

Guilherme Corteletti Pedrosa

**A Utilização do *Bootstrap* em um Sistema de  
Estoque Multi-Estágio**

Belo Horizonte

2012

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

# **A Utilização do *Bootstrap* em um Sistema de Estoque Multi-Estágio**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para obtenção do grau de Mestre em Engenharia de Produção.

Orientador: Professor Samuel Vieira  
Conceição

Belo Horizonte

2012

## RESUMO

Uma acurada previsão de demanda e um bom gerenciamento dos estoques de itens MRO (Manutenção, Reparação e Operação) são essenciais para as companhias devido aos altos custos de estoques associados a estes itens. Diversos autores na literatura desenvolveram métodos de previsão de demandas e discutiram qual a distribuição estatística mais adequada para a modelagem das demandas destes itens em um sistema multi-estágio. Entretanto, o *bootstrap*, técnica com excelente desempenho em sistemas de um estágio foi pouco utilizado em sistemas mais complexos. Motivado por essa informação, este trabalho propôs a inserção da técnica de *bootstrap* em uma simulação discreta para modelar as demandas dos itens de reparação em um sistema multi-estágio. Como demonstram os resultados, o *bootstrap* apresentou comportamento similar aos encontrados na literatura quando o mesmo é testado em um sistema de um estágio.

## **ABSTRACT**

An accurate forecast of demand and good management of inventories of items MRO (Maintenance, Repair and Operation) are essential for companies due to high inventory costs associated with these items. Several authors in the literature have developed methods of forecasting demand and discussed which statistical distribution best suited for modeling the demands of these items in a multi-echelon system. However, the bootstrap technique with excellent performance in single-echelon systems has been little used in more complex systems. Motivated by this information, this paper proposed the insertion of the bootstrap technique in a discrete simulation to model the demands of repair items on a multi-echelon system. As shown by the results, the bootstrap presented behavior similar as showed when it is tested in single-echelon systems.

## Sumário

1. Introdução .....	1
2. Revisão Bibliográfica .....	3
2.1 Sistema de estoque multi-estágio .....	3
2.2 Classificação das demandas .....	7
2.3 Suavização Exponencial .....	9
2.4 Método de Croston.....	9
2.5 Bootstrap.....	11
2.6 Algoritmo Genético.....	14
2.7 Simulação de Eventos Discretos.....	16
3. Metodologia.....	17
3.1 Simulação com Otimização.....	17
3.2 Simulação.....	19
3.2.1 Bootstrap no procedimento de simulação de eventos discretos...20	
3.2.2 Número de Replicações e o Dimensionamento do Tamanho das Corridas.....	22
3.2.3 Comparações entre soluções.....	24
3.3 Algoritmo Genético.....	25
3.3.1 Cromossomo.....	28
3.3.2 Solução Inicial.....	28
3.3.3 Cruzamento.....	29
3.3.4 Procedimento de Mutação 1.....	31
3.3.5 Procedimento de Mutação 2.....	32
3.3.6 Procedimento de Mutação 3.....	33
3.3.7 Procedimento de Renovação da População.....	35
3.4 Parâmetros dos testes.....	36
4. Resultados.....	37
5. Trabalhos Futuros.....	43
6. Conclusão.....	44
7. Anexo A.....	45
8. Bibliografia.....	54

## Lista de Figuras

Figura 1: Esquema do problema proposto por (SHERBROOKE 1968).....	3
Figura 2: Classificação segundo (SYNTETOS e BOYLAN, 2005).....	8
Figura 3: Modelo de Simulação com Otimização.....	17
Figura 4: Modelo de Simulação.....	19
Figura 5: Gerador de demanda <i>bootstrap</i> .....	21
Figura 6: Gerador de demanda Poisson.....	22
Figura 7: Cálculo do Regime Transiente.....	24
Figura 8: Exemplo de Cromossomo.....	28
Figura 9: <i>fill rate</i> – unif (40,20).....	52
Figura 10: <i>cicle service level</i> - norm(30,8).....	53

## Lista de Tabelas

Tabela 1: Resumo dos testes.....	37
Tabela 2: Solução do modelo com distribuição de Poisson.....	37
Tabela 3: Solução do modelo com a tecnica de <i>bootstrap</i> .....	38
Tabela 4: Solução do modelo com distribuição de Poisson.....	38
Tabela 5: Solução do modelo com distribuição de <i>bootstrap</i> .....	39
Tabela 6: Expectativa do Custo de Estoque – Poisson.....	39
Tabela 7: Expectativa do Custo de Estoque – <i>bootstrap</i> .....	39
Tabela 8: Expectativa do Custo de <i>backorder</i> – Poisson.....	40
Tabela 9: Expectativa do Custo de <i>backorder</i> – <i>bootstrap</i> .....	40
Tabela 10: Resumo dos Custos de Estoque e <i>Backorder</i> .....	41
Tabela 11: Resumo dos Custos de Estoque e <i>Backorder</i> .....	41
Tabela 12: Resumo dos resultados para <i>lead time</i> externo igual a 20 e interno igual a 10.....	41
Tabela 13: Resumo dos resultados para <i>lead time</i> externo igual a 10 e interno igual a 20.....	41
Tabela 14: Tempos de Execução.....	42
Tabela 15: Distribuições utilizadas para a geração das séries de demandas.....	48
Tabela 16: Resumo dos testes.....	49
Tabela 17: Médias dos resultados obtidos na Simulação.....	50
Tabela 18: Resultados obtidos para todas as distribuições de <i>lead time</i> para o nível de serviço 95%.....	51
Tabela 19: Média dos desvios absoluto.....	54

## 1. Introdução

Uma acurada previsão de demanda de itens MRO (Manutenção, Reparação e Operação) é essencial para as companhias devido aos altos custos de estoques associados a estes itens. Desde empresas de serviços a empresas manufatureiras, tais itens são responsáveis por grande parte dos custos de estoque e estão ligados diretamente à qualidade dos serviços e produtos oferecidos. As peças de reposição são estocadas para dar suporte às atividades de manutenção e para proteger os sistemas de falhas. Dessa forma, um bom gerenciamento dos estoques das peças de manutenção e da própria manutenção em si é importante atividade para qualquer empresa.

Um dos primeiros trabalhos sobre o controle de estoque para itens de reposição com enfoque na cadeia de suprimentos foi apresentado por (SHERBROOKE, 1968). A técnica de controle de itens de reparo de sistema de estoque multi-estágio (METRIC) foi um modelo desenvolvido por (SHERBROOKE, 1968) na força aérea americana. Este modelo assume que as demandas seguem uma distribuição de *Poisson* e, a partir deste problema, foram desenvolvidos diversos artigos considerando o sistema de estoque multi-estágio.

(MUCKSTADT, 1973) propôs uma modificação no modelo de (SHERBROOKE, 1968) denominada MOD-METRIC. Já (ROSENBAUM, 1981) propôs restrições ao nível de atendimento de serviço e (SINGH e VRAT, 1984) estudou o mesmo problema, porém adicionou o problema de localização de facilidades a ele. Ao longo dos anos foram apresentadas diversas heurísticas para o problema, (AXSATER, 1990) entre outros. Também foram propostos diversos modelos para políticas de estoques diferentes (HOPP, SPEARMAN, E ZHANG 1997) e para diferentes critérios de avaliação (FORSBERG 1995).

(GRAVES, 1985) utilizou a distribuição de *poisson* composta e (AXSATER, 2001) fez uma aproximação de uma técnica de itens *fast-moving* para itens *slow-moving* com *lead time* estocástico. (AXSATER, 2003) apresentou um método em que ele faz uma aproximação das demandas à distribuição normal para o cálculo da política de estoque. (DESMET, 2009) também assumiu a normalidade das demandas em seu trabalho, porém (BABAI, 2010) reforça a necessidade de uma técnica especial para os itens MRO. Padrões de demanda intermitente são comuns entre peças de reposição, ou seja, demandas aleatórias positivas com uma grande proporção de valores de zero (SILVER et al., 1971). Devido a sua natureza específica, a previsão acurada da demanda se torna uma tarefa bastante complicada (BOYLAN e SYNTETOS, 2010). Na prática, a demanda intermitente ocorre quando há muitos clientes de pequeno porte e alguns clientes de grande porte, ou quando a frequência de

pedidos dos clientes tem alta variabilidade.

Nos últimos anos, a área de previsão de demanda intermitente tem recebido muita atenção e avanços significativos foram feitos no campo. Pesquisadores têm dedicado seus esforços para melhorar a precisão dos procedimentos de previsão desses itens. (CROSTON, 1972) provou que a utilização da técnica de suavização exponencial simples (SES), proposta por (BROWN, 1959), era imprópria para uso em itens com demanda intermitente. Ele propôs uma nova técnica de previsão, na qual poderia lidar melhor com as dificuldades da demanda intermitente. Desde então, alguns autores propuseram diversas modificações do método de Croston, tais como (SEGERSTEDT, 1994; JOHNSTON e BOYLAN, 1996b; LEVÉN e SEGERSTEDT, 2004) entre outros.

Outro método eficiente para lidar com itens de demanda intermitente é o *bootstrap*. O método proposto por (WILLEMAIN, 2004) é bem aceito e produz previsões mais precisas da distribuição da demanda durante o *lead time* do que a suavização exponencial e, segundo o mesmo autor, do que o método de Croston.

Este trabalho propôs um método (Simulação com Otimização) para o problema de controle de estoques de sistemas multi-estágio para itens de reparação, assim como em (MARSEGUERRA, 2004), onde se combinou um algoritmo genético com a simulação de Monte Carlo para um problema de estoque de dois níveis. Neste trabalho utilizou-se a Simulação com Otimização para avaliar a técnica de *bootstrap* em um sistema de estoque multi-estágio.

Como a técnica de *bootstrap* tem obtido excelentes resultados para a modelagem da demanda de itens MRO durante o *lead time* (VISWANATHAN e ZHOU, 2008), o procedimento foi inserido em uma simulação de eventos discretos, como um gerador de demandas aleatórias, visando capturar melhor as variabilidades das demandas dos itens MRO.

Assim como os artigos (WILLEIMAN, 20004), (BOYLAN e SYNTETOS, 2010), (VISWANATHAN e ZHOU, 2008), entre outros, este trabalho, compara os pontos de ressuprimentos calculados utilizando o *bootstrap* com outras distribuições estatísticas, porém, em um sistema de estoque multi-estágio.

A heurística utilizada para o processo de otimização na técnica de Simulação com Otimização foi o algoritmo genético. Assim como em (PASANDIDEH, 2011), que resolveu um problema multi-estágio de itens de reparação em um sistema com política (r,Q), este trabalho também implementou um algoritmo genético para um problema de estoque multi-estágio porém, para política (s,s-1).

## 2. Revisão Bibliográfica

### 2.1 Sistema de estoque multi-estágio

Um dos primeiros trabalhos sobre controle de estoque para itens de reposição com enfoque na cadeia de suprimentos foi apresentado por (SHERBROOKE, 1968). A técnica de controle de itens de reparo de sistema de estoque multi-estágio (METRIC) foi um modelo desenvolvido por (SHERBROOKE, 1968) na força aérea americana enquanto o autor trabalhava na *RAND Corporation*. Diferentemente da abordagem de estoque básico (modelo do lote econômico), na qual a política de estoque é definida para cada item separadamente, o modelo desenvolvido por (SHERBROOKE 1968) considera a decisão em conjunto para todos os itens e limita suas aquisições a um orçamento. Ele é definido por várias bases onde as demandas são atendidas e por um único depósito central que alimenta as respectivas bases, conforme a figura 2.

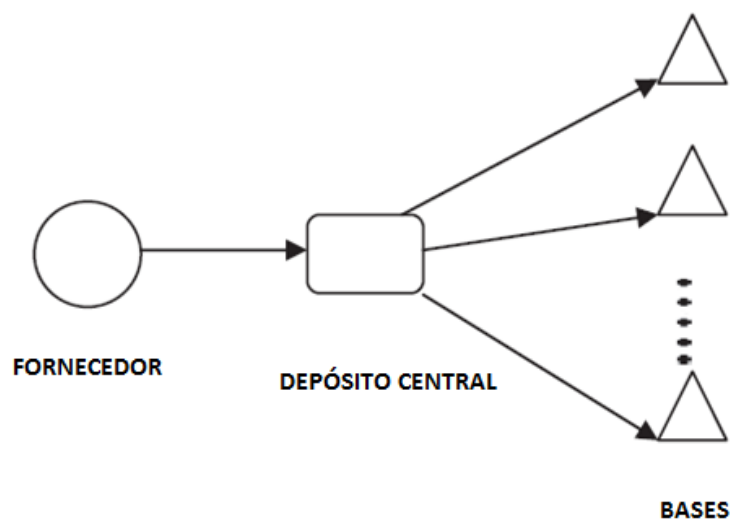


Figura 1: Esquema do problema proposto por (SHERBROOKE 1968)

O modelo do METRIC tem as seguintes premissas:

- As peças de reposição são estocadas nas bases e no depósito.
- Quando uma peça falha, o consumidor é atendido na base se a base possui a peça em estoque; quando a base não possui a peça para reposição, ela envia um pedido para o

depósito e é gerado um *backorder*.

- *Single Indenture*, ou seja, os itens de reparação, não são avaliados hierarquicamente (itens pais, itens filhos).
- A decisão de reparar a peça na base não depende do seu estoque ou da sua capacidade de trabalho.
- A base é ressuprida somente pelo depósito.
- A política de estoque  $(s - 1, s)$  é apropriada para todos os itens em todas as bases e depósito central.
- A demanda por peças de reposição nas bases permanecerão constantes ao longo do tempo.

O problema descrito é um problema np-completo (PASANDIDEH, 2011) e seu modelo matemático inteiro não linear, segundo (Sherbrooke 2004), pode ser definido da seguinte forma:

$$\text{Max } \sum_{j=1}^n A_j / \sum_{j=1}^n N_j \quad (1)$$

$$\text{s. a. } \sum_{i=1}^l \sum_{j=1}^n c_i s_{ij} \leq C \quad (2)$$

$$s_{ij} \in Z \quad (3)$$

Onde  $A_j$  é a disponibilidade da base  $j$ ,  $N_j$  o número de equipamentos na base  $j$ ,  $c_i$  o custo de aquisição do item  $i$ ,  $s_{ij}$  a quantidade estocada do item  $i$  na base  $j$ ,  $C$  o capital total a ser investido,  $n$  o número total de bases e  $l$  o número total de itens. Já  $j=0$  representa o depósito.

A função objetivo (8) maximiza a disponibilidade total do sistema, ou seja, a disponibilidade total de aeronaves (equipamentos) que estarão aptas a voar (Sherbrooke2004); a equação (9) restringe o total de itens a ser adquirido a um capital  $C$  e a equação (10) representa o domínio do problema.

O METRIC utiliza o conceito de canal de distribuição de demanda. Esta demanda é calculada em função das demandas de um mesmo item em todas as bases. A demanda no depósito é dada pela equação (11), onde  $m_{ij}$  é a demanda do item  $i$  na base  $j$  e  $r_{ij}$  é a probabilidade de reparo do item  $i$  na base  $j$ .

$$m_{i0} = \sum_{j=1}^n m_{ij} (1 - r_{ij}) \quad \forall i=1..l. \quad (4)$$

A demanda média do item  $i$  no canal de distribuição na base  $j$  é dada pela equação (12). Na equação (12),  $T_{ij}$  é o tempo médio de reparo do item  $i$  na base  $j$  e  $O_j$  é o tempo total decorrido entre o pedido e o transporte do item entre a base e o depósito.

$$\mu_{ij} = m_{ij} \left[ r_{ij} T_{ij} + (1 - r_{ij}) - \left( O_j + \frac{EBO[s_{i0}|m_{i0}T_{i0}]}{m_{i0}} \right) \right] \quad \forall i=1..l. j=1..n. \quad (5)$$

Assumindo que a demanda segue uma distribuição de *Poisson*, o número esperado de *backorders*,  $EBO$ , é o número esperado de demandas não atendidas pelas bases ou depósitos,  $s_{i0}$  é o ponto de ressuprimento de estoque e  $m_{i0}T_{i0}$  ou  $\mu_{ij}$ , a média da demanda a ser atendida (equação (13)).

$$EBO[s_{i0}|m_{i0}T_{i0}] = \sum_{x=s_{i0}+1}^{\infty} (x - s) P(DI = x | m_{i0}T_{i0}) \quad (6)$$

A disponibilidade na base é dada pela equação (14), onde  $Z_i$  é o número de unidades que são utilizadas no equipamento.

$$A_j = 100 \prod_{i=1}^l \left( 1 - \frac{EBO[s_{ij}|\mu_{ij}]}{N_j Z_i} \right) Z_i \quad (7)$$

(MUCKSTADT, 1973) propôs uma modificação no modelo de (SHERBROOKE, 1968) denominada MOD-METRIC (multi-item, multi-estágio e *multi-indenture*). Em seu artigo, o autor propôs um algoritmo para determinar a quantidade de armazenamento de armas de caças F-15 nas oficinas.

