PROGRAMAÇÃO GENÉTICA PARA OTIMIZAÇÃO DE SÉRIES TEMPORAIS COM DADOS FALTANTES

Márcio Mota Lopes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

_____
Prof. Nelson Francisco Favilla Ebecken, D.Sc

_____
Prof. Beatriz de Souza Leite Pires de Lima, D.Sc.

_____
Prof. Antônio César Ferreira Guimarães, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
JULHO DE 2007

LOPES, MARCIO MOTA

Programação Genética para Otimização de Séries Temporais com Dados Faltantes [Rio de Janeiro] 2007.

VIII, 87 p. 29,7 cm (COPPE/UFRJ, M. Sc., Engenharia Civil, 2007)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Série Temporal

2. Previsão

3. Algoritmo Genético

4. Nearest Neighbour in time

5. Programação Genética

I. COPPE/UFRJ II. Título (série)

Aos Meus Pais,

A minha namorada

AGRADECIMENTOS

A DEUS.

A minha Família por entenderem minha ausência e pelas orações.

A minha namorada Christiane, pelo incentivo e compreensão nos momentos mais difíceis.

Ao meu orientador Nelson Ebecken, pela confiança, paciência, pelo apoio e pelos ensinamentos tanto na vida profissional e quanto na pessoal.

Ao CNPQ pelo apoio financeiro.

Aos amigos do NTT, pelo enriquecimento profissional e pelo incentivo.

Aos amigos da iPixel, pela compreensão. Ao amigo Heder pela colaboração.

Um especial agradecimento a uma pessoa que me apoiou durante minha estadia no Rio de Janeiro, mas que não está mais presente entre nós - meu Tio Eduardo Lopes.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROGRAMAÇÃO GENÉTICA PARA OTIMIZAÇÃO DE SÉRIES TEMPORAIS COM DADOS FALTANTES

Márcio Mota Lopes

Julho/2007

Orientador: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

Esta dissertação apresenta uma técnica de inserção de dados cujo propósito é resolver o problema de valores faltantes em séries temporais, bem como, otimizar séries temporais, através da programação genética.

O método de inserção de dados é baseado no algoritmo dos vizinhos mais próximos. Essa análise é de grande importância, pois para a previsão de séries temporais os dados devem estar cronologicamente ordenados. Após as análises, estatística e dos dados faltantes da série, utiliza-se o Modelo de *Winters* para a previsão inicial. Com os resíduos gerados pelo modelo, usa-se a programação genética para identificar a melhor função que se adéqüe ao comportamento dos resíduos. Ao final, efetua-se testes e comparações dos resultados com outras implementações.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

GENETIC PROGRAMMING FOR OPTIMIZATION OF TIMES SERIES WITH MISSING DATA

Márcio Mota Lopes

July/2007

Adivisor: Nelson Francisco Favilla Ebecken

Departament: Civil Engineering

This work presents one technique of insertion of data whose intention is to solve the problem of missing values in time series, as well as, to optimize the prediction accuracy of time series, through the genetic programming approach.

The method of insertion of data is based on the nearest neighbours algorithm. This analysis is of great importance, because for forecast of time series the data must be sequentially inputed. After the analyses, statistics and the imputation of the missing data, the *Winters* Model for the initial forecast is used. With the residues generated by the model, genetic programming is used to identify the best function to minimize the behavior of the residues. Some tests and comparisons of the results with other strategies were presented.

ÍNDICE

CAPÍTULO 1
INTRODUÇÃO

Desde o início dos tempos o ser humano busca compreender o ambiente que o rodeia, visando desenvolvimento e ter condições competitivas em relação a seu semelhante. A seleção natural fez com que os seres humanos e desenvolvessem como seres cada vez mais inteligentes.

Hoje em dia, com o desenvolvimento tecnológico, o homem visa unir o processo da evolução natural com a tecnologia para ajudá-lo, dentre outros fatores, na tomada de decisões. Exemplos disso são a previsão de desastres ecológicos, evitando conseqüências maiores, os indicadores financeiros, dentre outros.

Esse cenário é de grande importância para as organizações na obtenção de informações que possam auxiliá-las na previsão do seu futuro. Essas previsões podem ocorrer nas mais diversas áreas da organização, tanto de marketing quanto de gestão de recursos humanos.

As mudanças econômicas ocorridas nos últimos tempos, como o processo de globalização, têm forçado as organizações empresarias a adaptarem-se continuamente para enfrentar os desafios de manterem-se no mercado de forma competitiva. Esta situação requer especial atenção das empresas no planejamento adequado das atividades envolvidas no processo de produção, a fim de melhorar o fluxo e a alocação de informações, materiais e pessoas, e atender satisfatoriamente à demanda de seus produtos.

O planejamento da produção tem a previsão de demanda como um dos seus grandes subsídios, uma vez que fornece informações sobre a demanda futura dos produtos possibilitando o planejamento com antecedência e, conseqüentemente, permitindo que os recursos produtivos estejam disponíveis na quantidade, momento e qualidade adequada. Uma boa previsão proporcionará menor estoque, custo financeiro e tempo de entrega, bem como, maior previsibilidade e satisfação do cliente.

Os erros de previsão são, evidentemente, inevitáveis, objetiva-se, contudo, minimizá-los. O desejo de compreender o passado e prever o futuro impulsiona a

procura de leis que expliquem o comportamento de certos fenômenos ou acontecimentos. Uma vez conhecidas as equações determinísticas que os explicam, estas podem ser utilizadas para prever o resultado de uma dada experiência, desde que sejam conhecidas as condições iniciais. No entanto, na ausência de regras que definam o comportamento de um sistema, determina-se o seu comportamento futuro a partir de observações concretizadas no passado. Nestas situações, uma das técnicas mais comuns é a de Previsão de Series Temporais, que se baseia em observações cronologicamente ordenadas da variável em estudo.

Antes de se trabalhar com a previsão de séries temporais deve-se primeiramente verificar a corretude dos dados, bem como a verificação dos valores faltantes (*missing values*) na série. Essa análise é de grande importância para a previsão de séries temporais, pois os dados devem estar cronologicamente ordenados.

O estudo de uma série temporal tem por um de seus objetivos a predição de demanda (também chamado de previsão), ou seja, antever de forma precisa o comportamento futuro da série específica a partir do modelo elaborado.

A previsão de demanda pode ser obtida por métodos qualitativos, quantitativos ou por ambos. Os métodos qualitativos são baseados em opiniões, intuições e na experiência acumulada; enquanto que os métodos quantitativos são baseados na análise de séries temporais e modelos causais.

O desenvolvimento de técnicas de previsão, cada vez mais sofisticadas, paralelamente ao rápido desenvolvimento de computadores e outras tecnologias de informação e manipulação de dados, têm levado diversas empresas a se interessarem cada vez mais pelo processo de previsão de demanda.

O presente trabalho tem por objetivo criar uma ferramenta que facilite o preenchimento de dados faltantes através do método dos vizinhos mais próximos. A implementação dessa ferramenta justifica-se pela importância de se manter o maior número de dados da série possível, assim como, mantê-los cronologicamente ordenados, já que a falta de dados dificultaria ou até mesmo impossibilitaria a criação do modelo.

A dissertação tem por objeto, ainda, verificar a viabilidade da utilização da programação genética para otimização dos resíduos gerados pela previsão do modelo de *Winters.*

Os resultados encontrados para os dois objetivos da dissertação serão apresentados e, no caso da otimização, também serão comparados com outras implementações.

Esta dissertação será organizada da seguinte forma:

O capítulo 1 – introdução; Capitulo 2 – Fundamentos de Series temporais: conceito, modelos estatísticos para previsão; Capítulo 3 – Algoritmo Genético e a Programação Genética: Conceito sobre algoritmos genético e a Programação Genética; Capitulo 4 – Tratamento de Dados Faltantes: aborda os algoritmos baseado nos vizinhos mais próximos no tempo. Capítulo 5: Ferramenta para Tratamento de dados Faltantes e Metodologia para previsão de resíduos usando Programação Genética: aborda a ferramenta computacional criada para tratamento de dados faltantes e para otimização dos parâmetros do modelo de *Winters* e também a metodologia criada para previsão dos resíduos gerado pelo modelo de *Winters* usando Programação Genética; Capitulo 6 – Estudo Comparativo: apresenta um estudo comparativo com outras metodologias para comprovar a viabilidade do modelo. Capítulo 7 – Conclusão: encerra o presente trabalho com as considerações finais e propostas para continuação dos trabalhos.

CAPÍTULO 2
FUNDAMENTOS DE SÉRIES TEMPORAIS

2.1 Conceito

Nos últimos anos, tem sido dada uma certa ênfase no melhoramento dos processos de tomada de decisão, tanto no que concerne ao mundo empresarial quanto no que pertence ao universo da política. Os métodos de gestão há vinte ou trinta anos atrás baseavam-se nos sentidos e intuição do gestor ou político. Hoje tal artifício é apoiado através de técnicas de tomada de decisão.

A utilização da previsão é comum nas organizações, especialmente em áreas como o marketing, produção e, principalmente, nas áreas financeira e contábil onde sua utilização tem sido mais intensa. As empresas não devem, no entanto, encarar a previsão como uma profecia, mas sim como a maneira mais eficaz de extrapolar as relações existentes nos dados, para fazer previsões. Os erros de previsão são inevitáveis, objetiva-se, contudo, minimizá-los.

2.2 Séries Temporais

Para Moretin, uma série temporal é qualquer conjunto de observações ordenadas no tempo [MORETIN 2004]. Exemplos:

(i) valores diários de poluição na cidade do Rio de Janeiro;

(ii) valores mensais de exportação de soja do Brasil;

(iii) índices diários da Bolsa de Valores de São Paulo;

(iv) registro de marés no porto de Santos.

Nos exemplo de (i) – (iii) são séries temporais discretas,ou seja, as observações são feitas em tempo específicos, enquanto (iv) é um exemplo de uma série continua, as observações são feitas continuamente no tempo.

Seja uma serie temporal $Z(t_1),...,Z(t_n)$, observada nos instantes $t_1,...t_n$, podemos estar interessado em:

(a) investigar o mecanismo gerador da série temporal; por exemplo, analisando uma série de alturas de ondas, podemos querer saber como estas ondas foram geradas;

(b) fazer previsões de valores futuros da série; estas podem ser de curto prazo, como para séries de vendas, produção ou estoque, ou a longo prazo, como para séries populacionais, de produtividades, etc;

(c) descrever apenas o comportamento da série; neste caso, a construção do gráfico, a verificação da existência de tendência, ciclos e variações sazonais, a construção de histogramas e diagramas de dispersão etc., podem ser ferramentas úteis;

(d) procurar periodicidades relevantes nos dados;

Como o presente trabalho presta-se a disponibilizar uma ferramenta que dê apoio ao usuário na tomada de decisão, o foco principal incidirá nos itens (b) e (c), uma vez que a previsão de séries temporais é um dos meios de fornecer informações para uma conseqüente tomada de decisão.

Os procedimentos de previsão utilizados na prática variam muito, podendo ser simples e intuitivos ou quantitativos e complexos. No primeiro caso, pouca ou nenhuma análise de dados é envolvida, enquanto no segundo caso esta análise pode ser considerável.

Os procedimentos de previsão podem ser obtidos por métodos qualitativos, quantitativos ou ambos. Os métodos qualitativos são baseados em opiniões, intuições e na experiência acumulada; enquanto que os métodos quantitativos são baseados na análise de séries temporais e modelos causais.

Os Métodos quantitativos utilizam dados históricos para prever a demanda em períodos futuros.

Na análise de séries temporais existem, basicamente, dois caminhos que o analista pode seguir: a análise no domínio do tempo e a análise no domínio da freqüência. Não há uma distinção formal sobre a utilização de cada uma dessas abordagens. A escolha se dá em função das características da aplicação, onde se o objetivo for para identificar ou filtrar um sinal, utiliza-se a análise no domínio do tempo.

Outra característica que define o enfoque da análise é o intervalo de tempo que a série é composta. Se o intervalo de tempo for frações de minutos, adota-se a análise no domínio da freqüência, caso contrário, se o intervalo for maior do que horas, aplica-se a análise no domínio do tempo. Em ambos os enfoques, existem vários modelos para atender às mais diversas aplicações. Este trabalho abordará apenas a análise da série no domínio temporal.

O período de tempo durante o qual se pretende que prevaleça uma dada tomada de decisão afeta, naturalmente, o processo de seleção do método de previsão. As previsões são feitas normalmente com um ou dois períodos de avanço. Os horizontes temporais podem ser classificados de curto, médio e longo prazo. O primeiro, até três períodos, é muito usado para decisões de gestão corrente, como previsão de estoque, enquanto que as previsões de médio prazo, de três meses a um ano, e de longo prazo, mais de dois anos, são usadas em planejamento, como exemplo, na medicina preventiva.

As séries temporais podem ser classificadas em modelos univariados, de função de transferência e multivariados. No primeiro grupo estão os modelos que se baseiam em uma única série histórica; já no segundo, a série de interesse é explicada pelo seu passado histórico e por outras séries temporais não correlacionadas entre si. O último grupo engloba os que modelam simultaneamente mais de uma série temporal, sem exigências com relação à direção da causalidade entre elas.

Nesse sentido, a ênfase do trabalho será em modelos univariados.

Uma série temporal é composta, segundo o modelo clássico, por até 4 componentes: tendência, ciclos, sazonalidade e irregularidade. Nem todos os

componentes estão, necessariamente, presentes em todas as séries. Pode-se encontrar séries com apenas 1, 2 ou 3 deles.

A tendência, está relacionada com o movimento dos dados, em longo prazo, para cima ou para baixo, sendo geralmente produzida em função do crescimento/decrescimento constante da série.

As variações cíclicas e a sazonalidade são padrões observáveis em períodos de tempo. A diferença básica reside no fato de a primeira ser de longo prazo – períodos superiores a um ano - enquanto a segunda trata dos casos em intervalos de tempo iguais ou inferiores a um ano.

Finalmente, as variações irregulares decorrem de fatos imprevisíveis, tais como guerras, greves, etc. Enquadra-se neste item, tudo que não for passível de classificação nos padrões anteriores.

O estudo de uma série temporal pode trazer informações importantes sobre o futuro, pois, normalmente, há correlação entre as variáveis em diversos instantes. É claro que algum grau de incerteza virá agregado, pois o futuro nunca refletirá exatamente as ocorrências passadas, mas a prática da previsão tem sido utilizada no auxílio ao planejamento e tomada de decisões.

2.3 Etapas da Previsão

Todos os métodos de previsão partem do princípio de que as experiências do passado serão usadas no futuro. Assume-se, assim, que as condições do passado serão válidas no futuro. A previsão envolve quatro etapas, a obtenção de dados históricos, redução dos dados, construção do modelo de previsão, e a extrapolação a partir desse modelo.

A primeira etapa exige um particular cuidado na leitura dos dados do problema em equação, de forma a se evitar distorções, por meio da verificação de dados faltantes e *outiliers*. A etapa seguinte, consiste em determinar os dados relevantes para o problema em equação, como por exemplo, pode-se estar apenas interessado em prever o

2.5 Análise e Visualização dos Dados Reais

Como descrito anteriormente, é de grande importância entender o comportamento da série, o que engloba tanto a visualização de gráficos quanto a verificação de análise analítica, a fim de facilitar a análise e a compreensão do problema.

A análise gráfica permite encontrar padrões aparentes (como tendências e sazonalidades), detectar erros grosseiros, valores ausentes, bem como identificar mudanças estruturais ou rupturas na série.

Com o estudo analítico é possível sintetizar o comportamento estatístico de séries estacionárias através dos cálculos da média, desvio padrão, mediana, etc., que serão descritos a seguir [ALLEMAO, 2004]:

2.5.1 Média Aritmética

Tem como objetivo representar toda a massa de dados a partir de um único número. É representada como:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (2.1)$$

2.5.2 Mediana

A principal característica da mediana é dividir um conjunto ordenado em dois grupos iguais; metade com valores inferiores e a outra metade com valores superiores à mediana. Em geral, a mediana ocupa a posição *(n + 1) / 2* do conjunto ordenado, onde n corresponde à quantidade de elementos do grupo.

Para conjuntos com quantidade par de elementos a mediana é calculada através da média dos dois valores do meio, ou seja, a mediana fica entre as duas observações centrais da disposição ordenada. A mediana é a média aritmética dos valores numéricos correspondentes àquelas duas observações centrais.

