Universidade Estadual de Campinas

Instituto de Matemática, Estatística e Computação Científica

Departamento de Matemática Aplicada

Uma introdução a problemas de dimensionamento de lotes

MS777 - Projeto Supervisionado I

Aluno: Angelo Gabriel Nicolau Dotta (RA 148338)

Orientadora: Dra Kelly Cristina Poldi

Campinas, 27 de novembro de 2015.

Sumário

Resumo

No Problema de Dimensionamento de Lotes (PDL) estamos interessados na organização dos processos produtivos visando minimizar os custos com a produção. Este problema é encontrado em indústrias dos mais variados setores, como por exemplo, têxteis, de bebidas, entre outras. Podemos descrevê-lo como decidir quando e quanto produzir, com o objetivo minimizar custos associados ao processo produtivo e considerando o atendimento da demanda por produtos finais. Consideramos como custos relacionados ao processo, o custo de produção, de estoque e de *setup* (preparação de máquinas). Este trabalho apresenta definições e modelos matemáticos para alguns problemas de dimensionamento de lotes. Métodos de solução para o PDL também foram estudados e são apresentados nesse trabalho.

Palavra-chaves: problema de dimensionamento de lotes, planejamento da produção, modelagem matemática.

Abstract

Lot Sizing Problems (LSP) are interested in the organization of production processes to minimize the cost of production. Such problems can be found in various sectors of industry, such as textiles, soft-drinks, among others. A Lot Sizing Problem can be described as the problem of deciding when and how much to produce in order to minimize costs associated with the production process, taking into account a demand to be fulfilled. The costs related to the process are the cost of production, inventory and setup. This study presents definitions and mathematical models for some lot-sizing problems. Solution methods for the LSP were also studied and presented in this work.

Key-words: lot sizing problem, production planning, mathematical modeling.

1 Introdução

O Problema de Dimensionamento de Lotes (PDL) foi formulado para auxiliar no planejamento da produção. Seguindo essa ideia, estamos interessados em determinar quando e quanto produzir minimizando os custos relacionados ao processo produtivo.

Os primeiros estudos foram feitos por Wagner e Whitin em 1913 e publicados somente em 1958. Eles estavam interessados em atender uma encomenda de maneira a satisfazer algumas exigências - modeladas como veremos adiante pelas restrições do problema. A solução que eles propuseram baseava-se em uma solução recursiva que funcionava bem, e ainda funciona, para o caso particular que consideraram.

Após esse estudos, os conhecimentos e as pesquisas no assunto não avançaram muito até o final da década de 80. Explicamos isto, supostamente, pela complexibilidade de encontrar uma solução otimizada. Esse bloqueio foi parcialmente eliminado com os avanços do computador, que ocorreram neste período, e que cada vez mais foi capaz de executar algoritmos sofisticados.

Modelando um problema encontrado na produção de uma indústria e o resolvendo de forma a minimizar os custos relacionados, podemos aproveitar melhor os recursos de uma empresa. Este problema é um problema clássico de Pesquisa Operacional, amplamente estudados nas referências apresentadas. Pode ser encontrado em várias indústrias dos mais variados setores, como por exemplo: indústria de bebida, de alimentos, química, de metalurgia entre outras.

2 Descrição do problema

O estudo do Problema de Dimensionamento de Lotes (PDL) é uma importante ferramenta para o planejamento da produção. Com o auxílio desse estudo pode-se obter resultados capazes de amparar a tomada de decisão, respondendo às perguntas: quando e quanto produzir. A quantidade de itens a ser produzida em um intervalo de tempo é chamada de lote, daí o nome do problema.

Atualmente as organizações desejam ampliar seus lucros dentre outros objetivos. Uma das formas de alcançar esse objetivo é reduzir custos, quaisquer que sejam eles. Utilizar seus recursos de forma otimizada é um importante passo para as empresas que desejam ocupar um lugar de destaque no mercado e manter-se vivas na economia mundial.

Se uma indústria consegue controlar melhor seus custos de produção, consequentemente poderá vender o produto mais barato que seus concorrentes, tornando-se mais competitiva, ou ainda, mesmo que vender os produtos por preços equivalentes, o lucro será maior, levando em conta que os custo dos produtos finais foram reduzidos.

Um empreendedor tem ciência que todas suas decisões possuem certas incertezas e riscos. Tomar essas decisões com auxílio de modelos matemáticos que comprovam otimalidade, ou então, aproximações dela, torna a tarefa de decisão um pouco menos complicada e arriscada.

No PDL estamos interessados em reduzir os custos associados ao processo produtivo. Dentre as despesas com a manufatura de uma indústria, podemos citar os gastos com a produção em si (matéria-prima, mão-de-obra, recursos em geral), os gastos com preparação de máquinas e equipamentos, denominado *setup*, e os custos de estoque.

Segundo Araújo e Rangel [1] os primeiros estudos do PDL foram propostos em 1913 e publicados em *Economic Order Quantily* (EOQ) por Ford Whitman Harris. Neste trabalho os autores abordaram um PDL não capacitado, contínuo com horizonte de planejamento infinito, único-item e, a partir deste, muitas variantes do problema foram modelados e resolvido. Por exemplo, podemos ver uma revisão de problemas de dimensionamento de lotes em [2] ou ainda os problemas capacitados estudados em [5] e [10]. Uma revisão e um estudo mais atual também podem serem encontradas em [8].

3 Classificações do problema

Nesta seção apresentamos de maneira sucinta algumas variantes encontradas no problema de dimensionamento de lotes. A seguir apresentamos um quadro comparativo com as derivações do PDL.

Itens	**Estágios**	**Capacidade**	**Horizonte**	**Demanda**
Único-Item	Multiestágio	Capacitado	Finito	Estática
				Dinâmica
Multi-Itens	Monoestágio	Não Capacitado	Infinito	Determinística
				Estocástica

Tabela 1: Quadro descritivo dos tipos de PDL.

Uma primeira classificação para o PDL está relacionada a quantidade de itens a ser produzida. Algumas empresas trabalham com portifólio restrito, com apenas um item e outras são mais abrangentes. Dessa maneira o problema pode ser classificado como **Único-Item** ou **Multi-Item**.

