# *LOCAL BRANCHING* APLICADO AO PROBLEMA DE DIMENSIONAMENTO DE LOTES

**Renato Andrade de Paiva**
**Franklina M. B. Toledo**
ICMC - Instituto de Ciências Matemátias e de Computação
Av. Trabalhador São-carlense, 400 - Centro, São Carlos, SP, Brasil
renatodepaiva@gmail.com, fran@icmc.usp.br

**RESUMO**

O problema de dimensionamento de lotes com restrições de capacidade (PDLC) é uma das tarefas centrais envolvidas no planejamento da produção. O objetivo é determinar o tamanho dos lotes a serem produzidos em um horizonte de planejamento finito. Este trabalho possui dois objetivos centrais. O primeiro trata do desenvolvimento e da aplicação de um método *local branching* para solução do PDLC com três características adicionais: 1) possibilidade de atraso para atender a demanda (*backlogging*); 2) possibilidade de manter a preparação para produção de um item de um período para o seguinte (*carry-over*); 3) possibilidade de iniciar uma preparação para produção de um item num período e concluí-la no período seguinte (*set-up crossover*). O segundo é avaliar a influência destas características na melhoria do custo dos planos de produção. Os resultados obtidos mostram que as soluções encontradas pelo método *local branching* são melhores com relação à qualidade que às soluções obtidas pelo software comercial CPLEX IBM ILOG. Também se conclui que a possibilidade de *backlogging*, *carry-over* e *set-up crossover* pode proporcionar economia significativa durante o planejamento da produção.

**Palavras-chave:** dimensionamento de lotes; *backlogging*; *carry-over*; *set-up crossover*; *local branching.*

**ABSTRACT**

The capacitated lot-sizing problem (CLSP) is one of the central tasks involved in the production planning. The objective is to determine the size of the lots to be produced in a finite planning horizon. This work has two main goals. The first one is the development and application of a local branching method to solve the CLSP including three additional characteristics: 1) possibility of backlogging the demand; 2) possibility to carry-over the set-up for production of one item from a period to the next one; 3) possibility to start the set-up for production of one item in a period and finish it in the next period (set-up crossover). The second one is to evaluate the performance of these characteristics to improve the production planning costs. The results show that the solutions found by the local branching method are better in quality than the ones obtained by the commercial software CPLEX IBM ILOG. Also, it is possible to conclude that the possibility of backlogging, carry-over and set-up crossover can significantly improve the economy in the production planning.

**Keywords: lot-sizing:** backlogging; carry-over; set-up crossover; local branching.

## 1. Introdução

Segundo Pochet e Wolsey (2006) uma importante atribuição do planejamento da produção é determinar o tamanho dos lotes de produtos que serão fabricados em cada intervalo de tempo, de modo que a demanda de produção seja atendida sobre um horizonte de tempo preestabelecido, conhecido como horizonte de planejamento. Este problema é conhecido na literatura como problema de dimensionamento de lotes (*lot-sizing problem*) e vêm sendo estudado desde 1958. Desta forma, diversos trabalhos de revisão e classificação do problema foram apresentados na literatura (Bahl *et al.,*1987; Wolsey, 1995; Buschkühl et al., 2010).

O problema de dimensionamento de lotes com restrições de capacidade (PDLC), que aqui é considerado como padrão, visa encontrar um plano de produção que minimize a soma dos custos de produção, de preparação para produção e de estoque. O horizonte de planejamento é finito e subdividido em períodos cuja capacidade produtiva é limitada. A capacidade de produção é compartilhada por múltiplos itens que consomem recursos tanto para sua produção quanto para a preparação do ambiente produtivo para sua fabricação. A demanda dos itens em cada período é conhecida a priori e deve ser atendida sem atraso. Este problema foi modelado na literatura como um problema de programação inteira mista e segundo Maes *et al.* (1991) é de difícil solução.