2.5.3 Variância

A variância é uma medida de dispersão que serve para representar, com um único número, quão próximos os valores observados em um conjunto de dados estão uns dos outros. Esta medida tem na média seu ponto de referência e o valor zero informa que não há dispersão no conjunto de observação.

A variância s^2 é definida como:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad (2.2)$$

Onde *n* corresponde ao número de dados do conjunto, $\bar{x}$ = média aritmética e x_i = valor da observação.

2.5.4 Desvio Padrão

O desvio padrão é definido como a raiz quadrada da variância e é estimado pela equação:

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad (2.3)$$

Onde n corresponde ao número de dados do conjunto, $\bar{x}$ = média aritmética e x_i = valor da observação.

A informação que o desvio padrão é representada relativamente a um ponto de referência – a média - e não propriamente a variabilidade dos dados, uns relativamente aos outros.

Se os dados se distribuem de forma normal, verifica-se que basicamente 68% dos dados estão no intervalo [$\bar{x}$ - s, $\bar{x}$ + s], 95% no intervalo [$\bar{x}$ - 2s, $\bar{x}$ + 2s] e 99% no intervalo [$\bar{x}$ - 3s, $\bar{x}$ + 3s]. Isto permite que se façam análises sobre o número de observações que caem longe da média aritmética, em termos de desvio padrão.

2.5.5 Função de Autocorrelação (FAC)

Um problema com o qual nos deparamos freqüentemente é o de estudar a correspondência existente entre duas variáveis, verificando se há alguma relação entre elas, e caso haja, em que grau está presente. Um método simples e objetivo para se tratar este problema é o uso do coeficiente de correlação. Esta técnica aplica-se a dados bivariados (x_i, y_i) e verifica a existência de associações lineares entre as amostras [MORETTIN, 2004];

Uma importante ferramenta para se identificar as propriedades de uma série temporal consiste de uma série de quantidades chamada coeficientes de autocorrelação amostral. A idéia é similar ao coeficiente de correlação.

A idéia é medir a correlação entre as observações de uma mesma variável em diferentes horizontes de tempo, isto é, correlações entre observações defasadas 1, 2, ... períodos de tempo. Assim, dadas *n* observações $x_1,...,x_n$ de uma série temporal discreta podemos formar os pares $(x_1, x_2),...,(x_{n-1}, x_n)$.

Seja *Z(t)* a série estudada, *Z(t-1)*, representa a variável com atraso de um tempo, ou seja, a série com lag = 1, e *Z(t-2)* a variável com atraso de dois tempos, isto é, a série com lag = 2.

A autocorrelação entre *Z(t)* e *Z(t-1)* indicará como os valores de *Z* estão relacionados com seus valores sucessores ou antecessores; e a autocorrelação entre *Z(t)* e *Z(t-2)* indicará como os valores de *Z* estão relacionados com os valores da série atrasada em dois períodos de tempo.

Em geral, para uma série de *n* observações, a função de autocorrelação com lag *k*, sendo *k*=1,2,3,... etc. é dada pela equação (Eq. 2.4) que indica a correlação entre *Z(t)* e *Z(t+k)*.

$$r_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \qquad (2.4)$$

Onde $\bar{x}$ é a média da série;

Na prática, para se obter uma boa estimativa do coeficiente de auto-correlação, deve-se dispor de pelo menos 50 observações de variável *x* [MONTGOMERY et al., 1990]. O número de autocorrelação de *lags* diferentes que se calcula para a análise da série temporal dever ser *n/4*, onde *n* é o número de observações na série.

O coeficiente de auto-correlação varia entre -1 e +1. O valor zero indica ausência de correlação entre as variáveis. Quanto maior for o módulo de *r*, maior será, o grau de associação linear existente entre eles.

2.6 Métodos de Análise de Séries Temporais

A seguir, são descritos alguns métodos tradicionais de análise de séries temporais comumente encontrados na bibliografia, sendo, contudo, mais detalhado o método de *Holt-Winters*, já que se trata do foco do trabalho.

2.6.1 Análise de Regressão Linear

É utilizada para definir um relacionamento funcional entre duas ou mais variáveis correlacionadas. O relacionamento é desenvolvido a partir de um dado observado no qual um parâmetro (a variável independente) é usado para predizer outro (a variável dependente). A equação para a regressão linear simples inclui somente uma variável independente [DAVIS et al, 1997]:

$$Y = a + bX \tag{2.5}$$

Onde: Y é a variável dependente, a é a interseção no eixo Y , b é a inclinação e X é a variável independente.

A grande desvantagem do método é considerar que os dados do passado e as projeções futuras devem se situar próximos a uma linha reta.

2.6.2 Análise de Regressão Polinomial

A regressão polinomial é representada pela seguinte equação [GIL, 2002]:

$$Y = \alpha_0 + \alpha_1 X + \alpha_2 X^2 + ... + \alpha_k X^k \tag{2.6}$$

Onde: Y é a variável dependente, X é a variável independente, α são os parâmetros (coeficientes do polinômio) e k é a ordem do polinômio.

Consiste em determinar os parâmetros de uma função que se ajuste a um conjunto de pontos, utilizando um método de otimização, como o método dos mínimos quadrados.

2.6.3 Métodos de Decomposição

Os métodos de decomposição separam a série temporal em seus componentes, ou seja, tentam prever um padrão para os elementos sazonal, tendência e ciclo, a fim de suavizar o erro. Fazer previsão usando estes métodos envolve extrapolar cada componente separadamente e recombiná-los em uma previsão final. Os métodos de decomposição são freqüentemente úteis, não somente em prover previsões, mas também em prover informações a respeito dos componentes da série de tempo e o impacto de vários fatores, com sazonalidade e ciclicidade, sobre os resultados observados. [Makridakis et al, 1983]

2.6.4 Modelos de Suavização Exponencial

Os modelos de suavização exponencial são amplamente utilizados para previsão de demanda devido a sua simplicidade, facilidade de ajustes e boa acurácia. Estes métodos usam uma ponderação distinta para cada valor observado na série temporal, de modo que valores mais recentes recebam pesos maiores. Assim, os pesos formam um conjunto que decai exponencialmente a partir de valores mais recentes.

Suavização Exponencial Simples

Se a série temporal mantém-se constante sobre um nível médio, uma suavização exponencial simples pode ser usada para a previsão de valores futuros da série. Sua representação matemática vem dada por [Makridakis et al., 1998].

$$\hat{Z}_{t+1} = \alpha Z_t + (1-\alpha)\hat{Z}_t \qquad (2.7)$$

onde $\hat{Z}_{t+1}$ é a previsão da demanda para o tempo *t+1*, feita no período atual *t*; α é a constante de suavização, assumindo valores entre 0 e 1; Z_t é o valor observado na série temporal para o tempo *t*; e, $\hat{Z}_t$ é o valor da previsão feita para o tempo *t*.

Uma forma de medir a acurácia da previsão, é calcular o erro gerado pela mesma, ou seja, $e_t = Z_t - \hat{Z}_t$.

O valor da constante de suavização α é arbitrário. A determinação do melhor valor para a constante pode ser feita iterativamente, utilizando alguma forma de comparação; como por exemplo, a média do quadrado dos erros, MQE. Desta maneira, seleciona-se aleatoriamente um valor inicial para a constante, a partir do qual previsões são geradas; comparam-se os valores previstos com os reais, e calcula-se a média do quadrado das diferenças entre os mesmos; o parâmetro que minimiza essa média é utilizado no modelo final.

A magnitude da constante α determina a velocidade de resposta do modelo frente a mudanças na demanda [Montgomery et al., 1990]. Valores pequenos de α fazem com que o modelo demore a assumir mudanças no comportamento da série; com valores grandes de α, o modelo reage rapidamente. Os modelos de suavização exponencial simples requerem uma estimativa inicial para $\hat{Z}_t$. Quando dados históricos estão disponíveis, pode-se usar uma média simples das *N* observações mais recentes como $\hat{Z}_t$; caso contrário, pode-se utilizar a última observação, ou fazer uma estimativa subjetiva.

Modelo de Holt

O modelo de Holt pode ser utilizado, de maneira satisfatória, em séries temporais com tendência linear. Este modelo emprega duas constantes de suavização, α e β (com valores entre 0 e 1), sendo representado por três equações [Makridakis et al., 1998].

$$L_t = \alpha Z_t + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (2.8)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (2.9)$$

$$\hat{Z}_{t+k} = L_t + kT_t \quad (2.10)$$

As equações (8) e (9) fazem uma estimativa do nível e da inclinação da série temporal, respectivamente. Já a equação (10), calcula a previsão da demanda para os próximos k períodos.

Assim como ocorre na suavização exponencial simples, o método de *Holt* requer valores iniciais, neste caso L_0 e T_0. Uma alternativa para estes cálculos iniciais é igualar L_0 ao último valor observado na série temporal e calcular uma média da declividade nas últimas observações para T_0. Uma outra forma de cálculo é a regressão linear simples aplicada aos dados da série temporal, onde se obtém o valor da declividade da série temporal e de L_0 em sua origem.

Os valores das constantes de suavização no modelo de *Holt* podem ser determinados de forma semelhante à usada na suavização exponencial simples; ou seja, uma combinação de valores para α e β que minimize a MQE.

Modelo de Winters

Os modelos de *Winters* descrevem apropriadamente dados da série onde se verifica a ocorrência de tendência linear, além de um componente de sazonalidade. Dados da componente sazonal caracterizam-se pela ocorrência de padrões cíclicos de variação, que se repetem em intervalos relativamente constantes de tempo.

Os modelos de *Winters* dividem-se em dois grupos: aditivo e multiplicativo. No modelo aditivo, a amplitude da variação sazonal é constante ao longo do tempo, ou seja, a diferença entre o maior e menor valor de demanda dentro das estações permanece relativamente constante no tempo. Já no modelo multiplicativo, a amplitude da variação sazonal aumenta ou diminui em função do tempo.

Modelo Sazonal Multiplicativo

O modelo multiplicativo de *Winters* é utilizado na modelagem de dados sazonais onde a amplitude do ciclo sazonal varia com o passar do tempo. Sua representação matemática vem dada por [Makridakis et al., 1998]

$$L_t = \frac{\alpha Z_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + T_{t-1}); \tag{2.11}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}; \tag{2.12}$$

$$S_t = \gamma \frac{Z_t}{L_t} + (1-\gamma)S_{t-s} \tag{2.13}$$

$$\hat{Z}_{t+k} = (L_t + KT_t)S_{t-s+k} \tag{2.14}$$

onde s é uma estação completa da sazonalidade (por exemplo, s é igual a 12 quando se tem dados mensais e sazonalidade anual); L_t, T_t e S_t representam o nível, a tendência e a sazonalidade da série, respectivamente; $\hat{Z}_{t+k}$ é a previsão para *k* períodos a frente; e, finalmente, γ é a constante de suavização que controla o peso relativo a sazonalidade, variando entre 0 e 1.

A equação (2.11) difere da equação que trata do nível da série no modelo de *Holt*, já que o primeiro termo é dividido por um componente sazonal, eliminando assim a flutuação sazonal de Z_t. A equação (2.12) é exatamente igual à equação da tendência no método de *Holt*. Já a equação (2.6), faz um ajuste sazonal nas observações Z_t.

Como todos os métodos de suavização exponencial, os modelos de *Winters* também necessitam de valores iniciais dos componentes (neste caso, nível, tendência e sazonalidade) para dar início aos cálculos. Para a estimativa do componente sazonal, necessita-se no mínimo de uma estação completa de observações, ou seja, s períodos [Makridakis et al., 1998]. As estimativas iniciais do nível e da tendência são feitas, então, no período s definido para o componente sazonal.

O estimador inicial para o nível da série é dado pela média da primeira estação

$$L_s = \frac{1}{s}(z_1 + z_2 + ... + z_s) \tag{2.14}$$

O cálculo da estimativa inicial para a tendência requer duas estações completas (2*s*)

$$T_s = \frac{1}{s}\left(\frac{z_{s+1} - z_1}{s} + \frac{z_{s+2} - z_2}{s} + ... + \frac{z_{s+s} - z_s}{s}\right) \tag{2.15}$$

Para o componente sazonal, utiliza-se *s* estimativas iniciais

$$S_1 = \frac{z_1}{L_s}, S_2 = \frac{z_2}{L_s}, ..., S_s = \frac{z_s}{L_s}. \tag{2.16}$$

Modelo Sazonal Aditivo

O modelo aditivo de *Winters* é utilizado na modelagem de dados sazonais onde a amplitude do ciclo sazonal permanece constante com o passar do tempo. Suas equações matemáticas são [Makridakis et al., 1998]

$$L_t = \alpha(z_t - S_{t-s}) + (1-\alpha)(L_{t-1} + T_{t-1}) \tag{2.17}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \tag{2.18}$$

$$S_t = \gamma(z_t - L_t) + (1-\gamma)S_{t-s} \tag{2.19}$$

$$\hat{Z}_{t+k} = L_t + kT_t + S_{t-s+k} \tag{2.20}$$

A equação da tendência permanece a mesma utilizada para o modelo multiplicativo equação (2.12). Nas demais equações, a única diferença é que o componente sazonal está efetuando operações de soma e subtração, ao invés de multiplicar e dividir.

Os valores iniciais de L_s e T_s são calculados de forma idêntica ao modelo multiplicativo. Já os componentes sazonais são calculados da seguinte forma:

$$S_1 = z_1 - L_s, S_2 = z_2 - L_s, ..., S_s = z_s - L_s \tag{2.21}$$

Podem ser encontrados na literatura vários outros métodos, como os métodos de *Box-Jenkins*, redes neurais, entre outros. Estes métodos são considerados mais complexos e não serão abordados em detalhes neste trabalho.

A metodologia *Box-Jenkins* gera previsões acuradas da série temporal e oferece uma abordagem bem estruturada para a construção e análise do modelo. [Pellegrini, 2000]

Porém, estes modelos possuem algumas limitações [MONTGOMERY et at, 1990]:

- De maneira geral, são necessárias pelo menos 50 observações para o desenvolvimento de um modelo aceitável de *Box-Jenkins*. Este fato pode impossibilitar a obtenção dos modelos em situações onde não existem muitas observações disponíveis;

- Não existe uma maneira fácil de modificar (ou melhorar) as estimativas dos parâmetros do modelo quando novas observações são acrescidas à série de dados;

- O tempo despendido na construção de um modelo satisfatório costuma ser grande. Existem situações em que centenas, ou talvez milhares de séries temporais estão em estudo, o que pode inviabilizar economicamente a realização de melhorias na acurácia das previsões.

2.6.5 Outros Modelos

Muitas técnicas têm sido testadas com o intuito de se efetuar previsões mais precisas e preferencialmente automatizadas. Muitos estudos na área da Inteligência Artificial (IA) foram dirigidos a este fim.

As Redes Neurais representam uma área relativamente nova e crescente de previsão. Diferentemente das técnicas de previsões estatísticas comuns, como análise de séries temporais, as redes neurais simulam o aprendizado humano. Desta forma, com o passar do tempo e com o uso repetido, as redes neurais podem desenvolver um

entendimento dos relacionamentos complexos que existem entre as entradas e saídas de um modelo de previsão [DAVIS et al, 1997].

A propriedade mais importante das redes neurais é a sua capacidade de "aprender", ou seja, reconhecer padrões e regularidades nos dados. Uma vez feito o aprendizado, a rede está apta a melhorar seu desempenho, e então extrapolar quanto a um comportamento futuro. [PELLEGRINI, 2000].

Em muitos casos, as redes neurais conseguem modelar irregularidades em séries temporais de forma superior a modelos tradicionais. Porém, este método necessita um número maior de dados observados na série temporal e nem sempre se consegue entender o que se passa na modelagem, pois não existe um modelo explícito (MAKRIDAKIS et al, 1998).

Além de Redes Neurais, Algoritmos Genéticos (AG) e Algoritmos Evolutivos (AE) e, dentre estes, a Programação Genética [KABOUDAN, 2000], apresentaram resultados promissores nesta área. E serão estudados com mais detalhes neste trabalho.

CAPÍTULO 3 ALGORITMO GENÉTICO E A PROGRAMAÇÃO GENÉTICA

3.1 Computação Evolucionista

A Computação Evolucionista compreende um conjunto de técnicas de busca e otimização inspiradas na evolução natural das espécies. Desta forma, cria-se uma população de indivíduos que vão reproduzir e competir pela sobrevivência. Os melhores sobrevivem e transferem suas características a novas gerações. As técnicas atualmente incluem [BANZAHAF 1998]: Programação Evolucionária, Estratégias Evolucionárias, Algoritmos Genéticos e Programação Genética. Estes métodos estão sendo utilizados, cada vez mais, pela comunidade de inteligência artificial para obter modelos de inteligência computacional [BARRETO 1997].