Alguns produtos necessitam de componentes ou processamentos prévios, neste caso, temos então uma demanda dependente de demandas anteriores o que caracteriza um PDL **Multiestágio**. No PDL **Monoestágio**, temos demandas independentes.

O problema é dito **Capacitado** quando há limitação na capacidade de algum recurso envolvido na modelagem, por exemplo, limitação na disponibilidade de mão-de-obra, de matéria-prima ou até mesmo na capacidade produtiva da(s) máquina(s). Na realidade toda modelagem, próxima da realidade, deveria ser capacitada em vista que os recursos são limitados na indústria. Entretanto, os primeiros métodos de solução não foram desenvolvidos tratando este caso. Antes de tratar o problema real, é necessário tratar um subproblema simplificado. Quando estamos no caso mais distante da realidade, onde temos recusos infinitos, estamos tratando do problema **Não Capacitado**. Restrições de capacidade limitam a produção de itens em cada período de tempo.

Chamamos de horizonte de planejamento o intervalo de tempo que está sendo considerado para o planejamento da produção. Pode ser **Finito** ou **Infinito**.

Um bom planejamento estratégico para a produção deve ser dividido em três estágios: planejamento a longo, médio e curto prazos. A partir daí vem a escolha do horizonte de planejamento que será trabalhado em cada problematização. Podemos encontrar mais detalhes de cada em desses estágio em [8].

Determinado o horizonte de planejamento, temos que produzir para atender uma demanda. A demanda pode ser **Estática**, quando é fixa independente do tempo; **Dinâmica**, quando é alterada para intervalos de tempos distintos; **Determinística**, quando sabemos os valores exatamente; **Estocástica**, quando não é conhecida exatamente e aí supomos que segue uma distribuição de probabilidade. Algumas modelagens podem admitir atrasos na entrega da demanda (*backlogging*).

Permitir atrasos na demanda (*backlogging*), a princípio, não parece muito lucrativo. Deixar de atender a demanda implica deixar de vender, o que leva a redução da receita bruta. Entretanto, observamos que às vezes se torna vantajoso, permitir atrasos.

Por exemplo, vamos supor que quando uma máquina é ligada ela tenha que produzir 100 itens no mínimo de uma só vez. Vamos supor também que estamos com um estoque quase suficiente para atender a demanda no período que estamos, apenas faltando 2 itens para cumpri-la de forma integral. Consideramos também que nos próximos períodos, a demanda é nula. Neste caso, não parece muito lucrativo para a indústria ter que desembolsar o *setup* desta máquina, sabendo que os 98 itens restantes irão para o estoque, onde podem perder valor. Podemos também ter um tanque, que a capacidade total seja 1000 litros. É mais compensador utilizar sua capacidade total do que apenas fabricar a quantidade que precisamos, visto que a preparação do tanque terá que ser feita independentemente se vamos prozuir 1 ou 1000 litros do material em questão.

Outras diferenças para o problema são possíveis de serem encontradas. Por exemplo, o produto final pode perder valor a medida que permanece no estoque, sendo então deteriorado. É possível que o custo de *setup* (preparação das máquinas) seja dependente das decisões anteriores. A medida que tentamos fazer com que o modelo matemático seja cada vez mais próximo da situação real, mais variáveis e restrições são consideradas no modelo, o que aumenta a dificuldade para solução. Para fins de estudos e familiarização com o problema, consideramos o problema mais simplificado, e então aperfeiçoamos.

Outra característica do PDL que até então não foi discutida, é o tempo de produção. O tempo gasto, segundo alguns autores, já está incorporado na modelagem do problema. Outros dizem que o fator mais importante não é o custo de produção em si, mas sim, o tempo gasto.

Segundo [10] a relevância de uma modelagem que considera o tempo é essencial, para um sistema de administração da produção que considera este como o principal pilar de decisão no processo produtivo, o *just-in-time*. Nele, o importante é produzir na hora certa, evitando custos de estoque. Essa modelagem foi amplamente estudada nesse artigo e também podemos encontrar uma modelagem em [1].

É relevante salientar que o PDL pode ser uma combinação entre uma classificações de cada coluna da Tabela 1. Dessa forma, por exemplo, um PDL único-item, pode ser multiestágio ou monoestágio; capacitado ou não; com horizonte infinito ou pré-determinado; e ainda ter demanda estática, dinâmica, deterministica ou estocástica.

4 Modelagens para o problema

A seguir apresentamos algumas modelagens matemáticas para o PDL de acordo com suas especificações. O objetivo em cada modelagem é basicamente o mesmo, consiste em determinar quanto e quando produzir, planejando a produção com custo mínimo, de acordo com o horizonte de planejamento adotado. Podemos observar na Tabela 1 da Seção 3 um quadro descritivo das diversidades para o problema. Os detalhes das diferentes classificações para o PDL também estão descritas na seção anterior.

4.1 PDL não capacitado

O PDL não capacitado foi o primeiro modelo proposto e resolvido em [11] em 1958. Embora estamos considerando que as máquinas não apresentam nenhuma restrição de capacidade, que a mão-de-obra, matéria-prima e todos os recursos de produção estão disponíves sem restrição, o que de fato não acontece, este modelo teve papel fundamental para maior estudo do assunto, pois a partir dele foi desenvolvido o Método de Wagner Whitin que será discutido adiante. Foi o primeiro método de solução para um PDL.

Considere o seguinte índice:

- $t, t = 1, \cdots, T$: para os períodos de tempo em um horizonte de planejamento finito T.

Definimos os seguintes parâmetros:

- s_t : custo de preparação para a produção no período t;
- c_t : custo unitário de produção (*setup*) no período t;
- h_t : custo unitário de estoque no período t;
- d_t : demanda no período t.

