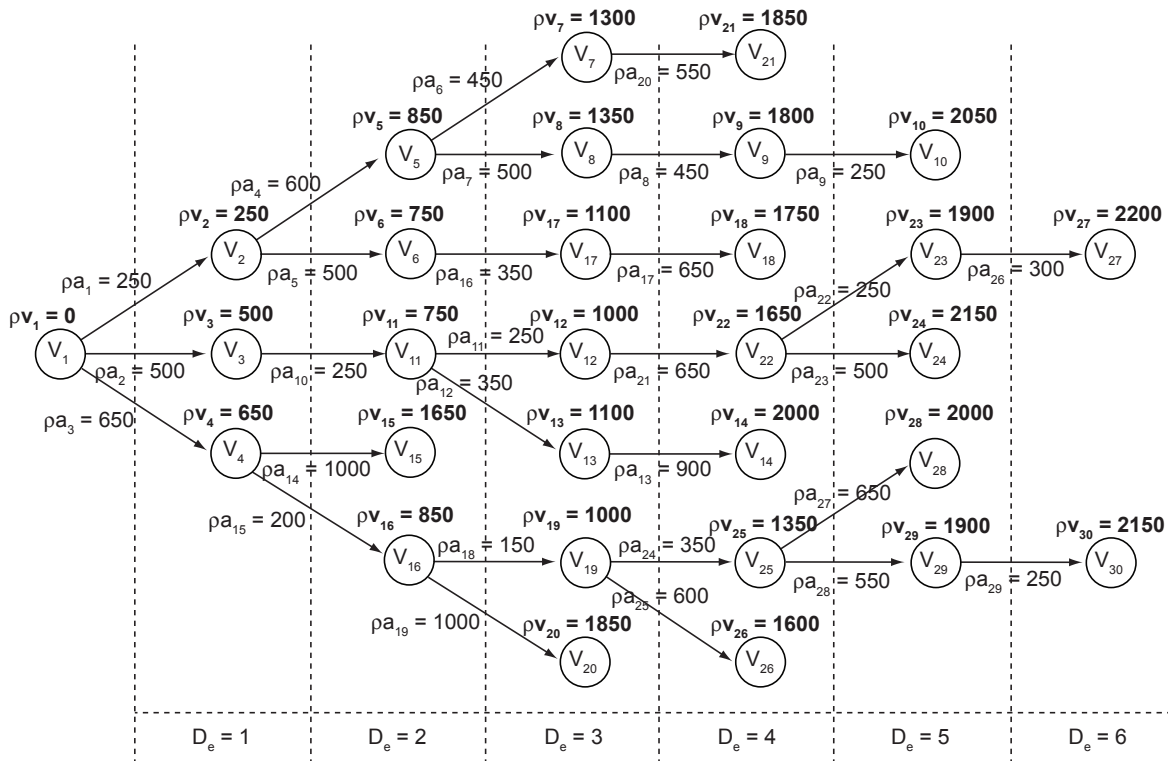


Figura 3. Estruturação de um problema hipotético na árvore de busca como proposto.

garantir previamente qual será o diâmetro mais econômico que levará ao caminho da solução ótima para cada caso particular, ou até mesmo, que exista um diâmetro que seja computacionalmente viável para a obtenção desta solução ótima. Por outro lado, o uso de diâmetros de busca suficientemente amplos e que ainda possibilitem a execução em um tempo viável poderá apresentar excelentes soluções a problemas reais.

6. Formulação do modelo baseado em algoritmos genéticos

Algoritmos genéticos (AG) são algoritmos adaptativos de busca de propósito geral, baseados no mecanismo de seleção natural e adaptação (Goldberg, 1989). Em um AG, soluções são representadas como indivíduos em uma população na qual as características de cada indivíduo, ou seja, atributos de uma solução, estão adequadamente codificados em um ou mais cromossomos. A idéia básica é partir de um conjunto inicial de soluções, ou seja, de uma população inicial e obter novas soluções por meio de um mecanismo de reprodução, *crossover* e mutação, evoluindo gradualmente estas soluções por várias gerações de indivíduos (Kimms, 1999).

Aplicações de AG em problemas de programação da produção também poderão ser encontradas, por exemplo, em Ponnambalam et al. (2001).

Para a definição do modelo baseado em algoritmos genéticos, este trabalho procurou adaptar tanto o problema de elaboração da programação da produção como também o AG, como defendem Cheng et al. (1999). Para isto, primeiramente, é necessário definir uma estrutura conveniente para estruturar a solução do problema como um ou mais cromossomos. Desta forma, são propostos dois cromossomos para representar um indivíduo *q*. O primeiro cromossomo armazena em cada “gene” uma corrida de forno programada, de forma que a

ordem dos “genes”, ou seja, seu “locus” será a ordem na qual as corridas de fornos irão acontecer no referido programa. Cada gene deste cromossomo possui três informações, quais sejam: a identificação do reator *r*, a identificação do forno *f*, e a identificação da liga metálica *l* que será fundida na referida corrida. O segundo cromossomo armazena em cada “gene” um item de pedido que será fundido neste programa, de forma que, o ordenamento de seus “genes” irá determinar a maneira na qual os fornos serão carregados. Neste segundo cromossomo, cada gene possui apenas uma informação armazenada, qual seja a identificação do item de pedido *i* correspondente a este “gene”.

O inter-relacionamento entre os cromossomos toma por base a ordem dos seus “genes”. Inicia-se com o primeiro “gene” do “cromossomo 1” conhecendo-se a capacidade do forno correspondente à sua corrida e o tipo de liga-metálica que irá ser fundido, verificando no “cromossomo 2”, em ordem, os “genes” que poderão ser relacionados, de acordo com o peso e o tipo de liga do item de pedido correspondente a cada “gene”, até que todos os “genes” do “cromossomo 2” tenham sido verificados. Este raciocínio é repetido para os outros “genes” do “cromossomo 1”, considerando agora, no “cromossomo 2”, somente os “genes” que ainda não tenham sido relacionados. Na Figura 4, pode ser visualizado um indivíduo hipotético no qual é ilustrado um exemplo de como seus dois cromossomos se inter-relacionam.