Computação Evolucionista (CE) é uma das áreas da Inteligência Artificial, que engloba um conjunto de métodos computacionais, inspirados na Teoria da Evolução das Espécies de Charles Darwin [DARWIN, 2000] para a solução de problemas. Segundo sua teoria, na natureza sobrevivem os indivíduos que possuem maior capacidade de se adaptarem ao meio ambiente, suas características genéticas são repassadas para as gerações seguintes e melhoradas. Assim a nova geração será composta por indivíduos com material genético melhor do que os da população anterior. Em 1975, Holland publicou "*Adaptation in Natural and Artificial Systems*", ponto inicial dos Algoritmos Genéticos (AGs). David E. Goldberg, aluno de Holland, nos anos 80 obteve seu primeiro sucesso em aplicação industrial com AGs. Desde então os AGs são utilizados para solucionar problemas de otimização e aprendizado de máquinas.

Existe uma série de variações em torno de algoritmos evolucionistas. Algumas recebem destaque e entre elas podem-se citar os algoritmos genéticos com codificação real, sistemas de classificação e programação genética.

3.2 Algoritmos Genéticos

Os métodos de otimização podem ser classificados em: métodos probabilísticos, numéricos e enumerativos, existindo ainda os híbridos. Os AG pertencem à classe dos métodos probabilísticos de busca e otimização, embora não sejam aleatórios. Usa-se o conceito de probabilidade, mas os AG não são simplesmente buscas aleatórias quaisquer. Pelo contrário, eles tentam dirigir a busca para regiões do espaço onde é provável que os pontos ótimos estejam.

Os AG podem ser definidos como métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. Nessa técnica, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores, à medida que o processo evolutivo continua.

O AG inicia-se com uma população de indivíduos, soluções possíveis para o problema, gerados de forma aleatória. A informação que um indivíduo disponibiliza, e que atende aos valores dos parâmetros do problema em equação, é representada por um cromossoma, análogo à estrutura vigente no DNA. Um cromossoma é, por sua vez, composto por um conjunto de genes (caracteres). Um valor possível para um gene é designado por alelo. A qualidade de cada solução (cromossoma) é medida por uma função chamada aptidão, sendo os indivíduos avaliados de acordo com esta. Em cada ciclo, parte-se da população atual.

O fluxo geral de um AG simples é formado pelas etapas:

- **Inicialização:** geralmente a população de N indivíduos é gerada aleatoriamente ou através de algum processo heurístico, onde se busca cobrir a maior área possível do espaço de busca.

- **Avaliação e adequabilidade:** uma função objetivo para cada membro da população é necessária para avaliar a adequabilidade do indivíduo e determinar, assim, a sua probabilidade de se perpetuar ou transmitir suas características às futuras gerações.

-

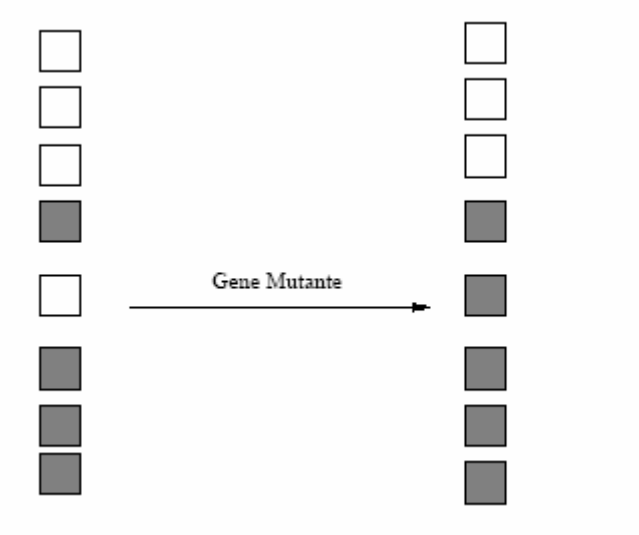

Figura 3.2 – Representação da Mutação

Como o processo é iterativo, um critério de parada deve ser determinado. É desejado que ele pare quando o ponto ótimo é atingido, mas não é essa a realidade na maioria dos problemas. Existem problemas inerentes aos métodos de otimização como, por exemplo, não ser possível afirmar que o ponto ótimo encontrado seja um ponto ótimo global. Para contornar esse problema, utiliza-se como critério de parada um número máximo de gerações ou um tempo limite de processamento. Estas informações podem ser descritas na figura 3.3 a seguir.

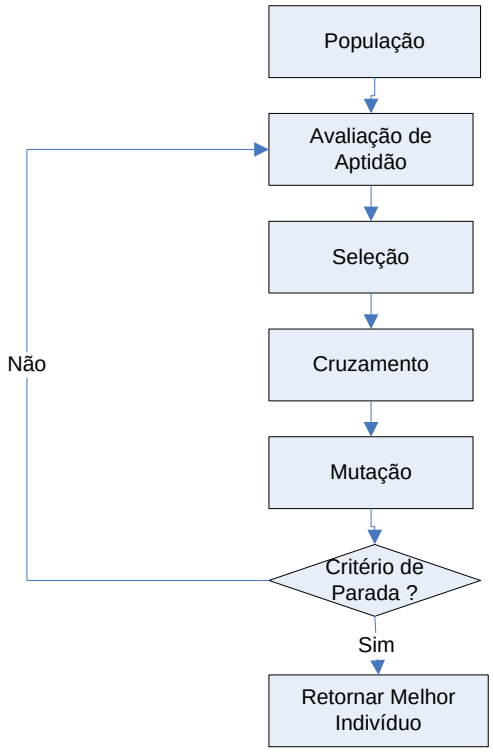

Figura 3.3 – Estrutura Básica do Um algoritmo

3.3 Programação Genética

A Programação Genética é uma abordagem para a geração automática de programas de computador desenvolvida por John Koza [KOZA 1989; KOZA 1992]. A técnica se baseia na combinação de idéias da teoria da evolução (seleção natural), genética (reprodução, cruzamento e mutação), inteligência artificial (busca heurística) e teoria de compiladores (representação de programas como árvores sintáticas). Basicamente, a Programação Genética é um algoritmo que busca, dentre um espaço relativamente grande, porém restrito de programas de computador, uma solução ou, pelo menos, uma boa aproximação para resolver determinado problema [BRUCE 1995].

O paradigma da Programação Genética foi desenvolvido por John Koza [KOZA 1989; KOZA 1992] com base nos trabalhos de John Holland em Algoritmos Genéticos [HOLLAND 1975]. Atualmente representa uma área muito promissora de pesquisa em Inteligência Artificial devido a sua simplicidade e robustez. Seu uso tem sido estendido a problemas de diversas áreas do conhecimento, como por exemplo: biotecnologia,

engenharia elétrica, análises financeiras, processamento de imagens, reconhecimento de padrões, mineração de dados, linguagem natural, previsão de séries temporais, dentre muitas outras [WILLIS 1997].

Na Programação Genética, o Algoritmo Evolutivo opera numa população de programas computacionais que variam de forma e tamanho [KOZA, 1992]. Esta população de indivíduos será evoluída de modo a gerar uma nova população constituída por indivíduos melhores, utilizando operadores de reprodução, cruzamento e mutação. O processo é guiado por uma função de aptidão (*fitness*) que mede o quanto o indivíduo está próximo da solução do problema. Indivíduos que possuem maior capacidade de adaptação têm melhores chances de sobreviver.

Por manipular programas diretamente, a Programação Genética lida com uma estrutura relativamente complexa e variável. Tradicionalmente, esta estrutura é uma árvore de sintaxe abstrata composta por funções em seus nós internos e por terminais em seus nós-folha. A especificação do domínio do problema é feita simplesmente pela definição dos conjuntos de funções e terminais [KOZA 1992].

3.3.1 Visão Geral do Algoritmo de Programação Genética

O algoritmo de Programação Genética é simples e pode ser descrito resumidamente como:

Criar aleatoriamente uma população de programas;

Executar os seguintes passos até que um Critério de Término seja satisfeito:

Avaliar cada programa através de uma função heurística (*fitness*), que expressa quão próximo cada programa está da solução ideal;

Selecionar os melhores programas de acordo com o *fitness*;

Aplicar a estes programas os operadores genéticos (reprodução, cruzamento e mutação);

Retornar com o melhor programa encontrado;

Cada execução deste laço representa uma nova geração de programas. Tradicionalmente, o Critério de Término é estabelecido como sendo encontrar uma solução satisfatória ou atingir um número máximo de gerações [KOZA 1992]. Porém, existem abordagens baseadas na análise do processo evolutivo, isto é, o laço permanece enquanto houver melhoria na população [KRAMER 2000].

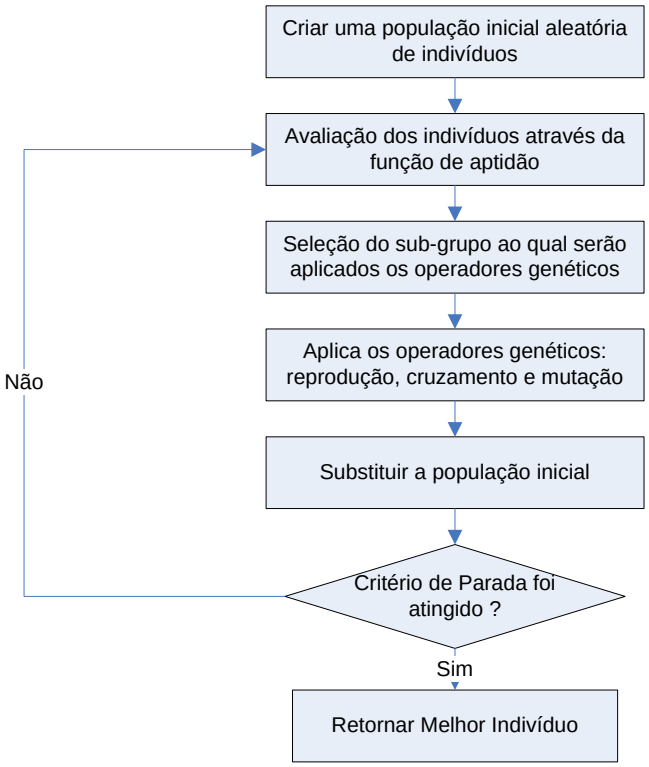

Figura 3.4 – Estrutura Básica do Algoritmo de Programação Genética

3.4 Representação de Programas

Na Programação Genética, os indivíduos são representados por árvores de sintaxe, ou seja, são formados por uma combinação dos conjuntos de Funções (*F*) e Terminais (*T*), de acordo com o domínio do problema. Um indivíduo da população que tem a forma: $x^2 + x + 2$, é representado na notação pré-fixa, utilizada pela PG de acordo com a equação (3.1) e sua representação em forma de árvore de sintaxe está mostrada na figura (3.5).

(+ (* *x x*) (+ *x* 2)) Eq (3.1)

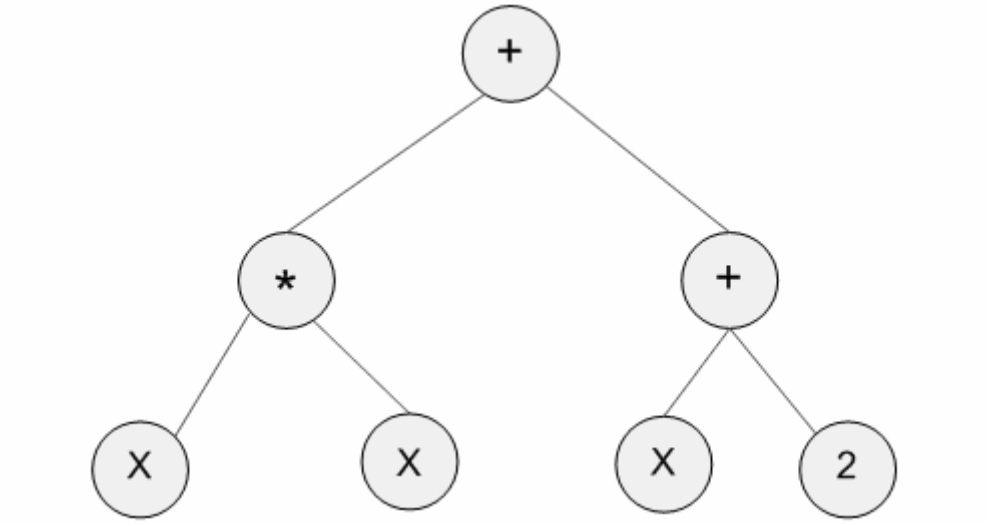

Figura 3.5 – Árvore de Sintaxe da Programação Genética

Em todo algoritmo de Programação Genética deve-se definir inicialmente os conjuntos *F*, de funções e *T*, de terminais. No conjunto *F*, definem-se os operadores aritméticos, funções matemáticas, operadores lógicos, entre outros. O conjunto *T* é composto pelas variáveis e constantes e fornece um valor para o sistema, enquanto que o conjunto de funções processa os valores no sistema. Juntos, os conjuntos de funções e terminais representam os nós.

Pode-se citar como exemplo, o conjunto *F*, dos operadores aritméticos, e o conjunto *T*, de terminais, da seguinte forma:

F = {+, -, *, /} e T = { x, 2}

Um indivíduo resultante da combinação destes dois conjuntos pode ser o indivíduo apresentado na equação (3.1). A escolha dos conjuntos *F* e *T* influenciam, consideravelmente, na solução apresentada pela Programação Genética. Se no conjunto *F* houver poucos operadores disponíveis, a Programação Genética provavelmente não será capaz de apresentar uma boa solução para o problema, por outro lado, ao disponibilizar muitas operações, o programa poderá ficar extenso, provocando esforço computacional desnecessário. O mais aconselhável é iniciar com os operadores básicos, tais como: adição, subtração, multiplicação, divisão, conjunção, disjunção e negação e ir adicionando outros operadores caso a solução apresentada não seja suficientemente boa.

Da mesma forma deve-se ter cuidado ao formar o conjunto das variáveis e constantes, pois o algoritmo de Programação Genética tem habilidade de combinar as variáveis, transformando-as em novas variáveis [BANZHAF,1998].

O espaço de busca da PG é constituído por todas as árvores que possam ser construídas através da combinação dos conjuntos *F* e *T*.

3.4.1 Fechamento e Suficiência

Para garantir a que a solução do problema ser viável, John Koza definiu a propriedade de Fechamento (*closure*) [KOZA 1992]. Para satisfazê-la, cada função do conjunto *F* deve aceitar, como seus argumentos, qualquer valor que possa ser retornado por qualquer função ou terminal. Esta imposição garante que qualquer árvore gerada pode ser avaliada corretamente.

Um caso típico de problema de Fechamento é a operação de divisão. Matematicamente, não é possível dividir um valor por zero. Uma abordagem possível é definir uma função alternativa que permita um valor para a divisão por zero. É o caso da função de divisão protegida (*protected division*) % proposta por [KOZA 1992]. A função % recebe dois argumentos e retorna o valor 1 (um) caso seja feita uma divisão por zero e, caso contrário, o seu quociente.

Para garantir a convergência para uma solução, John Koza definiu a propriedade de Suficiência (*sufficiency*) onde os conjuntos de funções *F* e o de terminais *T* devem ser capazes de representar uma solução para o problema [KOZA 1992]. Isto implica que deve existir uma forte evidência de que alguma composição de funções e terminais possa produzir uma solução.

3.4.2 População Inicial

O primeiro passo de uma PG é definir sua população inicial, ou seja, deve-se criar uma população de estruturas, ou estruturas de programas para posterior evolução.

Um dos principais parâmetros da Programação Genética é o tamanho máximo permitido para um programa, na PG este parâmetro é definido como sendo a

profundidade máxima da árvore, ou seja, o número máximo de nós da árvore. Esta profundidade é a maior profundidade que será permitida entre a raiz e todos os nós terminais de um mesmo indivíduo (TERADA, 1991). A profundidade de um nó em uma árvore é à distância do nó raiz até o nó *n*. A estrutura de árvore é construída a partir dos conjuntos *F* e *T*.

É muito usual a população inicial ser composta com árvores geradas aleatoriamente a partir faz funções *F* e terminais *T*. Porém, a "qualidade" da população inicial é um fator crítico para o sucesso do processo evolutivo [DAIDA 1999]. A população inicial deve ser uma amostra significativa do espaço de busca, apresentando uma grande variedade de composição nos programas, para que seja possível, através da recombinação de seus códigos, convergir para uma solução.