Estamos interessados em decidir o valor das seguintes variáveis:

- x_t : quantidade produzida no período t;
- I_t : estoque no final do período t (sem perda de generalidade, podemos supor o estoque inicial e final iguais a zero, ou seja, $I_0 = I_T = 0$).
- y_t : variável binária que assume o valor 1 se o item for produzido no período t e 0, caso contrário;

A seguir, apresentamos o modelo matemático para o PDL não capacitado:

$$
\begin{aligned}
\text{minimizar} \quad & \sum_{t=1}^{T} (c_t x_t + h_t I_t + s_t y_t) & & (1)\\
\text{sujeito a:} \quad & x_t + I_{t-1} - I_t = d_t, & \forall t \quad & (2)\\
& x_t \geq 0, I_t \geq 0, y_t \in \{0, 1\}, & \forall t \quad & (3)\\
& I_0 = 0, I_T = 0. & & (4)
\end{aligned}
$$

A função objetivo (1) considera a soma dos custos de preparação, de produção e de estoque. O primeiro conjunto de restrições (2) garantem que a demanda em cada período t seja atendida, conhecidas como restrições de balanceamento entre estoque e produção. O atendimento da demanda, sem atraso, é indicado pela não negatividade das variáveis I_t nas restrições (3), que também garantem que a quantidade produzida (x_t) é não negativa. Sem perda de generalidade, podemos supor o estoque inicial e final iguais a zero, ou seja, $I_0 = I_T = 0$.

4.2 PDL monoestágio único-item

Na modelagem seguinte estamos considerando o problema de dimensionamento de lotes monoestágio, único-item, com horizonte de planejamento finito, com demanda conhecida e que deve ser atendida sem atrasos. Essa modelagem é muito similar ao Problema clássico de Dimensionamento de Lotes Capacitado (PDLC), que será apresentado na proxima seção. A única diferença, é que este segundo é modelado para uma indústria que trabalha com vários itens.

Para determinação de um modelo para o PDL monoestágio único-item, considere o seguinte índice:

- $t, t = 1, \cdots, T$: para os períodos de tempo em um horizonte de planejamento finito T.

Definimos os seguintes parâmetros:

- s_t : custo de preparação para a produção no período t;
- c_t : custo unitário de produção (*setup*) no período t;
- h_t : custo unitário de estoque no período t;
- d_t : demanda no período t;
- M : um limitante superior.

Estamos interessados em decidir o valor das seguintes variáveis:

- x_t : quantidade produzida no período t;
- y_t : variável binária que assume o valor 1 se o item for produzido no período t e 0, caso contrário;

- I_t : estoque no final do período t (sem perda de generalidade, $I_0 = I_T = 0$, ou seja, podemos considerar o estoque inicial e final nulos).

A seguir, apresentamos o modelo matemático para o PDL monoestágio único-item:

$$\begin{aligned}
\text{minimizar} \quad & \sum_{t=1}^{T} (s_t y_t + c_t x_t + h_t I_t) && && (5)\\
\text{sujeito a:} \quad & x_t + I_{t-1} - I_t = d_t, && \forall t && (6)\\
& x_t \leq M y_t, && \forall t && (7)\\
& y_t \in \{0,1\}, x_t \geq 0, I_t \geq 0, && \forall t && (8)\\
& I_0 = 0, I_T = 0. && && (9)
\end{aligned}$$

A função objetivo (5) considera a soma dos custos de preparação, de produção e de estoque. O primeiro conjunto de restrições (6) garantem que a demanda em cada período t seja atendida. As restrições em (7) são as restrições de preparo (ou restrições de *setup*), ou seja, só pode haver produção se a linha de produção estiver preparada. O atendimento da demanda, sem atraso, é indicado pela não negatividade das variáveis I_t nas restrições (8), que também garantem que a quantidade produzida (x_t) é não negativa. Sem perda de generalidade, podemos supor os estoques inicial e final iguais a zero, ou seja, $I_0 = I_T = 0$.

4.3 PDL capacitado

A modelagem apresentada a seguir considera o problema capacitado, com demandas independentes de demandas prévias, caracterizando ainda um PDL monoestágio. As diferenças essenciais deste problema para a modelagem do problema apresentado na Seção 5.1, está baseada como o próprio nome sugere nos limitantes de capacidade que o problema está nos impondo. Outra diferença é quantidade de itens que iremos determinar a produção, neste caso não estamos trabalhando com apenas um único item. O problema na sua forma capacitada pode ser encontrado em [10].

Além disso vamos considerar o tempo de preparação de máquinas e de produção com o intuito de deixar o problema mais próximo da realidade industrial. Seguindo este mesmo aspecto, é interessante considerar que a demanda, bem como o custo de produção do item A para o item B, por exemplo, podem variar ou não, de acordo com o tempo que estamos considerando.

Considere os seguintes índices:

- $i, i = 1, \cdots, n$: para os itens;

- $t, t = 1, \cdots, T$: para os períodos de tempo em um horizonte de planejamento finito T.

Definimos os seguintes parâmetros:

- s_t : custo de preparação (*setup*) no período t;
- c_i : custo unitário de produção do item i;
- h_i : custo unitário de estoque do item i;
- d_{it} : demanda do item i no período t;
- b_i : tempo de processamento do item i;
- f_i : tempo de preparação (*setup*) para produção do item i;
- C_t : capacidade de produção em cada período t, em unidades de tempo;
- M : um limitante superior para x_{it}.

Estamos interessados em decidir o valor das seguintes variáveis:

- x_{it} : quantidade produzida do item i no período t;
- y_{it} : variável binária que assume o valor 1 se o item i for produzido no período t e 0, caso contrário;
- I_{it} : estoque do item i no final do período t (sem perda de generalidade, $I_{i0} = I_{iT} = 0$).

A seguir, apresentamos o modelo matemático para o PDL capacitado (PDLC):

$$
\begin{array}{lllr}
\text{minimizar} & \displaystyle\sum_{t=1}^{T}\sum_{i=1}^{n}(s_{it}y_{it} + c_i x_{it} + h_i I_{it}) & & (10) \\
\text{sujeito a:} & x_{it} + I_{i,t-1} - I_{it} = d_{it}, & \forall i, t & (11) \\
& \displaystyle\sum_{t=1}^{n}(b_i x_{it} + f_i y_{it}) \leq C_t, & \forall t & (12) \\
& x_{it} \leq M y_{it}, & \forall i, t & (13) \\
& y_{it} \in \{0, 1\}, x_{it} \geq 0, I_{it} \geq 0, & \forall i, t. & (13) \\
& I_{i0} = I_{iT} = 0. & & (14)
\end{array}
$$

A função objetivo (10) considera a soma dos custos de preparação, de produção e de estoque. O primeiro conjunto de restrições (11) garante que a demanda de cada item i em cada período t seja atendida, conhecidas como restrições de balanceamento entre estoque e produção. O conjunto de restrições (12) garante que a capacidade disponível para produção é limitada e compartilhada entre todos os itens. As restrições em (13) são as restrições de preparo (ou restrições de *setup*), ou seja, só pode haver produção se a linha de produção estiver preparada. O atendimento da demanda, sem atraso, é indicado pela não negatividade das variáveis I_{it} nas restrições (14), que também garantem que a quantidade produzida (x_{it}) é não negativa. Sem perda de generalidade, podemos supor o estoque inicial e final iguais a zero, ou seja, $I_{i0} = I_{iT} = 0$: restrição (15). Como podemos observar, o PDL capacitado é uma generalização para o PDL monoestágio único-item.