Quanto aos operadores genéticos de *crossover* e “mutação”, foram propostos, respectivamente para cada tipo de operação genética, dois operadores distintos, um para cada cromossomo. Na Figura 5, o esquema de funcionamento do operador genético de *crossover*, para o “cromossomo 1”, considerando indivíduos hipotéticos, é visualizado para o exemplo proposto.

Perceba que os “genes” externos aos pontos aleatórios de corte receberão sempre, em sua devida ordem, os “ge-

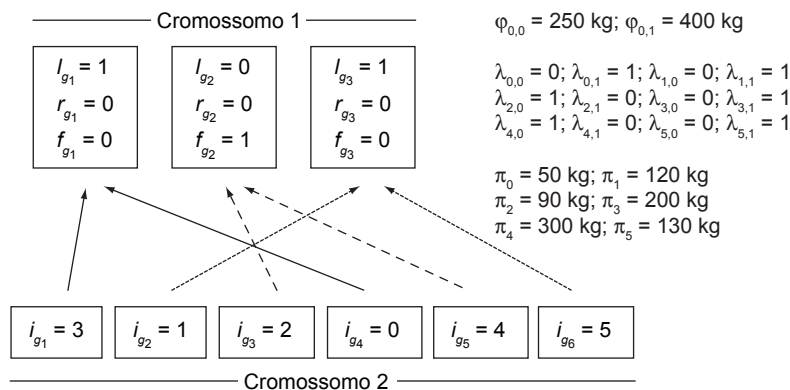


Figura 4. Inter-relacionamento entre os cromossomos de um indivíduo hipotético.

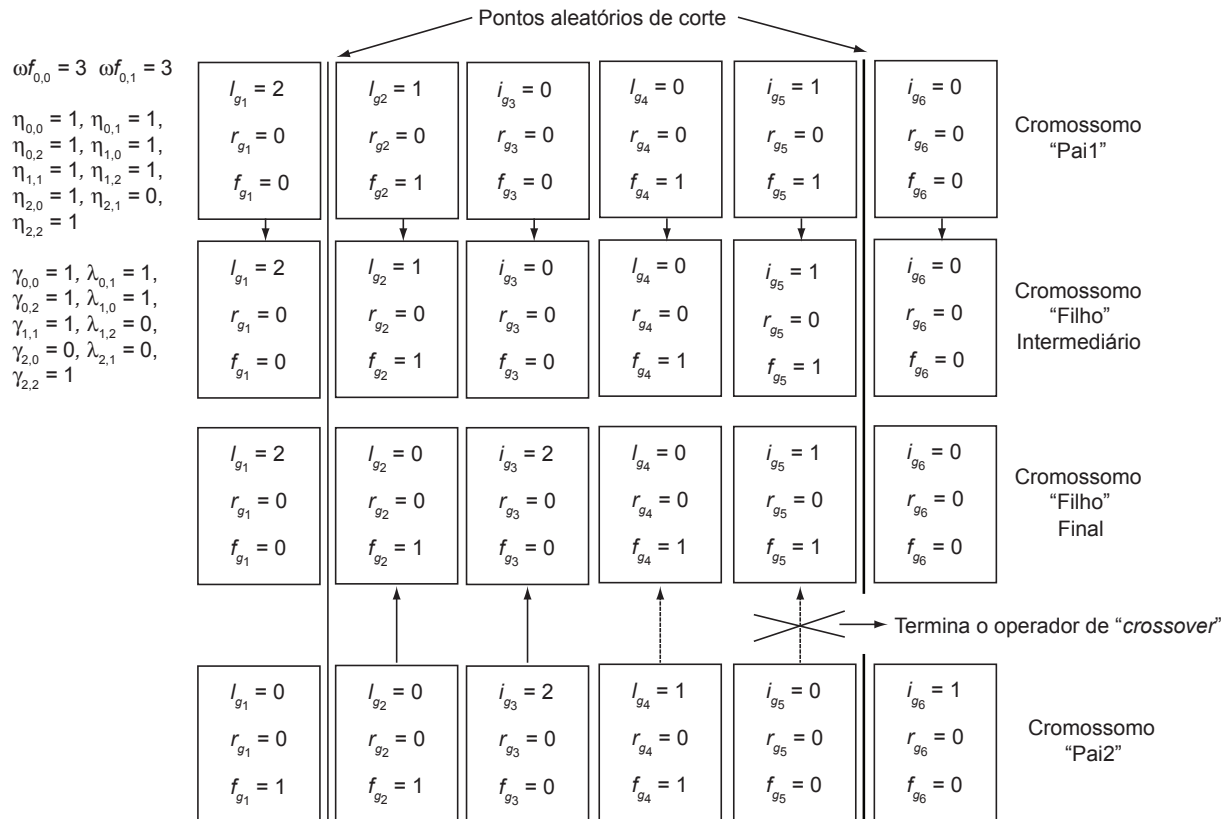


Figura 5. Esquema de funcionamento do operador genético de *crossover* para o “cromossomo 1”.

nes” do cromossomo “Pai1”. Quanto aos “genes” internos aos pontos aleatórios de corte, cada “gene” é analisado separadamente, de acordo com sua seqüência. Primeiramente todos os “genes” internos recebem, em sua devida ordem, os “genes” de “Pai1”. Em seguida, para cada um destes “genes” internos aos pontos de corte, é verificado se, caso seja permutado este “gene” com o correspondente “gene” do cromossomo “Pai2”, as restrições da capacidade máxima dos fornos e dos reatores serão obedecidas, assim como as restrições do ordenamento das ligas metálicas e, caso também seja econômico realizar esta permutação, o “gene” do cromossomo “filho” será permutado com o respectivo “gene” de “Pai2”, caso contrário, o cromossomo “filho” continuará com o “gene” de “Pai1” e termina-se o processo de *crossover*.