O problema de dimensionamento de lotes objeto de estudo deste artigo, assim como proposto por Sung e Maravelias (2008) inclui três características adicionais em sua modelagem: 1) a possibilidade de atrasar a produção (*backlogging*), ou seja, a demanda de um item em um determinado período pode ser atendida nos períodos seguintes; 2) possibilidade de preservar a preparação para produção de um dado item de um período para o imediatamente seguinte (*carry-over*); 3) possibilidade da preparação de um item pode começar em um período e terminar apenas no período seguinte (*set-up crossover*). A primeira característica visa permitir encontrar soluções para problemas com restrições de produção apertadas. Já a segunda característica permite a economia de tempos e custos de preparação para produção, o que é muito importante para problemas reais em que a preparação para a produção é muito custosa e/ou consome grande parte dos recursos de produção, como acontece, por exemplo, no setor de bebidas e no setor químico. Como a terceira característica se busca um melhor aproveitamento da capacidade produtiva, uma vez que uma preparação pode começar no final de um período e ser concluída no final do período seguinte. Isto é importante para setores produtivos com demandas elevadas. Por simplicidade ao longo deste artigo este problema é chamado de problema de dimensionamento de lotes com restrições de capacidade geral (PDLCG).

Este trabalho tem dois objetivos principais: a) propor um método *local branching* para a solução do PDLCG e avaliar sua eficiência; b) analisar a influência das três características consideradas no PDLCG com relação ao custo das soluções obtidas e a dificuldade em resolver o problema. Para avaliar o desempenho do método de *local branching* os resultados obtidos foram comparados com os resultados obtidos pelo software de otimização CPLEX 11 (IBM ILOG, 2010) com a estratégia de *local branching* habilitada. Quanto à análise das três características adicionais do PDLCG, a qualidade das soluções é comparada com a qualidade das soluções obtidas ao resolver o PDLC, assim pôde-se avaliar se é significativa, para a qualidade das soluções, a inclusão de cada uma destas três características.

O restante deste artigo é organizado da seguinte maneira. Na Seção 2 é apresentado o modelo proposto por Sung e Maravelias (2008). Na Seção 3 o método *local branching* é descrito. Na Seção 4 são apresentados os resultados e a discussão e na Seção 5, a conclusão do trabalho.

## 2. O Problema de Dimensionamento de Lotes Abordado

O problema de dimensionamento de lotes abordado neste artigo foi modelado por Sung e Maravelias (2008). Os autores consideram a possibilidade de atender a demanda com atraso, a possibilidade de preservação de preparação para produção e a possibilidade de iniciar a preparação para produção de um item em um período para que ela seja concluída no período seguinte. Vale destacar que para permitir que uma preparação seja iniciada num período e continuada no período seguinte, os autores propuseram flexibilizar o tamanho dos períodos, utilizando para isto variáveis de controle de ociosidade e de atraso nos períodos. A seguir é apresentado o modelo proposto pelos autores.

| | | |
|---|---|---|
| $Min \sum_t \sum_i \left(h_i^+ I_{it}^+ + h_i^- I_{it}^-\right) + \sum_t \sum_i K_i \alpha_{it}$ | *t=1..T;* | (2.1) |
| $x_{it} + I_{i,t-1}^+ + I_{it}^- = d_{it} + I_{it}^+ + I_{i,t-1}^-$ | *t=1..T;* | (2.2) |
| $\sum_i \beta_{it} = 1$ | *i=1..N, t=1..T;* | (2.3) |
| $\beta_{it-1} \leq z_{it}$ | *i=1..N, t=1..T;* | (2.4) |
| $\beta_{it} \leq z_{it}$ | *i=1..N, t=1..T;* | (2.5) |
| $y_{it} \leq \beta_{it-1}$ | *i=1..N, t=1..T;* | (2.6) |
| $y_{it} \leq \beta_{it}$ | *i=1..N, t=1..T;* | (2.7) |
| $y_{it} \leq \sum_{i' \neq i} z_{i't}$ | *i=1..N, i'=1..T, i,i'≠ i,t;* | (2.8) |
| $y_{it} \geq \beta_{it-1} + \beta_{it} + z_{i't} - z_{it} - 1$ | *i=1..N, t=1..T;* | (2.9) |
| $\alpha_{it} = z_{it} - \beta_{it-1} + y_{it}$ | *t=1..T;* | (2.10) |
| $\sum_i \frac{x_{it}}{r_i} + \sum_i s_i \alpha_{it} + Idle_t = G_t - Late_{t-1} + Late_t$ | *t=1..T;* | (2.11) |
| $Late_t \leq \sum_i (s_i - \delta)\varsigma_{it}$ | *t=1..T;* | (2.12) |
| $\varsigma_{it} \leq \beta_{it}$ | *i=1..N, t=1..T;* | (2.13) |
| $\varsigma_{it} \leq \alpha_{it}$ | *i=1..N, t=1..T;* | (2.14) |
| $y_{it} \leq \varsigma_{it}$ | *i=1..N, t=1..T;* | (2.15) |
| $\frac{x_{it}}{r_i} \leq (z_{it} - \varsigma_{it} + y_{it})G_t$ | *i=1..N, t=1..T;* | (2.16) |
| $\beta_{iT} = 0$ | *i=1..N;* | (2.17) |
| $x_{it}, I_{it}^+, I_{it}^- Idle_t, Late_t \geq 0$ | *t=1..T;* | (2.18) |
| $z_{it}, \beta_{it}, y_{it}, \zeta_{it} \in \{0, 1\}$ | *i=1..N, t=1..T;* | (2.19) |