4.4 PDL multiestágio

O que caracteriza um PDL multiestágio é a dependência de processos anteriores. Para exemplificar este problema, vamos considerar que o item final que estamos interessados em produzir são garrafas rotuladas de refrigerante. Cada garrafa após ser produzida deve ser imediatamente rotulada. Dessa forma temos que produzir as garrafas e os rótulos. Sendo assim, não devemos produzir rótulos, se as garrafas já não foram previamente produzidas. Para modelar este problema, usaremos uma variável binária de decisão de maneira similiar ao que fazemos para modelar o custo de *setup*.

Seguindo a formulação proposta em [4], o problema em questão pode ser modelado como segue abaixo. Com esta subseção em geral, estamos interessados apenas em apresentar o problema. Uma discussão mais aprofundada sobre este tipo de dimensionamento de lotes (multiestágio) pode ser encontrado em [4].

Considere os seguintes índices:

- $i, i = 1, \cdots, n$: para os itens;
- $t, t = 1, \cdots, T$: para os períodos de tempo em um horizonte de planejamento finito T.

Definimos os seguintes parâmetros:

- s_{it} : custo de preparação (*setup*) para a produção do item i no período t;
- c_{it} : custo unitário de produção do item i no período t;
- h_{it} : custo unitário de estoque do item i no período t;
- d_{it} : demanda do item i no período t;

- r_{ij} : quantidade necessária de itens do tipo i para fabricar uma unidade do item j;
- b_{it} : tempo de processamento do item i no período t;
- C_{it} : capacidade de produção em cada período t, em unidades de tempo, para cada item i.

Definimos também $S(i)$ como o conjunto de todos os itens i, tais que dependam da pré-fabricação de outros itens para o seu processamento.

Estamos interessados em decidir o valor das seguintes variáveis:

- x_{it} : quantidade produzida do item i no período t;
- y_{it} : variável binária que assume o valor 1 se o item i for produzido no período t e 0, caso contrário;
- I_{it} : estoque do item i no final do período t (sem perda de generalidade, $I_{i0} = I_{iT} = 0$).

A seguir, apresentamos o modelo matemático para o PDL multiestágio:

$$
\begin{array}{lll}
\text{minimizar} & \displaystyle\sum_{t=1}^{T}\sum_{i=1}^{n}(s_{it}y_{it} + c_{it}x_{it} + h_{it}I_{it}) & (16) \\
\text{sujeito a:} & x_{it} + I_{i,t-1} - I_{it} = d_{it} + \displaystyle\sum_{j \in S(i)} r_{it}x_{it}, \quad \forall i,t & (17) \\
& \displaystyle\sum_{t=1}^{n}(b_{it}x_{it} + s_{it}y_{it}) \leq C_t, \quad \forall t & (18) \\
& x_{it} \leq i_{it}y_{it}, \quad \forall i,t & (19) \\
& y_{it} \in \{0,1\}, x_{it} \geq 0, I_{it} \geq 0, \quad \forall i,t. & (20) \\
& I_{i0} = I_{iT} = 0. & (21)
\end{array}
$$

Uma observação pertinente que podemos fazer, é que se cada item não depende da fabricação de itens anteriores, temos $r_{ij} = 0, \forall i,j$, e ainda $S(i) = \{\}$, fazendo com que o somatório da restrição (17) seja igual a 0. Logo a restrição de balanceamento de estoque, e o problema em geral ficam bem similares ao problema monoestágio capacitado descrito na Seção 4.3, como era de se esperar, uma vez que estamos trabalhando com demandas independentes.

5 Métodos de solução

Vimos até então que os problemas vão aumentando suas restrições de acordo com as necessidades de se chegar o mais próximo da situação real. Conforme fomos permitindo na modelagem para os PDLs aumento de itens e dependência de processos anteriores, podemos enxergar que os problemas estão intimamente relacionados, a ponto de generalizá-los.

Se é possivel uma generalização para o problema, ou seja, escrevê-lo de forma mais abrangente, a pergunta mais esperada que fazemos é: Por que não familiarizar com o PDL generalizado e estudar um único método capaz de encontrar uma solução factível com boa aproximação? A resposta é simples. Pelas peculiaridades de cada problema. Por exemplo, alguns problemas são mais rápidos e melhores resolvidos por um método específico. Ou as vezes não é necessário ter um custo computacional[1] muito grande para um problema que seria facilmente resolvido usando heurística, por exemplo.

5.1 Método de Wagner Whitin

Este foi o primeiro método proposto, em 1958, para a solução de um problema de dimensionamento de lotes. Trata o PDL com único-item e sem restrição de capacidade. Chamado de Método WW, esse algoritmo recursivo descreve os passos para encontrar a solução de forma dinâmica para a formulação dada a seguir. Consideramos as definições de parâmetros, variáveis e índices como descrita na Seção 4.1.

$$\begin{aligned}
\text{minimizar} \quad & \sum_{t=1}^{T}(s_t y_t + c_t x_t + h_t I_t) && (22)\\
\text{sujeito a:} \quad & x_t + I_{t-1} - I_t = d_t, \qquad \forall t && (23)\\
& y_t \in \{0,1\}, x_t \geq 0, I_t \geq 0, \qquad \forall t. && (24)
\end{aligned}$$

Segundo [1], o método WW para encontrar solução para o problema (22)-(24) está baseado na Propriedade 6.1 sobre as condições ótimas do PDL.

Propriedade 6.1 *A solução ótima do problema de dimensionamento de lotes único-item sem capacidade satisfaz* $I_{t-1}\ x_t = 0$ *para* $t=1,\ldots,T$.