Existem vários métodos para inicializar uma população em estrutura de árvores, que podem melhorar a qualidade dos programas da população inicial. Os mais comuns são [LUKE; PAINAT, 2001]: *Full, Grow, ramped-half-and-half* [KOZA, 1992], que é uma combinação dos métodos *Full e Grow, random-branch* [CHELLAPILLA, 1997], *uniform* [BOHM; GEYER-SCHULZ, 1996], que serão brevemente descritos a seguir.

Método *Grow*: os nós são selecionados aleatoriamente dos conjuntos *F* e *T* (exceto para o nó raiz que é retirado do conjunto *F*), por este motivo o método produz árvores de formatos irregulares. Se uma ramificação contém um nó terminal, esta ramificação pára, mesmo que a profundidade máxima não tenha sido atingida.

Método *Full*: Ao invés de escolher aleatoriamente os nós do conjunto de funções e de terminais, o método *Full*, escolhe somente funções até que um nó de profundidade máxima seja selecionado, então ele passa a escolher somente terminais [BANZHAF,1998]. O resultado disso é que cada árvore atinge a profundidade máxima.

Método *Half-and-half*: o método Half-and-half é uma combinação dos métodos *Grow* e *Full*, ou seja, utiliza o método *Full* em 50% das vezes e o método *Grow* nas outras 50%, tem por objetivo gerar um número

outra metade pelo método *Grow*. As desvantagens deste método, segundo Luke, [LUKE; PAINAT, 2000] são:

- Se o conjunto de funções é maior que o conjunto de terminais, a tendência será de gerar a maior árvore possível;
- A escolha do parâmetro de profundidade máxima da árvore é realizada de forma proporcional e não aleatória;
- A faixa de profundidade é fixa (usualmente entre 2 e 6), independente do tamanho da árvore e dependendo do número de argumentos (aridade) de cada função, mesmo tendo a mesma profundidade, as árvores geradas pode ter tamanhos muito diferentes.

Método *Random-Branch*: neste método, ao invés de se informar a profundidade máxima da árvore, é informado seu tamanho máximo, *S*, este valor é igualmente dividido dentre as árvores de um nó-pai não terminal, o que faz com que muitas árvores não viáveis sejam geradas [CHELLAPILLA,1997], este método é menos restritivo, apesar de ter complexidade linear [LUKE; PAINAT, 2001];

Método *Uniform*: Criado por Bohm, o método *uniform* foi desenvolvido com o objetivo de criar árvores uniformes, geradas a partir do conjunto de todas as árvores possíveis [BOHM. 1996]. O algoritmo calcula várias vezes quantas árvores poderão ser geradas para cada tamanho desejado, por este motivo o método possui um alto custo computacional.

3.4.3 Função de Aptidão

A função de aptidão é a medida utilizada pela PG durante o processo de evolução, que irá dizer quão bem o programa aprendeu a predizer as saídas dentro de um processo de aprendizagem [BANZHAF, 1998].

A definição de uma função de aptidão é feita de acordo com o domínio do problema. Em geral, nos problemas de otimização esta função é definida como sendo a função objetivo, porém nada impede que se defina uma outra função. Uma boa escolha da função de aptidão pode ser responsável pelo bom funcionamento do algoritmo da

PG. Especificamente, no caso de Séries Temporais, pode-se utilizar como função de aptidão, a função que mede o erro calculado entre o valor previsto e o valor real, como por exemplo, o erro quadrático médio. Quanto menor for o erro obtido, melhor será o ajuste do modelo de previsão. O que se deseja, portanto, é minimizar a função de aptidão ou função objetivo.

A função de aptidão é uma forma de se diferenciar os melhores dos piores indivíduos. Se esta função for bem definida há uma grande probabilidade de que o algoritmo gere uma solução muito próxima da solução ótima. Algumas das funções de aptidão mais utilizadas são [KOZA, 1992]:

1. **Aptidão Nata (*raw fitness*)**: é uma maneira de calcular qualquer melhoria que o programa tenha obtido no conjunto de treinamento. A continuidade é uma propriedade importante da função de aptidão, pois isto permite que a PG melhore os programas iterativamente. O método mais comum de aptidão nata é através da avaliação do erro cometido, isto é, a soma de todas as diferenças absolutas entre o resultado obtido pelo programa e o seu valor correto.

2. **Aptidão Padronizada (*standardized fitness*)**: é uma função transformada da função de aptidão nata, na qual o valor zero é o valor designado ao melhor indivíduo. Devido ao fato da aptidão nata depender do domínio do problema, um valor bom pode ser um valor pequeno (quando se avalia o erro) ou um valor grande (quando se avalia a taxa de eficiência). A avaliação da aptidão padronizada é feita através de uma função de adaptação do valor da aptidão nata de forma que quanto melhor o programa, menor deve ser a aptidão padronizada.

3. **Aptidão Ajustada (*adjusted fitness*):** é obtida a partir da aptidão padronizada, seu valor varia entre zero e um, onde os maiores valores são associados aos melhores indivíduos. Se *f(i, t)* é a aptidão padronizada do indivíduo *i* na geração *t*, a aptidão ajustada, *a(i, t)*, é calculada pela equação (3.2):

$$a(i,t) = \frac{1}{1 + f(i,t)} \qquad \text{Eq (3.2)}$$

4. **Aptidão Normalizada (*normalized fitness*):** é uma função de aptidão na qual o seu valor está entre zero e um. A soma de todas as funções normalizadas dentro de uma população deve ser igual a um. Se *a(i,t)* é a aptidão ajustada do indivíduo *i* na geração *t*, sua aptidão normalizada, *n(i, t)*, será dada de acordo com a equação (3.3).

$$n(i,t) = \frac{1}{\sum_{k=1}^{m} a(k,t)} \qquad \text{Eq (3.3)}$$

3.4.4 Métodos de Seleção

O método de seleção tem por objetivo escolher quais programas deverão sofrer a ação dos operadores genéticos e compor uma nova geração. Sendo que a "qualidade" de um programa é dada pelo seu valor de aptidão, a seleção deve preferenciar, de alguma forma, os programas que apresentem os melhores valores de aptidão.

Existem diferentes operadores de seleção e a decisão de qual destes operadores será utilizado pela PG é uma tarefa importante durante a utilização do algoritmo. O método de seleção é responsável pela velocidade da evolução e geralmente citado como responsável pelos casos de convergência prematura que poderão determinar o sucesso do algoritmo evolucionário (BANZHAF, 1998). Alguns destes métodos são descritos a seguir.

Seleção Proporcional (*Proportional Selection*): este método de seleção é aplicado aos Algoritmos Evolutivos e especifica a probabilidade de que cada indivíduo seja selecionado para a próxima geração. Para o indivíduo *i*, a probabilidade de ser selecionado para a próxima geração é dada pela equação (3.4).

$$p_i = \frac{f_i}{\sum_j f_j} \qquad \text{Eq 3.4}$$

onde f_j representa o valor de aptidão do indivíduo e $\sum_j f_j$ representa o valor acumulado de aptidão. Os indivíduos que possuem maior aptidão possuem uma probabilidade maior de serem selecionados para a próxima geração, este pode ser

considerado, então, como um problema de maximização. Em geral o melhor indivíduo da população é copiado para a população seguinte, a esta escolha dá-se o nome de elitismo, que tem por objetivo privilegiar a melhor solução, de forma que este indivíduo propague suas características para a população seguinte. No entanto, se um indivíduo possui uma alta aptidão em relação aos demais, a probabilidade de que ele seja selecionado tende a ser alta, e como os demais indivíduos da população possuem uma aptidão bem menor, a tendência é que o indivíduo que possui maior valor de probabilidade, seja selecionado muitas vezes, fazendo com que haja uma convergência prematura da solução que poderá não ser a solução ótima, por outro lado, se os indivíduos apresentarem aptidões muito próximas, sua probabilidade de serem selecionados é a mesma, assim a população seguinte será basicamente a mesma, não havendo evolução [BANZHAF, 1998];

Truncamento *(truncation selection)*: este é o segundo método mais popular utilizado para seleção e provém dos algoritmos de Estratégias Evolucionárias [SCHWEFEL, 1995], com base em um valor de limiar (*threshold*) T que está no intervalo entre 0 e 1, a seleção é feita aleatoriamente entre os T melhores indivíduos [MUHLENBEIN; SCHIERKAMP-VOSSEN, 1993]. Se, por exemplo, $T = 0{,}6$, isto significa que a seleção é feita entre os 60% melhores indivíduos e os demais são descartados;

Ranqueamento *(Ranking Selection)*: no método de seleção por ranking [GREFENSTETTE; BAKER,1989] [WHITLEY, 1989] os indivíduos são ordenados de forma crescente de acordo com seu valor de aptidão. Assim, a cada indivíduo, é atribuído um número inteiro de acordo com sua posição no ranking, quanto melhor o ranking do indivíduo, melhor sua aptidão em relação aos demais indivíduos da população e, portanto, melhores são suas chances de ser sorteado.

Torneio: a seleção por torneio não é baseada na competição dentro da geração completa, mas apenas num sub-conjunto da população. Um certo número de indivíduos, que é o tamanho do torneio, é selecionado aleatoriamente, e é realizada uma competição seletiva. As características dos melhores indivíduos no torneio são substituídas pelas características dos piores indivíduos. No menor torneio possível é permitido que dois indivíduos participem da reprodução. O resultado da reprodução retorna à população substituindo o perdedor do torneio [BANZHAF, 1998]. Sua ordem de complexidade é

linearmente proporcional ao tamanho da população, pois independe de uma ordenação prévia dos elementos e do cálculo das probabilidades de seleção [BICKLE, 1995].

3.4.5 Operadores Genéticos

Após os indivíduos terem sido selecionados por um dos métodos de seleção, os operadores genéticos são aplicados a estes indivíduos para então gerar a nova população. Diversos operadores genéticos foram criados, porém os mais importantes e mais utilizados são [KOZA, 1992]: cruzamento, mutação e reprodução, que serão descritos a seguir:

Reprodução: um programa é selecionado e copiado para a próxima geração sem sofrer nenhuma mudança em sua estrutura.

Cruzamento: dois indivíduos pais são selecionados e seu material genético é combinado, gerando assim um novo indivíduo, que espera-se que seja melhor do que os anteriores, pois foram criados a partir da combinação das melhores partes de cada indivíduo. O operador de cruzamento visa guiar a solução de maneira a combinar as melhores soluções na busca da solução ótima. Basicamente, o operador funciona da seguinte maneira:

- Escolhe dois indivíduos através do valor de sua função de aptidão;
- Seleciona aleatoriamente, em cada indivíduo, um ponto de cruzamento;
- Permuta as sub-árvores dos dois indivíduos gerando os filhos, que farão parte da nova população. Um exemplo do operador de cruzamento pode ser visto na figura 3.6.

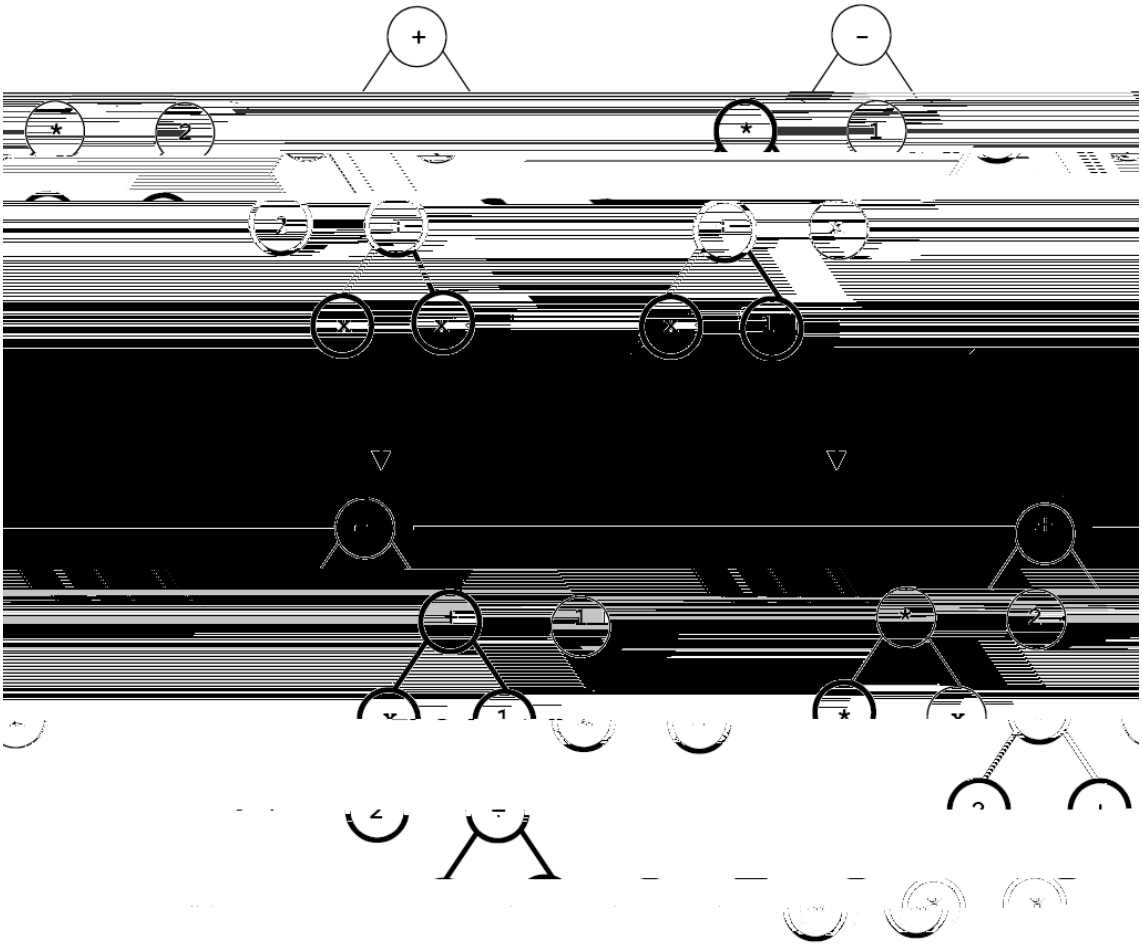

Figura 3.6 - Exemplo de Cruzamento Entre Dois Programas

Mutação: o operador efetua alterações em um indivíduo somente, ou seja, nenhum novo indivíduo é gerado. Normalmente, após ter sido efetuado um cruzamento, a probabilidade de que o indivíduo gerado seja submetido a uma operação de mutação é baixa, em geral, este índice varia na faixa de 0,1% a 0,5%. A probabilidade de mutação é um parâmetro definido em cada execução. Nesta operação, seleciona-se aleatoriamente, um ponto do indivíduo e substitui-se a sub-árvore, cujo nó foi selecionado por uma nova sub-árvore gerada aleatoriamente. Esta sub-árvore está sujeita às mesmas limitações quanto à profundidade e tamanho, do que as árvores geradas na população inicial. Este novo indivíduo é inserido novamente na população. A função deste operador é inserir diversidade na população, fazendo com que os novos indivíduos explorem novas áreas do espaço de busca [MICHALEWICZ, 1997], evitando máximos e mínimos locais. Porém ao se inserir muita diversidade numa

população, a mesma poderá não convergir para um ótimo global, ou mesmo poderá não convergir, oscilando indefinidamente, razão pela qual a taxa de mutação deve ser baixa.

3.4.6 Parâmetros utilizados na programação genética

As definições dos parâmetros a serem utilizados podem ter grande influência nos resultados obtidos através da PG, outros parâmetros também podem influenciar no tempo computacional. Como os citados a seguir:

Tamanho da População: neste parâmetro deve ser informado o número de indivíduos que a população deverá conter. A escolha deste parâmetro deve ser criteriosa, pois de sua escolha depende a qualidade dos resultados. Uma população pequena restringe o espaço de busca, enquanto que se a população for grande, poderá estar provocando um esforço computacional excessivo, sem grandes alterações nos resultados [BANZHAF, 1998];

Taxa de cruzamento: este parâmetro define a taxa de utilização do operador de cruzamento que deve ocorrer em cada geração. Se esta taxa for alta, pode haver uma convergência pré-matura do algoritmo, por outro lado se for muito pequena, o algoritmo poderá levar muito tempo para obter a convergência necessária para que uma boa solução para problema;

Taxa de mutação: este parâmetro define o percentual de mutações que deverão ocorrer para cada geração, se esta taxa for alta poderá tornar a busca completamente aleatória;

Número de gerações: o número de gerações define quantas vezes o processo evolutivo será executado, pode funcionar como critério de parada.