**Parâmetros:**

| | | |
|---|---|---|
| $N$ | = | número de itens; |
| T | = | número de períodos do horizonte de planejamento; |
| $h_i^+$ | = | custo unitário de estoque do item $i$; |
| $h_i^-$ | = | custo unitário de atraso do item $i$; |
| $K_i$ | = | custo de preparação para produção do item $i$; |
| $d_{it}$ | = | demanda do item $i$ no período $t$; |
| $r_i$ | = | Taxa de produção do item $i$; |
| $s_i$ | = | tempo de preparação para produção do item $i$; |
| $\delta$ | = | número positivo pequeno; |
| $G_t$ | = | intervalo de tempo disponível para produção no período $t$; |

**Variáveis:**

| | | |
|---|---|---|
| $I_{it}^{+}$ | = | quantidade de itens do tipo *i* em estoque no final do período *t*; |
| $I_{it}^{-}$ | = | total da demanda não atendida do item *i* no final do período *t*; |
| $x_{it}$ | = | quantidade produzida do item *i* no período *t*; |
| $\alpha_{it}$ | = | número de preparações do item *i* que começam no período *t*; |
| $Idle_t$ | = | intervalo de tempo ocioso no período *t;* |
| $Late_t$ | = | intervalo de tempo em que o período *t* é ampliado; |
| $y_{it}$ | = | 1 caso o período *t* esteja operando em modo *return-product*, 0 caso contrário; |
| $\beta_{it}$ | = | 1 caso *i* seja o último item produzido no período *t*, 0 caso contrário; |
| $z_{it}$ | = | 1 se existe produção do item *i* é no período *t*, 0 caso contrário. |

A função objetivo (2.1) visa minimizar a soma dos custos de estoque, de atraso e de preparação para produção dos itens ao longo do horizonte de planejamento. As equações (2.2) asseguram que a demanda produzida do item *i* no período *t* acrescida de sua quantidade em estoque e da demanda que não é atendida no período *t* é igual à demanda do item *i* no período *t* mais a quantidade estocada acrescida da demanda do item que não foi atendida no período anterior (*t*-1). As restrições (2.3) garantem que apenas um item é o último a ser produzido no período *t.* As restrições (2.4) asseguram que $z_{it}$ = 1 se houver preparação para produção do item *i* no final do período *t*-1. Enquanto as restrições (2.5) impõem que a variável $z_{it}$ = 1 se houver preparação para produção do item *i* no final do período *t.* As restrições (2.6-2.9) permitem que um mesmo item seja produzido no início e no final do período se mais de um item for produzido no período (*return product*), ou seja, a variável $y_{it}$ é igual a 1. As restrições (2.6) garantem que só existirá *return product* se o item *i* é o primeiro a ser produzido no período *t.* Em (2.7) impõe-se que só pode ocorrer a volta a produção do item no final do período, ou seja, se $\beta_{it}$ = 1. Em (2.8) é assegurado que pelo menos um segundo item deve ser produzido no período. As restrições (2.9) impõe a condição de *return product* se o item *i* foi o primeiro a ser produzido no período *t*, é o último e existe a produção de outro item neste período. As restrições (2.10) impõem que para a produção de um dado item *i* a preparação seja contabilizada, ou seja, ou a preparação foi carregada do período anterior, ou ela será efetivada no período corrente.