Esta propriedade está nos dizendo que a produção, em cada período, será a quantidade total necessária para satisfazer a demanda no exato momento, ou então não deve-se produzir nada no período que estamos, pois o estoque do periodo anterior é suficiente para satisfazer a demanda. Com outras palavras, a produção em cada período será zero ou a soma das demandas futuras (de todo o horizonte de planejamento ou não).

Observamos que é fundamental o problema não ser capacitado e não existir custo de *setup*, uma vez que a Propriedade 6.1 é responsável por satisfazer a única restrição existente no

[1] Ententa-se por custo computacional a dificuldade do método retornar a solução, bem como o número de iterações e o tempo de execução.

problema.

A ideia na qual o método se baseia segue uma lógica simples. Como não temos custo de preparação de máquina, não tem custo adicional se produzir em todos os períodos ou não. Dessa forma estamos balanceando apenas o custo de produção e estoque em cada período explicito na nossa função objetivo.

Nosso problema então, é decidir o valor de x_t para todo período período t. Para resolver isto Wagner Whitin propõem uma técnica de programação dinâmica, onde obtemos a solução ótima usando a equação recursiva (singular) dada a seguir proposta em [11] e reformulada em [1].

$$\mathrm{f}_t = min_{j>t}(C_{ij} + f_j) \qquad \forall t \in \{1, 2, ..., T\}, j \in \{2, 3, ..., T+1\}$$

Onde fazemos as seguintes definições:

- f_t é o custo mínimo para o período de planejamento t. O que devemos fazer é encontrar o período j, tal que esse mínimo ocorra, $\forall j > t$.

- $f_{T+1} := 0$;

- $C_{tj} = s_t + h_t d_{t+1} + (h_t + h_{t+1}) d_{t+2} + (h_t + h_{t+1} + h_{t+2}) d_{t+3} + ... + (h_t + h_{t+1} + ... + h_{t-2}) d_{t-1}$;

Dado os vetores que representam o custo unitário, o custo de *setup* e o custo de estoque, podemos realizar o procedimento descrito acima para encontrar os períodos que teremos produção.

Passo 1: Calcular C_{tj}, $\forall j > t$;

Passo 2: Definir $f_{T+1} := 0$;

Passo 3: Calcular f_T e verificar em qual período j o mínimo aconteceu;

Passo 4: Calcular f_{T-1} e ir diminuindo, até obter f_1, sempre armazenando o $j-$ésimo período que o mínimo ocorreu em cada f_t;

Passo 5: Teremos os valores de $f_t, \forall t$, e os j$-$ésimos períodos correspondentes ao mínimo de cada um. Basta então decidir quando vamos produzir somente para a demanda do período correspondente, ou para o período mais demandas futuras e armazenar.

5.2 Heurísticas

Heurísticas são algoritmos utilizados para obtenção de soluções factíveis quando se tem grande dificuldade computacional. Alguns métodos para iniciar suas iterações precisam de aproximações iniciais, para a solução procurada, ou de limitantes (superiores e inferiores), e a partir daí melhorar a solução. Visto isso, as heurísticas também podem ser usadas para esta finalidade.

Na seção anterior, discutimos um método que pode ser usado apenas para problema de dimensinamento de lotes não capacitado. Agora, vamos apresentar algumas ideias de procedimentos heurísticos para resolver PDLs monoestágio único-item, cuja modelagem foi descrita na Seção 4.2.

O uso de heurísticas não nos garante que chegaremos em um resultado ótimo, apenas nos garante que encontraremos uma solução factível. Apresentamos a seguir as percepções principais de duas heurísticas de um PDL monoestágio único-item. Essas heurísticas também podem ser enontradas em [1].

5.2.1 Heurística lote-por-lote (LL)

Como o próprio nome sugere, nesta heurística estamos interessados apenas em atender a demanda no período que estamos produzindo. Desta forma, sempre que tivermos uma demanda, teremos produção e o estoque será nulo. Produzimos x itens no período t_0, a demanda neste período será exatamente x. Não parece uma heurística com qualidade de solução muito conveniente, visto que teremos produção em todos os períodos. Trabalhar desconsiderando a existência do estoque, faz com que perdemos a ideia principal do PDL. Poder usar produtos estocados, ou adiantar a produção para atender demandas futuras. Como temos dito, as heurísticas não nos garantem otimalidade, porém podem nos fornecer limitantes. A qualidade de cada uma deve ser estudada considerando as especificações do problema que está sendo resolvido.

5.2.2 Heurística de balanço por partes (BP)

A noção que esta heurística nos sugere é um balanceamento entre os custos de produção e custo de estocagem. Ou seja, consiste em verificar se no período que estamos atualmente, vale a pena produzir para os períodos subsequentes. Se sim, até qual período vale a pena adiantar a demanda.

Isso é uma ideia inteligente, pois às vezes vale a pena produzir, sem perda de generalidade, em t_1, a demandas requisitadas em t_1, t_2 e t_3, do que ter que pagar um novo custo de *setup* em cada um desses dois últimos períodos.

A determinação de quanto vamos produzir e quanto vamos estocar, pode ser obtida fazendo os passos descritos a seguir:

Passo 1: Para cada período t_j, calcular o custo de estoque $h(t_j)$ e somar o custo dos estoques anteriores, ou seja, $\sum_{i=1}^{t_j} h(i) = \mu_j$;

Passo 2: Quando $\mu_j \geq s_j$ (maior que o custo de preparação que deve ser fixo);

Passo 3: Comparar os período para verificar qual deles μ_j está mais próximo de s_j;

Passo 4: Produzir até o período em questão, fazer $t_j \leftarrow t_{j+1}$ e voltar ao passo 1, até que tenha sido decidido a produção para todo $t_j, \forall j \in 1, 2, ..., T$.

5.3 Métodos exatos

Na modelagem do PDLC, vide Seção 4.3, usamos variáveis binárias para decidir sobre a preparação, ou não, das máquinas (*setup*). Desta forma estamos com um problema inteiro misto. Segundo [8] pela dificuldade de solução do modelo em questão, as técnicas de solução mais indicadas para um PDLC também são heurísticas. Entretanto podemos usar algumas técnicas de programação inteira para obter solução, como por exemplo, *branch-and-cut*, ou *branch-and-bound*[2].