Já o operador de *crossover* para o “cromossomo 2” é realizado em duas etapas distintas. Em uma primeira etapa é criado um repositório intermediário em que os “genes” serão armazenados em ordem, para que, em uma segunda etapa, a partir deste depositório, seja criado o cromossomo “Filho”. Durante a primeira etapa, primeiramente, realiza-se o corte dos cromossomos “Pais” a partir de dois pontos aleatórios. Feito isto, inicia-se preenchendo o repositório com os “genes” de “Pai1”, em sua devida

da ordem, até o ponto que corresponde ao primeiro corte aleatório, tal que, os itens de pedidos correspondentes a estes “genes” provenientes do cromossomo de “Pai1” não estejam nos “genes” da parte interna do corte aleatório do cromossomo de “Pai2”. Feito isto, todos os “genes” da parte interna de corte do cromossomo de “Pai2” são transferidos para este repositório, em sua devida ordem. Neste momento, os “genes” de “Pai1” correspondentes a itens de pedidos que ainda não estejam nos “genes” do repositório são adicionados ao repositório, também em sua devida ordem. Por fim, este mesmo procedimento é realizado com os “genes” restantes de “Pai2”. Finalmente, o repositório de “genes” é completado com novos “genes” correspondentes a itens de pedidos que não estavam nos “genes” de “Pai1” nem de “Pai2”, até que todos os itens de pedidos a programar tenham um gene correspondente neste repositório. Na segunda etapa, tomando por base o “cromossomo 1”, que já foi gerado para este novo indivíduo, segue-se o esquema de inter-relacionamento entre os cromossomos como ilustrado na Figura 4. Este esquema garantirá que todas as restrições de capacidade dos fornos e das operações produtivas sejam satisfeitas, “marcando” os cromossomos que foram relacionados neste procedimento. Por fim, o “cromossomo 2” para o

indivíduo “filho” pode ser criado adicionando em um cromossomo inicialmente vazio, respeitando seu ordenamento, os “genes” “marcados” do repositório. Este procedimento descrito está ilustrado na Figura 6 considerando um exemplo hipotético.

O operador genético de “mutação” é mais simples, atuando da mesma forma para os dois tipos de cromossomos. Para isto, dois genes são sorteados e suas posições são trocadas. A única restrição é que, para o “cromossomo 1”, esta troca satisfaça as restrições de ordenamento de fornos e, também, que a nova seqüência de “genes” resultem, nas posições alteradas, em corridas econômicas. Caso estas restrições não sejam obedecidas, a “mutação” não acontece. É importante observar o fato de que a mutação para o “cromossomo 1” deve ocorrer antes que se aplique o operador genético de *crossover* para o “cromossomo 2”, uma vez que este operador depende do “cromossomo 1” para ser realizado. Já para o “cromossomo 2”, o processo de “mutação” acontece no repositório intermediário, antes de “marcar” os “genes” que serão selecionados para o cromossomo “filho”. Desta forma, a posterior marcação dos “genes” que irão fazer parte do cromossomo “filho” já irá ser influenciada por esta “mutação”.

Por fim, é aplicado um procedimento de melhoria local nos cromossomos “filhos” resultantes dos operado-

res de “*crossover*” e de “mutação”. Este procedimento de melhoria procura avaliar se ainda é possível inserir novos “genes” no “cromossomo 1”, ou seja, se a nova programação gerada ainda possui recursos produtivos suficientes para novas corridas de forno e se existem itens de pedidos a programar disponíveis. Para isto, todas as possibilidades de corridas de fornos são avaliadas, considerando as restrições de fornos, reatores e de ordenamento de ligas e também os itens de pedidos disponíveis, de acordo com sua ordem, no repositório de “genes” montado para o *crossover* do “cromossomo 2”, selecionando os que ainda não tiverem sido marcados pelo operador de *crossover*. Este procedimento de melhoria tem por objetivo evoluir a solução apresentada pelo indivíduo “filho” resultante, antes mesmo de inseri-lo na nova população gerada, entretanto, este procedimento caracteriza o modelo proposto como um AG híbrido.

Por fim, considere para o AG híbrido proposto os seguintes dados: SQ_{Ger} o conjunto de indivíduos de uma geração; $SG1_o$ o conjunto de “genes” no “cromossomo 1”; $SG2_o$ o conjunto de “genes” no “cromossomo 2”; e S_{GR} o conjunto de “genes” no repositório. N_{Ger} é o número de gerações de indivíduos que serão criadas; NP é o número de indivíduos de uma geração; $txcross$ é a taxa de *crossover*; $txmut$ é a taxa de mutação; $g1_y$ é um “gene” y do “cromossomo 1”; $g2_w$ é um “gene” w do