O modelo proposto permite que uma preparação seja iniciada num dado período e concluída no período seguinte (*set-up crossover*). Para tanto os autores propuseram que os períodos do horizonte de planejamento tenham dimensões diferentes. Desta forma, a preparação é contabilizada num único período, no entanto, uma variável de atraso o excesso de tempo gasto para o próximo período. No modelo, as restrições (2.11-2.14) controlam os mecanismos de tempo num dado período, tamanhos de período não-uniformes (período de tempo modificado) e *set-up crossover.* Em (2.11), a soma dos tempos de produção, de preparação e de máquina parada (ociosidade) são iguais ao comprimento do período de tempo modificado pelo atraso do período anterior e pelo atraso do período atual. As restrições (2.12) estabelecem uma conexão entre o tempo de atraso e o tempo de preparação de um item que terá sua preparação concluída no período seguinte. As restrições (2.13) indicam que, se a preparação para a produção do item *i* está sendo preservada do período *t* para o período *t* + 1, então esta preparação é a preparação final do período *t.* As restrições (2.14) determinam que uma preparação deve começar em *t* para atravessar o limite de tempo do período *t* e continuar no período seguinte. Restrições (2.15) indicam que se no período *t* um item é produzido mais de uma vez, então a preparação do estado final deve continuar até o período seguinte, fazendo com que *i* seja o estado inicial do período *t* + 1. A produção do item *i* no período *t* é limitada pelas restrições (2.16). As restrições (2.17) impedem que ocorra preparação após o horizonte de planejamento e, por fim, as restrições (2.18) e (2.19) representam as condições de não-negatividade e domínio das variáveis.

**3. Método *Local Branching***

O PDLC pertence à área de otimização combinatória e é, na maioria das vezes, intratável de maneira exata em situações reais. Em Fischeti e Lodi (2003), os autores propuseram uma estratégia genérica com a finalidade de tratar problemas inteiros mistos. Esta estratégia, chamada de *local branching*, tem como base o princípio da busca local associado à ideia de separação (*branching*) do método *branch&bound.* A partir de uma solução factível determina-se uma vizinhança que permitirá escrever uma restrição de separação das soluções factíveis do espaço de solução em dois conjuntos: o primeiro composto pelas soluções que pertencem à vizinhança; e o segundo composto pelas demais soluções.

Como proposto em Fischeti e Lodi (2003), a partir do valor das variáveis binárias na solução factível inicial $\bar{y}$, defini-se um conjunto $\bar{S}$ chamado de suporte binário de $\bar{y}$, tal que:

$$\bar{S} = \{ j \in B \mid \bar{y} = 1 \}.$$

A partir do conjunto $\bar{S}$ e de um valor *k* define-se que toda solução com no máximo *k* variáveis binárias com valores diferentes dos valores destas variáveis na solução factível inicial pertence à vizinhança desta solução $N(\bar{y}, k)$. A restrição de *local branching* é definida por:

$$\sum_{j \in \bar{S}} (1 - y_j) + \sum_{j \in B \setminus \bar{S}} y_j \leq k \, .$$

Nesta restrição a primeira somatória contabiliza as variáveis binárias que terão seus valores alterados de 1 para 0, enquanto a segunda somatória contabiliza as variáveis que terão seus valores alterados de 0 para 1. Note que a soma das duas somatórias indica o número total de variáveis binárias que tiveram seu valor alterado. Logo, quanto maior o valor de *k* mais ampla a vizinhança considerada, no entanto, para valores pequenos de *k* a vizinhança é reduzida. A ideia é determinar um valor para *k* tal que a vizinhança seja suficientemente grande para conter soluções melhores que a solução atual e, ao mesmo tempo, suficientemente pequena para que o subproblema inteiro-misto resultante seja resolvido em tempo computacional aceitável.