A seguir apresentamos os pontos principais para os métodos exatos discutidos por Karimi et al. [8]. Segundo os autores, existem dois métodos, basicamente, que nos fornecem boa aproximação para o 'resultado exato'. O primeiro foi proposto por Barany e Leung e o outro por Eppen e Martin.

5.3.1 Barany e Leung

Barany et al. [3] e Leung et al. [9] publicaram seus artigos em 1984 e 1989, respectivamente. A proposta dos dois seguem a mesma linha de raciocínio. Usar a técnica de geração de cortes para encontrar uma aproximação da solução.

Com essa técnica deseja-se fazer uma relaxação para o problema e resolvê-lo. Relaxação significa, por exemplo, permitir que as varíaveis inteiras sejam reais positivas, os ainda, que as variáveis binárias sejam reais entre 0 e 1. Após a solução para o problema inicial relaxado deseja-se adicionar uma desigualdade válida.

[2] Conhecido como BB, ou ainda B&B, é um algoritmo usado para resolver problemas de programação inteira e problemas de programação inteira mista. A ideia desta técnica é dividir para conquistar. Resolvemos o problema inicial usando uma relaxação (linear ou lagrangiana, por exemplo), e então dividimos o problema original em subproblemas. Na definição destes subproblemas aumentamos uma restrição, a restrição que forçará uma variável que deveria ser inteira no problema original ser. Podemos encontrar os detalhes desta técnica, bem como outros algoritmos de solução para problemas inteiros mistos em [12].

Desigualdade válida são novas restrições adicionadas ao problema. Elas podem ser geradas a partir da técnica de plano de corte, adicionando por exemplo um corte de Gomory. Mais sobre técnicas de plano de corte, bem como a definição de um corte de Gomory pode ser vista em [12].

Após a incorporação da desigualdade válida, deseja-se resolver o novo subproblema gerado. Este novo subproblema não será tão dificil de solucionar, em vista que é o problema anterior acrecido de uma restrição. Dessa forma com poucas iterações chegaremos a uma boa aproximação para solução deste novo subproblema.

Podemos conjecturar envoltório convexo, como o menor espaço factível para o problema original. Portanto sabemos que a solução do problema está em um vértice desse envoltório. A cada desigualdade válida adicionada, estamos aproximando cada vez mais do envoltório convexo. Dessa forma concluímos que adicionada uma desigualdade estaremos mais proximo da solução procurada.

5.3.2 Eppen e Martin

Eppen e Martin [6] descreveram esta aproximação para encontrar um resultado em 1987. Como foi discutido em [8], essa estimativa utiliza a técnica de redefinição de variáveis para reformular o PDLC descrito na Seção 4.3 para um problema com representação baseada em grafo.

Ainda segundo este artigo, lançar mão desta técnica faz com que fiquemos com uma formulação com relaxação mais apertada do que seria o problema original. E independentemente de possuir mais variáveis e mais parâmetros o tempo de solução é reduzido.

6 Linguagem de modelagem OPL

A *Optimization Programming Language* (OPL) é uma linguagem de programação matemática algébrica usada para modelar problemas de otimização e resolvê-los utilizando o CPLEX[3]. Trata-se de uma importante ferramenta para a realização de testes computacionais analisando na prática o que tem sido discutido de forma teórica. Um dos maiores benefícios dessa linguagem é na proximidade da sintaxe com a modelagem matemática.

[3]É um *solver* de problemas de programação matemática de alta performance que resolve problemas de programação linear, problemas de programação inteira mista e problemas de programação quadrática. É um dos mais importantes e melhores *solvers* encontrados atualmente. Mais detalhes podem ser encontrados em [7].

Modelado um problema no OPL, quando queremos encontrar solução, o CPLEX é invocado. Este por sua vez, é um solver tão eficiente, que resolve o problema da melhor forma possível, utilizando das mais variadas técnicas de otimização, linear, inteira e inteira mista estudadas até hoje.

Podemos configurar o OPL, de modo que ele atenda algumas exigências do usuário. Podemos admitir uma certa distância do ótimo (*gap*). Às vezes queremos que o CPLEX utilize uma técnica específica ao invés de outra, com o objetivo de comparar os resultados e analisar qual a melhor solução, e a mais rápida. Podemos também escolher qual tipo de relaxação será utilizada, entre muitas outras configurações.

6.1 Modelando o PDLC no OPL

A seguir apresentamos o código do PDLC descrito na Seção 4.3. A partir dos dados de entrada dos vetores custos (de produção, de estoque e *setup*) e do vetor demanda, indicando as quantidades requisitadas de cada item em cada período correspondente, o CPLEX é invocado e o problema é resolvido.

Após encontrar solução o OPL nos retorna um arquivo com o valor assumido pela função objetivo, o tempo de execução e o *gap* utilizado para cada PDLC executado.

6.2 Código

A Figura 1 a seguir representa o arquivo principal do projeto. Neste arquivo fazemos as ligações entre a modelagem para o PDLC (descrita no arquivo PDLC.mod, Figura 2) e os arquivos de teste (arquivos .dat: Figuras 3, 4 e 5). Este arquivo também é responsável pela criação de outro (arquivo Figura 6) com os resultados da execução do algoritmo.

A Figura 2 é o arquivo PDLC.mod. Neste arquivo encontra-se a modelagem do problema no OPL, com todas as restrições que tem que satisfazer. O domínio de cada variável de decisão, não é colocado nas restrições, pois elas já são definidas no espaço factível de cada uma.