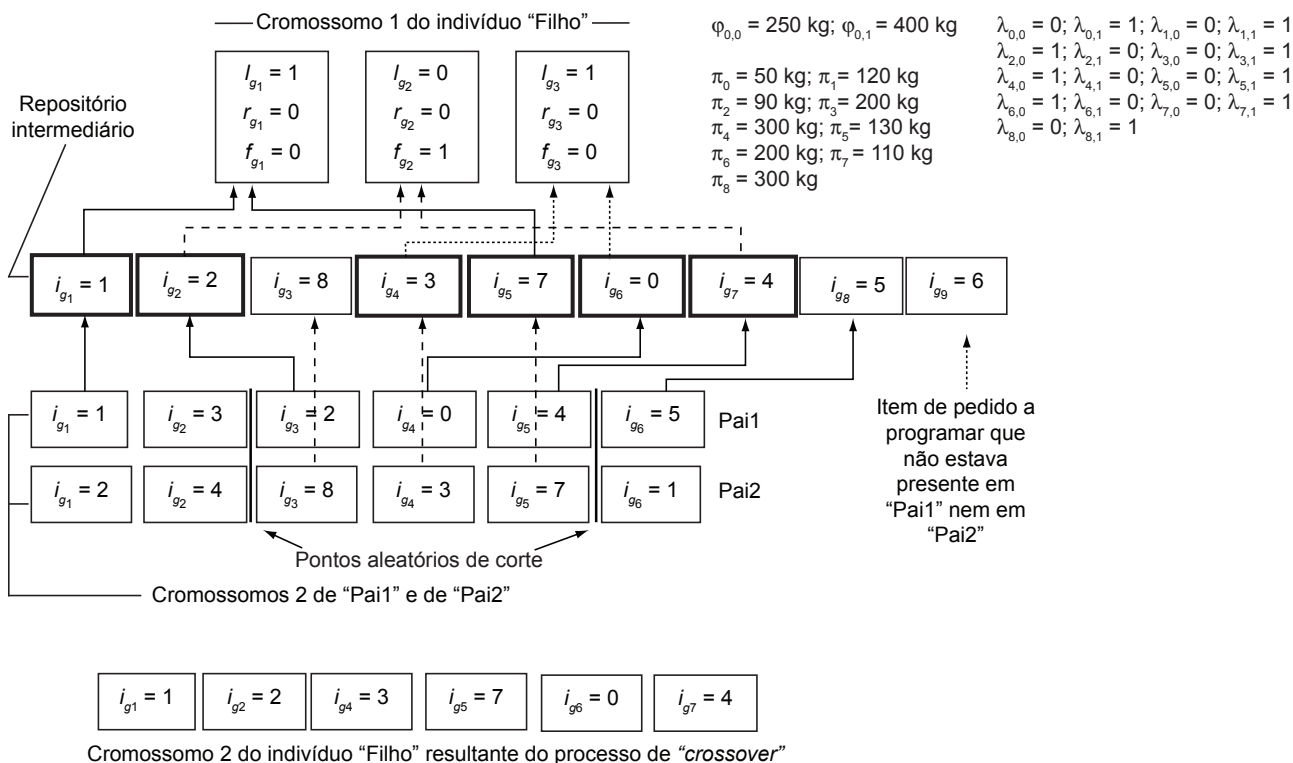


Figura 6. Esquema de funcionamento do operador genético de *crossover* para o “cromossomo 2”.

“cromossomo 2”; $\rho g1_y$ é o valor de um “gene” y do “cromossomo 1”; ρq_{q_x} é o valor de um indivíduo; ρnq_{q_x} é o valor normalizado de um indivíduo; $g1ant_{y,r,f}$ é o “gene” anterior ao “gene” y que usa o reator r e o forno f (caso não exista então $g1ant_{y,r,f} = -1$); $g1pos_{y,r,f}$ é o “gene” posterior ao “gene” y que usa o reator r e o forno f (caso não exista $g1pos_{y,r,f} = -1$); $i_{g_{iu}}$ é o índice do item de pedido do “gene” g_{iu} do conjunto de “genes” S_{GR} ; $i_{g_{2w}}$ é o índice do item de pedido do “gene” g_{2w} do conjunto de “genes” $SG2_Q$; $\bar{l}_{g_{1y}}$ é a liga metálica, $\bar{r}_{g_{1y}}$ é o reator e $\bar{f}_{g_{1y}}$ é o forno do “gene” g_{1y} do conjunto de “genes” $SG1_Q$; $\bar{\Psi}_{om}$, $\bar{\tau}_{ea,oa}$ são analogias, para fins de cálculos, aos dados Ψ_{om} , $\tau_{ea,oa}$, respectivamente.

7. O sistema de apoio à decisão proposto

Os sistemas de apoio à decisão (SAD) são sistemas computacionais que visam sistematizar e apoiar os processos decisórios empresariais, sendo comumente compostos pelos mesmos componentes básicos como: interface com o usuário, banco de dados e banco de modelos, visando fornecer apoio em decisões semi-estruturadas e desestruturadas, estando sob controle do tomador de decisões e disponibilizando um conjunto de ferramentas para estruturar e aumentar a efetividade das decisões. Informações gerais sobre SAD podem ser encontradas, por exemplo em Marakas (1999), Mallach (2000), Shim et al. (2002) e Kim (2001). Eom (1996) revisa artigos sobre SAD e aplicação de SAD em programação da produção pode ser vista, por exemplo, em Chryssoulouris et al. (2000).

O SAD proposto tomará por base os modelos decisórios previamente criados para disponibilizar ao tomador

de decisões, ao término do estágio de acúmulo de pedidos em cada ciclo, o apoio necessário para a elaboração de um programa de produção viável para o ciclo. Este processo pode ser mais bem visualizado na Figura 7.

É importante ressaltar que o tempo de execução dos métodos de solução seja menor que poucos minutos, de modo que a interatividade com o tomador de decisões não seja comprometida. Além disso, suas interfaces deverão ser simples e intuitivas, para que os resultados apresentados possam ser rapidamente avaliados e as devidas análises concluídas e, caso necessário, um novo ciclo decisório possa ser iniciado.