3.4.7 Critério de Parada

O critério mais utilizado é limitar o número máximo de gerações ou até que uma boa solução seja encontrada [KOZA,1992], porém existem outros critérios baseados no acompanhamento do processo evolutivo, ou seja, enquanto houver melhoria na população, o processo evolutivo prossegue [KRAMER; ZHANG; GAPS, 2000].

CAPÍTULO 4 TRATAMENTO DE VALORES FALTANTES

Ao se trabalhar com séries temporais diversas várias dificuldades podem surgir durante o processo, seja na preparação dos dados, na definição dos operadores, na análise do algoritmo a ser adotado ou, mesmo, na interpretação dos conhecimentos gerados. E a preparação dos dados é de grande importância, na qual são despendidos esforços para garantir a qualidade aos dados, que é de fundamental para uma análise eficiente.

A fase de pré-processamento de dados inicia-se após a coleta e organização dos dados. Podem existir diversos objetivos na fase de pré-processamento de dados. Um deles é solucionar problemas nos dados, tais como identificar e tratar dados corrompidos, atributos irrelevantes e valores desconhecidos.

A ausência de dados representa um problema para os algoritmos de análise de séries temporais, uma vez que pode dificultar sua aplicação a problemas reais com bancos de dados incompletos.

4.1 Valores Faltantes

A inexistência ou falta de valores na base de dados esta presente em grande parte das pesquisas e, no estudo de séries temporais, este processo pode ser um problema crítico. Este fenômeno pode induzir a erros, comprometendo todo o esforço desprendido, e muitas são as razões para a sua ocorrência, tais como: recusa em se prestar a informação, incapacidade do entrevistado, indisponibilidade do meio pesquisado, não retorno de questionários, impossibilidade de contato com a fonte, perda de registros, erro de digitação, falha no equipamento de coleta, dentre outros.

A inexistência de dados pode ocorrer por muitas razões. Existe um procedimento que é bem simples para estimar os dados inexistentes é de eliminar os registros que

tenham dados ausentes, ou seja, só se trabalha com registros completos. Esse item pode prejudicar a análise da serie já que pode acontecer perda no tamanho no conjunto final.

Outro procedimento usado trata-se de métodos utilizados para se incorporar dados no conjunto de registros. Esta técnica utiliza basicamente a estatística, ou seja, utiliza-se probabilidades, cálculos de correlações, médias e desvios padrões para gerar o valor faltante.

Uma metodologia será empregada neste trabalho substituindo os valores faltantes pelos vizinhos mais próximos. Esse método envolve o preenchimento dos dados faltantes baseado na substituição do valor por um vizinho mais próximo no tempo.

4.2 Método dos Vizinhos mais próximos no tempo

Antes de falar sobre o método dos vizinhos mais próximos no tempo, será explicado o algoritmo do vizinho mais próximo (*KNN*), já que o método é baseado nele.

O "*KNN*" é baseado no conceito estatístico de buscar os objetos vizinhos mais similares. Como usualmente utiliza medidas diretas de distância para calcular a similaridade entre os objetos, torna-se muito sensível a dados fora do padrão, ou seja, objetos que tenham um valor extremamente alto ou baixo. Outro cuidado que deve ser tomado é com a unidade (m3, kg, $) em que os dados estão expressos: valores de atributos coerentes entre si e com os objetos que representam, medidos com unidades diferentes podem necessitar de alguma normalização. [ANDRADE,2004]

Antes de iniciar seu processo, o algoritmo, necessita receber o parâmetro k, que representa o número de objetos mais próximos que serão utilizados Assim, o primeiro passo do processamento propriamente dito será selecionar k objetos do conjunto de dados.

Baseado nesse conceito [LATINI, PASSERINI, 2004] sugere o método dos vizinhos mais próximos no tempo que consiste em criar uma matriz de periodicidade. Exemplo, se a série tem uma periodicidade diária e os valores são coletados a cada hora

do dia, a matriz terá o número de linhas igual ao número de dias coletados e o número de colunas será igual número de horas do dia.

A matriz ficaria igual à tabela abaixo.

$$\begin{bmatrix} x_{11}, x_{12} \ldots x_{1n} \\ x_{21}, x_{22} \ldots x_{2n} \\ \ldots \\ x_{k1}, x_{k2} \ldots x_{kn} \end{bmatrix}$$

Onde *n* é o número de horas do dia e *k* é o número de dias.

Neste trabalho são usadas as seguintes técnicas:

- Substituição *(Replacement)*;
- Vizinho mais próximo com interpolação das colunas;
- Vizinho mais próximo com interpolação de linhas;
- Preenchimento suavizado Simples *(Smooth Fill with a matrix 3x3 and 4 elements);*
- Preenchimento Suavizado Ponderado *(Smooth Fill with a matrix 3x3 and 8 elements);*

4.2.1 Substituição *(Replacement)*

O algoritmo de substituição restaura dados faltantes com valores ocorridos no mesmo momento em períodos diferentes. Exemplo: Dada uma série de dados coletados por hora durante 3 dias, a matriz 3 x 24 seria formada.

X_1 = valor da mesma hora coletada no dia anterior

X_2 = valor da mesma hora coletada no dia posterior

Y = valor da hora anterior a hora que deve ser preenchida.

X = a hora a ser preenchida.

Se X_1 ouX_2 for nulo, a substituição é simples selecionando o valor não nulo, caso ambos sejam nulos este algoritmo não poderá ser usado.

Caso ambos sejam não nulos o calculo será da seguinte maneira:

$$X = X_1 \text{ se } Z_1 \leq Z_2 \text{ ou } X = X_2 \text{ se } Z_1 > Z_2 \qquad (4.1)$$

Onde Z_1e Z_2 são as distâncias de Y, ou seja,

$$Z_1 = |Y - X_1| \qquad (4.2)$$

$$Z_2 = |Y - X_2| \qquad (4.3)$$

Exemplo seja a série 3 x 24 (tabela 1)

10	12	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	x	x	x	x	x	x	x	x	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

Tabela Matriz 3 x 24 contendo valores faltantes (x)

O primeiro preenchimento será o ponto (2,15). Usando as equações 4.1, 4.2 e 4.3 temos:

$X_1 = 25;\ X_2 = 34;\ Y = 20$

$Z_1 = |25 - 20| = 5\ ;\ Z_2 = |34\text{-}20| = 14$

Com isso a menor distância é Z_1, então $X = X_1$.

A figura 4.1 mostra o valor identificado tracejado em azul:

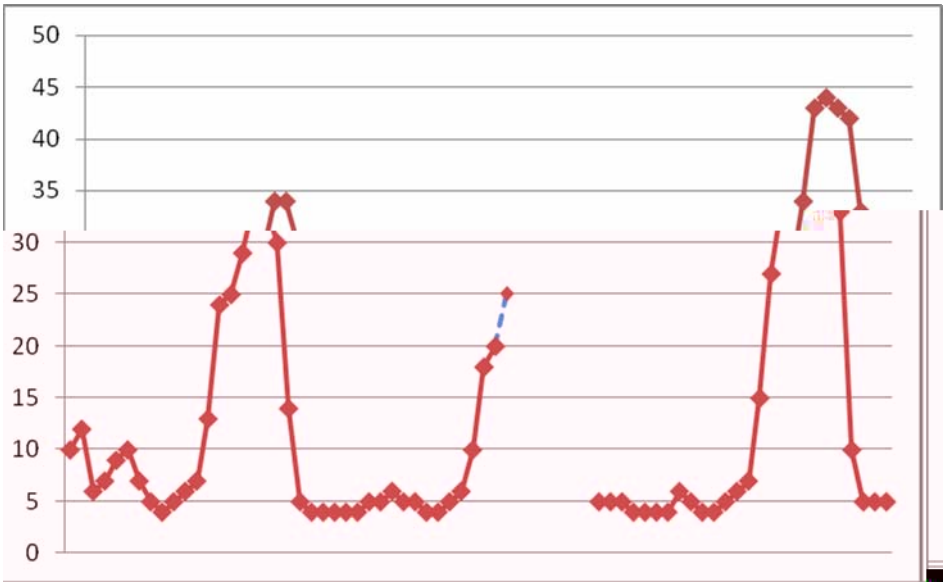

Figura 4.1– Gráfico Da Série *replacement*.

Fazendo o cálculo para todos os dados faltantes o gráfico ficará (Figura 4.2):

10	7	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	25	29	34	34	30	33	10	5	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

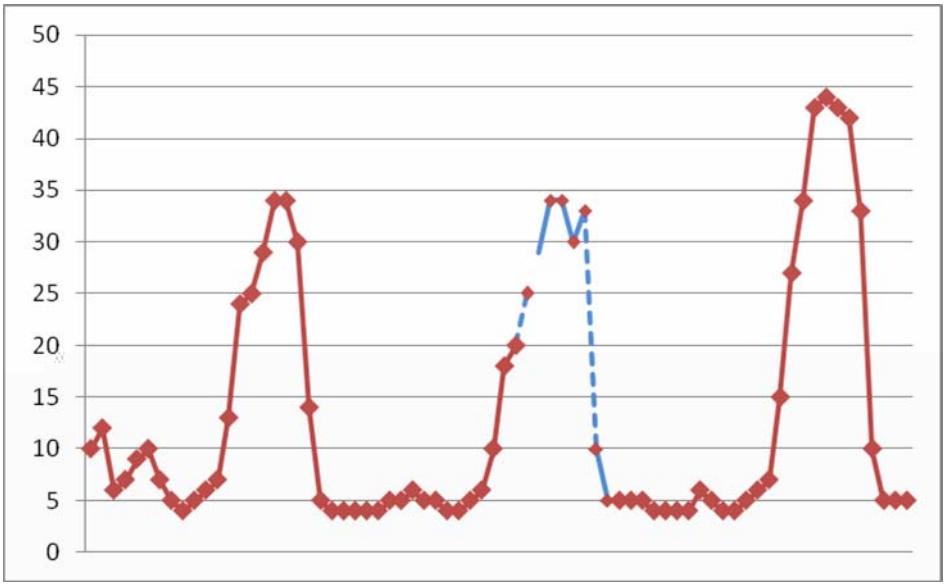

Figura 4.2– Gráfico Da Série usando *replacement*.

4.2.2 Vizinho mais próximo com interpolação da Coluna

O algoritmo da interpolação da coluna trata o valor faltante através da média dos valores da linha anterior e posterior da mesma coluna do valor faltante. Usando o exemplo anterior,

X_1 = valor da mesma hora coletada no dia anterior

X_2 = valor da mesma hora coletada no dia posterior

Y = média dos vizinhos.

X = a hora a ser preenchida.

Se X_1 ou X_2 for nulo, pode-se calcular um dos valores usando o algoritmo *replacement*.

A tabela seria:

10	7	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	29	36	39	38	36	23	7	4	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

E o gráfico (figura 4.3):

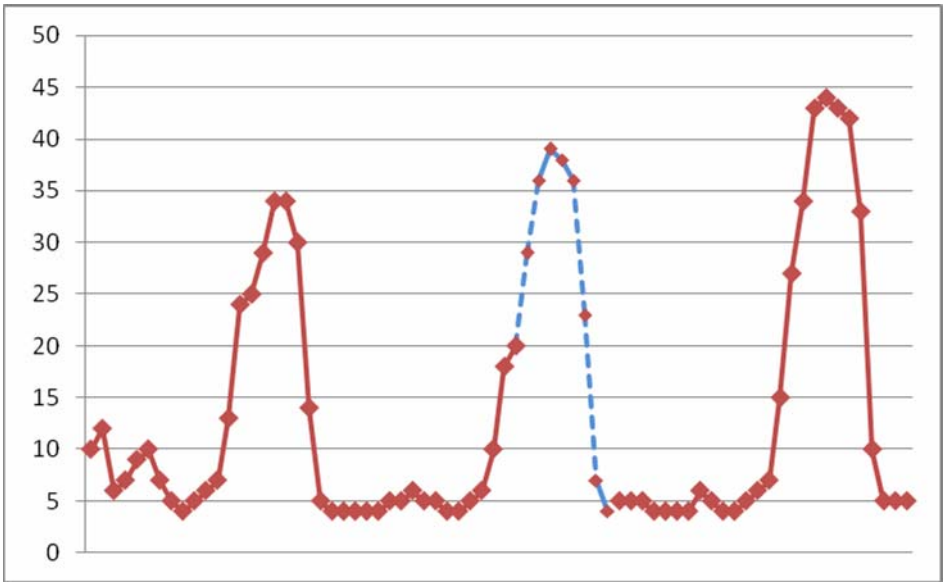

Figura 4.3– Gráfico Da Série usando interpolação da coluna.

4.2.3 Vizinho mais próximo com interpolação das linhas

O algoritmo do vizinho mais próximo com interpolação das linhas usa a média do valor anterior e o posterior para calcular o valor do dado faltante. Baseado no exemplo estudado até o momento:

X_1 = valor de uma hora anterior ao dado faltante

X_2 = valor da uma hora posterior ao dado faltante

Y = média dos vizinhos.

X = a hora a ser preenchida.

Se o valor faltante estiver no início da linha da matriz ou no final da linha este algoritmo não poderá ser usado. Como o cálculo usa somente a linha, o valor de X_1 não poderá ser nulo. Caso seja aplica-se primeiramente o algoritmo de interpolação de colunas para definir o valor de X_1.

Com essas informações a tabela ficaria:

10	7	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	18	16	14	13	12	10	8	6	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

Resultado pode ser conferido no gráfico (Figura 4.4):

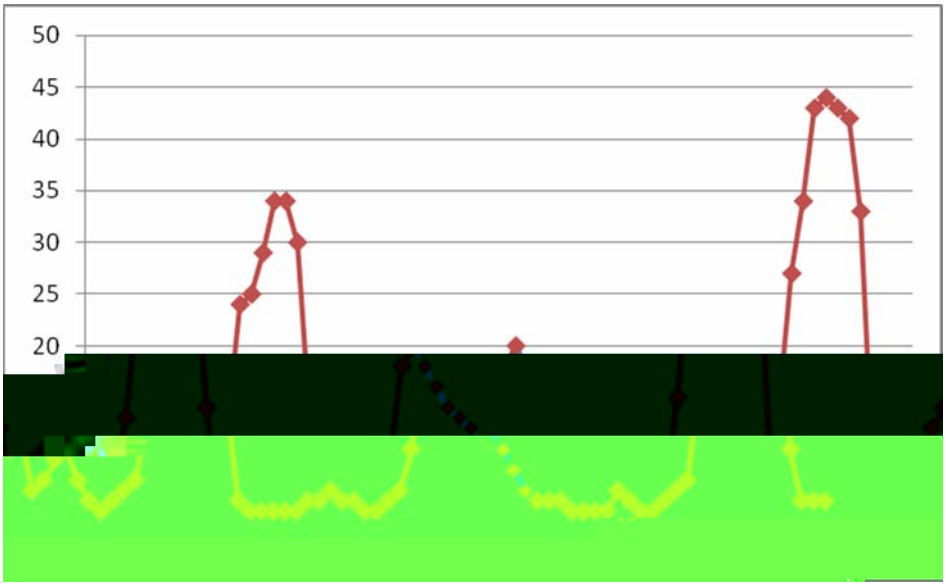

Figura 4.4 Gráfico da Série usando interpolação das linhas.

4.2.4 Preenchimento suavizado Simples *(Smooth Fill with a matriz 3x3 and 4 elements)*

Este algoritmo calcula o valor faltante através da média dos valores ao seu redor. Três variáveis devem ser satisfeitas:

(1) A dimensão da matriz

(2) O número de elementos da matriz

(3) Os pesos de cada elemento no cálculo da média

A matriz abaixo apresenta os quatro elementos que serão usados para calcular o valor faltante:

$$\begin{vmatrix} 0 & X_1 & 0 \\ X_2 & H & X_3 \\ 0 & X_4 & 0 \end{vmatrix}$$

Onde H é o valor a ser encontrado, X_i representa os quatros elementos a serem considerados no cálculo. Elementos não nulos têm o mesmo peso. Elementos nulos são descartados. Caso todos os elementos sejam nulos pode-se usar tanto o algoritmo *replacement* quanto a interpolação por colunas.

O valor de H é calculado usando somente valores não nulos:

$$H = \sum_{i=1}^{n} \frac{X_i}{n} \tag{4.4}$$

Usando o exemplo anterior:

1° Passo: Calcular o primeiro valor faltante encontrado (começando da esquerda para a direita)

24	25	29
20	X	x
27	34	43

A média seria (25+20+34)/3 = 26; Só foram utilizados 3 elementos porque um deles é valor nulo.