(ROSENBAUM, 1981) estudou o problema de estoque de itens de reparação para 2 níveis, centros de distribuição e centros regionais de distribuição, tendo como restrições o nível de atendimento de serviço do depósito central e dos regionais. (SINGH e VRAT, 1984) estudaram o mesmo problema, porém, adicionou o problema de localização de facilidades a ele. (SVORONOS e ZIPKIN, 1988) propuseram alguns refinamentos para o problema de estoque multi-estágio avaliando a demanda média e sua respectiva variância no depósito e nas bases.

(GRAVES, 1985) também estudou o problema, entretanto considerou que as falhas

nos equipamentos eram geradas por uma distribuição de *Poisson* composta e os tempos de transporte fixos. Ele gerou uma aproximação da distribuição do nível de estoque na rede e calculou a demanda e a variância dos *backorders* nas bases. (DEBOT e GRAVES, 1985) apresentaram um modelo que consideravam os custos de estocagem, de armazenagem e de pedido.

AXSATER (1990) apresentou uma heurística simples e exata para o problema de estoques de dois níveis com reabastecimento um a um, *lead times* constantes e demandas distribuídas de acordo com a distribuição de *Poisson*. Neste trabalho, a avaliação dos custos de estoques é feita a partir dos custos de estoques médios. Em (AXSATER e ZHANG, 1996) os autores consideraram a distribuição de *Poisson* composta para as demandas nas bases e avaliaram o modelo de estoque em relação aos custos de armazenagem e de rupturas. Já (FORSBERG, 1995) utiliza um modelo de inventário em que a política de revisão contínua  $(s-1,s)$  é aplicada para todos os itens em um sistema de dois níveis, e também avalia os custos de inventário como em (AXSATER e ZHANG, 1996).

(HOPP, SPEARMAN, e ZHANG, 1997) formularam um modelo utilizando a política de revisão contínua  $(r,Q)$  considerando as restrições de frequências de pedidos e nível de serviço. (AXSATER, 2000) desenvolveu um procedimento exato para avaliação do sistema com política de revisão contínua  $(r,Q)$  e com demanda distribuída de acordo com a distribuição de *Poisson* composta.

(RUSTENBURG 2000) focou seu trabalho na disponibilidade total de um sistema complexo de vários componentes. A capacidade de armazenamento dos itens de reparação é limitada, assim como a quantidade de capital para a aquisição de peças de reposição. Neste artigo, o autor desenvolve estratégias para a aquisição de itens de reparo e avalia qual quantidade de capital é necessária para atingir certo nível de disponibilidade.

(SEO, JUNG, E HAHM, 2001) desenvolveram uma política ótima de estoque para ser utilizada em um depósito central com bases idênticas múltiplas. (AXSATER, 2001) fez uma aproximação de uma técnica de itens *fast-moving* para itens *slow-moving* com *lead time* estocástico. (AXSATER, 2003) apresentou um método em que ele faz uma aproximação das demandas nas bases à distribuição normal para encontrar os pontos ótimos da política de estoque.

(CAGLAR, CHUNG-LUN, e SIMCHI-LEVI, 2004) desenvolveram um modelo matemático para sistema de controle de estoque de peças de reposição com 2 níveis e multi-item, com função objetivo que minimiza os custos que estão sujeitos aos tempos de repostas de cada base. (AXSATER, 2005) determinou os custos de ruptura no depósito central e

desenvolveu uma nova perspectiva descentralizada para a otimização dos custos de armazenagem e de ruptura em um sistema com política de revisão contínua  $(r,Q)$ . (JOKAR e ZANGENEH, 2006) desenvolveram um modelo com um único depósito central, com muitas bases e com perda de demanda não atendida e substituição de pedidos de demandas. Em seu modelo foi considerado a política de revisão contínua  $(s-1,s)$ . Em (HAJI, NEGHAB, e BABOLI, 2008) foi desenvolvido uma nova política de controle de estoque de peças de reposição.

Em (DESMET, 2009) foi apresentado um modelo em que se assume a normalidade das demandas nas bases e *lead times* nominais. No artigo referido é apresentada uma simulação discreta onde são avaliadas as médias e as variâncias das demandas e dos *lead times*, assim como o modelo aproximado. (BABAI, 2010) desenvolveu um procedimento numérico para calcular a política de estoque  $(s-1,s)$ , considerando a função objetivo com custos de armazenagem e de ruptura. Neste estudo, o autor reafirma a necessidade de um método específico para o controle de itens MRO.

## 2.2 Classificação das demandas

A Classificação das demandas é referente à variabilidade inerente da série temporal (GHOBBAR e FRIEND, 2003) e existem quatro categorias principais que são distintas por sua variabilidade. Os padrões de demandas são classificados em: intermitentes, na qual as demandas positivas aparecem aleatoriamente, porém com períodos sem demanda; errática, caracterizada pelo tamanho de demanda altamente variável; demanda irregular, que é tanto intermitente quanto errática, e em itens *slow movings*, que apresentam a característica da demanda intermitente, porém as demandas positivas têm valores próximos da unidade.

Poucos autores têm apresentado uma metodologia consistente para classificar os itens de acordo com os padrões de demanda. (SILVER, 1998) et al. não apresentou uma metodologia, mas estabeleceu alguns limites entre os padrões *slow-moving* e itens com demanda regular. Com base em experimentos computacionais, eles propuseram um limite superior de 10 unidades para a média do *lead time* para a demanda *slow moving*, sendo a distribuição de *Poisson* indicada para demandas com coeficiente de variação entre os valores 0.9 e 1.1, e fora desse parâmetro, a distribuição de Laplace é a melhor indicação. Para valores maiores ou iguais a 10, a distribuição normal deve ser considerada quando o coeficiente de variação for menor ou igual a 0,5. Caso contrário, a distribuição Gama deve ser utilizada.

Um método analítico foi desenvolvido por (WILLIAMS, 1984) para classificar a

demanda em três categorias: suave, baixa movimentação e intermitente. Estas categorias foram baseadas na decomposição da variância da demanda durante o *lead time* em variabilidade do *lead-time*, variabilidade da transição entre demandas e variabilidade do tamanho da demanda. No entanto, de acordo com (EAVES, 2002), a classificação da demanda de (WILLIAMS, 1984) não era adequada para descrever a estrutura da demanda observada na Royal Air Force (RAF).

(JOHNSTON e BOYLAN, 1996) propuseram uma outra definição de demanda intermitente utilizando o conceito de média dos intervalos entre demandas positivas  $p$ . De acordo com o artigo citado, para  $p$  maior do que 1,25, o método de Croston deve ser utilizado em detrimento da suavização exponencial simples. (SYNTETOS e BOYLAN, 2005) levou esse trabalho à frente propondo novos valores para  $p$ , e usou o coeficiente de variação do tamanho da demanda elevado ao quadrado ( $CV^2$ ) para ser usado como uma nova fronteira (“*lumpiness*”). Com esta nova metodologia, os autores classificaram as demandas em diferentes categorias: demanda intermitente, errática, *slow-moving* e *lumpy*. A classificação das demandas e seus respectivos pontos de corte são mostrados na figura 1. Os valores encontrados no referido artigo são  $p=1,32$  e  $(CV^2)=0,49$ . Finalmente, (BOYLAN, 2008) mostrou empiricamente a insensibilidade do valor  $p$  de corte, no intervalo aproximado 1,18 - 1,86, para fins de classificação da demanda.

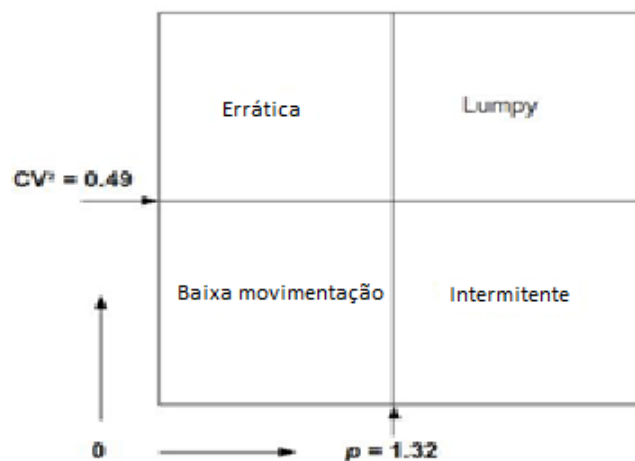


Figura 2: Classificação segundo (SYNTETOS e BOYLAN, 2005)

### 2.3 Suavização Exponencial

A suavização exponencial é uma técnica muito popular que pode ser usada para produzir uma série de dados temporais suavizados ou para fazer previsões. Muitas vezes, é aplicada no mercado financeiro para dados econômicos e para previsão de vendas para controle de estoque (BROWN, 1959). Vários estudos empíricos têm sido feitos com diversos modelos de suavização exponencial para previsão de peças de reparação. A forma mais simples da suavização exponencial é dada por:

$$M_t = \alpha X_t + (1 - \alpha)M_{t-1}, t = 1 \dots T \quad (8)$$

$$V = \left(\frac{1}{T}\right) \sum_1^T (X_t - M_{t-1})^2 \quad (9)$$

Onde  $X_t$  é a demanda real por um item no período  $t = 1 \dots T$ ;  $M_t$  é a média estimada por meio da suavização exponencial; e  $\alpha$  é a constante de amortecimento entre 0 e 1. Já  $V$  é a variância da demanda.

Três variações básicas de suavização exponencial são comumente usadas: suavização exponencial simples (BROWN, 1959); suavização exponencial com correção de tendência (HOLT, 1957), e método de Holt-Winters (WINTERS, 1960). A diferença entre estas abordagens é que as séries temporais são construídas a partir de dados observados, tais como o nível, o crescimento e os efeitos sazonais, respectivamente, e esses componentes são adaptados à série ao longo do tempo (Billah, 2006). Essas técnicas não são aplicadas para os itens *slow-moving* (itens MRO) porque, se vários períodos de demanda nulas forem observados a previsão vai diminuir e, eventualmente, se aproximará de zero (WALLTROM e SEGERSTEDT, 2010).

### 2.4 Método de Croston

(CROSTON, 1972) propôs um método para previsão da demanda para itens de reparação, pois a suavização exponencial não era apropriada para este tipo de item. Este método aplica a técnica de suavização exponencial separadamente para o tamanho das demandas e para o intervalo entre as demandas positivas. A previsão é atualizada somente quando houver demanda. Se não há ocorrência de demanda no período  $t$ , então a suavização exponencial para a previsão do tamanho médio da demanda  $\hat{Z}_t$  e a estimativa do tamanho dos

intervalos entre demandas positivas  $\hat{P}_t$  não são alteradas. Seja  $Q_t$  o intervalo de tempo entre demandas e  $D_t$  o valor da demanda positiva, o método Croston pode ser assim descrito pelas seguintes fórmulas:

$$Z_t = \alpha D_t + (1 - \alpha)Z_{t-1} \quad (10)$$

$$P_t = \alpha Q_t + (1 - \alpha)P_{t-1} \quad (11)$$

$$E(Y_t) = Z_t / P_t \quad (12)$$

De acordo com (SYNTETOS e Boylan, 2005), esse método é amplamente aplicado na indústria e foi incorporado em vários pacotes de *software* de previsão de demanda. A maioria dos autores concorda que o método de Croston é mais adequado para a demanda intermitente do que os métodos tradicionais, tais como média móvel e suavização exponencial simples.

(SYNTETOS e Boylan, 2001) mostraram que o método original de Croston acarreta em um erro positivo na estimativa da demanda a cada unidade de tempo. Eles também propuseram um novo estimador não enviesado (eq. 6), a esperança da demanda no período  $t$ , que é adicionado ao método original. As atualizações ocorrem da mesma maneira que no método de Croston original.

$$\hat{d}_t = \left(1 - \frac{\beta}{2}\right) \frac{\hat{z}_t}{\hat{p}_t} \quad (13)$$

(SYNTETOS, 2001) observou que o método proposto por (SYNTETOS e BOYLAN, 2001) ainda era enviesado e propôs uma modificação para o método Croston. Esse procedimento foi novamente avaliado por (TEUNTER e SANI, 2009) que verificaram o seu bom funcionamento.

(LEVÉN e SEGERSTEDT, 2004) propuseram uma modificação do método original (LVCR), que pode lidar tanto com a demanda para itens *slow-moving* quanto para itens *fast-moving*. Segundo os autores, em experimentos utilizando simulação, o novo estimador superou a suavização exponencial e pode lidar com várias demandas em um único dia. O estimador novo para a demanda média é:

$$\hat{d}_t = \hat{d}_{t-1} + \alpha \left( \frac{x_t}{t_t - t_{t-1}} - \hat{d}_t \right) \quad (13.a)$$

(LEVÉN e SEGERSTEDT, 2004) afirmaram que seu estimador evitava o viés encontrado por (SYNTETOS e BOYLAN, 2001), no entanto, (BOYLAN e SYNTETOS, 2007) mostraram que essa afirmação não era verdadeira.

Como foi demonstrado por outros autores que o método proposto por (BOYLAN e SYNTETOS, 2001) possuía um viés positivo, (BOYLAN e SYNTETOS, 2005) propuseram um novo estimador e compararam com o método (syCR) com a média móvel simples, (SES) e (orgCR) em 3000 séries de demandas intermitentes reais de uma indústria automotiva. Eles mostraram que o método (syCR) é o estimador mais preciso, considerando-se os estimadores erro médio de escala e média dos erros relativos da raiz quadrada geométrica. (BOYLAN e SYNTETOS, 2006) usaram uma mesma série de dados em um procedimento de simulação para comparar o desempenho dos métodos para o controle de estoque. Eles mostraram que método syCR possui um desempenho global superior aos demais, seguido pela média móvel e orgCR.

(BOYLAN e SYNTETOS, 2007) avaliaram os métodos (orgCR), (leCR), (syCR) e (SES). Eles simularam séries com cinco intervalos entre demandas diferentes e concluíram que o (IVCR) tem pior desempenho que os métodos (CR) e (orgCR) em relação à média da raiz quadrada do erro. (BOYLAN e SYNTETOS, 2007) também afirmam que (syCR) tem o melhor desempenho quando os intervalos entre demandas são pequenos.

(TEUNTER e DUNCAN, 2009) usaram um grande conjunto de dados da Força Aérea do Reino Unido para fazer um estudo comparativo. Os autores utilizaram uma nova perspectiva para avaliar os métodos (orgCR), (syCR), (leCR) e o *bootstrap*.

(BOYLAN e SYNTETOS, 2010) avaliaram os métodos presentes na literatura dando foco para a análise de suas respectivas variâncias. Nesse artigo, os autores afirmam que essa perspectiva é fundamental para a análise dos modelos de previsão de demanda para itens de reparação.

## **2.5 Bootstrap**

A técnica de *bootstrap* clássica, introduzida por (EFRON, 1979), envolve a amostragem consecutiva, com substituição, da demanda histórica. Depois das diversas replicações é gerada uma distribuição empírica. Segundo (BOYLAN e SYNTETOS, 2010) são usadas usualmente mais de 10 mil replicações no procedimento de *bootstrap*. As duas principais desvantagens do *bootstrap* clássico são a não consideração de qualquer auto-correlação potencial entre os dados da demanda histórica e os dados gerados nas replicações

serem sempre os mesmos que os da amostra original.

Para contornar estas desvantagens do *bootstrap* clássico, (WILLEIMAN, 2004) propôs o mais conhecido método de *bootstrap* para previsão da demanda de itens de reparação. Este procedimento capta melhor a auto-correlação entre as ocorrências de demandas, especialmente para a demanda intermitente com alta proporção de valores zero. Ele utiliza o modelo de Markov para avaliar as probabilidades de transições entre as demandas nulas e não nulas.

Inicialmente, o método avalia as probabilidades de transição entre as demandas zero/não-zero (processo de Markov). Com base nestas probabilidades de transição, uma sequência de valores zero e não zero é gerada para todo o horizonte de previsão (demanda durante o *lead time*). Os valores diferentes de zero gerados são substituídos pelo processo de *jittering*, que permite uma variação em torno das demandas presentes na amostra original.

Sendo  $X^*$  um valor da demanda histórica selecionado aleatoriamente,  $Z$  os valores do inverso da distribuição normal padrão também selecionados aleatoriamente, e  $S$  a demanda diferente de zero, então a previsão da demanda é dada por:

$$S = \begin{cases} INT\{X^* + Z\sqrt{X^*}\} , if S > 0 \\ X^* , caso contrário \end{cases} \quad (14)$$

Segundo (WILLEIMAN, 2004) o método de *bootstrap* produz previsões mais precisas sobre a distribuição da demanda durante um *lead time* fixo do que a técnica de suavização exponencial e o método de Croston modificado. De acordo com (HUA e ZHANG 2006), a técnica de *bootstrap* de Willeiman é mais útil quando os dados históricos são limitados.