8. Resultados experimentais

Visando garantir a aplicação prática dos modelos propostos sob a forma de um SAD, foram empregados vários casos experimentais que visam refletir com fidelidade situações produtivas reais deste setor industrial, tomando por base situações de produção reais da indústria utilizada como referência neste trabalho. Desta forma, foram gerados 8 cenários diferentes possíveis de acontecer em uma fundição de mercado característica. Cada um destes 8 cenários procura avaliar situações específicas de um ambiente produtivo real como, por exemplo: níveis diferentes de consumo das capacidades das corridas de fornos, de consumo das capacidades das operações produtivas; de complexidade dos itens de pedidos encomendados; e de distribuição dos itens de pedidos pelas ligas metálicas disponíveis. Para cada cenário experimental, foram gerados quatro casos experimentais, em que a diferença entre cada caso, para cada cenário, reside principalmente

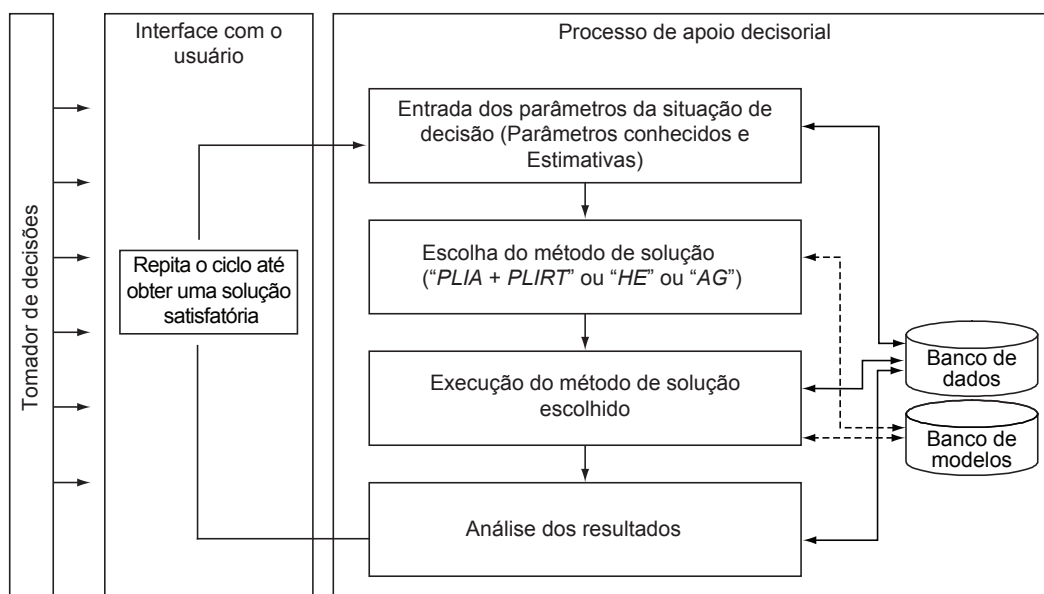


Figura 7. Sistema de apoio à decisão proposto.

no número de itens de pedidos diferentes a programar (variando entre 200 a 1000 itens de pedido a programar nos casos de cada cenário), totalizando, então 32 casos experimentais para análise.

Para cada caso experimental, foram analisadas as soluções apresentadas para cada modelo proposto, de acordo com seus respectivos métodos de solução. Visando facilitar o entendimento, o modelo *PLIRT*, quando aplicado utilizando como entrada os dados do método de solução do modelo *PLIRX*, será tratado como *PLIRT1*. Já quando o modelo *PLIRT* utilizar como dados de entrada os resultados apresentados pelo método de solução do modelo *PLIA*, este será tratado como *PLIRT2*. A análise dos métodos de solução dos modelos propostos toma por base, para cada caso experimental, o limite superior de solução, proveniente da solução do modelo *PLIRX*, e o limite inferior de solução, proveniente do modelo *PLIRT1*. Desta forma, contando a solução do limite superior e do limite inferior, *PLIRT2*, com tempo de solução de 360 e 3600 segundos, *AG* e *HE* diâmetros 1, 2 e 3, serão realizados, para cada caso experimental, 8 experimentos, totalizando então 256 experimentos realizados. Para a solução dos modelos de programação inteira binária, foi empregada a ferramenta comercial CPLEX 7.0. Já para as implementações computacionais dos modelos *HE* e *AG* foi empregada a ferramenta de desenvolvimento DELPHI 7.0. Por fim, todos os testes computacionais foram realizados em um computador portátil Toshiba® Satellite A25-S207, equipado com um processador Intel® Pentium® 4 de 2,66 GHz e 512 MB de memória RAM.

Os resultados experimentais revelam que em média as primeiras soluções apresentadas para o modelo *PLIRX* estão 15,2% distantes da estimativa da solução ótima, enquanto que, para o modelo *PLIRT1*, as primeiras soluções apresentadas estão, em média, 2,6% distantes da estimativa de uma possível solução ótima. Uma vez que o método de solução para o modelo *PLIRT* utiliza como entrada a solução do modelo *PLIRX*, é possível avaliar que o uso da primeira solução apresentada para *PLIRX* poderá comprometer sua performance. Os experimentos revelaram que o modelo *PLIRX* possui uma dificuldade de conversão, em tempos menores, da solução apresentada, principalmente quando comparado com o modelo *PLIRT*, uma vez que, dos 32 casos experimentais, para 22 casos, o método de solução não apresentou soluções iniciais com uma diferença menor que 10% da estimativa da solução ótima, e destes, 12 casos experimentais necessitaram em torno de 1 hora para apresentar uma solução com uma proximidade de 10% da estimativa da solução ótima. Estes resultados justificam o fato de não se utilizar a aplicação conjunta dos modelos *PLIRX* e *PLIRT* no SAD proposto.