2° Passo: Calcular o segundo valor faltante encontrado

25	29	34
26	X	X
34	43	44

A média seria (29+26+43)/3 = 33; Da mesma forma do primeiro passo só foram utilizados 3 elementos porque um deles era valor nulo.

Fazendo essas interações até o final dos valores faltantes o resultado será:

10	7	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	26	33	37	38	37	28	14	8	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

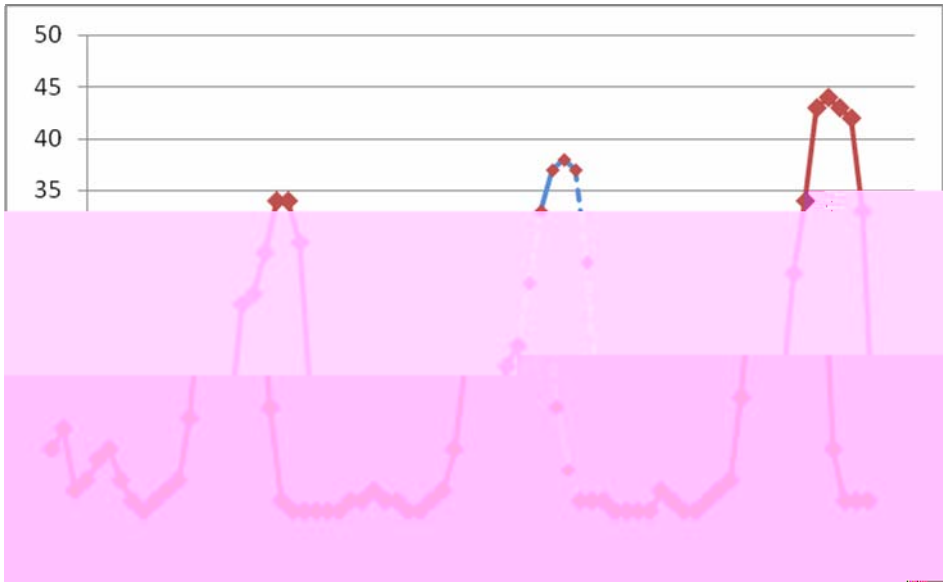

Figura 4.4 Gráfico da Série usando matriz 3x3 e 4 elementos da esquerda para a direita.

Pode-se usar este algoritmo iniciando o cálculo com o primeiro e o ultimo valor faltantes caminhando o cálculo para o centro.

Exemplo:

$$\begin{vmatrix} - & 25 & - \\ 20 & x & x \\ - & 34 & - \end{vmatrix} \quad \begin{vmatrix} - & 4 & - \\ X & X & 5 \\ - & 5 & - \end{vmatrix}$$

A média seria respectivamente (25+ 20+ 34)/3 = 26 e (4 + 5 + 5)/3 = 5;

$$\begin{vmatrix} - & 25 & - \\ 20 & 26 & x \\ - & 34 & - \end{vmatrix} \quad \begin{vmatrix} - & 4 & - \\ X & 5 & 5 \\ - & 5 & - \end{vmatrix}$$

Executando esse procedimento até o final o resultado seria:

$$\begin{vmatrix} 10 & 7 & 6 & 7 & 9 & 10 & 7 & 5 & 4 & 5 & 6 & 7 & 13 & 24 & 25 & 29 & 34 & 34 & 30 & 14 & 5 & 4 & 4 & 4 \\ 4 & 4 & 5 & 5 & 6 & 5 & 5 & 4 & 4 & 5 & 6 & 10 & 18 & 20 & 26 & 33 & 37 & 38 & 30 & 18 & 7 & 5 & 5 & 5 \\ 5 & 4 & 4 & 4 & 4 & 6 & 5 & 4 & 4 & 5 & 6 & 7 & 15 & 27 & 34 & 43 & 44 & 43 & 42 & 33 & 10 & 5 & 5 & 5 \end{vmatrix}$$

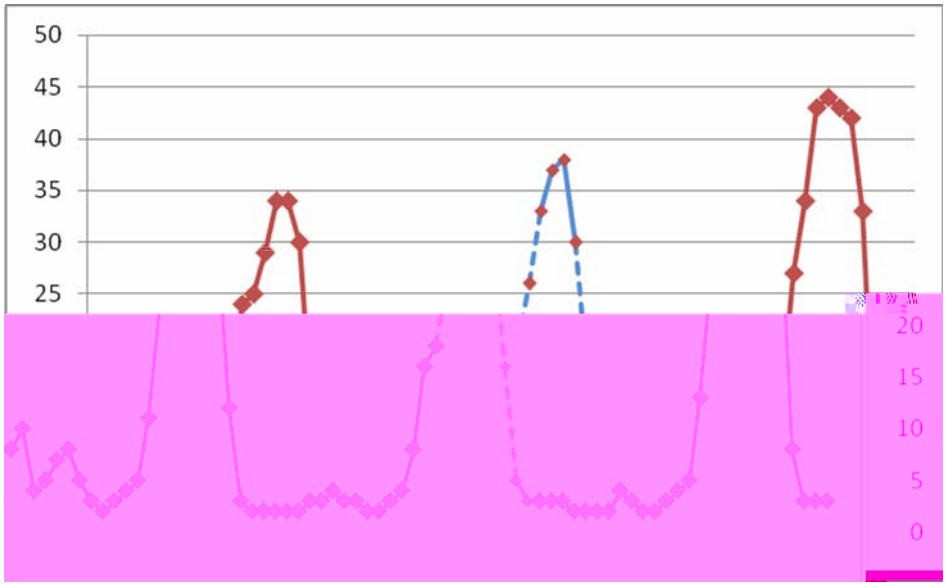

Figura 4.5 – Gráfico da Série usando matriz 3x3 e 4 elementos calculando o primeiro e último e deslocando para o centro.

4.2.5 Preenchimento suavizado Ponderado *(Smooth Fill with matrix 3x3 and 8 elements)*

Este algoritmo usa a média dos oito vizinhos mais próximos do valor faltante. A matriz abaixo mostra como serão usado os elementos.

$$\begin{vmatrix} B_1 & A_1 & B_2 \\ A_2 & H & A_3 \\ B_3 & A_4 & B_4 \end{vmatrix}$$

Os elementos A_i têm distancia 1 de H, enquanto os elementos B_i tem distância $\sqrt{2}$, com isso existe diferença de pesos entre os elementos para calcular a média. O valores nulos serão descartados e a são calculadas as média de V_1 e V_2 da seguinte forma:

$$V_1 = \sum_{i=1}^{n} \frac{A_i}{n} \tag{4.5}$$

$$V_2 = \sum_{j=1}^{n} \frac{B_j}{n} \tag{4.6}$$

O valor de H será calculado somando V_1 e V_2 com seus respectivos pesos:

$$V = V_1 * 0.7071 + V_2 * 0.2929$$

Exemplo:

1° Passo: Calcular o primeiro valor faltante encontrado (começando da esquerda para a direita)

24	25	29
20	X	x
27	34	43

$V_1 = (25+20+34)/3 = 26$ $V_2 = (24+29+27+43)/4 = 31$,

$V = 26 * 0.7071 + 31*02929 = 27$

O resultado seria:

24	25	29
20	27	x
27	34	43

2° Passo: Calcular o segundo valor faltante encontrado

25	29	34
27	X	x
34	43	44

$V_1 = (29+27+43)/3 = 33$ $V_2 = (25+34+44+34)/4 = 34$,

$V = 33 * 0.7071 + 34*02929 = 33$

Os pesos valores dos pesos foram retirados de [LATINI, PASSERINI, 2004].

O resultado seria:

25	29	34
27	33	x
34	43	44

Fazendo o procedimento até o ultimo dado faltante a matriz e o gráfico (Figura 4.6) ficariam:

10	7	6	7	9	10	7	5	4	5	6	7	13	24	25	29	34	34	30	14	5	4	4	4
4	4	5	5	6	5	5	4	4	5	6	10	18	20	27	33	37	38	35	26	14	7	5	5
5	4	4	4	4	6	5	4	4	5	6	7	15	27	34	43	44	43	42	33	10	5	5	5

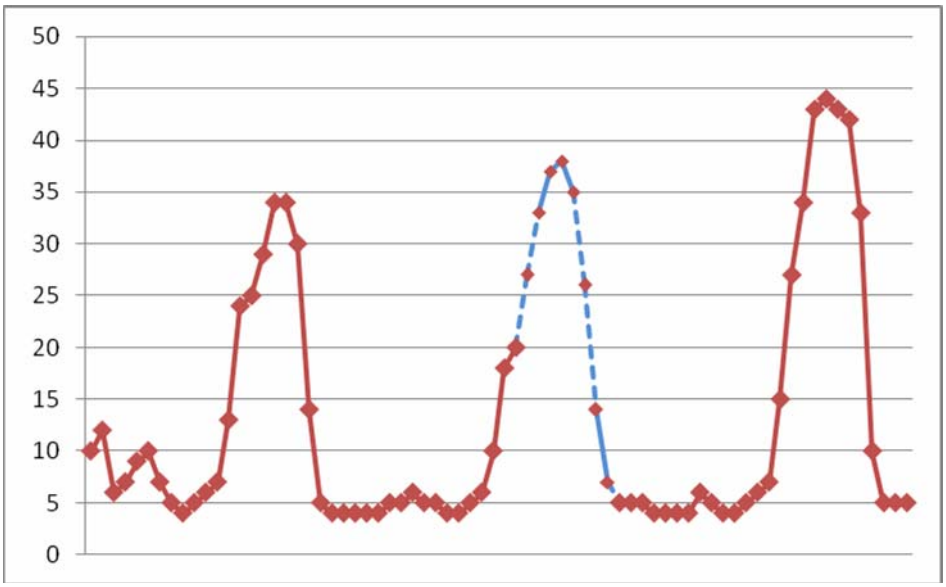

Figura 4.6 – Gráfico da Série usando matriz 3x3 e 8 elementos.

CAPÍTULO 5
FERRAMENTA PARA TRATAMENTO DE DADOS FALTANTES E METODOLOGIA PARA PREVISÃO DE RESÍDUOS USANDO PROGRAMAÇÃO GENÉTICA

O presente capítulo aborda a ferramenta computacional criada para substituição de dados faltantes mencionada no capítulo 4, bem como a metodologia criada para otimizar séries temporais utilizando previsão de resíduos gerado pelo modelo de *Winters*.

5.1 Aplicativo para Substituição de Dados Faltantes

Para executar o tratamento de dados faltantes, foi criada uma ferramenta amigável que permite ao usuário escolher o algoritmo que melhor complemente os dados faltantes na série temporal em estudo.

A aplicação foi desenvolvida na linguagem C# ASP.NET , a qual denominaram de *iPredition*, e interface para *web.*

Para utilizar o aplicativo, é mister que o usuário escolha a base de dados que possua o dado faltante, a qual que deve ser em formato – *txt*. Os valores faltantes serão interpretados pelo aplicativo através dos seguintes caracteres (-), (?) , ( ) - espaço.

Ao carregar a base de dados, o sistema automaticamente apresenta, em forma de gráfico, os dados da série e exibe, em tela, os resultados relativos à média, desvio padrão, mediana, assim como os pontos de valor máximo e mínimo. Os dados faltantes são marcados com um "x" em cor cinza no gráfico, conforme demonstrado na figura 5.1.

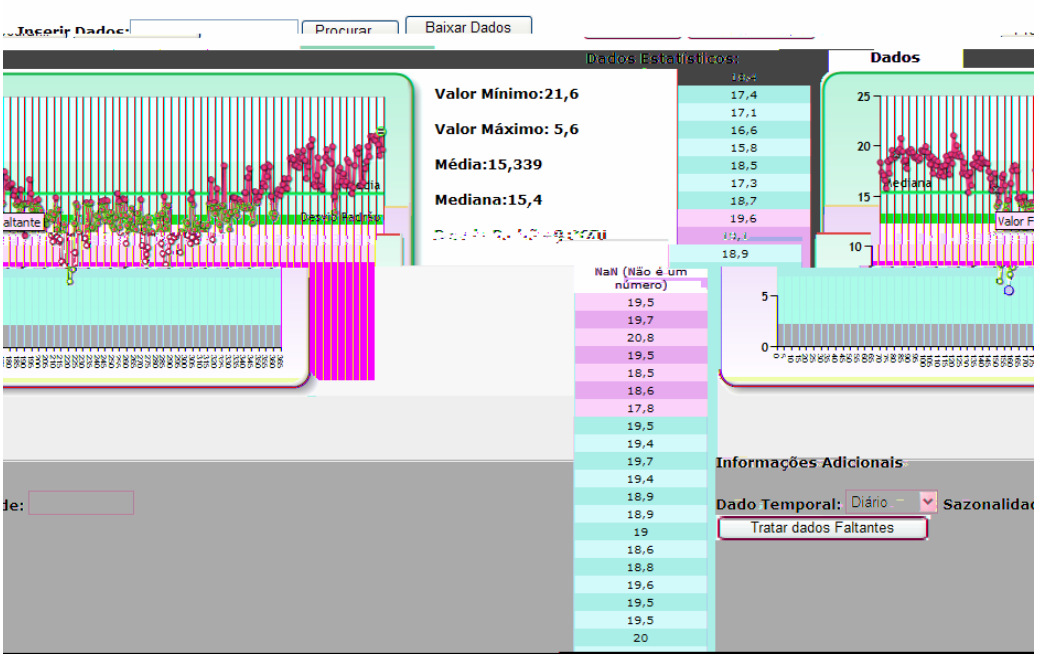

Figura 5.1 – Tela inicial mostrando dados da série Atmosfera

Para realizar a substituição dos dados faltantes por um dos algoritmos citados no capítulo 4, o usuário deve selecionar o período no qual a série foi coletada (diário,

semanal, mensal, trimestral, semestral ou anual). Ao clicar no botão “tratar dados faltantes”, são apresentados os gráficos e os dados estatísticos de todos os algoritmos propostos nessa dissertação; os dados que foram substituídos são mostrados através de um círculo vermelho. Veja Figura 5.2.

Após a análise, o usuário poderá selecionar o método mais apropriado e clicar no botão “Fazer Previsão”.

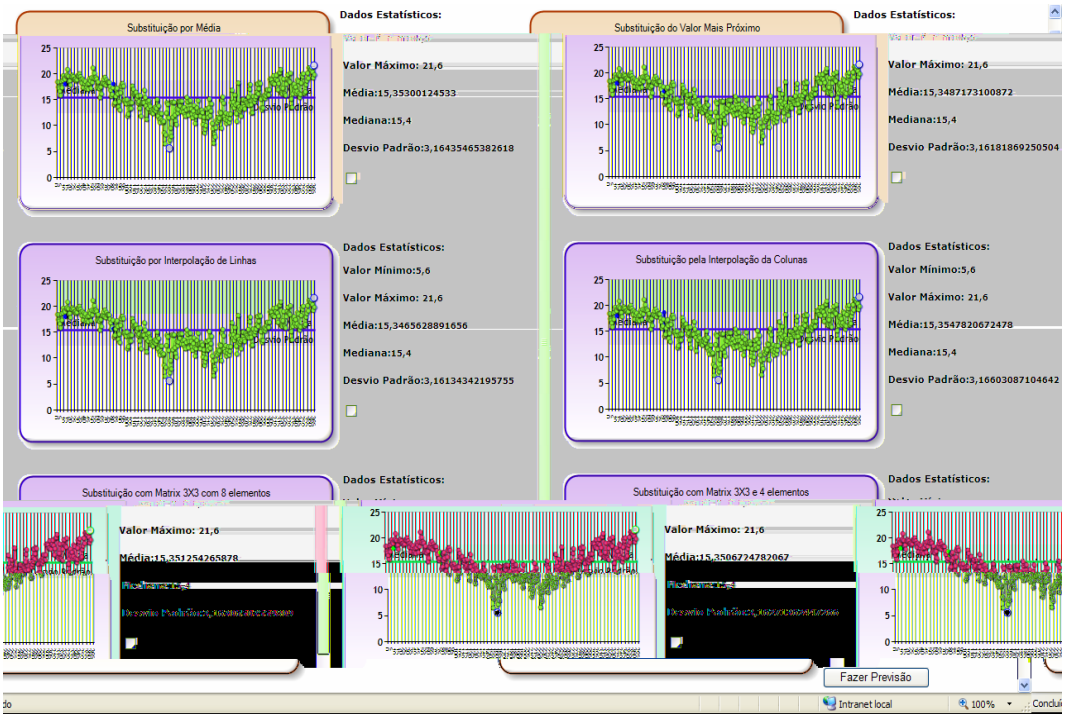

Figura 5.2 – Tela com os resultados dos algoritmos para tratar dados faltantes – Série Atmosfera

Na previsão utiliza-se o método de *Winters*, gerando os resultados conforme a figura 5.3.