O pseudo código do *bootstrap* de (WILLEIMAN, 2004) é mostrado passo a passo a seguir:

#### **Método de *bootstrap* (WILLEIMAN, 2004)**

1. Obtenha as demandas históricas de acordo com a unidade de tempo escolhida (ex. dias, semanas, meses);
2. Estime as probabilidades de transição de acordo com a cadeia de Markov de dois estados;
3. Levando em consideração a última demanda observada, use o modelo de Markov para gerar uma sequência de valores zeros e não zeros para todo o horizonte de

- previsão;
4. Substitua todas as demandas não zero por um valor de demanda positiva obtida nas demandas históricas;
  5. Utilize o processo de *jittering* nas demandas positivas;
  6. Some todos os valores de demandas positivas de todo horizonte a ser avaliado e obtenha um valor da previsão da demanda durante o *lead time*;
  7. Repita os passos 3 – 6 muitas vezes; (Geralmente 1000 vezes)
  8. Construa uma distribuição empírica com os valores de previsão da demanda durante o *lead time*.

Recentemente (VISWANATHAN e ZHOU, 2008) desenvolveram um método de *bootstrap*, no qual, por meio de experimentos computacionais, mostrou ser superior ao método de (WILLEMAIN, 2004) tanto para testes com dados simulados, quanto para dados de demandas reais. A diferença entre os métodos é que em (VISWANATHAN e ZHOU, 2008) os intervalos entre demandas positivas são gerados por meio da distribuição histórica dos intervalos entre demandas, em vez de gerados a partir de modelo de Markov de dois estados. (VISWANATHAN e ZHOU, 2008) também não utilizaram o processo de *jittering*.

O pseudo código do método (ZHOU e VISWANATHAN, 2008) é mostrado a seguir:

#### **Método de *bootstrap* (ZHOU e VISWANATHAN, 2008)**

1. Gere os histogramas com os dados da demanda histórica (tamanho das demandas positivas e intervalo entre demandas);
2. Gere um intervalo entre demandas de acordo com um dos histogramas obtido no passo 1. Atualize a variável horizonte de tempo que é usado para contar o tempo usado do *lead time*;
3. Se a variável horizonte de tempo é igual ou menor que o *lead time*, gere aleatoriamente o valor da demanda positiva de acordo com o seu histograma e vá para o passo 2. Senão, some os valores dos tamanhos da demanda até o momento e obtenha uma previsão para o valor da demanda durante o *lead time* e vá para o passo 5;
4. Repita os passos 2 – 3 1000 vezes;
5. Construa uma distribuição empírica com os valores de previsão da demanda durante o *lead time*.

(VISWANATHAN e ZHOU, 2010) comparam o método proposto por (VISWANATHAN e ZHOU, 2008) com o método paramétrico proposto por (BABAI e SYNTETOS, 2007). Segundo esses resultados, o *bootstrap* tem desempenho melhor para dados simulados (grande série de dados), mas tem resultado pior que o método paramétrico para dados de demandas reais (pequena série de dados).

## 2.6 Algoritmo Genético

Existem diversas aplicações de meta-heurísticas para a solução de problemas de otimização combinatória, sendo muitas delas baseadas em fenômenos naturais. Uma classe de heurísticas é aquela baseada no princípio da evolução.

O algoritmo genético foi introduzido por (HOLLAND, 1975) e desde então foi aplicado com sucesso nos mais diversos problemas tais: problema do caixeiro viajante, problema de sequenciamento de tarefas em máquinas, etc. Segundo (PASANDIDEH, 2011), apesar de o algoritmo genético ter sido utilizado para alguns problemas de controle de estoque, a aplicação para os problemas de estoque multi-estágio ainda são raros.

Um algoritmo evolucionário pode ser descrito, segundo (MICHALEWICZ, 2000), de acordo com o pseudo-código:

**Procedimento:** Algoritmo genético

**begin**

$t = 0$

Inicializar a população  $P(t)$

Avaliar a população  $P(t)$

**While** (condição do término do loop)

**Begin**

Alterar  $P(t)$

$t = t + 1$

Selecionar  $P(t)$  de  $P(t-1)$

Avaliar  $P(t)$

**end**

**end**

A população é composta por um conjunto de respostas (cromossomos) que vão sendo modificados de acordo com as regras de evolução da heurística, até encontrar uma solução ótima (melhor solução encontrada pelo procedimento de busca). Para todos os passos do algoritmo apresentados acima, existem uma infinidade de variações na literatura.

A representação do cromossomo, resposta do problema de otimização combinatória, pode ser feita por meio de vetores de símbolos de tamanho fixo (representação binária), permutações, da representação dos possíveis estados de respostas e de expressões simbólicas (árvores binárias de tamanhos variáveis). Apesar de um mesmo problema poder ser representado por várias formas diferentes, todas essas formas têm eficiências iguais, e os pesquisadores costumam utilizar a representação natural do problema (MICHALEWICZ, 2000).

Assim como todas as etapas básicas, a geração da população inicial (conjunto de respostas inicial) pode ser feita de diversas maneiras, sendo a mais comum a geração aleatória dos cromossomos (PASANDIDEH, 2011).

A qualidade das soluções da população é avaliada por meio da função de avaliação (*fitness*), que tem como conceito básico representar de forma concisa as respostas. A função de avaliação também pode conter penalidades para a violação de restrições dos problemas de otimização combinatória.

As principais regras de seleção dos indivíduos que permanecerão vivos na próxima geração  $P(t + 1)$  podem ser divididas em determinísticas e probabilísticas. As regras determinísticas, como o próprio nome diz, selecionam os cromossomos de acordo com regras fixadas. Já as probabilísticas, permitem uma variação nessas regras. O conceito geral do procedimento de seleção é a permanência das melhores soluções.

Os operadores dependem da representação dos cromossomos e, conseqüentemente, do espaço de soluções do problema. Os operadores podem ser das mais diversas formas e tem como objetivo alterar as soluções da população. O procedimento de cruzamento combina duas soluções da população para gerar uma nova solução. A mutação altera uma solução de acordo com uma distribuição de probabilidade.

As regras de paradas também são definidas de acordo com o problema. As regras mais usuais são definidas quanto ao número de gerações e quanto à variação da melhoria das soluções.

(PASANDIDEH, 2011) usou um algoritmo genético para resolver um problema de estoque de peças de reposição, dois estágio (um depósito e N bases), sendo a política de revisão contínua (r,Q). O problema foi formulado para minimizar o custo total de inventário

sujeito as restrições de custo total de investimento, frequência de ordens de ressuprimento e esperança de *backorders*.

Outro uso de algoritmo genético para controle de inventário de itens de reposição pode ser encontrado em (MARSEGUERRA, 2004). Os trabalhos feitos nessa área de pesquisa utilizam a simulação como técnica para a avaliação das respostas “função de avaliação” devido à alta complexidade dos custos envolvidos para controle dos itens de reparação.

Segundo (CHWIF, MEDINA, 2006), o trabalho de (MARSEGUERRA, 2004) pode ser definido como uma Simulação com Otimização (problema de otimização em que a função objetivo, as restrições, ou ambas, são respostas que somente podem ser avaliadas pela simulação computacional).

(FANGGENG, 2010), em uma conferência, propôs um algoritmo genético para a solução do problema METRIC descrito por (SHERBROOKE, 1968). Em outra conferência, os autores propuseram novas modificações nos operadores do algoritmo e concluíram ter encontrado melhores resultados.

## **2.6 Simulação de Eventos Discretos**

O processo de simulação de eventos discretos pode ser descrito como um modelo representativo de um sistema real e a condução de experimentos para avaliar o comportamento de suas variáveis (SHANNON, 1975).

Sistemas reais com complexidade elevada entre as suas variáveis são difíceis de serem avaliados por meio de modelos matemáticos algébricos ou modelos probabilísticos, sendo o uso da simulação um recurso valioso na análise destes.

De acordo com (LAW, 2007), a simulação de eventos discretos lida com a modelagem de sistemas que evoluem no tempo, por meio de uma representação em que as variáveis de estado mudam instantaneamente em pontos separados no tempo. Para este contexto, um sistema pode ser definido como um conjunto de elementos que interagem entre si no desempenho de uma função, visando alcançar algum objetivo.

Segundo (CHWIF, MEDINA, 2006) os modelos de simulação são definidos como estáticos (onde a variável tempo não é considerada para avaliar o sistema), dinâmicos (quando o sistema é influenciado pela variável tempo); determinísticos ou estocásticos (se as variáveis do sistema analisadas podem ser descritas como variáveis aleatórias) e contínuos ou discretos ( como é realizada a contagem do tempo no processo de simulação).

Também em (MEDINA e SHWIF, 2006), os autores apresentaram um conjunto de técnicas de simulação de eventos discretos que mostra os requisitos básicos para a sua modelagem e execução. Os autores, entre muitos critérios, definem conceitos básicos como regime transiente, regime permanente e comparação entre alternativas diferentes.

### 3. Metodologia

#### 3.1 Simulação com Otimização

Para a avaliação da técnica de *bootstrap* como em (VISWANATHAN e ZHOU, 2008), em um modelo de estoque multi-estágio, foi utilizada a técnica de Simulação com Otimização. Esta técnica se diferencia das heurísticas puras, pois as soluções geradas durante os passos do procedimento de otimização são avaliadas em um processo de simulação e não avaliadas por uma função exata.

A figura mostra a relação entre o procedimento de otimização e o processo de avaliação. A heurística utilizada neste trabalho foi o algoritmo genético (MICHALEWICZ, 2000) e o processo que avalia as soluções, a simulação de eventos discretos (MEDINA e SHWIF, 2006) combinado com o *bootstrap*. No procedimento deste trabalho, sempre que é criada uma nova solução, a função objetivo é avaliada pelo procedimento de simulação.

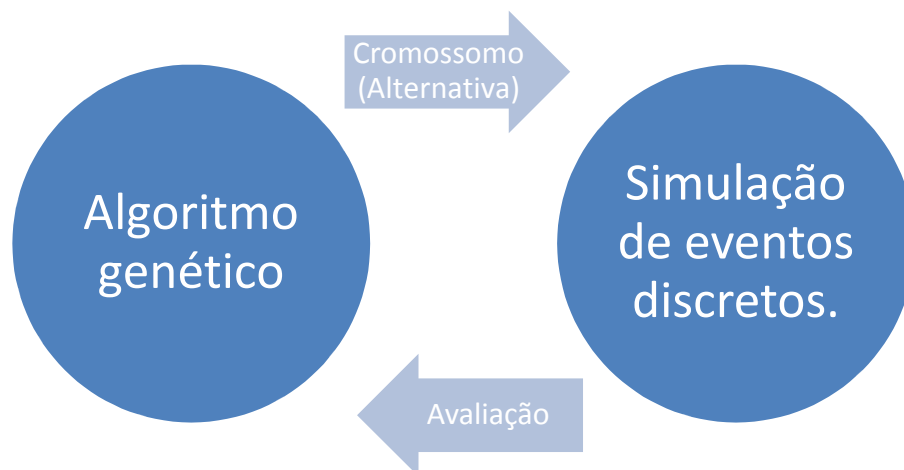


Figura 3: Modelo de Simulação com Otimização

Como é feito usualmente, as comparações em relação à qualidade da previsão de demandas de itens de MRO entre os métodos de estimativas da média e variância (suavização exponencial, método de Croston, método de Croston modificado) e as técnicas de *bootstrap*

(VISWANATHAN e ZHOU, 2010) e (WILLEIMAN, 2004) são realizadas através de resultados de simulações discretas, a técnica, Simulação com Otimização, mostrou-se uma opção razoável para avaliar a técnica de *bootstrap* em um modelo multi-estágio.

Outro fator importante que justifica o uso da Simulação com Otimização para resolver o problema de alocação de estoque multi-estágio, é que este se trata de um problema de (np completo) (PASANDIDEH, 2011).

O modelo de programação inteira utilizado para otimizar o sistema de estoque multi-estágio considera dois dos três custos básicos da política de controle de estoque. Os custos considerados são a expectativa de custo *backorder* e a expectativa de custo de manutenção de estoque.

A formulação matemática do modelo está descrita abaixo:

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n C_B E(\text{Backorder}_{ij}) + \sum_{i=1}^n \sum_{j=1}^n C_E E(s_{ij}) \quad (15)$$

$$\text{s. a. } \sum_i \sum_{j=1}^n c_i s_{ij} \leq C \quad (16)$$

$$s_{ij} \in Z \quad (17)$$

$s_{ij}$  = quantidade de produto  $i$  na base  $j$

$c_i$  = custo de alocar um produto  $i$

$C$  = custo total de produtos a serem alocados

$E(\text{Backorder}_{ij})$  = expectativa de *backorder* do produto  $i$  na base  $j$

$E(s_{ij})$  = expectativa de quantidade de estoque do produto  $i$  na base  $j$

$C_B$  = Custo de *backorder*

$C_E$  = custo de manutenção de estoque

A função objetivo minimiza a quantidade esperada de custo de estoque de produto  $s_{ij}$  em todas as bases, incluindo o depósito central, e também minimiza a expectativa de custo de *backorder* de todos os produtos em todas as bases, excluindo o depósito central. A restrição (1) do problema limita a quantidade de produtos alocados no sistema a um capital  $C$ .

A restrição que limita a quantidade de capital total do sistema foi proposta por (SHERRBROOKE, 1968) e esta representa um problema da mochila. O autor em seu trabalho alega que os gestores de estoque não possuem um orçamento ilimitado o que crítica

políticas que tem como alvo níveis de atendimentos de demandas pré-fixados.

### 3.2 Simulação

Para a avaliação das soluções criadas durante as iterações do algoritmo genético foi utilizada a técnica de simulação de eventos discretos descrita em (CHWIF, MEDINA, 2006). O modelo de simulação foi criado baseado nas condições descritas em (SHERBROOKE, 2004) para o problema de estoque multi-estágio de itens MRO.

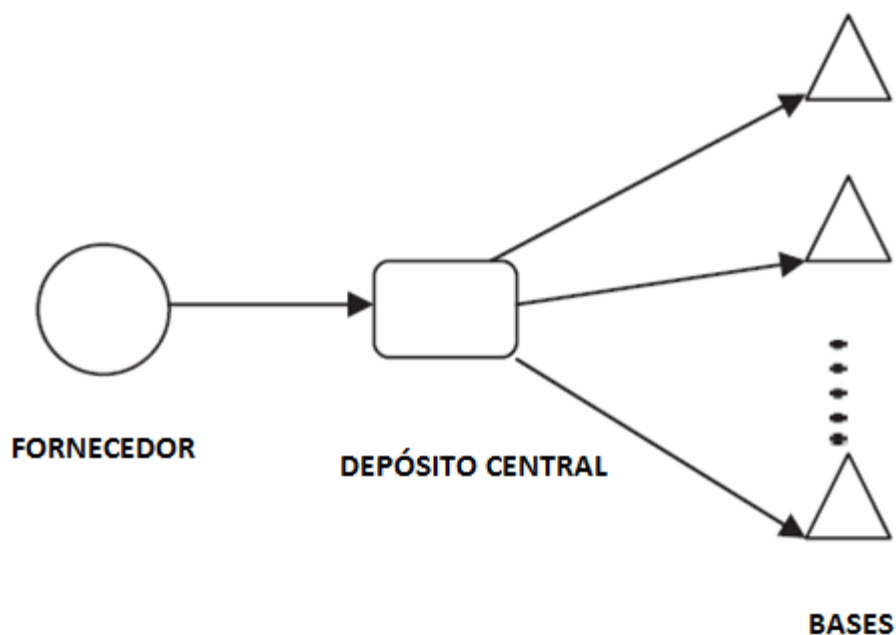


Figura 4: Modelo de Simulação

A figura acima retirada de (SHERRBROOKE, 2004) mostra uma visão geral do problema de estoque descrito pelo mesmo autor. Nesta figura, as setas representam o fluxo de ressurgimento dos itens MRO e a relação entre os níveis do problema.

O depósito central concentra todos os pedidos de ressurgimento feitos ao fornecedor e distribui os itens para as bases com um *lead time*  $T_1$ . Os pedidos feitos pelo depósito central possuem *lead time*  $T_2$  e ambos os *lead times* são independentes dos produtos  $i$  requisitados.

As condições de contorno do processo de simulação de eventos discretos são as mesmas utilizadas por (SHERBROOKE, 20006) e estão resumidas abaixo:

- As peças de reposição são estocadas nas bases e no depósito.
- Quando uma peça falha, o consumidor é atendido na base se ela possui a peça em

estoque; quando a base não possui a peça para reposição, ela envia um pedido para o depósito e é gerado um *backorder*.