6.3 Exemplos computacionais

Podemos criar vários arquivos de teste (arquivos .dat) e executar o projeto. O OPL irá gerar um arquivo .txt com todos os dados de saída (Figura 6) para os arquivos .dat (Figuras 3, 4 e 5) especificados no arquivo main.mod (Figura 1). As figuras 3, 4 e 5 representam os dados dos três testes realizados.

```
main{
var nome_model = "exercicio5_CLSP.mod";

var models = new Array();

models[models.length] = "clsp1.dat";
models[models.length] = "clsp2.dat";
models[models.length] = "clsp3.dat";


var modelCurr = 0;

      while (modelCurr < models.length ) {
            var oplrun = new IloOplRunConfiguration(nome_model,
models[modelCurr]);
            var ofile = new IloOplOutputFile("saida.txt", true);
            oplrun.oplModel.generate();
            oplrun.cplex.solve();
            ofile.write(" ", models[modelCurr]);
            ofile.write("\t OBJ =  ", oplrun.cplex.getObjValue());
            ofile.write("\tTempo =  ", oplrun.cplex.getCplexTime());
            ofile.write("\tGAP =  ", oplrun.cplex.getMIPRelativeGap());
            ofile.write("\n");
            ofile.close();
            modelCurr++;
      }
}
//var models = new Array("clsp1.dat","clsp2.dat", "clsp3.dat");
```

Figura 1: Arquivo main.mod

```
int N = ...;
int T = ...;

range Items = 1..N;
range Periods = 1..T;

float CAP[Periods] = ...;
float PT[Items] = ...;
float H[Items] = ...;
float ST[Items] = ...;
float S[Items] = ...;
float D[Items][Periods] = ...;

dvar float+ X[Items][Periods];
dvar float+ I[Items][0..T];
dvar boolean Y[Items][Periods];

//Objective function
minimize
      sum(i in Items, t in Periods) (H[i]*I[i][t] + S[i]*Y[i][t] +
C[i]XY[i][t]);

subject to {
      //Estoque nulo no primeiro e ultimo periodo
      forall(i in Items)
          I[i][0] == 0;

      forall(i in Items)
          I[i][T] == 0;

      //Atendimento de demanda
      forall(i in Items, t in Periods)
          I[i][t-1] + X[i][t] - I[i][t] == D[i][t];

      //Restriçoes de capacidade de maquina
      forall(t in Periods)
        sum(i in Items)(B[i]*X[i][t] + F[i]*Y[i][t]) <= CAP[t];


      //Preparação pra produção
      forall(i in Items, t in Periods)
          X[i][t] <= CAP[t]*Y[i][t];

}
```

Figura 2: Arquivo PDLC.mod

```
N = 6;
T = 15;

CAP = [ 1000.000, 1000.000, 1000.000, 1000.000, 1000.000, 1000.000,
1000.000, 1000.000, 1000.000, 1000.000, 1000.000, 1000.000, 1000.000,
1000.000, 1000.000, ];

B = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

C = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

H = [ 4.000, 2.000, 2.000, 4.000, 3.000, 4.000, ];

F = [ 120.000, 120.000, 90.000, 30.000, 30.000, 60.000, ];

S = [ 1000.000, 200.000, 600.000, 400.000, 1000.000, 200.000, ];

D = [
 [      0,      95,       0,      91,     108,     103,     110,
101,      90,     101,      97,     106,     103,     106,     106,
],
 [    106,       0,      92,     104,      91,     106,      95,
100,     107,     103,      93,      98,      93,     104,     107,
],
 [    107,       0,       0,       0,      93,     107,     101,
95,       91,     105,     105,      96,      94,     108,     107,  ],
 [     91,       0,       0,     104,     100,     106,      91,
109,      99,     103,      94,      91,     110,      93,      93,
],
 [     99,      95,     104,     109,     101,      95,      97,
96,       94,      93,      95,     102,     108,      90,      99,  ],
 [      0,     104,       0,     102,     106,      94,      98,
95,       98,      92,      91,      94,     104,     103,      97,  ],
];
```

Figura 3: Arquivo PDLC_Ex1.dat

```
N = 6;
T = 15;

CAP = [ 750.000, 750.000, 750.000, 750.000, 750.000, 750.000, 750.000,
750.000, 750.000, 750.000, 750.000, 750.000, 750.000, 750.000,
750.000, ];

B = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

C = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

H = [ 3.000, 3.000, 4.000, 3.000, 1.000, 5.000, ];

F = [ 40.000, 40.000, 30.000, 50.000, 30.000, 30.000, ];

S = [ 200.000, 400.000, 1000.000, 200.000, 400.000, 400.000, ];

D = [
 [    108,       0,     129,     102,     109,     103,       0,
145,     147,     118,     141,     108,     144,     105,     150,
],
 [    122,       0,     115,     137,     142,     135,     117,
0,       118,     113,     140,     132,     119,     117,     146,  ],
 [    113,     121,     129,     101,     115,     136,     134,
112,     134,     117,     119,     122,       0,       0,     126,
],
 [    130,       0,     143,       0,     118,       0,     150,
0,       141,     136,       0,       0,     148,     115,     142,  ],
 [      0,     146,     146,     126,     119,       0,       0,
122,     138,     116,     124,     139,     117,     135,     120,
],
 [      0,     123,       0,     107,     104,     111,     104,
107,     110,     123,     130,     133,     118,     128,       0,
],
];
```

Figura 4: Arquivo PDLC_Ex2.dat

```
N = 8;
T = 15;

CAP = [ 3750.000, 3750.000, 3750.000, 3750.000, 3750.000, 3750.000,
3750.000, 3750.000, 3750.000, 3750.000, 3750.000, 3750.000, 3750.000,
3750.000, 3750.000, ];

B = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

C = [ 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, ];

H = [ 1.000, 1.000, 2.000, 2.000, 2.000, 3.000, 3.000, 4.000, ];

F = [ 20.000, 25.000, 20.000, 25.000, 30.000, 30.000, 25.000, 30.000,
];

S = [ 1600.000, 1500.000, 1150.000, 1250.000, 1750.000, 1500.000,
1250.000, 1500.000, ];

D = [
 [     264,     239,     195,     211,     235,     293,     336,
354,     367,     373,     330,     299,     265,     240,     332,
],
 [     335,     324,     261,     286,     324,     381,     441,
474,     481,     501,     437,     391,     348,     320,     468,
],
 [     259,     240,     193,     218,     239,     282,     330,
366,     365,     375,     322,     308,     258,     237,     344,
],
 [     332,     312,     267,     280,     313,     376,     441,
483,     478,     510,     438,     409,     344,     315,     457,
],
 [     516,     484,     384,     430,     482,     570,     654,
726,     713,     751,     652,     600,     513,     477,     662,
],
 [     253,     243,     192,     208,     246,     285,     339,
357,     363,     372,     333,     305,     250,     249,     332,
],
 [     503,     480,     400,     411,     476,     578,     665,
722,     724,     759,     658,     605,     507,     486,     663,
],
 [     342,     313,     262,     283,     320,     382,     450,
490,     487,     509,     446,     400,     345,     312,     442,
],
];
```