Já para a aplicação conjunta dos modelos *PLIA* e *PLIRT*, a análise dos resultados experimentais revela que

o modelo *PLIA* apresentou menores variações nos tempos de execução que o modelo *PLIRT*. Apesar de o modelo *PLIRT* ter apresentado, em tempos menores que 1 minuto, soluções factíveis para 28 dos 32 casos experimentais solucionados. De forma geral, o modelo *PLIA* apresentou um tempo médio de 16 segundos para a primeira solução, e o modelo *PLIRT* apresentou, também para a primeira solução, um tempo médio de 45 segundos. Quanto à diferença entre esta primeira solução apresentada e a estimativa da solução ótima para os modelos *PLIA* e *PLIRT*, para cada caso experimental, foi possível avaliar que *PLIA* é o modelo que apresenta as soluções iniciais de menor qualidade, entretanto, estão, em média, apenas 5,7% distantes das estimativas de suas soluções ótimas. Já para o modelo *PLIRT*, as primeiras soluções apresentadas estão, em média, 2,4% distantes das estimativas das possíveis soluções ótimas.

Quanto ao modelo heurístico *HE*, considerando os diâmetros de busca igual a 1, 3 e 10, seu método de solução apresenta um acréscimo aproximadamente linear do tempo de solução, de forma geral, para os casos considerados, tanto em relação ao aumento do diâmetro de busca como em relação ao aumento do número de itens de pedidos considerados. A análise dos resultados apresentados mostra que, em média, o método de solução para o modelo *HE*, considerando o diâmetro igual a 1, não excedeu um minuto para os casos considerados. Proporcionalmente, estes tempos serão, em média, 3 e 10 minutos para os diâmetros iguais a 3 e 10, respectivamente.

Já para o modelo meta-heurístico *AG*, é necessário realizar uma avaliação prévia, com base em testes experimentais, dos parâmetros envolvidos com seu método de solução. Tomando por base testes estatísticos, com várias opções de configuração, para esta análise (empregou-se o teste ANOVA para análise dos resultados, ver Triola (1999)), foi possível definir os seguintes parâmetros para execução do *AG*: taxa de *crossover* = 0,9; taxa de “mutação” = 0,1; número de indivíduos em cada geração populacional = 100; número de gerações que serão criadas = 100; será considerado sobreposição populacional; será feito uso da estratégia de hibridização com métodos de melhoria. Devido, também, à natureza probabilística do modelo *AG*, foram realizadas 30 replicações de seu método de solução. Em média, o tempo de solução apresentado para o modelo *AG* foi baixo. É possível avaliar também que o modelo *AG* sofre influência do número de itens de pedidos diferentes a programar, apresentando, em média, tempos de solução maiores em cenários com maior disponibilidade de itens de pedidos em aberto, mas, de forma geral, este acréscimo no tempo de solução foi pouco expressivo. Por fim, para a maioria dos casos experimentais avaliados, o método de solução do modelo

AG apresentou tempos de solução abaixo de um minuto.

A avaliação da performance obtida pelos métodos de solução tomou por base o limite superior de solução para cada caso experimental. Este limite foi fornecido pelo método de solução do modelo *PLIRX*. Para isto, empregou-se a ferramenta *CPLEX 7.0* com um limite de tempo de solução de 14.400 segundos, utilizando como solução, findado o tempo de execução, a estimativa da solução exata que foi apresentada pela ferramenta. Para os casos experimentais analisados, esta estimativa estava em média apenas 6,1% distante da solução exata estimada para o modelo *PLIRX*. Já os valores da performance do método de solução para o modelo *PLIRT2* foram obtidos considerando o tempo de solução de 180 segundos para *PLIA* e 180 segundos para *PLIRT*, totalizando 360 segundos.

A análise dos experimentos revelou, sem considerar casos isolados, que o modelo de programação matemática *PLIRT2* apresenta uma melhora na performance em casos com maior número de itens de pedido em aberto, enquanto os modelos *AG* e *HE*, inversamente, apresentam uma melhora na performance em casos com menos itens de pedido em aberto. Pode-se analisar, também, que o modelo de programação matemática *PLIRT2* apresenta uma performance relativamente menor nos cenários com itens de pedidos mais complexos. Quanto ao modelo *HE*, pode-se analisar que sua performance será tão maior quanto menor for a concorrência por recursos produtivos. O modelo *AG*, por sua vez, conseguiu performances superiores em alguns casos isolados, conseguindo obter, também, resultados satisfatórios para todos os casos considerados.

Quanto à análise do impacto do tempo de solução na performance apresentada pelo modelo *PLIRT2*, é possível analisar considerando 10 vezes o tempo usado para os testes de performance anteriores, ou seja, 1.800 segundos para *PLIA* e 1.800 segundos para *PLIRT2*, que, no geral, este impacto não foi significativo, apresentando resultados apenas 1,2% superiores.

Já para o modelo heurístico *HE*, quanto ao impacto do diâmetro nas performances apresentadas pelo seu método de solução, é possível avaliar que diâmetros de solução maiores possibilitaram um aumento apenas sensível nas performances apresentadas pelo método de solução de *HE*. É possível avaliar também, apesar de não ser comum, que diâmetros maiores podem convergir para regiões de ótimos locais inferiores. A análise dos resultados apresentados revela que o aumento do diâmetro de busca de 1 para 3 possibilitou uma melhoria média de 1,1% nas performances de solução, enquanto que, o aumento do diâmetro de 3 para 10 causou uma melhoria média de apenas 0,6% nas performances medidas.