- *Single Indenture*, ou seja, os itens de reparação, não são avaliados hierarquicamente (itens pais, itens filhos).

- A decisão de reparar a peça na base não depende do seu estoque ou da sua capacidade de trabalho.

- A base é ressuprida somente pelo depósito.

- A política de estoque  $(s - 1, s)$  é apropriada para todos os itens em todas as bases e depósito central.

- A demanda por peças de reposição nas bases permanecerá constantes ao longo do tempo.

### **3.2.1 *Bootstrap* no procedimento de simulação de eventos discretos**

O principal objetivo do trabalho é avaliar o procedimento de *bootstrap* em um sistema de estoque de itens de reparação multi-estágio. A técnica de amostragem possui várias adaptações na literatura, (VISWANATHAN e ZHOU, 2008) e (WILLEMAIN, 2004), e várias comparações com métodos paramétricos, (BOYLAN e SYNTETOS, 2006) et al.

Nos artigos pesquisados, as comparações são realizadas sempre em sistemas de estoques simples, nos quais o ponto de reposição do produto depende somente da sua demanda direta. Este trabalho procura testar o *bootstrap* em um sistema multi-estágio onde o ponto de reposição dependerá da sua demanda direta e de demandas ocorridas em outros pontos do sistema.

Como descrito ao longo da metodologia, a técnica de *bootstrap* foi testada em um problema estoque multi-estágio com função objetivo que minimiza os custos de *backorders* e de manutenção de estoques e utiliza as premissas encontradas em (SHERBROKE, 2006).

O *bootstrap* foi inserido no processo de simulação de eventos discretos como um gerador de demandas de itens MRO. Ao longo da simulação, em cada base de cada produto, é gerada uma demanda de acordo com o processo de *bootstrap* baseado (VISWANATHAN e ZHOU, 2008).

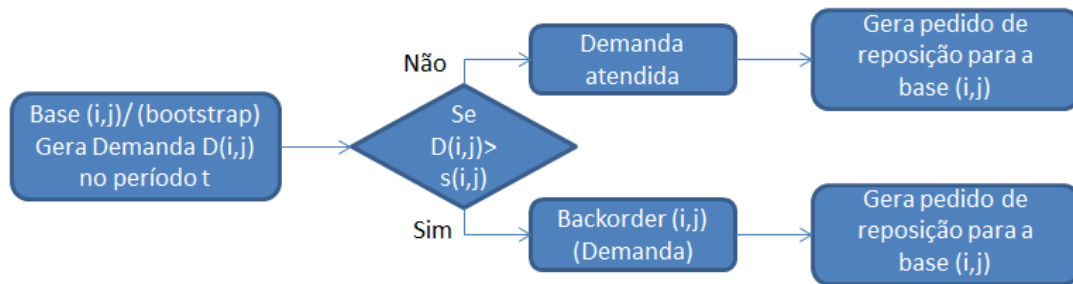


Figura 5: Gerador de demanda *bootstrap*

A figura 5 mostra como o procedimento de *bootstrap* é inserido na simulação e como a demanda gerada por ele percorre o processo de simulação. Sempre que há chegada de demanda no sistema acontece um pedido de reposição no depósito central, o que influencia diretamente sua demanda.

Se a demanda é maior que o estoque atual do produto  $i$  da base  $j$ , ela é atendida prontamente e acontece um pedido de reposição de estoque. Se a demanda é menor, um *backorder* e um pedido de reposição de estoque são gerados.

Em (VISWANATHAN e ZHOU, 2008), os autores propõe a inserção de dois histogramas para gerar a demanda durante o lead time, um para o tamanho das demandas e outro para o tamanho dos intervalos entre demandas positivas. Este trabalho também utiliza estes dois histogramas para gerar as demandas dos itens de reposição.

Utilizando o histograma com as frequências dos intervalos entre as demandas, o gerador de demandas calcula o tempo de chegada da demanda na simulação. Quando o tempo da simulação coincide com o tempo de chegada da demanda, o procedimento calcula o tamanho da demanda por meio do segundo histograma.

O método proposto e baseado na técnica de (VISWANATHAN e ZHOU, 2008) é descrito a seguir:

**Método de *bootstrap* modificado baseado na técnica de (VISWANATHAN e ZHOU, 2008)**

1. Gere os histogramas com os dados da demanda histórica (tamanho das demandas positivas e intervalo entre as demandas);
2. Gere o período que ocorrerá a demanda no sistema de acordo com o histograma do intervalo entre demandas obtido no passo 1.

3. Quando o tempo da simulação for igual ao calculado no passo anterior, gere o tamanho da demanda de acordo com o histograma do tamanho das demandas positivas;
4. Repita os passos 2 – 3 ao longo da simulação;

As demandas positivas e os tempos de chegadas das demandas são gerados por meio das funções inversas dos histogramas encontrados no passo 1. Já os valores das duas variáveis são gerados através de números aleatórios que correspondem às frequências dos valores das variáveis nos seus respectivos histogramas.

Para comparar o modelo de simulação discreto com o *bootstrap* como gerador de demandas foi também utilizada a distribuição de Poisson. A distribuição de Poisson assim como a técnica de amostragem foi inserida como geradora de demandas de itens M.R.O em cada base do sistema de estoque multi-estágio como mostra a figura 6.

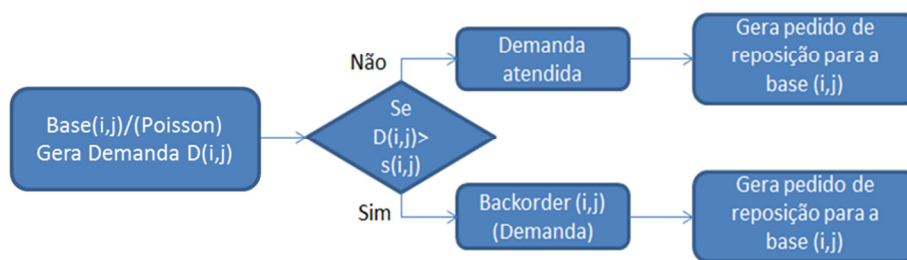


Figura 6: Gerador de demanda Poisson

Para a geração das demandas dos itens M.R.O com a distribuição de Poisson foi calculada a média aritmética das demandas históricas e essa média foi utilizada como parâmetro da distribuição de Poisson.

### 3.2.2 Número de Replicações e o Dimensionamento do Tamanho das Corridas

Dois parâmetros fundamentais, em um processo de simulação de eventos discretos, são o número de replicações de uma simulação e o dimensionamento do tamanho da corrida da replicação. O primeiro parâmetro, trata do número de corridas necessárias para a construção do intervalo de confiança dos valores a serem avaliados na simulação, e o segundo parâmetro refere-se ao intervalo de tempo necessário para que a simulação

represente adequadamente o processo real que está sendo avaliado.

Para o cálculo do número de replicações em uma rodada de simulação foi utilizada a técnica descrita em (CHWIF, MEDINA, 2006). A técnica denominada Amostra Piloto, consiste em rodar uma amostra piloto com um número de replicações pré-determinado e calcular a precisão (metade do intervalo de confiança) da rodada pela fórmula:

$$h = t_{n-1; \alpha/2} \frac{s}{\sqrt{n}} \quad (18)$$

onde  $t_{n-1; \alpha/2}$  é o  $(1 - \alpha/2)$  percentil da distribuição  $t$  de *Student* com  $n - 1$  graus de liberdade,  $s$  desvio padrão e  $n$  é o número de replicações da amostra.

Em seguida, de acordo com a técnica descrita pelos autores, pode-se encontrar o tamanho do número de replicações para a precisão desejada pela fórmula:

$$n^* = \left\lceil n \left( \frac{h}{h^*} \right)^2 \right\rceil \quad (19)$$

onde  $n^*$  é o número de replicações para a precisão desejada,  $n$  o número de replicações da amostra piloto,  $h^*$  a precisão desejada e  $h$  a precisão da amostra piloto. Caso a precisão encontrada não seja a desejada, deve-se repetir o cálculo 19 até se encontrar o número de replicações adequado.

O resultado encontrado para o número ideal de replicações para a realização dos testes com o *bootstrap* em um sistema multi-estágio foi 22 replicações.

O segundo ponto refere-se ao intervalo de tempo necessário para que a simulação represente adequadamente o processo real que está sendo avaliado. Para o dimensionamento do tamanho da corrida, um conceito fundamental é o período transiente e o período permanente.

Segundo os mesmos autores (CHWIF, MEDINA, 2006), o período transiente é o início do processo de simulação, na qual as condições iniciais exercem grandes influências. Já o regime permanente é aquele em que as condições iniciais não interferem nos valores encontrados no processo de simulação de eventos discretos.

Para o cálculo do dimensionamento de corridas (eliminação do período transiente), os autores (CHWIF, MEDINA, 2006) sugerem que sejam feitas corridas longas e que se calcule médias móveis dos valores observados. Com os valores das médias móveis calculadas, o

ponto de início do regime permanente será aquele em que os valores se estabilizam. Neste trabalho foram calculadas médias móveis para 10, 100 e 500 dias e foi utilizado o valor do custo total do sistema.

A figura abaixo mostra que o regime permanente começa a partir de 1000 dias de simulação (as médias móveis se aproximam de 80 unidades monetárias), ou seja, as corridas devem possuir um descarte de 1000 dias na contabilização dos valores simulados.

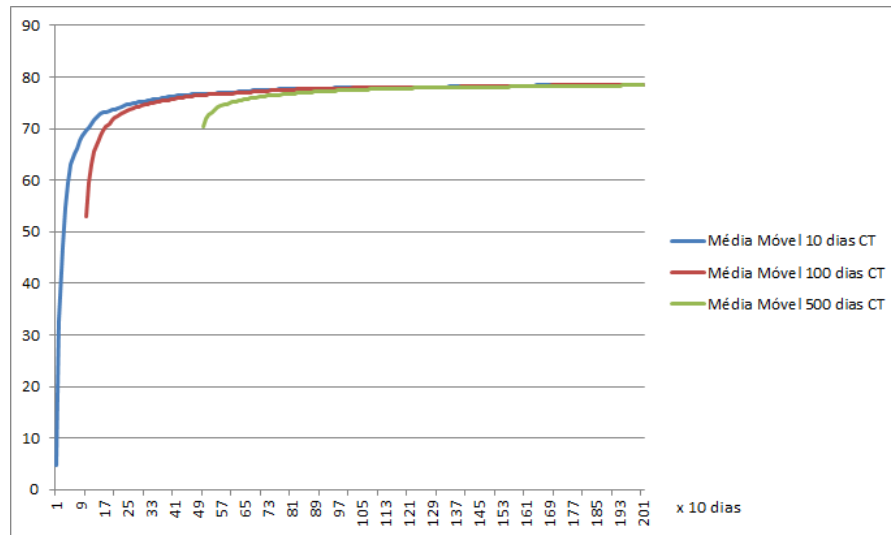


Figura 7: Cálculo do Regime Transiente

Como os itens de reposição têm valores elevados e possuem baixa demanda foi determinado o tamanho real da corrida em 10 anos (1000 dias + 3650 dias).

### 3.2.3 Comparações entre soluções

Durante o processo de otimização é necessário comparar as soluções do conjunto  $P(t)$  para encontrar a melhor solução do problema de estoque. A alternativa escolhida, descrita em (CHWIF, MEDINA, 2006), para comparar as soluções, constrói um intervalo de confiança das diferenças das replicações, par a par do custo total, entre as duas soluções da população e de acordo com este intervalo define qual solução é a melhor.

Com o intervalo com 95% de confiança das diferenças das médias do custo total (Função objetivo) construído, a técnica escolhe qual é a melhor opção: se o intervalo possui seus dois limites negativos (primeira alternativa é a melhor); se possui os dois limites positivos (segunda alternativa é a melhor); se possui os dois limites com sinais diferentes

(alternativas são equivalentes ou diferentes).

Este procedimento é usado correntemente durante toda Simulação com Otimização, sempre que algum procedimento procura ordenar um conjunto de qualquer elemento do conjunto  $P(t)$ .

1- Calcular as diferenças:

$$d_i = x_{1i} - x_{2i} \quad (18)$$

2- Calcular a média e o desvio padrão das diferenças das soluções

$$\bar{d} = \frac{\sum_{i=1}^b d_i}{b} \quad (19)$$

$$s_d = \sqrt{\frac{\sum_{i=1}^b (d_i - \bar{d})^2}{b-1}} \quad (20)$$

3- Construir um intervalo de confiança  $[\theta_1, \theta_2]$  para as médias das diferenças;

$$\bar{x} \pm t_{n-1; \alpha/2} \frac{s}{\sqrt{n}} \quad (21)$$

4- Se o intervalo de confiança  $[\theta_1, \theta_2]$ :

Contiver o 0, isto é:  $\theta_1 < 0$  e  $\theta_2 > 0$ , então nada pode ser concluído sobre a diferença entre as médias das alternativas.

Se o intervalo cair totalmente à direita do 0, isto é,  $\theta_1 > 0$  e  $\theta_2 > 0$ , então a média da alternativa 1 é a maior.

Se o intervalo cair totalmente à esquerda do 0, isto é,  $\theta_1 < 0$  e  $\theta_2 < 0$ , então a média da alternativa 2 é a maior.

### 3.3 Algoritmo Genético

Existem várias categorias de heurísticas na literatura e suas aplicações são diversas. Existe aplicação de meta-heurísticas em problemas clássicos, como o problema de estoque, o do caixeiro viajante e da alocação de facilidades. Devido a esta grande flexibilidade das

meta-heurísticas, as mais diversas podem ser usadas como o algoritmo de otimização em um processo de Simulação com Otimização.

Neste trabalho, a heurística utilizada no processo de otimização foi o algoritmo genético. Assim como em (MARSGUERRA, 2004), que em um processo de Simulação com Otimização utilizou um algoritmo genético em conjunto com a simulação de Monte Carlo, este trabalho fez o uso deste mesmo algoritmo, porém utilizou a simulação discreta como processo de simulação.

O algoritmo genético foi apresentado por (HOLLAND, 1975) e foi aplicado nos mais diversos problemas. Esta classe de algoritmo tem como características principais trabalhar com um conjunto de respostas, denominado população, e com procedimento de cruzamento e mutação.

O algoritmo genético utilizado neste trabalho é baseado no pseudo-código descrito em (MICHALEWICZ, 2000) e está descrito abaixo:

**Procedimento:** Algoritmo genético / Simulação com Otimização

**begin**

$t = 0$

Gerar um conjunto de soluções iniciais aleatórias  $P(0)$ ;

Avaliar cada solução da população  $P(0)$  com o processo de simulação e ordenar as respostas;

**While** (  $t \leq \text{No. de gerações}$  )

**Begin**

$t = t + 1$

Selecionar as melhores respostas de acordo com a probabilidade  $p$ ;

Fazer operações de cruzamento com as soluções selecionadas;

Avaliar as novas soluções com o processo de simulação e atualizar população  $P(t)$ ;

Fazer os procedimentos de mutação de acordo com a probabilidade  $p_2$  nos candidatos da população  $P(t)$ ;

Avaliar as novas soluções com o processo de simulação e atualizar população  $P(t)$ ;

**If** ( $t = \textit{critério para renovar a população}$ )

Guardar as melhores soluções de  $P(t)$ ;

Gerar novas soluções e avaliá-las de acordo com o processo de simulação;

Inserir as melhores respostas no conjunto de soluções  $P(t)$ ;

**end**

**end**

**end**

O algoritmo genético possui cada iteração marcada pela variável  $t$  e o procedimento inicia esta variável em  $t = 0$ . Esta variável marca a quantidade de gerações criada pelo algoritmo. Também como condição inicial, o pseudo-código cria a população  $P(0)$  por meio de um procedimento aleatório e avalia e ordena as soluções contidas na população inicial de acordo com os resultados obtidos na simulação discreta.

O processo evolucionário do algoritmo genético está representado no primeiro e único *loop* do algoritmo. Neste *loop* acontece o procedimento de cruzamento, o procedimento de mutação e o procedimento de renovação da população.

O procedimento de cruzamento gera uma nova solução a partir do cruzamento de duas soluções contidas na população atual. O procedimento de mutação causa uma pequena alteração também em uma solução contida na população  $P(0)$ . O procedimento de renovação da população renova o conjunto de soluções quando o número de gerações atinge um valor pré-fixado.

O algoritmo genético termina quando o número de gerações  $t$  atinge um determinado valor. Ao longo das iterações do algoritmo, as respostas vão melhorando e o resultado final é a melhor solução encontrada até o critério de parada.

Por se tratar de um processo de Simulação com Otimização, ao longo de todos os passos descritos acima, as respostas são avaliadas de acordo com o processo de simulação discreta e ordenadas de acordo com o procedimento de comparação de soluções descrito em (MEDINA e SHWIF, 2006).