Figura 5: Arquivo PDLC_Ex3.dat

De acordo com os arquivos de testes das Figuras 3, 4 e 5, foi gerado o arqruivo de saída que segue na Figura 6. Cada linha representa dos dados de saída para cada arquivo de teste respectivamente.

```
clsp1.dat   OBJ =  34393    Tempo =  0.219   GAP =  9.09306e-005
clsp2.dat   OBJ =  29290    Tempo =  1.531   GAP =  9.83566e-005
clsp3.dat   OBJ =  137641   Tempo =  2.125   GAP =  9.80979e-005
```

Figura 6: Arquivo Saída.txt

Observamos que os PDLC's das Figuras 3 e 4 tem a mesma dimensão. Entretanto o tempo de execução do PDLC descrito na Figura 4 foi cerca de sete vezes maior que o da Figura 3. As mudanças mais bruscas de um problema para o outro foram nos vetores que representam a capacidade da máquina, tempo de preparação de máquinas e custo de prepação da mesma. Podemos inferir que com a diminição do espaço factível do Exemplo 1 Figura 3, para o Exemplo 2 Figura 4 fez com que o tempo de encontrar solução para o PDL aumentasse tanto.

Para o Exemplo 3 Figura 5, o tempo foi maior pois aumentamos o horizonte de planejamento e a quantidade de itens a ser planejado.

7 Conclusão

O planejamento da produção é uma importante chave para o corte de gastos de uma indústria. No Problema de Dimensionamento de Lotes estamos interessados de maneira geral em decidir quando e quanto produzir a fim de minimizar os custos de produção.

Dentre os mais variados custos associados ao processo produtivo, podemos citar três: o custo de preparação de máquinas (*setup*), custo de produção e custo de estoque. E a priori, queremos atender a demanda, afim de maximinizar o lucro. O problema de maneira sucinta tenta modelar e otimizar este objetivo, com estas restrições.

Entretanto para cada tipo indústria podemos acrescentar restrições, bem como modificar a função objetivo. Dessa forma podemos criar um modelo, que considere até mesmo o tempo que a matéria-prima fica armazenada gerando gastos de manutenção do espaço físico em que se encontra. Nem sempre, satisfazer a demanda completa é a melhor opção, podemos assim admitir atrasos na entrega. O tempo que o produto fica no estoque, assim como os custos gerados com funcionários, e infinidades de singularidades que podemos acrescentar à ideia do PDL original podem ser incorporadas. Grande parte desses acréscimos podem ser estudados e modelados a fim de aproximar o problema da realidade, e assim ter melhor resultado da otimização.

Em contrapartida, pagamos um preço por deixar o problema mais próximo da realidade, que é a dificuldade de solução. De acordo com cada modelagem, podemos ter variáveis de decisões reais, inteiras e binárias. Pode ser que o número de variáveis seja muito grande. Podemos ter um conjunto de restrições muito apertado, isto é, um conjunto extremamente restrito de valores que a função objetivo pode assumir (região de factibilidade do problema).

Em vista do que foi apresentado, não existe um método que podemos classificar como o melhor para resolver o PDL. Dependendo das peculiaridades de cada modelagem, e de cada problema em si, podemos usar uma técnica diferente para encontrar a solução.

O que podemos apresentar de forma sucinta, é que nunca, ou quase nunca chegaremos no ótimo efetivo. É indispensável o uso de técnicas de relaxação, lagrangiana ou linear, lançar mão de desigualdades válidas, usar *branch-and-bound*, planos de corte e entre outras práticas, entretanto elas fazem com que conseguimos, no máximo, uma solução aproximada do que seria a solução exata. Outro fator que pode contribuir para os erros na solução, é pelo fato dos cálculos na máquina serem realizados e representados por ponto flutuante.

Concluimos ainda que aumentar o horizonte de planejamento ou a quantidade de itens a serem produzidos, bem como diminuir o espaço fáctivel, ou seja, o espaço no qual varíaveis de decisão podem assumir valores, fazem com que o tempo de execução do OPL aumente.

Para os casos testados o tempo não foi um agravante. Porém existem casos nos quais o OPL pode demorar horas e horas para encontrar uma solução. Desta forma o estudo de heurísticas para o problema se tornam indispensáveis para a agilidade de solucionar o problema de dimensionamento de lotes.

Referências

[1] Araujo, S.A.; Rangel, S. (2014). *Matemática aplicada ao planejamento da produção logística.* Notas em Matemática Aplicada, SBMAC, 2014.

[2] Bahl, H.C.; Ritzman, L.P.; Gupta, J.N.D (1987). *Determining lot sizes and resource requirements: a review.* Operations Research, 35(3): 329–345.

[3] Barany, I.; Van Roy, T. J.; Wolsey, L. A. (1984). *Strong formulations for multi-item capacitated lot sizing.* Management Science, 30(10):1255–1261.

[4] Billington, P.; Mcclain, L.; Thomas, L. (1983). *Mathematical programming approaches to capacity mrp systems: Review, formulation and problem reduction.* Management Science, 29:1126-1141.

[5] Buschkühl, L.; Sahling, F.; Helber, S.; Tempelmeier, H. (2010). *Dynamic capacited lot-sizing problems: a classification and review of solutions approaches.* OR Spectrum, 32:231-261.

[6] Eppen, G. D.; Martin, R. K. (1987). *Solving multi-item capacitated lot sizing problems using variable redefinition.* Operations Research, 35(6):832–48.

[7] IBM-OPL - Optimization Programming Language. http://www-03.ibm.com/software/products/ en/ibmilogcpleoptistud/

[8] Karimi, B.; Ghomi, S.F.; Wilson, J. (2003). *The capacited lot sizing problem: a review of models and algorithms.* Omega, 31(5):365-378.

[9] Leung, J. M. Y.; Magnanti, T. L.; Vachani, R. (1989). *Facets and algorithms for capacitated lot sizing.* Mathematical Programming, 45:331–59.

[10] Trigeiro, W. W.; Thomas, J.; Mcclain, J. O. (1989). *Capacitated lot sizing with setup times.* Management Science, 35:353-366.

[11] Wagner, H.M.; Whitin, T.M. (1958). *Dynamic version of the economic lot size model.* Management Science, 5(1):89-96.

[12] Wolsey, L.A. (1998). *Integer programming.* New York: John Wiley & Sons.