A separação do problema original é determinada por um lado pelo problema original com a adição da restrição de *local branching* acima. Para obter o segundo subproblema acrescenta-se ao problema original a restrição complementar:

$$\sum_{j \in \bar{S}} (1 - y_j) + \sum_{j \in B \setminus \bar{S}} y_j \geq k + 1 \, .$$

Desta forma, a união das soluções factíveis dos dois subproblemas resulta no conjunto de soluções factíveis do problema original. Como para o método *branch&bound*, os subproblemas são resolvidos e novamente divididos.

O *local branching* é um método exato. Entretanto, o custo computacional para resolver os nós intermediários pode ser muito elevado, uma vez que em cada nó é resolvido um problema inteiro misto. Isto pode tornar o método computacionalmente inviável. Desta forma, Fischetti e Lodi (2003) propuseram algumas modificações para o método, tais como definir um limite de tempo para a solução dos nós (limite_de_sol_nó) ou incluir mecanismos de diversificaçãoo (por exemplo, Passo 6c do pseudocódigo a seguir). Assim, o local branching passa a se comportar como uma heurística, melhorando cada vez mais a solução atual e tentando se aproximar do valor ótimo.

Como estratégia para obter a primeira solução factível para o *local branching* e, por conseqüência, o primeiro limitante superior do problema, a proposta aqui apresentada foi obter a melhor solução para o PDLC limitando o tempo de execução do software de otimização CPLEX 11 em 50 segundos. A partir desta solução a estratégia *local branching*, como proposto por Fischetti e Lodi (2003) é executada. Em seus testes, os autores identificaram que uma boa faixa de valores para k seria um número no intervalo [10, 20]. Para o problema aqui estudado o valor adequado é 13, que foi definido a partir de testes computacionais.

Vale destacar que cada um dos subproblemas gerados é resolvido utilizando o software de otimização CPLEX 11.0 e que três critérios de parada foram adotados. O primeiro critério de parada é atingido se a solução ótima do subproblema foi encontrada. A segunda condição ocorre quando o problema é infactível. A terceira condição é ativada quando um tempo limite de execução pré-estabelecido é atingido.

**Método *Local Branching* para o *PDLCG***

Passo 1. Resolva o PDLC. Utilize a solução obtida ($Y^1$) como solução inicial para o *local branching*.

Passo 2. Faça $k = 13$; limite_de_tempo_sol_nó = 50 segundos; limite_tempo = 250 segundos – tempo Passo 1. Obtenha $\bar{S}$ para a solução $Y^1$.

Passo 3. Divida o problema em dois subproblemas, $M^1$ e $M^2$, onde $M^1$ representa o lado esquerdo da árvore *do local branching* e $M^2$ representa o lado direito.

- Restrição 1: $M^1$: $\sum_{j\in \bar{S}}(1-y_j^1)+\sum_{j\in B\setminus \bar{S}} y_j^1 \le k$
- Restrição 2: $M^2$: $\sum_{j\in \bar{S}}(1-y_j^1)+\sum_{j\in B\setminus \bar{S}} y_j^1 \ge k+1$

Passo 4. Faça o subproblema atual igual a $M^1$.

Passo 5. Resolva o subproblema atual respeitando o limite_de_tempo_sol_nó.

Passo 6. Avalie as seguintes possibilidades:

Passo 6a. Caso a solução ótima ($Y^2$) do subproblema atual seja encontrada dentro do limite de tempo estabelecido e seu valor seja melhor que o limitante superior atual, atualize a solução atual.
Remova a Restrição 1 do subproblema atual. Faça o problema atual igual ao subproblema $M^2$. Em seguida, adicione uma restrição do tipo Restrição 1 ao subproblema atual, porém utilizando $Y^2$ para definir $\bar{S}$. Adicione ao subproblema $M^2$ a Restrição 2 utilizando $Y^2$. **Volte ao Passo 5.**