O número total de gerações utilizado para otimizar o sistema de estoque multi-estágio utilizando o *bootstrap* inserido no procedimento de simulação de eventos discretos e a distribuição de Poisson como geradora de demandas foi cento e 20 gerações.

### 3.3.1 Cromossomo

Uma das questões fundamentais na construção de uma heurística é a forma de representação das respostas. Assim, como nas pesquisas de (AXSATER, 2003) (SHERRBROOKE, 1968) e (PASANDIDEH, 2011), este trabalho utiliza uma matriz para representar uma solução do problema de estoque multi-estágio.

Na matriz resposta as linhas  $i$  representam os produtos e as colunas  $j$  representam as bases. O depósito central é representado por  $j = 0$  e a complexidade do problema cresce com o número de produtos e/ou número de depósitos.

	Depósito	base 1	base 2	base 3	base 4
item 1	2	2	2	3	2
item 2	3	3	2	4	3
item 3	4	4	2	2	1
item 4	4	5	2	1	3
item 5	1	3	2	3	1
item 6	2	2	1	2	2

Figura 8: Exemplo de Cromossomo

A figura acima mostra um exemplo de resposta do problema de estoque multi-estágio criado pelo procedimento implementado. No exemplo, foi alocadas 4 unidades do produto  $i = 4$  no depósito central  $j = 0$ , 5 unidades na base  $j = 1$ , 2 unidades na base  $j = 2$ , 1 unidade na base  $j = 3$  e 3 unidades na base  $j = 4$ .

### 3.3.2 Solução Inicial

Existem várias maneiras de construir o primeiro conjunto de soluções da população  $P(0)$  do algoritmo genético. Este passo pode ser escrito com heurísticas simples que geram respostas aproximadas (constroem respostas com as características do problema) ou por procedimentos aleatórios (criam soluções sem considerar as características do problema).

O procedimento aleatório implementado está descrito no pseudo-código abaixo:

**Procedimento:** Algoritmo genético / Solução inicial

**Begin**

Criar um conjunto  $P(t)$  com soluções vazias;

**Para** toda solução pertencente a  $P(t)$

**Begin**

Selecionar uma solução  $S$  de  $P(t)$ ;

**While** ( $\sum_i^n \sum_{j=1}^n c_i s_{ij} \leq C$ )

**Begin**

$i = \text{sortear um produto};$

$j = \text{sortear uma base ou depósito};$

Atribuir valores a solução  $S$ :  $s_{ij} = s_{ij} + 1$ ;

**end**

Avaliar a resposta  $S$  de acordo com o processo de simulação;

**end**

**end**

O procedimento de solução inicial do algoritmo genético, que constrói uma solução aleatória, procura alocar todo o capital  $C$  nas soluções criadas. Este procedimento sorteia um produto  $i$  e uma base  $j$ , aloca uma unidade de estoque aos respectivos  $i$  e  $j$  selecionados e desconta o valor gasto nessa operação. O procedimento para quando todo capital  $C$  é alocado.

### 3.3.3 Cruzamento

O procedimento de Cruzamento, em um algoritmo evolucionário, combina duas soluções da população  $P(t)$  escolhidas por meio de um critério que, geralmente, é baseado no

*fitness* de cada solução. A combinação entre duas soluções pais é feita por meio de mecanismos que aproveitam as suas características para gerar uma nova solução (solução filho).

**Procedimento:** Algoritmo genético / Procedimento de cruzamento

**begin**

Criar um conjunto  $P^*(t)$ ;

**Para** 1 até tamanho de  $P(t)/2$ ;

**begin**

Selecionar uma solução  $S^1$  do conjunto de resposta  $P(t)$  de acordo com  $p_1$ ;

Selecionar uma segunda solução  $S^2$  do conjunto de resposta  $P(t)$  de acordo com  $p_2$ ;

Criar uma solução vazia  $S^*$ ;

**Para** 1 até o número de produtos  $i$ ;

**Para** 0 até o número de bases  $j$ ;

**Begin**

**Begin**

Calcular o coeficiente  $f_c$  da relação das soluções  $S^1$  e  $S^2$  de acordo com

a equação  $f_c = \frac{fitness^1}{fitness^1 + fitness^2}$ ;

Atribuir valores a solução  $S^*$ :  $s^*_{ij} = f_c s^1_{ij} + (1 - f_c) s^2_{ij}$ ;

**end**

**end**

Avaliar a resposta  $S^*$  de acordo com o processo de simulação;

Inserir a resposta  $S^*$  no conjunto  $P^*(t)$ ;

**end**

Somar os conjuntos  $P(t)$  e  $P^*(t)$ ;

Ordenar esse novo conjunto de acordo com as melhores respostas e selecionar  $P(t)$ ;  
**end**

O procedimento de cruzamento implementado seleciona as candidatas a soluções pais de acordo com o critério da roleta (este critério associa uma probabilidade  $p$  a cada solução de acordo com o peso que seu *fitness* representa para a população  $P(t)$ ). Quanto maior o peso da solução maior a sua probabilidade de ser selecionada e quanto menor o seu peso menor a probabilidade.

No pseudo-código descrito acima, são gerados dois números aleatórios que corresponderam às probabilidades das soluções serão selecionadas.

Para combinar as duas soluções pais o primeiro passo é calcular o valor  $f_c$  do par. O cálculo é realizado através da equação  $f_c = \frac{fitness^1}{fitness^1 + fitness^2}$ , que representa quanto melhor uma solução é em relação à outra.

O segundo passo é realizado pela equação  $s^*_{ij} = f_c s^1_{ij} + (1 - f_c) s^2_{ij}$  que atribui a cada célula da matriz da solução filho o valor correspondente a uma porcentagem da solução 1 e uma porcentagem da solução 2 de acordo com o valor de  $f_c$  calculado.

O número de iterações do procedimento é determinado pelo tamanho da população  $P(t)$  dividido por dois. As novas soluções criadas são somadas ao conjunto  $P(t)$  inicial e este novo conjunto é ordenado e posteriormente reduzido até o tamanho inicial de  $P(t)$ .

### 3.3.3 Procedimento de Mutação 1

Um dos procedimentos mais importantes dentro de um algoritmo genético é o procedimento de mutação. Este procedimento tem como objetivo principal propor pequenas alterações em algumas soluções da população  $P(t)$  para que o algoritmo genético não fique preso em mínimos locais.

Assim como todos os passos de um algoritmo evolucionário, existem diversas maneiras de escrever um procedimento de mutação. Neste trabalho são utilizadas 3 rotinas de mutações gulosas, ou seja, rotinas que procuram as melhores mudanças possíveis de acordo com algum critério pré-estabelecido.

**Procedimento:** Algoritmo genético / Procedimento de *mutação 1***begin**

Selecionar uma solução  $S$  de  $P(t)$  de acordo com  $p_{mut-1}$ ;

Criar uma lista  $L$  com os  $s_{ij}$  da solução selecionada ordenados de acordo com os custos de manutenção de estoque;

Criar uma lista  $L^*$  com os  $s_{ij}$  da solução selecionada ordenados de acordo com os custos de *backorder*;

**Para** 1 até tamanho No. De iterações;

**begin**

Selecionar um  $s_{ij}$  da lista  $L$ ;

Selecionar um  $s^*_{ij}$  da lista  $L^*$ ;

$$s_{ij} = s_{ij} - 1;$$

$$s^*_{ij} = s^*_{ij} + 1;$$

**end**

Avaliar a resposta  $S$  de acordo com o processo de simulação;

**end**

A primeira rotina de mutação procura tirar uma unidade da célula da matriz solução  $s_{ij}$  com o maior custo de manutenção de estoque, e alocar esta mesma unidade de produto a célula  $s^*_{ij}$  com o maior custo de *backorder*.

Este procedimento seleciona uma solução de  $P(t)$  de acordo com a probabilidade  $p_{mut-1} = 0,16666$  e tem o número de iterações limitados a 3.

**3.3.4 Procedimento de Mutação 2**

O segundo procedimento de Mutação procura alocar uma unidade de produto  $i$  da célula  $s_{ij}$  com o maior valor de manutenção de estoque, ao seu respectivo depósito central

$S^*_{i0}$ .

Esta rotina também é limitada em 3 iterações, tem objetivo de testar qual é a vantagem de alocar uma unidade de produto  $i$  em seu depósito central e escolhe qual elemento da população passará pela mutação, de acordo com a probabilidade  $p_{mut-2} = 0,16666$ .

**Procedimento:** Algoritmo genético / Procedimento de *mutação 2*

**begin**

Selecionar uma solução  $S$  de  $P(t)$  de acordo com  $p_{mut-2}$ ;

Criar uma lista  $L$  com os  $s_{ij}$  da solução selecionada ordenados de acordo com os custos de manutenção de estoque;

**Para** 1 até tamanho No. De iterações;

**begin**

Selecionar um  $s_{ij}$  da lista  $L$ ;

$s_{ij} = s_{ij} - 1$ ;

$s^*_{i0} = s^*_{i0} + 1$ ;

**end**

Avaliar a resposta  $S$  de acordo com o processo de simulação;

**end**

### 3.3.4 Procedimento de Mutação 3

O terceiro procedimento de mutação divide o problema de estoque multi-estágio em sub-problemas  $i$ , ou seja, considera cada linha da matriz um problema independente. Dessa forma, esta rotina procura balancear o gasto com cada produto  $i$  e sua contribuição ao custo total do problema.

Como neste trabalho o problema de estoque multi-estágio é avaliado de acordo com os custos de manutenção de estoque e de *backorder* em cada célula, a relação

$\frac{\sum_0^j s_{ij}}{\sum_0^j s_{ij} + b_{ij}}$  mostra o quanto de recurso está sendo utilizado pelo sub-problema para o atendimento das demandas.

Sendo assim, sub-problemas que possuem a relação descrita acima próximas de um, possuem excesso de produtos alocados e relações próximas de zero, escassez de produtos alocados. O procedimento de mutação retira uma unidade de produto do sub-problema com excesso de recursos e aloca esta unidade ao sub-problema com escassez de recursos.

Esta rotina também é limitada em 3 iterações, tem objetivo de balancear a quantidade de recursos alocados entre os produtos  $i$  como descrito acima e escolhe qual elemento da população passará pela mutação de acordo com a probabilidade  $p_{mut-3} = 0,16666$ .

**Procedimento:** Algoritmo genético / Procedimento de *mutação 3*

**begin**

Selecionar uma solução  $S$  de  $P(t)$  de acordo com  $p_{mut-3}$ ;

Avaliar o custo total de manutenção de estoque para cada produto  $i$  em relação ao seu

custo total  $\frac{\sum_0^j s_{ij}}{\sum_0^j s_{ij} + b_{ij}}$  ;

Criar uma lista  $L$  com os  $i$  da solução selecionada ordenados de acordo com a relação acima;

**Para** 1 até tamanho No. De iterações;

**begin**

Selecionar um  $i$  da lista  $L$  entre os primeiros candidatos;

Selecionar um  $s_{ij}$ ;

Selecionar um  $i^*$  da lista  $L$  entre os últimos candidatos;

Selecionar um  $s_{i^*j}$ ;

$s_{ij} = s_{ij} - 1$ ;

$s^*_{i*j} = s^*_{i*j} + 1;$

**end**

Avaliar a resposta  $S$  de acordo com o processo de simulação;

**end**

### 3.3.4 Procedimento de Renovação da População

Outro mecanismo utilizado para fugir de mínimos locais é o procedimento de renovação da população. Como ao longo das iterações do algoritmo genético, o conjunto de soluções vai se tornando homogêneo, este procedimento tem como objetivo renovar grande parte das soluções presentes no conjunto  $P(t)$ .

O procedimento de renovação da população guarda as 5 melhores soluções do conjunto  $P(t)$ , gera um novo conjunto com o procedimento de solução inicial e, por meio do procedimento de cruzamento utiliza um candidato do novo conjunto de soluções e um candidato escolhido entre as 5 melhores soluções para gerar uma nova solução.

Ao término das iterações é gerado um novo conjunto  $P(t)$  com as soluções criadas através do procedimento de *crossove*., acrescida das 5 melhores soluções encontradas. Este procedimento ocorre a cada 10 iterações do algoritmo genético.

**Procedimento:** Algoritmo genético / Procedimento de *renovação da população*

**begin**

Criar uma lista  $L$  com as 5 melhores soluções da população  $P(t)$ ;

Excluir todas as soluções da população  $P(t)$ ;

Criar uma nova população  $P^*(t)$  de acordo com o procedimento de *Solução inicial*;

**Para** 1 até número de iterações;

**begin**

Selecionar uma solução de  $L$  de acordo com  $p1$ ;

Selecionar uma solução de  $P^*(t)$  de acordo com  $p2$ ;

Fazer o procedimento de cruzamento com as soluções selecionadas;

Inserir a nova solução em  $P(t)$ ;

**end**

Inserir  $L$  na população  $P(t)$ ;

**end**

### 3.1.5 Parâmetros dos testes

Para o teste do *bootstrap* inserido em uma simulação de eventos discreto utilizou-se diferentes níveis de capital  $C$  com objetivo de mostrar se há diferença de comportamento da técnica para essas diferentes quantidades. Neste trabalho foram utilizadas as quantidades de capital 5000, 10000, 15000 unidades monetárias.

As demandas históricas utilizadas para testar o método de *bootstrap* inserido na simulação de eventos discretos foram geradas de acordo com (VISWANATHAN e ZHOU, 2010). Assim como no referido artigo, os intervalos entre demandas e os tamanhos de demanda (demandas diferentes de zero) foram gerados separadamente.

Para gerar cada série de demanda, foi utilizada uma distribuição estatística para gerar o intervalo entre demandas e outra distribuição estatística para gerar os valores das demandas positivas. Todas as séries de demandas possuem 2000 pontos de demandas positivas. As demandas históricas foram as séries de demandas geradas até a 1000º ponto de demanda positiva. Os testes de simulação foram realizados com as séries de demandas entre a 1001ª e 2000ª demandas positivas.

Neste trabalho foi utilizada a distribuição de Poisson com média 10 para gerar os intervalos entre demandas positivas e a mesma distribuição estatística com médias 1, 3 e 9 para gerar os tamanhos das demandas positivas.

Os custos de todos os itens possuem o mesmo valor de 100 unidades monetárias e o custo de estocagem é igual a uma unidade monetária e o custo de ruptura 5 unidades monetárias.

Outro variável analisada são os valores de *lead time* interno e externo do sistema multi-estágio. O *lead time* interno é o tempo de resposta do atendimento de uma ordem de ressurgimento emitida por uma base para o depósito central e o *lead time* externo o tempo de resposta do atendimento de uma ordem emitida pelo depósito central para um fornecedor externo. Neste trabalho foram utilizados 2 valores de *lead time* interno 10 e 20 dias e *lead*

*time* externo 10 e 20 dias.

Os teste foram separados em 2 grupos como mostra a tabela abaixo, primeiro grupo com *lead time* externo 20 dias e *lead time* interno 10 dias e o segundo grupo com *lead time* externo 10 dias e *lead time* interno 20 dias. Ao todo foram realizadas 36 execuções do algoritmo genético.

Todos os testes e procedimentos foram implementados em java no compilador Netbeans e foram rodadas na plataforma WINDOWS VISTA em um computador pentium dual core com 4 gigabytes de memória ram.

Resumo dos Testes				
grupo	intervalo entre demandas	demandas	lead times	C
1	Poisson (10)	Poisson (1)	lead time externo=20	5000
		Poisson (3)	lead time interno=10	10000
		Poisson (9)		15000
2	Poisson (10)	Poisson (1)	lead time externo=10	5000
		Poisson (3)	lead time interno=20	10000
		Poisson (9)		15000

Tabela 1: Resumo dos testes

#### 4. Resultados

Os parâmetros utilizados para testar a técnica de *bootstrap* comparada a distribuição de Poisson mostram que o algoritmo genético e a simulação de eventos discretos tiveram comportamentos diferentes para cada gerador de demandas de itens M.R.O. Os parâmetros mostram diferenças entre os tempos de execução do algoritmo genético e da simulação assim como diferenças entre a qualidade das soluções ótimas encontradas.

Outro importante resultado encontrado foi que a técnica de *bootstrap* estima padrões mais elevados de demandas do que a distribuição de Poisson combinada com um estimador de sua média.

Assim como nos trabalhos de (WILLEIMAN, 2004) e no anexo A, os resultados mostram que há uma diferença entre modelar a demanda de itens de reposição com a distribuição de Poisson e modelar a demanda com a técnica de *bootstrap*.