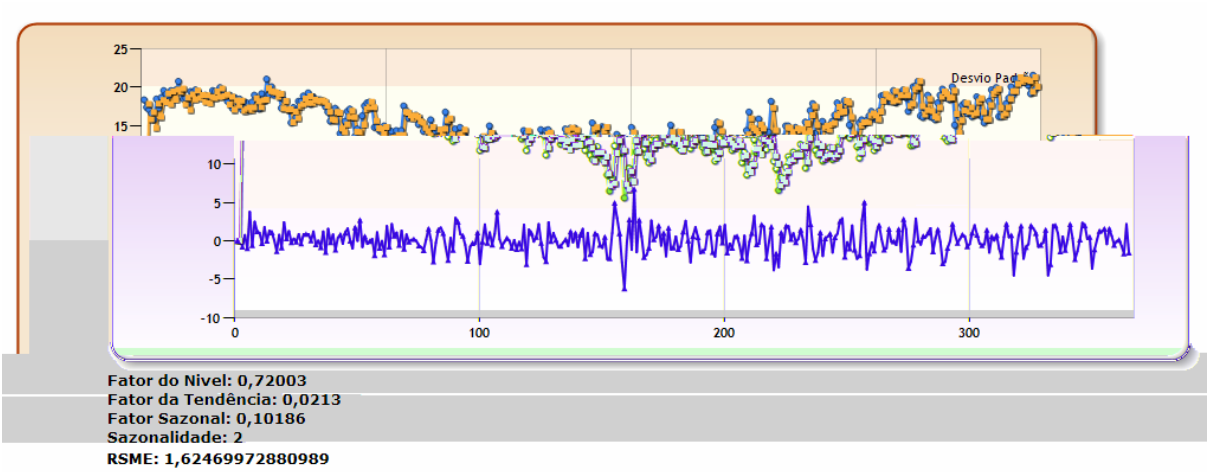

Figura 5.3 – Resultado da Previsão da Série Atmosfera

Os itens em amarelo da figura 5.3, representam os resultados encontrados pelo modelo *Winters*, os itens em vermelho os ruídos encontrados que servirão de base para otimização da série, através da programação genética e o item em azul representa a série original.

Os resultados dos fatores do modelo *Winters* encontrados, quais sejam, fator do nível, fator da Tendência, fator sazonal, sazonalidade e a Raiz do Erro Quadrático Médio (RSME) são apresentados abaixo do gráfico.

Para encontrar os melhores parâmetros do modelo de *Winters*, foi implementado um algoritmo genético que minimiza o erro quadrático médio (equação 2.20) mostrado no capítulo 1. Como descrito no mesmo capítulo, para calcular o modelo de *Winters* é mister conhecer, antecipadamente, a sazonalidade da série; entretanto, para essa implementação não é necessário, vez que o algoritmo genético também identifica o valor da sazonalidade da série.

Os parâmetros do algoritmo genético usados para a otimização do modelo *Winters* foram:

- Tamanho da População: 500;
- Tipo de Seleção: Roleta;
- Taxa de *Crossover*: 70%;

- Taxa de Mutação: 10%
- *Fitness*: RSME;

Importante ressaltar, que para a previsão da série temporal deve-se ter um mínimo de pontos, já que no caso da sazonalidade igual a 12, é preciso ter os doze pontos anteriores para se conhecer o próximo valor. Esse fator é crítico para séries com poucos dados.

A interpretação dos resultados e a otimização dos resíduos de *Winters,* utilizando programação genética, serão descritos no capítulo a seguir.

5.2 Metodologia para Otimização de Resíduos usando Programação Genética

Essa metodologia é inspirada em problemas de regressão, cujo objetivo é encontrar uma aproximação para determinados valores de uma variável contínua, onde dada uma função *f(x)*, num certo intervalo, deve-se encontrar uma função *g(x)*, tal que $f(x_i) = g(x_i) \forall x_i \in X$, onde *X* é um conjunto de valores do intervalo considerado. Para tal, deve-se aplicar um algoritmo de aprendizagem, com base no conjunto de pontos conhecidos, para encontrar uma função *g* próxima da função *f* no domínio desejado. Uma forma de constatar esta proximidade é verificar o menor erro quadrático médio.

Após a previsão com o modelo de *Winters,* os resíduos encontrados serão tratados pela programação genética, com a finalidade de se obter o melhor comportamento dos mesmos e chegar a um resultado final que minimize, ainda mais, o erro global de previsão.

Tanto o modelo de *Winters*, quanto o modelo dos resíduos gerado pela Programação Genética, que melhor se ajuste a série, serão obtidos através da minimização da raiz erro quadrático médio.

Por fim, os resultados são somados e comparados com a série original para identificar o real erro da série.

A ferramenta JGAP (JAVA GENETIC ALGORITHMS PACKAGE) versão 3.0, disponível no endereço http://jgap.sourceforge.net/, foi utilizada para realizar a previsão dos resíduos através da Programação Genética.

O algoritmo utiliza como terminal, o último valor da série temporal para calcular o próximo (T_i), já que a proposta do modelo de *Winters* é decompor a série nos três componentes principais, e também até quatro constantes reais entre 0 a 100. A opção por até quatro, deve-se a realização de sucessivas tentativas, onde se constatou que são suficientes para a obtenção de valores satisfatórios.

As funções utilizadas no conjunto *F* são: sen, cos, raiz, exponencial, multiplicação, divisão, soma, subtração e um número elevado a outro. A utilização dessas funções permite que a Programação Genética se ajuste aos dados com modelos não lineares. Estes modelos, em geral, se ajustam melhor aos dados, pois normalmente as séries analisadas não são oriundas de modelos lineares. O tamanho da população, o número de gerações e as taxa de reprodução, mutação e cruzamento, a função de aptidão são descritos na lista abaixo.

- T = $\{T_i, C_1, C_2, C_3, C_4\}$
- F = {+ , - , * , / , sem, cos, raiz, pow, exp}
- População = 2000;
- Gerações: 500;
- Taxa de Cruzamento: 90 %
- Taxa de Reprodução: 10%;
- Função de Aptidão: *RMSE*
- Método de Inicialização da População Inicial: *Grow*

O tópico a seguir aponta um exemplo aplicando a metodologia criada.

5.3 Exemplo Prático

A série usada para demonstrar a metodologia foi retirada de [MORETTIN & TOLOI, 2004], e representa as temperaturas em graus centígrado. Os dados foram coletados todos os dias, às 12:00h, na cidade de São Paulo, durante o período de primeiro de janeiro a trinta e um de dezembro de 1997. Esses dados podem ser obtidos em http://www.ime.usp.br/~pam/ST.html ou no apêndice A.

Os dados originais podem ser visualizados na figura 5.1, já os resultados e os resíduos na figura 5.4. Após a identificação dos parâmetros do modelo de *Winters*, os resíduos são utilizados pela Programação Genética, para se encontrar a melhor solução.

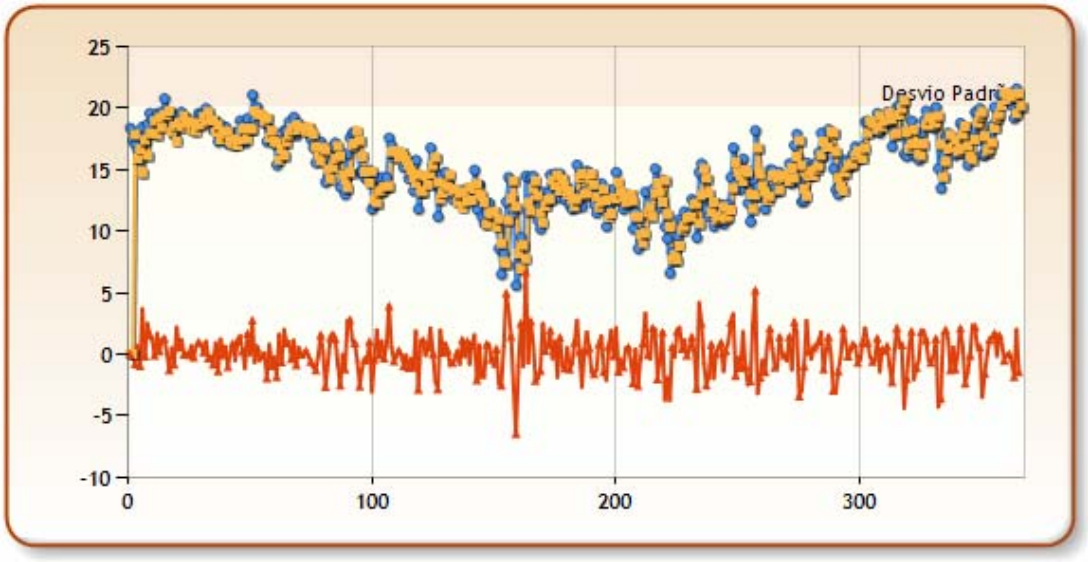

Figura 5.4 – Valores Previstos pelo Modelo de *Winters* e o resíduos gerados da Série Atmosfera

Os parâmetros que melhor se ajustaram para o modelo foram: Tendência: 0.60708, Nível: 0.02981, Fator Sazonalidade: 0.11331. A sazonalidade encontrada na série foi o valor 2.

Ao utilizar a Programação Genética para calcular os resíduos o melhor resultado encontrado é mostrado a seguir:

Best solution fitness: 1.5753950012917008
Best solution: cosine ((sine (43.41551 - 15.73465)) + (((Exp(X)) ^ (cosine X)) / (Exp((cosine 43.41551)))))
Depth of chromosome: 5

Os resultados mostram o melhor *fitness*, a árvore gerada e a profundidade da árvore. A figura 5.5, mostra a diferença do valor previsto para os dados reais (resíduos).

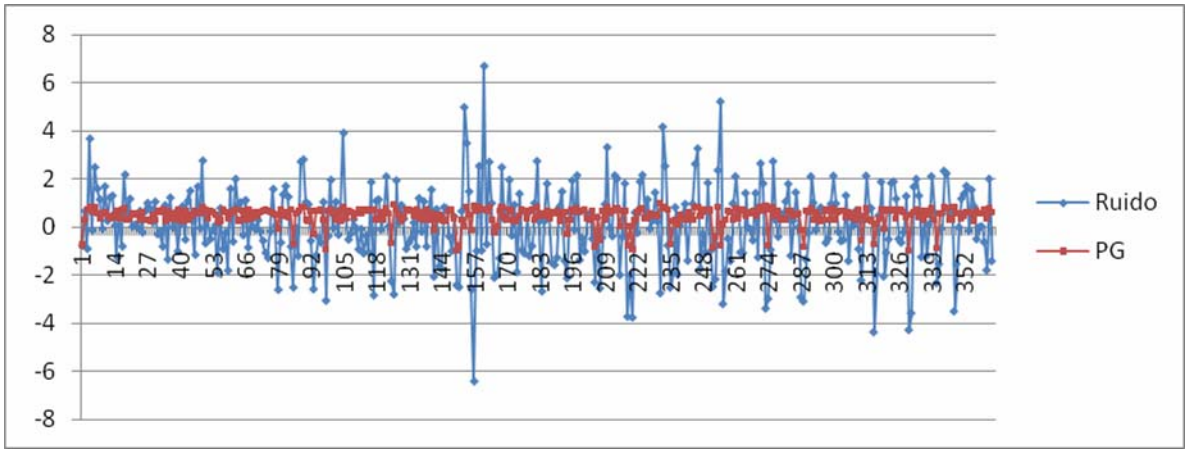

Gráfico 5.5 – Resíduos x Valor Previsto pela PG

Nota-se que a Programação Genética não conseguiu identificar os picos dos resíduos, mas o comportamento foi identificado, já que verificando a soma dos resíduos com o valor previsto de *Winters* a raiz do erro quadrático médio diminuiu para RSME igual a 1.56010188. A figura 5.6 mostra os resultados consolidados.

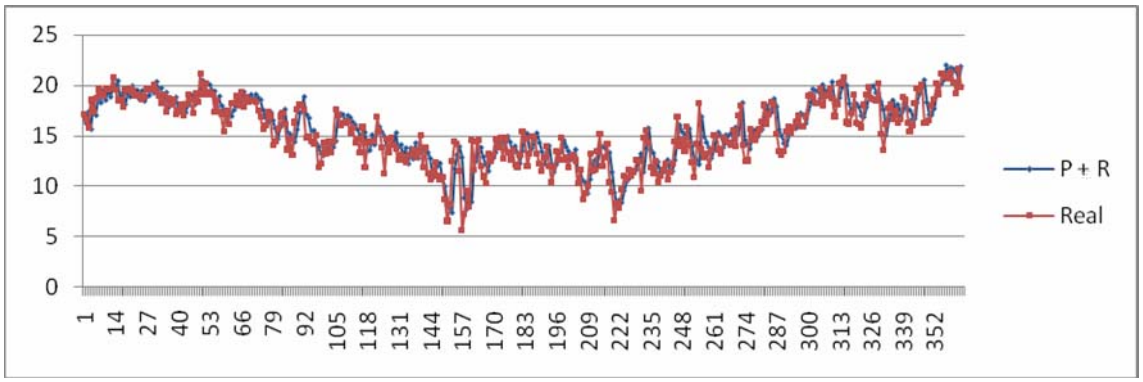

Gráfico 5.6 – Série Real x Previsão do modelo de *Winters* + Programação Genética

Os resultados encontrados foram comparados com outros modelos, apresentados no capítulo a seguir.

CAPÍTULO 6 ESTUDO COMPARATIVO

No presente capítulo, há um estudo comparativo da previsão usada neste trabalho com os métodos tradicionais feitos com metodologia de ARMA, com a programação Genética Original, e a metodologia criada por Souza [SOUZA, 2006] baseado na metodologia de *Boosting* usando programação genética.

6.1 Experimentos

A metodologia de Souza, ou previsão dos erros (BCIGP), citada no item anterior, é baseada na idéia de sucessivas previsões de uma determinada série, sendo melhor descrita a seguir.

Luzia Vidal, [SOUZA,2005], realiza uma execução com a programação genética para se encontrar um preditor para a série temporal, o resíduo é então calculado e obtém-se:

$$Z_t = \hat{Z}_t + \varepsilon_1 \qquad \text{Eq(6.1)}$$

Os resíduos ε_1 formam então, o novo conjunto de dados que serão utilizados na próxima execução da programação genética. O problema agora se resume a minimizar o erro de previsão, ou seja, determinar a melhor previsão para o resíduo ε_T, de forma que:

$$\varepsilon_1 = Z_t - \hat{Z}_t \qquad \text{Eq (6.2)}$$

Para minimizar o erro de previsão da série, é então realizada a previsão do resíduo ε_1, ou seja:

$$\varepsilon_1 = \hat{\varepsilon}_1 + \varepsilon_2 \qquad \text{Eq (6.3)}$$

ou seja:

$$\varepsilon_2 = \varepsilon_1 - \hat{\varepsilon}_1 \qquad \text{Eq (6.4)}$$

O resíduo ε_2 é previsto pela programação genética. Este procedimento repete-se até que um critério de parada tenha sido atingido. Foram realizadas dez predições para o resíduo, sendo selecionado como melhor preditor, aquele que apresentou o menor RMSE.

A solução final da previsão do resíduo será então dada por:

$$\varepsilon_1 = \hat{\varepsilon}_1 + \hat{\varepsilon}_2 + \hat{\varepsilon}_3 + ... + \hat{\varepsilon}_n \qquad \text{Eq (6.5)}$$

E a nova previsão da série será dada por:

$$Z_t = \hat{Z}_t + \varepsilon_1 \qquad \text{Eq (6.6)}$$

As séries estudadas serão divididas em um conjunto de treinamento de 90% e um conjunto de teste de 10% dos dados.

6.2 Experimento I – Base da Atmosfera

Com relação à base em estudo, já descrita no capítulo anterior, são apresentados apenas os resultados relacionados ao conjunto de teste.

A base de dados completa possui 365 registros, sendo que 36 destes, correspondente a 10%, foram usados como teste.

A figura 6.1 apresenta o resultado comparativo dos valores real com o previsto, usando a metodologia apresentada neste trabalho.