Passo 6b. Caso o limite de tempo seja atingido e uma solução factível melhor do que a solução atual tenha sido encontrada ($Y^2$), substitua a Restrição 1 do subproblema atual por:

$$\sum_{j\in \bar{S}}(1-y_j^2)+\sum_{j\in B\setminus \bar{S}} y_j^2 \le k$$

**Volte ao Passo 5**.

Passo 6c. Caso o limite de tempo seja atingido e nenhuma solução factível tenha sido encontrada, substitua a Restrição 1 do subproblema atual por:

$$\sum_{j\in \bar{S}}(1-y_j^1)+\sum_{j\in B\setminus \bar{S}} y_j^1 \le k/2$$

**Volte ao Passo 5**.

Passo 6d. Caso o subproblema atual seja infactível faça:
Passo 6d1. Se $M^2 \neq \emptyset$, faça o subproblema atual igual a $M^2$ e $M^2 = \emptyset$.
**Volte ao Passo 5.**
Passo 6d2. **PARE.**

## 4. Resultados e Discussão

Nesta seção são apresentados os resultados computacionais para o problema de dimensionamento de lotes detalhado na seção anterior. Dois pontos são analisados: a qualidade das soluções obtidas utilizando o método *local branching*; e a influência das características consideradas (possibilidade de *backlogging*, *carry-over* e *set-up crossover*) no custo dos planos de produção.

Para avaliar a qualidade das soluções obtidas pelo método *local branching*, o valor destas soluções é comparado com o valor das soluções obtidas utilizando o software comercial de otimização CPLEX 11.

A influência nos custos ao considerar a possibilidade de *backlogging, carry-over e set-up crossover* para os problemas foi avaliada em etapas. Para tanto foi incluída uma caracaterística por vez e os resultados obtidos foram comparados ao PDLC.

O método *local branching* foi codificado em linguagem C++ e os subproblemas inteiros mistos foram resolvidos utilizando bibliotecas do software de otimização CPLEX. Todos os testes computacionais foram realizados em um microcomputador Core 2 Duo 2.33 Ghz, 3GB de memória RAM e sistema operacional Linux Debian 4. O tempo computacional dedicado à resolução de cada um dos problemas teste limitado a 250 segundos.

Foram adotados como problemas testes (instâncias), o benchmark proposto por Trigeiro *et al.* (1989) acrescentando aos problemas, quando necessário, custos para atender a demanda com atraso. Foram propostas pelos autores 751 instâncias, cujo número de itens varia de 6 à 30 e o número de períodos é igual a 15, 20 ou 30. Os tempos de preparação foram gerados uniformemente no intervalo de 5 a 150. Os custos de preparação foram gerados no intervalo de 200 a 1500, enquanto os custos de estoque foram gerados no intervalo de 0,8 a 5,0. Para os problemas que consideram atraso na demanda, o atraso foi calculado multiplicando-se o custo de estoque por 1,5.

O método *local branching* obteve para 93% das instâncias solução de igual ou melhor qualidade que o software de otimização CPLEX 11.0. Logo, pode-se afirmar que a estratégia de *local branching* é eficiente para a solução do problema de dimensionamento de lotes aqui estudado. Na Tabela 1, os resultados são apresentados de forma mais detalhada, em que as instâncias são dividas em GRUPOS como proposto em (Trigeiro *et al.*, 1989).

| Grupos | CPLEX | *Local Branching* |
|---|---|---|
| EW | 43% | 100% |
| F | 26% | 100% |
| G | 23% | 96% |
| X | 55% | 91% |

Tabela 4.1 – Porcentagem de soluções de melhor qualidade.