O primeiro resultado exemplificado mostra uma solução encontrada para o problema de estoque de itens multi-estágio com demanda gerada com intervalo entre demandas com a

distribuição de Poisson com média igual a 10 e tamanhos de demandas com Poisson com média igual a um para os 2 geradores de demandas. Os tempos de *lead time* usados para estes 2 testes foram externo igual a 20 e interno igual a 10.

poiss 10000 D=1 Int=10				
6	3	3	3	3
6	3	3	3	3
6	3	3	3	3
7	3	3	3	3
7	3	3	3	3
CT=48,52245				

Tabela 2: Solução do modelo com distribuição de Poisson

boot 10000 D=1 Int=10				
6	3	3	3	3
7	3	3	3	3
7	3	3	3	3
6	3	3	3	3
7	3	2	3	3
CT=48,96875				

Tabela 3: Solução do modelo com a técnica de *bootstrap*

Ambos os resultados mostram que grande parte do estoque de itens de reposição foi alocado para o depósito central (coluna 1) e que todas as bases possuem ponto de reposição igual a 3. Outro ponto importante é que as soluções geradas apresentam quantidades de alocações idênticas, fato comum a outras soluções encontradas com outros padrões de demandas e outros parâmetros.

As figuras também mostram que os custos totais encontrados pelas respostas apresentam apenas uma pequena diferença, custo total da distribuição de Poisson igual a 48,70597 e custo total da técnica de *bootstrap* igual 48,96875.

As tabelas 4 a 9 mostram as matrizes dos custos esperados de estoque e do custo esperado de *backorder* das soluções encontradas com *lead time* externo igual a 10 e *lead time* interno igual a 20 com distribuição de Poisson com média igual a um para a geração dos tamanhos de demandas.

poiss 10000 D=1 Int=10				
3	4	4	4	4

3	4	4	4	4
5	4	4	4	4
3	4	4	4	4
2	4	4	5	4
CT=52,8426				

Tabela 4: Solução do modelo com distribuição de Poisson

boot 10000 D=1 Int=10				
6	3	3	3	3
6	3	3	3	3
6	3	3	3	3
7	3	3	3	3
7	3	3	3	3
CT=54,94955				

Tabela 5: Solução do modelo com distribuição gerada pelo *bootstrap*

Expectativa do Custo de Estoque					Soma
0,42355	1,9469	1,9712	2,0332	1,94705	8,3219
0,42435	1,97745	1,91885	1,9851	1,982	8,28775
1,5779	2,0846	2,21565	2,0839	2,0659	10,02795
0,40225	1,9003	1,9188	1,878	1,9598	8,05915
0,151	1,8905	1,8833	2,6718	1,76075	8,35735
Custo Total de Estoque = 43,0541					

Tabela 6: Expectativa do Custo de Estoque – Poisson

Expectativa do Custo de Estoque					Soma
0,4782	1,98875	1,943	1,94535	1,98425	8,33955
0,4714	1,9787	1,9663	1,93185	1,9362	8,28445
0,48755	1,98885	2,0803	1,9832	1,96615	8,50605
0,49475	1,94605	1,95515	1,9346	2,007	8,33755
0,18195	1,8284	2,7861	1,7934	2,63585	9,2257
Custo Total de Estoque = 42,6933					

Tabela 7: Expectativa do Custo de Estoque – *bootstrap*

As tabelas dos custos esperados de estoque e de *backorder* dos dois modelos apresentam os seus respectivos custos para cada produto em todas as bases e depósitos centrais. O custo esperado de estoque utilizando a técnica de *bootstrap* é menor que o custo

esperado de estoque utilizando a distribuição de Poisson. Já as matrizes com os custos esperados de *backorder* mostram que o custo de *backorder* utilizando a técnica de *bootstrap* é maior.

Expectativa do Custo de <i>Backorder</i>					Soma
0,513563	0,5075	0,5895	0,398	0,55925	2,05425
0,505937	0,52125	0,544	0,4455	0,513	2,02375
0,360438	0,40025	0,32425	0,34825	0,369	1,44175
0,546687	0,4615	0,61175	0,6075	0,506	2,18675
0,5205	0,547	0,562	0,27325	0,69975	2,082
					9,7885

Tabela 8: Expectativa do Custo de *backorder* – Poisson

Expectativa do Custo de <i>Backorder</i>					Soma
0,605563	0,57825	0,57725	0,63125	0,6355	2,42225
0,643063	0,625	0,576	0,679	0,69225	2,57225
0,587438	0,656	0,445	0,57475	0,674	2,34975
0,678	0,651	0,72525	0,764	0,57175	2,712
0,55	0,7325	0,33525	0,80925	0,323	2,2
					12,25625

Tabela 9: Expectativa do Custo de *backorder* – *bootstrap*

As tabelas a seguir, apresentam os custos de estoque e de *backorder*, de todos os resultados encontrados excluídos os testados com tamanhos de demandas gerados com a distribuição de Poisson com média 9.

Assim como as tabelas 6 a 9, as tabelas a seguir mostram que as médias dos custos de estoque para a distribuição de Poisson são menores que as alcançadas pela técnica de *bootstrap*. Elas também mostram que as médias dos custos de *backorder* são maiores para a técnica de *bootstrap*. Outro fator importante apresentado é que à medida que o capital  $C$  cresce, crescem os custos esperados de estoque e diminuem os custos esperados de *backorder*.

Resumo dos Testes					Poisson		Bootstsrp	
grupo	intervalo entre demandas	demandas	lead times	C	Estoque	Backorder	Estoque	Backorder
1	poisson (10)	poisson (1)	led time externo=20	5000	9,1576	70,49175	8,41555	71,0875
			led time interno=10	10000	39,3372	9,18525	38,3585	10,61025
				15000	82,1673	0,391	81,4597	0,861
2	poisson (10)	poisson (3)	led time externo=20	5000	0,2955	576,6135	0,12215	582,2363
			led time interno=10	10000	0,37955	350,9815	0,46705	343,5085
				15000	14,88235	147,9855	15,14375	156,8458

Tabela 10: Resumo dos Custos de Estoque e *Backorder*

Resumo dos Testes					Poisson		Bootstsrp	
grupo	intervalo entre demandas	demandas	lead times	C	Estoque	Backorder	Estoque	Backorder
1	poisson (10)	poisson (1)	led time externo=20	5000	9,8503	72,56075	9,7031	77,53725
			led time interno=10	10000	43,0541	9,7885	42,6933	12,25625
				15000	80,3953	1,548	85,54645	1,2875
2	poisson (10)	poisson (3)	led time externo=20	5000	0,0156	581,50075	0,05695	583,2548
			led time interno=10	10000	2,8634	337,62075	3,94585	351,1053
				15000	15,3739	151,563	17,9457	170,8643

Tabela 11: Resumo dos Custos de Estoque e *Backorder*

As tabelas 12 e 13 mostram todos os resultados dos custos totais de todos os testes realizados. A tabela 12 mostra os resultados encontrados com *lead time* externo igual a 20 e *lead time* interno igual a 10. Já a tabela 13, os custos totais, com *lead time* externo igual a 10 e *lead time* interno igual a 20.

Resumo dos Testes						
grupo	intervalo entre demandas	demandas	lead times	C	Poisson	Bootstrap
1	poisson (10)	poisson (1)	led time externo=20	5000	79,64935	79,50305
			led time interno=10	10000	48,52245	48,96875
				15000	82,5583	82,3207
2	poisson (10)	poisson (3)	led time externo=20	5000	576,909	582,3584
			led time interno=10	10000	351,36105	343,97555
				15000	162,86785	171,9895
3	poisson (10)	poisson (9)	led time externo=20	5000	1476,7598	1359,2638
			led time interno=10	10000	1634,8654	1657,3876
				15000	1719,0243	1739,5221

Tabela 12: Resumo dos resultados para *lead time* externo igual a 20 e interno igual a 10

Resumo dos Testes						
grupo	intervalo entre demandas	demandas	lead times	C	Poisson	Bootstrap
1	poisson (10)	poisson (1)	led time externo=10	5000	82,41105	87,24035
			led time interno=20	10000	52,8426	54,94955
				15000	81,9433	86,83395
2	poisson (10)	poisson (3)	led time externo=10	5000	581,51635	583,3117
			led time interno=20	10000	340,48415	355,0511
				15000	166,9369	188,80995
3	poisson (10)	poisson (9)	led time externo=10	5000	1354,9126	1401,2764
			led time interno=20	10000	1578,9076	1564,7634
				15000	1836,8934	1860,8954

Tabela 13: Resumo dos resultados para *lead time* externo igual a 10 e interno igual a 20

Comparando as tabelas acima, podemos notar que os custos totais da primeira tabela são menores que o da segunda tabela. Essas diferenças mostram que diferentes valores de *lead time* externo e interno provoca alteração nos padrões de demandas. Quando se avalia o problema de estoque multi-estágio se avalia qual é a demanda no depósito central.

Os tempos de execução dos dois modelos foram completamente diferentes. Enquanto o modelo que utiliza a técnica de *bootstrap* apresentou tempo de execução com média de 115 minutos e variância 28 minutos, o modelo com a distribuição de Poisson apresentou média de 432 minutos e variância de 89 minutos.

Tempo de Execução			
Poisson		Poisson	
Média	Variância	Média	Variância
115 min	28 min	432 min	89 min

Tabela 14: Tempos de Execução

A principal causa para essa grande diferença nos tempos de execução foi que a distribuição de Poisson durante o processo de simulação emite mais ordens de ressuprimento que a técnica de *bootstrap*. Essa grande quantidade de ordens, geram grandes tamanhos de filas na simulação diminuindo o tempo de processamento.

Os testes com o tamanho de demandas geradas pela distribuição de Poisson com média igual a 9, foram desconsiderados, pois, a qualidade das soluções, principalmente dos modelos com a distribuição de Poisson, foram bastante insatisfatórios.

## 5. Trabalhos Futuros

Outro problema onde se pode testar o *bootstrap* em um sistema de estoque multi-estágio está descrito nas equações 22 a 24. Este problema, diferentemente do testado ao longo do trabalho, minimiza a quantidade total de capital investido e o custo de manutenção de estoque do sistema.

$$\text{Min } \sum_{i=1}^n \sum_{j=1}^n C_E E(s_{ij}) + \sum_i^n \sum_{j=1}^n c_i s_{ij} \quad (22)$$

s. a.

$$E(\text{Backorder}_{ij}) = \text{NSL}_{ij} \quad (23)$$

$$s_{ij} \in \mathbb{Z} \quad (24)$$

$s_{ij}$  = quantidade de produto  $i$  na base  $j$

$c_i$  = custo de alocar um produto  $i$

$C$  = custo total de produtos a serem alocados

$E(\text{Backorder}_{ij})$  = expectativa de *backorder* do produto  $i$  na base  $j$

$E(s_{ij})$  = expectativa de quantidade de estoque do produto  $i$  na base  $j$

$C_E$  = custo de manutenção de estoque

$\text{NSL}_{ij}$  = Nível de serviço do produto  $i$  na base  $j$

A restrição deste problema limita a expectativa de *backorder* ao nível de serviço  $\text{NSL}_{ij}$ , ou seja, ao contrário do problema descrito em (SHERBROOKE, 2006) e do presente problema, o capital é ilimitado e o seu objetivo é encontrar quais quantidades de estoque que atenderam as demandas dos itens MRO para cada nível de serviço pré-especificado.

Comparando esta formulação com a formulação encontrada neste trabalho, pode-se notar que a restrição de capital  $\sum_i^n \sum_{j=1}^n c_i s_{ij} = C$  aparece na função objetivo e a parte da função objetivo  $\sum_{i=1}^n \sum_{j=1}^n C_E E(\text{Backorder}_{ij})$ , aparece como restrição.

Com esta formulação, acredita-se que as diferenças entre a técnica de *bootstrap* e as distribuições estatísticas usadas para modelar a demanda se tornarão mais evidentes. Como o

anexo A mostra, quando se utiliza o *bootstrap* para alcançar determinados níveis de serviços comparado a diversas distribuições estatísticas, a técnica de *bootstrap* aloca maiores quantidades de estoque. Essa é a diferença que se pode utilizar para mostrar que também em um sistema multi-estágio a técnica de *bootstrap* atinge melhores resultados que outras distribuições estatísticas.

Outro fator que deve ser mencionado é que neste trabalho também ocorre pequenas diferenças entre a distribuição de Poisson e a técnica testada inserida no procedimento de simulação.

Para trabalhos futuros recomenda-se a utilização de heurísticas gulosas e que se resolva o problema para cada produto  $i$  separadamente, pois as quantidades de produtos a serem alocados são ilimitadas, o que torna cada linha um problema independente.

Então, deve-se procurar o nível de serviço  $NSL_{ij}$  para um  $i$  fixo para todas as suas bases  $j$  já que a única característica a ser observada é qual quantidade de estoque do depósito central que permitirá que todas as outras bases alcancem seus respectivos níveis de serviços pré-especificados.

Está provado em (WILLEIMAN, 2004) que a técnica de *bootstrap* alcança melhores resultados que as outras distribuições estatísticas para sistemas de um estágio sem restrição de capital e com *lead time* fixo. Como se infere no anexo A (onde mostra uma comparação entre o *bootstrap* e outras distribuições estatísticas, porém com *lead time* estocástico), a técnica de *bootstrap* perde força quando existem outras variáveis a serem analisadas. Dessa forma deve-se mostrar como a técnica de amostragem se comporta em um sistema multi-estágio.

## 6. Conclusão

Uma acurada previsão de demanda de itens MRO (Manutenção, Reparação e Operação) é essencial para a determinação da política de estoque de itens de reparação, devido à grande variabilidade da demanda desses itens. Estes itens geralmente são responsáveis diretamente por grande parte dos custos de estoques incorridos pelas empresas e são de fundamental importância para as operações das mesmas.

Os trabalhos existentes na literatura referentes à previsão de demanda de itens de reparação, (WILLEIMAN, 2004), (TEUNTER, 2009) e (SYNTETOS e BOYLAN, 2010), geralmente avaliam os resultados em um sistema de estoque simples (um único nível) e esses são avaliados segundo um procedimento de simulação discreta. No entanto, vale ressaltar que

os trabalhos desses autores, utilizam medidas de dispersão estatísticas para a avaliação das técnicas de previsão.

Também na literatura existem vários autores que discutem qual a melhor distribuição estatística para a modelagem da demanda dos itens MRO em um sistema de estoque multi-estágio.

Os principais autores que publicaram nesta área foram (GRAVES, 1985), (AXSATER, 2001), (AXSATER, 2003) e (DESMET, 2009). Eles propuseram o uso da distribuição de Poisson (sendo esta a mais utilizada nos artigos encontrados na literatura), da distribuição de Poisson composta, da aproximação à distribuição normal, do uso da distribuição gamma e do uso de outras técnicas de aproximação.

Como demonstrado em (SYNTETOS E BOYLAN, 2005), as demandas dos itens de reparação podem ser divididas em quatro classes de demandas e cada classe tem características próprias. Devido às características de cada classe, a modelagem da demanda destes itens se torna um desafio. (WILLEMAN, 2004) propôs um método utilizando a técnica de *bootstrap* no qual, segundo o autor, consegue captar muito bem as transições entre os estados de demandas e a grande variação das demandas positivas. A técnica de *bootstrap*, proposta por (WILLEIMAN, 2004), gera uma distribuição empírica que, segundo o mesmo autor, tem como principal concorrente a distribuição de Poisson composta, essa comprovadamente com um bom desempenho para a modelagem de itens de reparação em um sistema multi-estágio.

(MARSEGUERRA, 2004) utilizou uma Simulação com Otimização para resolver um problema de estoque de dois níveis, devido à grande dificuldade em modelar os custos incorridos nesses sistemas. O presente trabalho propôs o uso da mesma técnica, porém com o objetivo de utilizar a distribuição empírica resultante do método proposto em (VISWANATHAN e ZHOU, 2010) para a modelagem/previsão das demandas dos itens MRO.

A dissertação mostrou que é provável que a técnica de *bootstrap* possa ser utilizada para modelar/estimar a demanda dos itens de reposição em um sistema multi-estágio, porém, é necessário, que seja construído outros modelos de testes.

## **Anexo A**

### **7. *Bootstrap* com *Lead Time* Estocástico (Modelos de Previsão de demandas)**

Para um modelo de estoque de um estágio, sem restrição orçamentária, com backlog, foram testados duas diferentes aproximações para calcular a política de estoque com revisão contínua (s-1,s) para itens com demanda intermitente e lead time estocástico. A primeira aproximação foi um método baseado em (VISWANATHAN and ZHOU, 2008) e o segundo método o método de Croston modificado. Este modelo avalia os custos em função dos custos de manutenção, pedido e de ruptura.

### 7.1 Método de Croston Modificado

A modificação que foi utilizada para prever a média foi apresentada por (SYNTETOS e BOYLAN, 2001), equação (6) e a variância foi calculada de acordo com a equação (2) encontrada na seção 2.2 em (WILLEIMAN, 2004).