Por fim, os resultados experimentais podem ser apresentados resumidamente: quanto menor for a competição

por recursos produtivos maior será a performance apresentada pelo modelo heurístico *HE*, apresentando, quando se tem baixos níveis de competição por recursos, performances superiores que os modelos *PLIRT2* e *AG*. Quando se consideram níveis de médios a altos de competição por recursos, o modelo heurístico *HE* apresentará performance superior ao modelo de programação matemática *PLIRT2* somente em situações de elevada complexidade dos itens de pedidos e/ou de maior distribuição dos itens de pedido pelas ligas metálicas que a indústria trabalha. O modelo *AG*, por outro lado, não apresentou grandes variações de performance nos diferentes cenários, o que confirma a robustez de seu método de solução. Desta forma, o modelo *AG* conseguiu performances superiores nos cenários produtivos em que os modelos *HE* e *PLIRT2* tiveram mais dificuldade, quais sejam os ambientes de média ou alta competição por recursos, com itens de produção de complexidade maior.

A Figura 8 permite avaliar a performance média geral (expressa em % alcançada do limite superior) apresentada pelos métodos de solução para os modelos propostos em todos os casos experimentais avaliados.

Por fim, os três modelos se completam quando aplicados em um SAD, apresentando respostas distintas, para as várias situações decisórias possíveis de acontecer na realidade de um ambiente de produção de uma fundição de mercado.

9. Análise de adequação do sistema proposto nas fundições de mercado

A análise da adequação do sistema proposto tomou por base uma amostra representativa das fundições de mercado existentes nos principais pólos deste setor industrial no interior do Estado de São Paulo (Indaiatuba, Itu, Piracicaba e São Carlos). Foram selecionadas

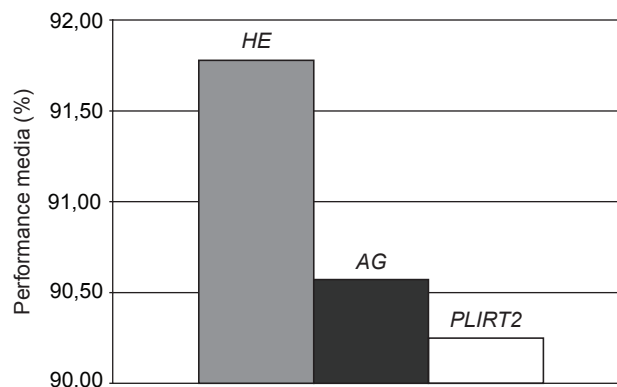


Figura 8. Performance média geral dos métodos de solução para os modelos propostos expressa em porcentagem alcançada do limite superior.

2 fundições de mercado mais representativas em cada um destes pólos, totalizando uma amostra de 8 empresas. Aplicou-se um questionário estruturado versando sobre as seguintes características de seu sistema produtivo: forma de produção, variabilidade da produção, forma de elaboração da programação da produção incluindo restrições e objetivos considerados e, também, principais prioridades de melhorias do ambiente produtivo que estas empresas almejam que sejam decorrentes de um método mais sofisticado de elaboração da programação da produção.

As respostas obtidas permitiram avaliar que todas as restrições atualmente consideradas para elaborar a programação da produção para a etapa de vazamento são atendidas, assim como todos os possíveis objetivos que atualmente norteiam este processo de programação da produção. A diversidade atual de objetivos considerados simultaneamente durante a elaboração de um programa de produção, assim como o grau de subjetivismo envolvido por estes objetivos, poderá ser convenientemente atendida pelo procedimento de valoração proposto por este trabalho.

Identificou-se que a principal prioridade de melhoria é a estimativa de prazos de entrega mais realísticos. Ela poderá ser devidamente atendida pelo sistema de controle da produção *PBC* proposto; o *PBC* permitirá ampliar consideravelmente o horizonte de programação da produção praticado pela indústria e avaliar com boa precisão o possível ciclo produtivo futuro no qual um item de pedido será programado, permitindo conhecer, por conseguinte, sua data de término.

Já a consideração das capacidades dos fornos de fundição, assim como, do número máximo de corridas disponível para cada forno e para cada reator e, também, as restrições de ordenamento de ligas e das seqüências de ligas que agilizam o processo de fundição, permitem atender à melhoria desejada de otimização da etapa de vazamento. Enquanto que, a consideração das estimativas de

consumo das operações produtivas pelos itens de pedido em aberto, assim como a consideração das estimativas das capacidades das operações nas etapas de moldagem e acabamento, permite atender às melhorias desejadas quanto à otimização das capacidades das operações das etapas de moldagem e acabamento.

Com o uso do *PBC*, espera-se, também, uma redução dos níveis de estoques em processo, impactando positivamente a qualidade devido a uma menor movimentação e empilhamento dos moldes. Será possível também o aumento da flexibilidade produtiva por meio de um intercâmbio entre operações similares ou de uma mesma etapa.

10. Conclusão

Os experimentos permitem observar que os três modelos propostos atingiram níveis significativos de performance quando comparados ao limite superior de solução para cada caso analisado. Uma vez que este limite superior de solução é uma estimativa igual ou superior ao limite máximo de uma possível solução exata, pode-se avaliar o quão próximo estas soluções obtidas se encontram das possíveis soluções exatas para os casos experimentais tratados. Com isto, é possível confirmar a viabilidade e efetividade dos três modelos propostos; mas, no cômputo geral, o método que mais se destaca é o *beam search HE*. Leitores interessados no detalhamento dos algoritmos genético e *HE* propostos podem entrar em contato com um dos autores por meio de e-mail.

Pode-se também, com base na amostra representativa de fundições de mercado analisada, confirmar a aplicação prática do sistema proposto. Espera-se assim, com base nos resultados obtidos, que este trabalho represente uma solução viável e efetiva para o processo de elaboração da programação da produção em fundições de mercado. Isso deverá ser comprovado num futuro trabalho cujo objetivo é implantar e avaliar o sistema proposto em uma fundição representativa.