Analisando o problema quanto às características, percebe-se que o PDL com possibilidade de atraso na demanda apresenta resultados iguais ou melhores que o PDLC com praticamente o mesmo custo computacional. Ao ser acrescentada a possibilidade de preservação de preparação, o tempo computacional necessário para solução dos problemas aumenta, fazendo com que todas as instâncias tivessem sua execução interrompida ao atingir o limite de tempo, indicando que as soluções obtidas não foram comprovadamente ótimas. Entretanto, os custos dos melhores planos de produção encontrados são menores do que os planos ótimos obtidos ao resolver o PDLC. Esta melhoria se acentua quando as possibilidades de atraso e de preservação de preparação são consideradas simultaneamente. Porém, a combinação de características que resultou nos melhores resultados para a maioria das instâncias foi o PDLCG. Considerar estas características em conjunto resultou em melhorias consideráveis no custo total da solução, em média a economia é igual a 14%, no entanto, para alguns casos foi obtida uma redução de aproximadamente 63% no valor da função objetivo. Estes resultados são detalhados na Tabela 4.2. Vale destacar que a solução inicial (Passo 1 do pseudocódigo) equivale a solução do PDLC, seu valor foi omitido, pois os resultados da Tabela 4.2, mostram a melhoria em relação ao PDLC, logo a melhoria em relação a solução inicial.

| Grupos | % média de ganho | % mínima de ganho | % máxima de ganho |
|---|---|---|---|
| EW | 23% | 16% | 32% |
| F | 27% | 18% | 63% |
| G | 22% | 2% | 31% |
| X | 10% | 2% | 30% |

Tabela 4.2 - Grau de melhoria das soluções obtidas pelo PDLCG em relação ao PDLC.

## 5. Conclusão

Este trabalho tem como objeto de estudo o problema de dimensionamento de lotes com restrições de capacidade. O PDLC abordado contém múltiplos itens e tempos e custos de preparação para produção. Três características adicionais são consideradas: 1) possibilidade de atraso para atender a demanda (*backlogging*); 2) possibilidade de manter a preparação para produção de um item de um período para o seguinte (*carry-over*); 3) possibilidade de iniciar uma preparação para produção de um item num período e concluí-la no período seguinte (*set-up crossover*).

O método *local branching* proposta para a resolução do PDLCG mostrou-se eficiente quando comparado ao software de otimização CPLEX 11 (IBM ILOG). Para os problemas resolvidos, o método obteve solução melhor que o CPLEX para a maioria dos exemplos (ver Tabela 4.1)..

Quanto às características adicionais consideradas concluí-se que estas proporcionam economia significativa de custos do plano de produção (ver Tabela 4.2). Logo, para os ambientes de produção em que é possível atender a demanda com atraso e preservar a preparação para produção de um período para o período seguinte ou ainda iniciar a preparação para produção de um item em um período para concluí-la no período seguinte, é economicamente importante incluir estas possibilidades na concepção do planejamento da produção.

## Agradecimentos

Os autores agradecem à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) e ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro para o desenvolvimento desta pesquisa.

## Referências

Bahl, H. C., L. P. Ritzman, e J. N. D. Gupta (1987) "Determining lot sizes and resources requirements: a review". *Operations Research*, v. 35, p. 329-345.

Buschkühl, L., F. Sahling, S. Helber e H. Tempelmeier (2010) "Dynamic capacitated lot-sizing problems: a classification and review of solution approaches". *OR Spectrum*, 32, p.231–261.

Fischetti, M. e A. Lodi (2003) "Local branching", *Mathematics Programming*, ser. B 98, p. 23-47.

IBM ILOG (2010) *Solver CPLEX*, www-01.ibm.com/software/integration/optimization/cplex-optimizer/, acesso em 20.05.2010.

Maes J., McClain, J. O. e Van Wassenhove, L. N. (1991) "Multilevel Capacitated Lotsizing Complexity and LP Based Heuristic", *European Journal of Operational Research* 53, 131-148.

Pochet Y. e L.A. Wolsey (2006) Production planning by mixed integer programming, Springer.

Sung, C. e C. T. Maravelias (2008), "A mixed-integer programming formulation for the general capacitated lot-sizing problem", *Computers and Chemical Engineering*, v. 32, p. 244-259.

Trigeiro, W. W., L. J. Thomas, J. O. McClain (1989) "Capacitated Lot Sizing With Setup Times", *Management Science*, v. 35, p. 353-366.

Wolsey, L.A. (1995) "Progress with single-item lot-sizing", *European Journal of Operational Research* (1995), v. 86, p. 395-401.