A variância e a demanda média durante o *lead time* são calculadas segundo as seguintes fórmulas:

$$E(D_{lt}) = E(D)E(LT) \quad (25)$$

$$Var(D_{lt}) = Var(D)E(LT) + (E(D))^2 Var(LT) \quad (26)$$

Existem várias distribuições estatísticas para se trabalhar com controle de estoque de itens de reposição. (LEVÉN e SEGERSTEDT, 2004) afirmam que a distribuição normal é a distribuição mais usada para determinar a probabilidade de ruptura. (SYNTETOS et al. 2010) argumentam que "apesar da inadequação da distribuição normal para representar os tamanhos das demandas positiva, a utilização da normal constitui uma suposição razoável para a demanda durante o *lead time*, especialmente para longos *lead times*."

O uso da distribuição Gamma é justificado em (BURGIN, 1975) e a análise prática reforçada por (LEVEN e SEGERSDT, 2004) entre outros. Para calcular o ponto de ressuprimento da política de estoque (s-1,s) para as distribuições gamma e normal, é escolhido o menor  $s_t$  que satisfaz  $\sum_0^{s_t} \Phi_D(x) \geq CSL$ , onde  $CSL$  é o nível de serviço e  $\Phi_D(x)$  é a distribuição de probabilidade.

### 7.2 Uma versão modificada de (ZHOU e VISWANATHAN, 2008)

O método utilizado para analisar o impacto do *lead time* estocástico sobre o desempenho do modelo de controle de estoque foi baseado no método desenvolvido por

(WILLEIMAIN et al. 2004), mais precisamente foi realizada uma adaptação ao método de (ZHOU e VISWANATHAN, 2008). Este método utiliza o histograma das frequências dos tamanhos dos intervalos entre demandas para gerar as demandas positivas, ao invés de utilizar a cadeia de Markov de dois estados, como proposto por (WILLEIMAN et al. 2004).

A adaptação proposta está de acordo com (FRICKER JR e ROBBIN, 2000), onde os autores usaram histogramas do *lead time*, do tamanho da demanda e dos tamanhos dos intervalos entre demandas para realizar a previsão. Assim, o procedimento utilizado para os testes é baseado no procedimento proposto por (ZHOU e VISWANATHAN, 2008), porém com uma diferença no passo 2 onde é utilizado um valor de acordo com o histograma gerado a partir dos dados históricos do *lead time*.

Para o método *bootstrap* foi calculado o valor da política de estoque com a distribuição empírica resultante do processo de *bootstrap* e, também, foram usados os estimadores  $E(D_{lt})$  e  $Var(D_{lt})$  resultantes do mesmo processo para calcular a política de estoque de acordo com as distribuições normais e gamma. O cálculo é feito da mesma forma que o feito para o método de Croston modificado.

### 7.3 Processo de Simulação

Para comparar o método de *bootstrapping* com *lead time* estocástico com a abordagem paramétrica foram utilizados dados de demandas simulados. Os intervalos entre demandas e os tamanhos de demanda (demandas diferentes de zero) foram gerados separadamente.

Para os testes foram utilizadas três distribuições diferentes com três parâmetros diferentes para os valores do intervalo entre demandas, e também, três distribuições diferentes com três parâmetros diferentes para gerar as demandas positivas. As distribuições e os parâmetros são os mesmos utilizados no trabalho de (VISWANATHAN e ZHOU, 2010). Para avaliar o *lead time* estocástico sobre os métodos de previsão foram utilizadas quatro distribuições diferentes.

Para gerar cada série de demanda foi utilizada uma distribuição estatística para gerar o intervalo entre demandas e outra distribuição estatística para gerar os valores das demandas positivas. As nove distribuições do tamanho do intervalo entre demandas e as nove distribuições do tamanho das demandas positivas utilizadas para gerar as séries de dados com seus respectivos coeficientes de variação são apresentados na tabela 1. Para a realização dos testes foram geradas um total de 81 séries de demanda, sendo utilizado todas as combinações

possíveis dos pares de demanda; intervalo entre demandas x tamanho das demandas. Por meio de uma rápida análise, podemos notar, de acordo com (SYNTETOS e BOYLAN, 2005), que as séries de demandas são caracterizadas como demanda intermitente e demanda *lumpy*.

Intervalo entre demandas	CV	Tamanho das demandas	CV
Uniform(2,9)	0.37	Uniform(2,23)	0.485
Uniform(3,8)	0.26	Uniform(5,20)	0.364
Uniform(4,7)	0.16	Uniform(8,17)	0.208
2+Exp(3)	0.6	Normal(20,16)	0.2
2.5+Exp(2.5)	0.5	Normal(20,9)	0.15
3+Exp(2)	0.4	Normal(20,4)	0.1
2+Gamma(0.25,1)	0.667	Lognormal(2.0,0.4)	0.701
2+Gamma(0.5,2)	0.471	Lognormal(2.0,0.3)	0.591
2+Gamma(1,4)	0.333	Lognormal(2.0,0.2)	0.471

Tabela 15: Distribuições utilizadas para a geração das séries de demandas

Todos os testes e todas as gerações das séries de demandas foram realizados no MATLAB 7.0, sendo que toda série de demanda possui 2000 pontos de demandas positivas. O cálculo da política de estoque para *bootstrap* e para o método de Croston foi realizado com as séries de demandas geradas até o 1000º ponto de demanda positiva. Os testes de simulação foram realizados com as séries de demandas entre a 1001ª e 2000ª demandas positivas. A unidade de tempo escolhida no processo de simulação é dia.

No cálculo de todos os métodos foi utilizado quatro distribuições diferentes de *lead times*: Uniform (20,40), Normal (30,8), exponencial (30) e *lead time* fixo de 30 dias. Para peças de reposição, o tempo de espera para o reabastecimento geralmente é longo e por isso foi considerado como o seu valor médio 30 dias. Não é razoável pensar em um fornecedor com uma grande variabilidade em seu *lead time*, com exceção de alguns casos específicos devido à natureza do processo. Decidimos considerar a variabilidade do *lead time* usando diferentes distribuições como a distribuição normal e uniforme. A distribuição exponencial foi utilizada para analisar o comportamento da técnica, quando sujeita a uma situação mais extrema, e o *lead time* fixo em uma situação ideal.

As medidas usadas para comparar o desempenho dos diferentes métodos foram o custo total do estoque (custo de estocagem, custo de pedidos, custo de ruptura), o nível de serviço por ciclo (CSL) e *fill rate* FR, sendo essas duas últimas como medidas de qualidade.

Supõe-se que uma ordem é colocada imediatamente após a ocorrência de uma demanda e que elas não se cruzam.

Para todas as simulações realizadas, o custo de armazenagem foi 1, o custo de ruptura foi fixado em 5 e os custos de pedidos igual a 0. Para cada série de dados foi simulados os níveis de serviços 80%, 85%, 90% e 95%. Os valores considerados para os custos de estoques também são consistentes com o trabalho de (VISWANATHAN e ZHOU, 2010). Um resumo de todos os testes realizados pode ser encontrado na tabela 2.

	Valores
Tamanho das demandas	Uniform,lognormal,normal
Intervalos entre demandas	Uniform,gamma,exponential
Lead times	Fixo.Uniform,normal,exponential
Custo de ruptura	5
Custo de estocagem	1
Custo de pedido	0
Nível de Serviço	80% , 85% , 90% , 95%
Parâmetro da Suavização	0.1

Tabela 16: Resumo dos testes

As etapas do processo de simulação podem ser resumidas em:

- 1- Recebimento de pedidos marcados para data corrente e atualização dos níveis de estoques.
- 2- Se a demanda ocorre, ela é atendida prontamente. Se não há estoque suficiente, a demanda, a demanda não atendida é colocada em *backlog*.
- 3- Se o valor do nível de estoque está abaixo do parâmetro  $s$ , uma ordem de pedido no valor de  $(s - \text{nível de estoque} + \text{backlog})$  é realizada.

#### 7.4 Testes da Simulação

A tabela 2 mostra os resultados de cada método, considerando a média das medidas para todas as distribuições do *lead time* (exceto para o *lead time* exponencialmente distribuído) para os níveis de serviço de 80%, 85%, 90% e 95%. Foram excluídos os *lead times* exponencialmente distribuídos porque seus resultados apresentaram comportamento totalmente diferentes dos demais.

A tabela 2 mostra que para o nível de serviço 95%, o FR e CSL para os métodos baseados na técnica de *bootstrap* alcançaram médias maiores do que o nível de serviço pré-

especificado. Os métodos baseados no método Croston alcançaram FR e CSL próximos, porém inferiores ao nível de serviço pré-especificado.

Para os níveis de serviço de 80%, 85% e 90%, os FR dos métodos *bootstrap* apresentaram resultados maiores do que o nível de serviço pré-especificado. Da mesma forma as medidas CSL também apresentaram resultados superiores, porém as medidas CSL estão mais próximas do nível de serviço pré-especificado. Os resultados da simulação das medidas FR e CSL para os métodos de Croston pioram a medida que o nível de serviço diminui, atingindo valores abaixo do nível de serviço pré-especificado.

Como o valor do custo de pedido foi igualado a zero, o custo total de estocagem é composto apenas pelos custos de armazenagem e de ruptura. Apesar disso, o custo total sofre maior influência por parte do custo de armazenagem. As diferenças nos custos totais entre os dois grupos caem quando o nível de serviço diminui, indicando que os métodos com previsão *bootstrap* alcançam FR e CSL mais elevados, com menores custos totais. O método mais preciso e que apresenta os menores custos, foi o *bootstrap* com distribuição normal. Já o método Croston com distribuição normal atingiu o pior resultado. A distribuição normal alcança custos totais superiores sempre maiores que as técnicas com *bootstrap*.

AVERAGE MEASURES FOR SERVICE LEVEL				
SL=0.95	AIL	FR	CSL	TC
boot/emp	57	98.05%	97.08%	253248
boot/gamma	57	98.12%	97.15%	254171
boot/norm	51	96.93%	95.44%	228696
croston/gamma	55	94.65%	92.55%	255021
croston/norm	48	92.83%	90.11%	224814
norm/norm	58	97.88%	97.00%	263781
SL=0.90	AIL	FR	CSL	TC
boot/emp	43	94.30%	91.86%	194130
boot/gamma	41	93.50%	90.72%	186883
boot/norm	40	92.67%	89.62%	180208
croston/gamma	38	87.97%	84.02%	182858
croston/norm	37	86.98%	82.78%	176311
norm/norm	44	94.45%	92.33%	205174
SL=0.85	AIL	FR	CSL	TC
boot/emp	34	89.41%	85.44%	158570
boot/gamma	32	87.42%	82.81%	149206
boot/norm	32	87.74%	83.20%	150761
croston/gamma	29	80.51%	74.94%	143805

croston/norm	30	81.11%	75.63%	146959
norm/norm	36	90.02%	86.60%	168933
SL=0.80	AIL	FR	CSL	TC
boot/emp	28	83.75%	78.27%	134082
boot/gamma	25	80.55%	74.36%	124703
boot/norm	27	82.28%	76.39%	129960
croston/gamma	22	72.74%	65.85%	119083
croston/norm	24	75.14%	68.60%	126527
norm/norm	29	84.77%	79.99%	142750

Tabela 17: Médias dos resultados obtidos na Simulação

O comportamento dos cinco métodos para o nível de serviço 95% é mostrado na tabela 3. Para este nível e *lead time* fixo igual a 30 dias, todos os métodos atingiram o FR de 95%. Na verdade, a taxa de *fill rate* e o nível ciclo de serviço foram acima de 95% para todos os métodos, exceto os CSL dos métodos Croston.

Para todos os níveis de serviço e todos os métodos, quando a variância da distribuição do *lead time* aumenta, as medidas FR e CSL obtidas nos testes de simulação diminuem. Para o *lead time* exponencialmente distribuído, as medidas de qualidades são realmente baixas comparadas com o nível de serviço pré-especificado.

AVERAGE MEASURES FOR LEAD TIME DISTRIBUTION					
SL=0.95	AIL	FR	CSL	TC	
boot/emp	53	99.12%	98.42%	234772	LT=30
	58	98.20%	97.20%	255854	unif(20,40)
	60	96.85%	95.62%	269120	norma(30,8)
	86	74.17%	72.80%	408711	exp(30)
boot/gamma	53	99.08%	98.35%	232729	LT=30
	58	98.17%	97.15%	255531	unif(20,40)
	62	97.11%	95.96%	274252	norma(30,8)
	85	73.99%	72.60%	403918	exp(30)
boot/norm	48	98.47%	97.33%	213912	LT=30
	52	96.95%	95.36%	229686	unif(20,40)
	54	95.37%	93.64%	242490	norma(30,8)
	60	64.27%	62.51%	297274	exp(30)
croston/gamma	51	95.33%	93.00%	238683	LT=30
	55	94.69%	92.62%	256749	unif(20,40)
	58	93.93%	92.02%	269630	norma(30,8)
	72	68.21%	66.63%	351327	exp(30)
croston/norm	46	94.24%	91.36%	213504	LT=30
	49	92.76%	90.03%	226352	unif(20,40)
	50	91.49%	88.95%	234584	norma(30,8)

	51	58.38%	56.51%	260722	exp(30)
norm/norm	57	99.27%	98.79%	258045	LT=30
	58	98.00%	97.04%	264010	unif(20,40)
	59	96.39%	95.16%	269288	norma(30,8)
	56	62.06%	60.29%	284296	exp(30)

Tabela 18: Resultados obtidos para todas as distribuições de *lead time* para o nível de serviço 95%

As figuras 2 e 3 mostram um resumo do *fill rate* e do nível de serviço por ciclo de todos os testes para o *lead time* uniformemente distribuído e normalmente distribuído, respectivamente. O *bootstrap* com distribuição empírica apresenta o maior CSL em todos os níveis de serviço sendo este comparado à distribuição normal. Os mesmos padrões são mostrados em todos os níveis de serviço, os métodos *bootstrap* com resultados superiores aos alcançados pelos métodos de Croston. A distribuição normal apresenta valores superiores a todos os métodos, sendo para o *fill rate* ou para o *cycle service level*. Todas as afirmações feitas para a figura 2 valem para a figura 3.

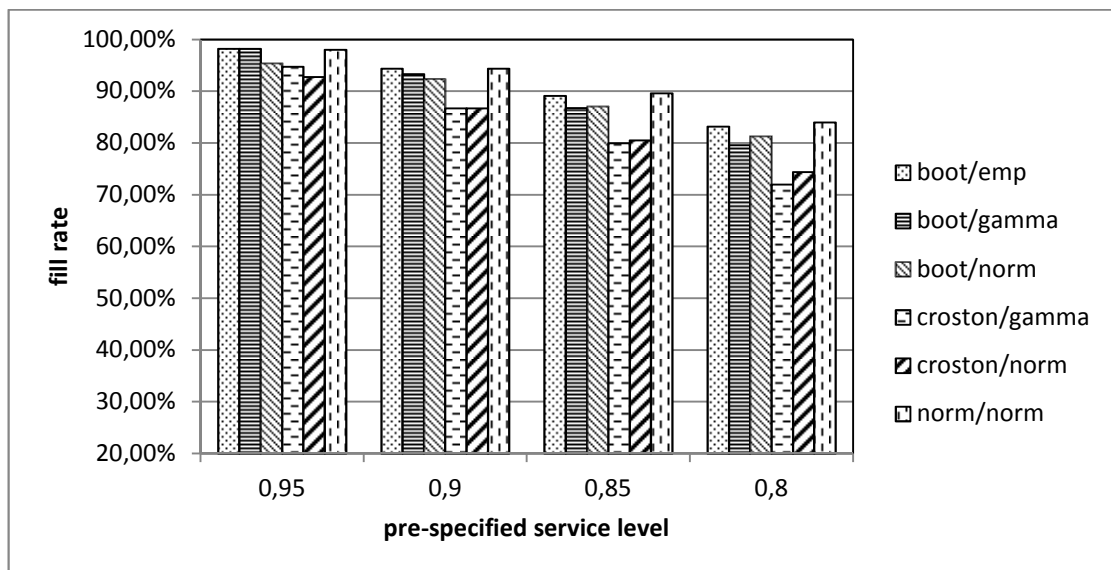


Figura 9: *fill rate* - unif(40,20)

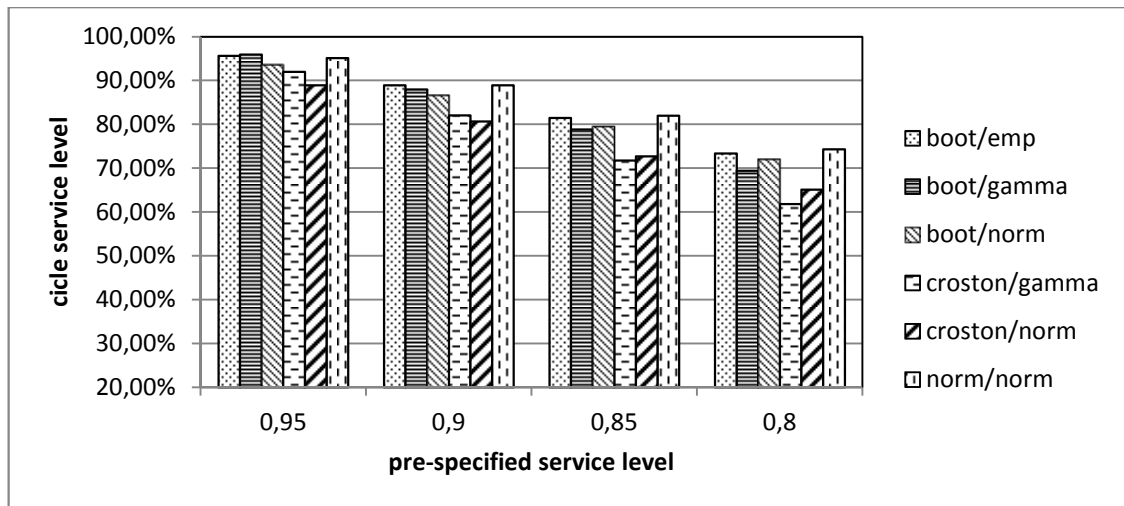


Figura 10: *cycle service level* - norm(30,8)

Outra medida que foi utilizada para avaliar a acurácia dos métodos propostos foi o MAD (Média dos desvios absolutos) como visto em (PORRAS 2007). O MAD pode ser definido como a diferença absoluta entre o nível de serviço especificado e o nível de serviço apresentado pelo *fill rate* e pelo nível de serviço por ciclo.