Referências Bibliográficas

- ARAUJO, S. A.; ARENALES, M. N. **Dimensionamento de lotes e programação do forno numa fundição automatizada de porte médio**. Dissertação de mestrado, Depto. De Ciências de Computação e Estatísticas – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo – Campus São Carlos, 2003.
- BURBIDGE, J. L. **Period Batch Control**. Oxford: Clarendon Press, 1996.
- CHENG R.; GEN M.; TSUJIMURA Y., “A tutorial survey of job-shopscheduling problems using genetic algorithms: Part II. Hybrid Genetic Search Strategies”, **Computers & Industrial Engineering**, v. 37, n. 1, p. 51-55, 1999.
- CHRYSSOLOURIS, G.; PAKAKOSTAS, N.; MOURTZIS, D. A decision-making approach for nesting scheduling: a textile case. **International Journal of Production Research**, v. 38, n. 17, p. 4555-4564, 2000.
- COLIN, E. C. Beam search filtrado com inserção de ociosidade na programação de uma máquina em ambiente do tipo JIT. **Gestão & Produção**, São Carlos, v. 5, n. 2, p. 117-132, 1998.
- CROCE, F. D.; GHIRARDI, M.; TADEI, R. Recovering beam search: Enhancing the beam search approach for combinatorial optimization problems. **Journal of Heuristics**, v. 10, n. 1, p. 89-104, 2004.

- EOM, S. B. Mapping the intellectual structure of research in decision support systems through author cocitation analysis (1971-1993). **Decision Support Systems**, v. 16, n. 4, p. 315-338, 1996.
- FERNANDES, F. C. F.; LEITE, R. B. Automação industrial e sistemas informatizados de gestão da produção em fundições de mercado. **Gestão & Produção**, São Carlos, v. 9, n. 3, p. 313-344, 2002.
- GHIRARDI, M.; POTTS, C. N. Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach. **European Journal of Operational Research**, v. 165, n. 2, p. 457-467, 2003.
- GOLDBARG, M. C.; LUNA, H. P. **Otimização combinatória e programação linear: modelos e algoritmos**. Ed. Campus, Rio de Janeiro, 2000.
- GOLDBERG, D. E., “**Genetic Algorithms in Search, Optimization & Machine Learning**”, Addison Wesley Longman Inc., New York, Estados Unidos, 1989.
- KIM, H. An XML-based modeling language for the open interchange of decision models. **Decision Support Systems**, v. 31, n. 4, p. 429-441, 2001.
- KIMMS, A. “A genetic algorithm for multi-level, multi-machine lot sizing and scheduling”, **Computers and Operations Research** – PERGAMON, v. 26, n. 8, p. 829-848, 1999.
- MALLACH, E. A. **Decision support and data warehouse systems**. McGraw-Hill: Boston, Estados Unidos, 2000.
- MARAKAS, G. M. **Decision Support Systems in The Twenty-First Century**. Prentice Hall, New Jersey, Estados Unidos, 1999.
- McMULLEN, P. R.; TARASEWICH, P. A beam search heuristic method for mixed-model scheduling with setups. **International journal of production economics**, v. 96, n. 2, p. 273-283, 2003.
- PACCIARELLI, D.; PRANZO, M. Production scheduling in a steelmaking-continuous casting plant. **Computers and Chemical Engineering**, v. 28, n. 12, p. 2823-2835, 2004.
- PONNAMBALAM, S. G.; RAMKUMAR, V.; JAWAHAR, N. A multiobjective genetic algorithm for job shop scheduling. **Production Planning & Control**, v. 12, n. 8, p. 764-774, 2001.
- SABUNCUOGLU, I.; KARABUK, S. A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. **IIE Transactions**, v. 30, n. 2, p. 179-191, 1998.
- SHERALI, H. D.; DRISCOLL, P. J. Evolution and State-of-the-art in integer programming. **Journal of Computational and Applied Mathematics**, v. 124, n. 1-2, p. 319-340, 2000.
- SHIM, J. P.; WARKENTIN, M.; COURTNEY, J. F.; POWER, D. J.; SHARDA, R.; CARLSSON, C. Past, present, and future of decision technology. **Decision Support Systems**, v. 33, n. 2, p. 111-126, 2002.
- SHIVAPPA, D. N.; BABU, S. Evolving requirements for a reengineering model for ferrous foundries in India - An experience. **International Journal of Production Economics**, v. 50, n. 2-3, p. 225-243, 1997.
- SILVA, R. J. **Programação de cargas de forno em uma fundição de mercado**. Dissertação de mestrado – Universidade Federal de São Carlos, Departamento de Engenharia de Produção, São Carlos, 2001.
- TOTH, P. Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems. **European Journal of Operational Research**, v. 125, n. 2, p. 222-238, 2000.
- TRIOLA, M. F. **Introdução à estatística**, 7ª ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos S.A. Editora, 1999.
- VIANNA, A. C. G.; ARENALES, M. Um problema de programação da produção em uma fundição automatizada. **Gestão & Produção**, São Carlos, Universidade Federal de São Carlos, v. 2, n. 3, 1995.
- VOORHIS, T. V.; PETERS, F.; JOHNSON, D. Developing software for generating pouring schedules for steel foundries. **Computers & Industrial Engineering**, v. 39, n. 3-4, p. 219-234, 2001.

DECISION SUPPORT SYSTEM FOR PRODUCTION SCHEDULING IN STEEL CASTING FOUNDRIES

Abstract

This work proposes a Decision Support System (DSS) for production scheduling in steel casting foundries. To this end, the productive environment of these foundries is organized by means of a production control system known as Period Batch Control (PBC). Three viable decision models for production scheduling are proposed: a binary integer programming model, a model based on classical approximation methods, specifically the method known as beam search, and a model based on a meta-heuristic known as genetic algorithm. Also presented here are the experimental results obtained from the application of the three models and an analysis of the fitness of the proposal for eight representative steel casting foundries in the state of São Paulo, Brazil.

Keywords: decision support systems, production scheduling, steel casting foundries.