O MAD pode ser calculado da seguinte forma:

$$MAD_{CSL} = \frac{1}{N} \sum_{i=1}^N CSL_i - SL \quad (27)$$

$$MAD_{FR} = \frac{1}{N} \sum_{i=1}^N FR_i - SL \quad (28)$$

onde  $SL$  e  $o$  é o nível de serviço pré-especificado,  $N$  é o número total de séries de demandas avaliadas e  $CSL_i$  e  $FR_i$  os valores simulados de nível de serviço por ciclo e *fill rate* de acordo com os métodos propostos.

Os MAD's encontrados mostram que os métodos com o *bootstrap* como estimador da demanda média e da variância apresentam os melhores resultados e os métodos utilizando o método de Croston apresentam os piores resultados. A distribuição normal apresenta um resultado intermediário, com erros sempre maiores que todos os métodos utilizando o *bootstrap*. Um dos motivos para a distribuição normal apresentar resultados satisfatórios e que segundo (SYNTETOS et al. 2010) *lead times* longos permitem o uso do teorema do limite central por meio das somas individuais das demandas durante o período calculado. Os erros aumentam na medida em que o nível de serviço especificado diminui.

CSL = 0.8	boot/emp		boot/gamma		boot/norm		croston/gamma		croston/norm		norm/norm	
	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL
AVERAGE	6.37	6.10	5.58	7.62	5.43	6.21	13.25	17.28	12.18	15.62	7.57	6.36
LT=30	9.62	5.90	7.34	4.85	8.22	4.59	12.62	15.69	12.22	14.73	11.21	8.11
unif(20,40)	5.21	5.21	3.98	7.34	3.92	5.85	12.86	17.05	11.77	15.45	6.42	4.53
norma(30,8)	4.28	7.21	5.42	10.67	4.15	8.18	14.27	19.09	12.54	16.70	5.07	6.44
CSL = 0.85	boot/emp		boot/gamma		boot/norm		croston/gamma		croston/norm		norm/norm	
	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL
AVERAGE	5.72	4.88	4.82	5.07	4.84	4.83	10.85	13.96	10.60	13.50	6.87	5.62
LT=30	8.67	5.86	7.29	4.23	7.38	4.21	10.70	13.08	10.74	12.80	9.80	7.62
unif(20,40)	5.10	3.99	3.77	4.54	3.86	4.33	10.58	13.84	10.28	13.38	6.17	4.53
norma(30,8)	3.39	4.78	3.41	6.45	3.28	5.95	11.27	14.97	10.79	14.31	4.66	4.71
CSL = 0.9	boot/emp		boot/gamma		boot/norm		croston/gamma		croston/norm		norm/norm	
	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL
AVERAGE	4.87	3.79	4.27	3.43	3.93	3.62	8.22	10.23	8.56	10.96	5.53	4.69
LT=30	6.91	5.18	6.34	4.36	5.89	3.88	8.72	10.40	8.76	10.78	7.56	6.25
unif(20,40)	4.77	3.29	3.84	2.89	3.39	3.06	8.04	9.95	8.36	10.76	5.24	4.31
norma(30,8)	2.94	2.90	2.62	3.05	2.51	3.91	7.90	10.33	8.55	11.34	3.79	3.50
CSL = 0.95	boot/emp		boot/gamma		boot/norm		croston/gamma		croston/norm		norm/norm	
	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL	FR	CSL
AVERAGE	3.25	2.63	3.24	2.52	2.56	2.20	5.13	6.25	5.67	7.40	3.40	3.01
LT=30	4.14	3.48	4.11	3.41	3.56	2.64	5.58	6.94	5.78	7.43	4.34	3.89
unif(20,40)	3.29	2.54	3.26	2.43	2.40	1.90	5.20	6.23	5.62	7.28	3.30	2.81
norma(30,8)	2.32	1.87	2.35	1.72	1.71	2.07	4.62	5.58	5.62	7.48	2.55	2.32

Tabela 19: Média dos desvios absolutos

## 8. Bibliografia

**Axsäter, S. (1990).** *Simple solution procedures for a class of two echelon inventory problems.* Operations Research, 38: 64–69.

**Axsäter, S., & Zhang, W. (1996).** *Recursive evaluation of order-up-to-S policies for two-echelon inventory systems with compound Poisson demand.* Naval Research Logistics, 43:151–157.

**Axsäter, S. (2000).** *Exact analysis of continuous review (R, Q) policies in two-echelon inventory systems with compound Poisson demand.* Operations Research, 48:686–696.

**Axsäter, S. (2001).** *Scaling down multi-echelon inventory problems.* International Journal of Production Economics, 71:255–261.

**Axsäter, S. (2003).** *An approximate optimization of a two-level distribution inventory system.* International Journal of Production Economics, 81–82, 545–553.

**Axsäter, S. (2005).** *A simple decisions rule for decentralized two-echelon inventory*

*control*. International Journal of Production Economics, 93–94, 53–59.

**Babai, M. Z.; Jemai, Z. e Dallery, Y. (2010).** *Analysis of order-up-to-level inventory systems with compound poisson demand*. 8th International Conference of Modeling and Simulation, Hammamet, Tunisia.

**Billah, B.; King, M.; Snyder, R. e Koehler, A. (2006).** *Exponential smoothing model selection for forecasting*. International Journal of Forecasting, 22:239–247.

**Boylan, J. e Syntetos, A. (2007).** *The accuracy of a modified croston procedure*. International Journal of Production Economics, 107:511–517.

**Boylan, J. E. e Syntetos, A. A. (2010).** *Spare parts management: a review of forecasting research and extensions*. IMA Journal of Management Mathematics, 21:227–237.

**Boylan, J. E.; Syntetos, A. A. e Karakostas, G. C. (2008).** *Classification for forecasting and stock control: A case study*. Journal of the Operational Research Society, 59:473–481.

**Brown, R. G. (1959).** *Statistical forecasting for inventory control*. McGraw-Hill, New York.

**Burgin, T. (1975).** *The gamma distribution and inventory control*. Operations Research Quarterly, 26:507–525.

**Caglar, D., Chung-Lun, L., e Simchi-Levi, D. (2004).** *Two-echelon spare parts inventory system subject to a service constraint*. IIE Transactions, 36,655–666.

**Chwif, L. e Medina, A.C.** *Modelagem e Simulação de Eventos Discretos: Teoria & Aplicações*. São Paulo: Bravarte, 2006.

**Croston, J. D. (1972).** *Forecasting and stock control for intermittent demands*. Research Quarterly, 23(3):289–303.

**Deboedt, M. A., e Graves, S. C. (1985).** *Continuous review policies for a multi-echelon inventory problem with stochastic demand*. Management Science, 31:1286–1299.

**Eaves, A. e Kingsman, B. (2004).** *Forecasting for the ordering and stock-holding of spare parts*. Journal of the Operational Research Society, 55:431–437.

**Eaves, A. H. C. (2002).** *Forecasting for the ordering and stock holding of consumable spare parts*. PhD thesis, Lancaster University, UK.

**Efron, B. (1979).** *Bootstrap methods: another look at the jackknife*. Annal of Statistics, 7:1–26.

**Fanggeng, Z.; Jiangsheng, S.; Lianwu1, Z. e Zhenshu M., (2010).** *Genetic Algorithm for the Multi-Echelon Inventory Problem of Weapon Equipment Repairable Spare Parts*. Proceedings - 3rd IEEE International Conference on Computer Science and

Information Technology, ICCSIT 2010 2, art. no. 5565161, pp. 619-622.

**Forsberg, R., (1995).** *Optimization of order-up-to-S policies for two-level inventory systems with compound Poisson demand.* European Journal of Operational Research 81:143-153.

**Fricker Jr, R. e Robbin, M. (2000).** *Retooling for the logistics revolution. designing marine corps inventories to support the warfighter.* 2000 - oai.dtic.mil.

**Graves, S. C. (1985).** *A multi-echelon inventory model for a repairable item with one-for-one replenishment.* Management Science, 31:1247–1256.

**Ghobbar, A. e Friend, C. (2003).** *Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model.* Computers & Operations Research, 30:2097–2114.

**Haji, R., Neghab, M. P., e Baboli, A. (2008).** *Introducing a new ordering policy in a two-echelon inventory system with Poisson demand.* International Journal of Production Economics, 117: 212–218.

**Holland, J. H. (1975).** *Adaption in natural and artificial systems.* Ann Arbor, Michigan:University of Michigan Press. re-issued by MIT Press (1992).

**Holt, C. C. (1957).** *Forecasting trends and seasonal by exponentially weighted averages.* ONR Memorandum No. 52, Carnegie Institute of Technology, Pittsburgh, USA (published in International Journal of Forecasting 2004, 20, 5–13).

**Hopp, W. J., Spearman, M. L., e Zhang, R. Q. (1997).** *Easily implementable inventory control policies.* Operations Research, 45:327–340.

**Hopp, W. J., e Spearman, M. L. (2001).** *Factory physics.* 2nd ed.. New York: McGraw-Hill.

**Holt, C. C. (1957).** *Forecasting trends and seasonal by exponentially weighted averages.* ONR Memorandum No. 52, Carnegie Institute of Technology, Pittsburgh, USA (published in International Journal of Forecasting 2004, 20, 5–13).

**Hua, Z. e Zhang, B. (2006).** *A hybrid support vector machines and logistic regression approach for forecasting intermittent demand of spare parts.* Applied Mathematics and Computation, 181:1035–1048.

**Johnston, F. e Boylan, J. (1996a).** *Forecasting for items with intermittent demand.* Journal of the Operational Research Society, 47:113–121.

**Johnston, F.; Boylan, J. e Shale, E. (2003).** *An examination of the size of orders from customers, their characterisation and the implications for inventory control of slow moving items.* Journal of the Operational Research Society, 54:833–837.

**Johnston, F. R. e Boylan, J. E. (1996b).** *Forecasting for items with intermittent demand.* Journal of the Operational Research Society, 47:113–121.

**Jokar, M. A., e Zangeneh, S. (2006).** *Developing a model for a two-echelon two-item inventory system with lost sale and demand substitution.* Management of Innovation and Technology, 2:926–930.

**Levén, E. e Segerstedt, A. (2004).** *Inventory control with a modified croston procedure and erlang distribution.* International Journal of Production Economics. 90:361–367.

**Marseguerra, M.; Zio, E. e Podofilini, L., (2004).** *Multiobjective spare part allocation by means of genetic algorithms and Monte Carlo simulation.* Reliability Engineering & System Safety 87:325-335.

**Muckstadt, J. A. (1973).** *A model for a multi-item, multi-echelon, multi-indenture inventory system.* Management Science, 20:472–481.

**Michalewicz, Z. e Fogel, D. (2000).** *How to Solve It: Modern Heuristics.* Springer, Verlag, Berlin, Heidelberg, New York, Segunda edição.

**Pasandideh, S.H.R.; Niaki, S.T.A. e Tokhmehchi, N., (2010).** *A parameter-tuned genetic algorithm to optimize two-echelon continuous review inventory systems.* Expert Systems with Applications 38 (9): 11708-11714.

**Porrás, E. e Dekker, R. (2007)** *Inventory control system for spare parts at a refinery: An empirical comparison of different re-order point methods.* European Journal of Operational Research 184 (2008) 101–132.

**Rosenbaum, B.A., (1981).** *Inventory placement in a two-echelon inventory system: An application.* TIMS Studies in the Management Sciences, 16 :195–207.

**Segerstedt, A. (1994).** *Inventory control with variation in lead times, especially when demand is intermittent.* International Journal of Production Economics, 35:365–372.

**Seo, Y., Jung, S., e Hahm, J. (2001).** *Optimal reorder decision utilizing centralized stock information in a two-echelon distribution system.* Computers and Operations Research, 29:171–193.

**Sherbrooke, C. (2004).** *METRIC: A multi-echelon technique for recoverable item control.* Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow, Segunda edição.

**Sherbrooke, C. (1968).** *Optimal Inventory Modeling of Systems: MultiEchelon Techniques.* Operations Research, 12:122–141.

**Silver, E.; Ho, C.-M. e Deemer, R. (1971).** *Cost minimizing inventory control of*

*items having a special type of erratic demand pattern.* INFOR, 9:198–219.

**Silver, E.; Pyke, D. e Peterson, R. (1998).** *Inventory Management and Production Planning and Scheduling.* Wiley, New York, London, Sidney, third Edição.

**Singh, N. e Vrat, P., (1984).** *A location–allocation model for a two–echelon repair–inventory system.* IIE Transactions 16 (3):222–228.

**Svoronos, A., e Zipkin, P. (1988).** *Estimating the performance of multi-level inventory systems.* Operations Research, 36:57–72.

**Syntetos, A. (2001).** *Forecasting of intermittent demand.* PhD thesis, Brunel University of Buckinghamshire, New University, UK.

**Syntetos, A. e Boylan, J. (2006).** *On the stock control performance of intermittent demand estimators.* International Journal of Production Economics, 103:36–47.

**Syntetos, A. A.; Babai, M. Z. e Altay, N. (2010).** *Modelling spare parts demand: an empirical investigation.* 8th International Conference of Modeling and Simulation, Hammamet, Tunisia.

**Syntetos, A. A. e Boylan, J. E. (2001).** *On the bias of intermittent demand estimates.* International Journal of Production Economics, 71:457–466.

**Syntetos, A. A. e Boylan, J. E. (2005).** *The accuracy of intermittent demand estimates.* International Journal of Forecasting, 21:303–314.

**Syntetos, A. A.; Boylan, J. E. e Croston, J. D. (2005).** *On the categorization of demand patterns.* Journal of the Operational Research Society, 56:495–503.

**Teunter, R. e Duncan, L. (2009).** *Forecasting intermittent demand: A comparative study.* Journal of the Operational Research Society, 60:321–329.

**Teunter, R. e Sani, B. (2009).** *On the bias of croston's forecasting method.* Operational Research, 194:177–183.

**Viswanathan, S. e Zhou, C. (2008).** *A new bootstrapping based method for forecasting and safety stock determination for intermittent demand items.* Working paper, Nanyang Business School, Nanyang Technological University, Singapore.

**Wallström, P. e Segerstedt, A. (2010).** *Evaluation of forecasting error measurements and techniques for intermittent demand.* International Journal of Production Economics, 128:625–636.

**Watson, R. B. (1987).** *The effects of demand-forecast fluctuations on customer service and inventory cost when demand is lumpy.* Journal of the Operational Research Society, 38:75–82.

**Willemain, T. R.; Smart, C. N. e Schwarz, H. F. (2004).** *A new approach to*

*forecasting intermittent demand for service parts inventories*. International Journal of Forecasting, 20:375–387.

**Williams, T. M. (1984)**. *Stock control with sporadic and slow-moving demand*. Journal of the Operational Research Society, 35:939–948.

**Winters, P. R. (1960)**. *Forecasting sales by exponentially weighted moving averages*. Management Science, 6:324–342.

**Zhou, C. e Viswanathan, S. (2010)**. *Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory*. International Journal of Production Economics, p. doi:10.1016/j.ijpe.2010.09.021.