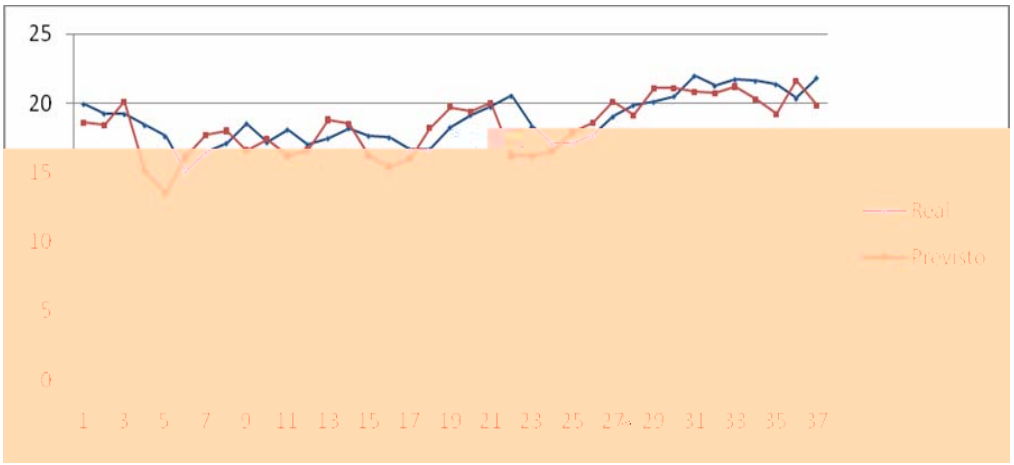

Figura 6.1 – Valores da Série Real com a Série Prevista

O RSME encontrado foi 1.64415124, que comparado com os outros modelos gerou um resultado mais satisfatório. O modelo ARMA encontrou 6.244017, a programação genética original encontrou 5.938722 e a previsão dos erros encontrou 2.467783.

6.3 Experimento II – Base de Dados de Bebida

Esta base de dados, retirada de [MORETTIN & TOLOI, 2004], corresponde à produção física industrial (PIB). O conjunto de dados foi coletado mensalmente no período de janeiro de 1985 a julho de 2000. O conjunto de dados total corresponde a 187 dados.

O primeiro passo é utilizar a ferramenta *iPredition* para conhecer o comportamento da série e realizar a previsão com o modelo de *Winters*. A figura 6.2 aponta os dados da série e os dados estatísticos.

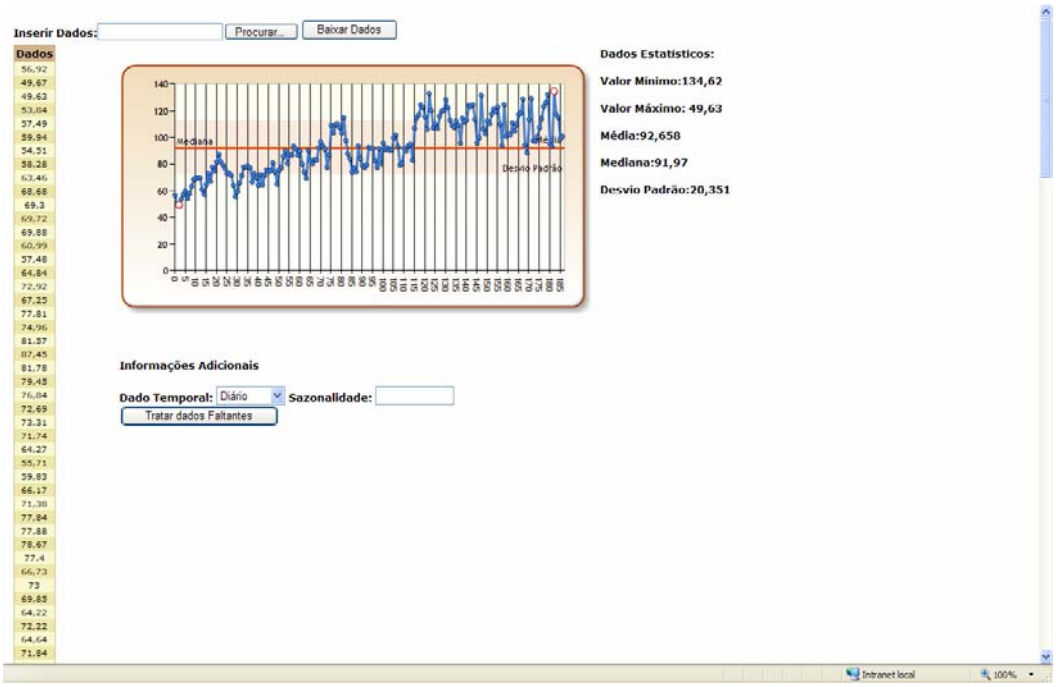

Figura 6.2 – Visualização dos dados originais de Bebidas

O modelo de *Winters* gerou os seguintes resultados: Tendência (T): 0.31612, Nivel (N) 0.01845 e Fator Sazonalidade (0.30925), sazonalidade da série 12, igual à anual.

A figura 6.3 mostra o comportamento da previsão e os resíduos gerados.

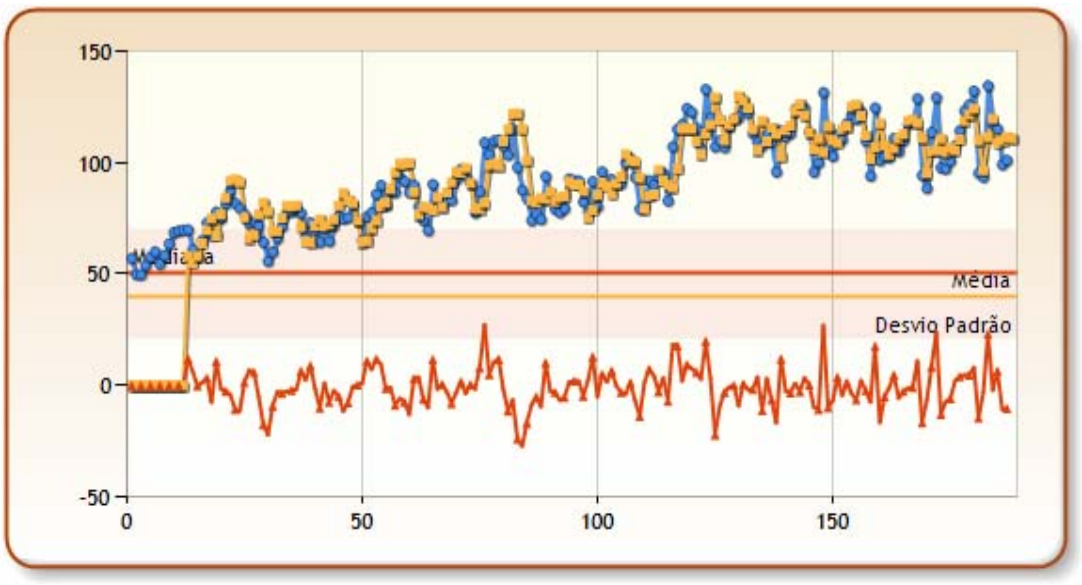

Figura 6.3 – Previsão do Modelo de *Winters* e os resíduos da Série Bebidas

A seguir, são tratados os resíduos com a Programação Genética, tendo como resultados:

```
Best solution fitness: 8.659004241896266
 Best solution: X * (sine ((52.791996 - (45.05967 - 47.79933)) / (X +
45.05967)))
 Depth of chromosome: 5
```

A figura 6.4 mostra que a Programação Genética identifica o comportamento de todos o conjunto de dados. Nota-se, que somente os grandes picos da série não foram identificados.

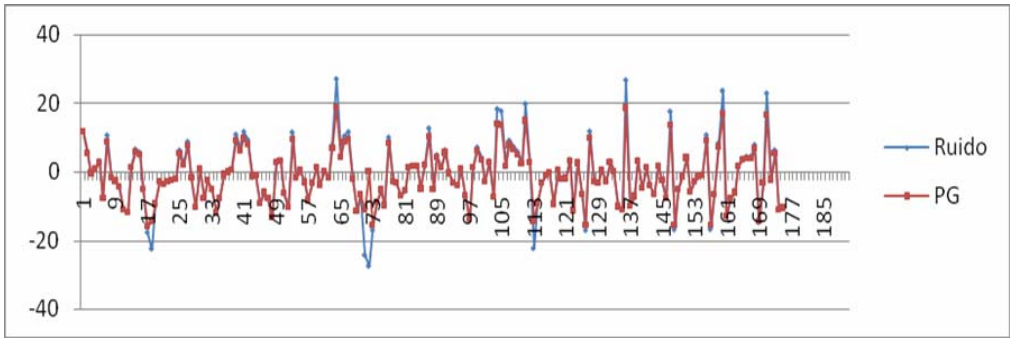

Figura 6.4 Resíduos da Série x Previsão com Programação Genética

Somando a previsão dos resíduos com o modelo de *Winters*, o resultado obtido é conhecido na figura 6.5. Nota-se que o comportamento foi identificado, no qual a previsão, praticamente, se sobrepôs a série real.

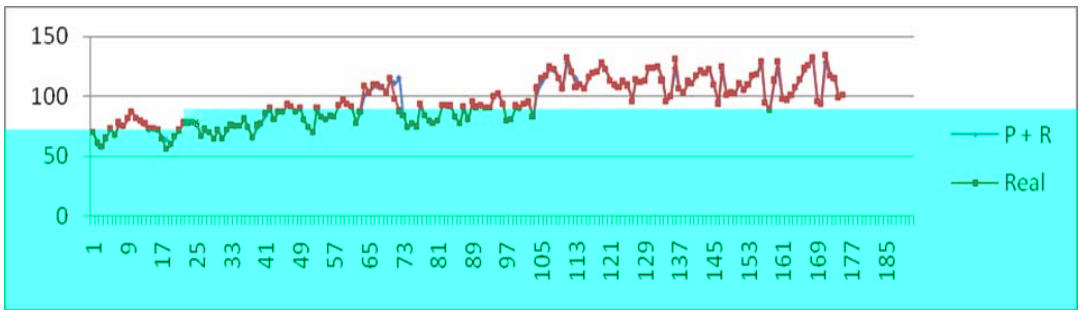

Figura 6.5 – Comparação dos Valores Reais com a Previsão de *Winters* e PG dos Resíduos

Verificando o conjunto de Testes, o modelo de *Winters* somado a Programação Genética dos Resíduos gerou o RMSE de 2.24590241. Este resultado mostra que a Programação Genética propiciou uma melhora significativa do modelo de *Winters*, fato

que demonstra sua eficiência em relação aos demais modelos. A Programação Genética Tradicional gerou o RMSE 14.7947, a metodologia de BCIGP gerou 6.613493 e o ARMA gerou 29.747780. A figura 6.6 mostra a previsão gerada pela metodologia e os dados reais do conjunto de testes.

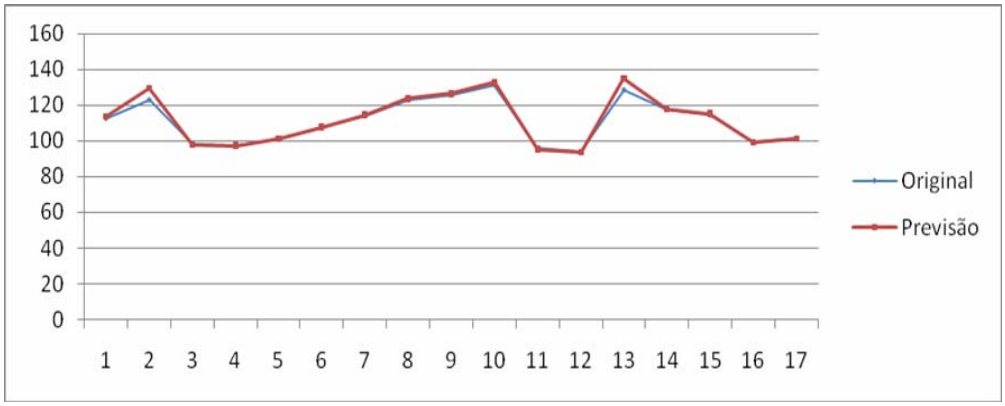

Figura 6.6 Comparação dos dados da Previsão com o Original

6.4 Experimento III – Base de Dados de Consumo

A base de dados de Consumo, corresponde às vendas físicas na região metropolitana de São Paulo. A coleta dos dados foi realizada no período de janeiro de 1984 a outubro de 1996, periodicidade mensal.

A figura 6.7 mostra os dados reais da série.

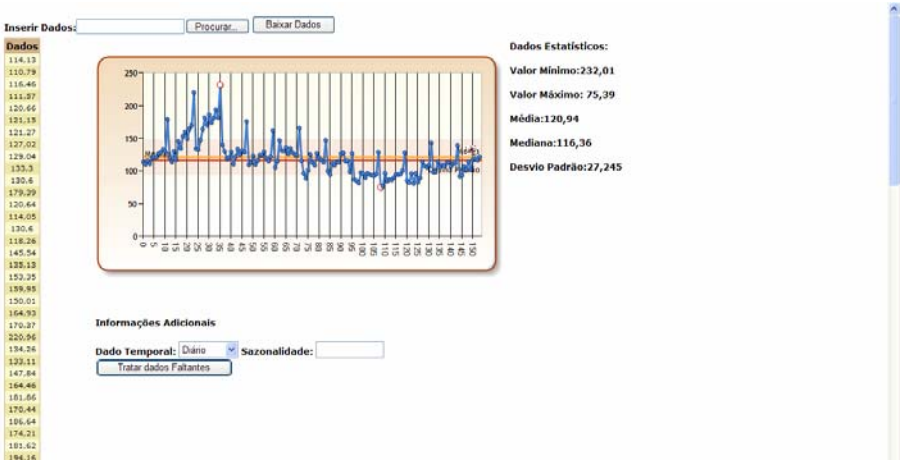

Figura 6.7 – Dados Originais da Série de Consumo

Uma vez conhecido o comportamento da série e seus dados estatísticos, inicia-se o uso da metodologia realizando a previsão da série através do modelo de *Winters*. Os resultados foram T: 0.74344; N: 0.03087; S: 0.34368; a sazonalidade é de 12, periodicidade anual. A previsão pode ser observada na figura 6.8.

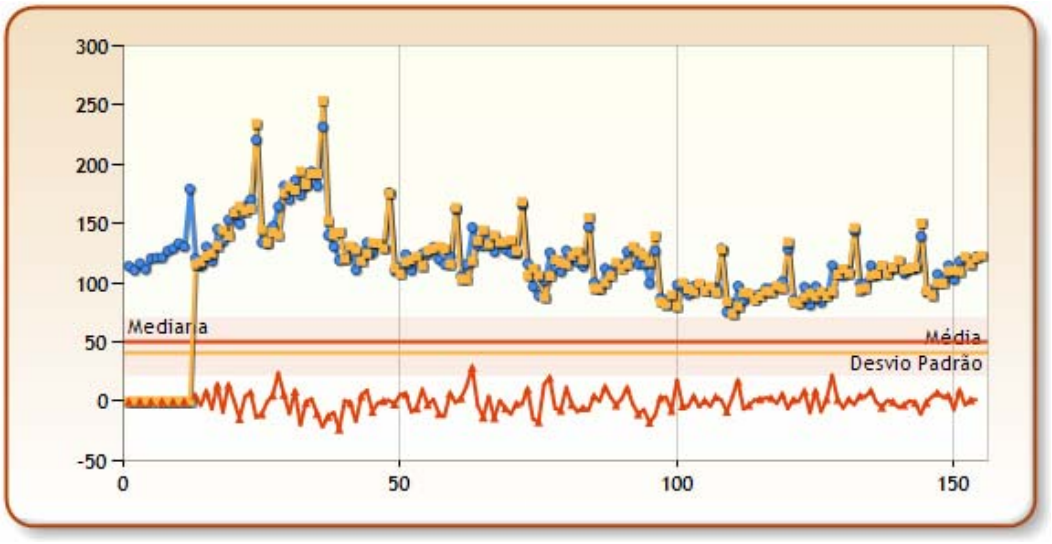

Figura 7.8 – Resultado da previsão de *Winters* e os resíduos gerados

Usando o tratamento dos resíduos com a Programação Genética foram encontrados os seguintes resultados:

```
Best solution fitness: 8.768588874113423
 Best solution: (((cosine 15.571521) + (cosine 15.571521)) * (Exp((cosine 69.25403))))
/ ((sine (sine X)) - (46.579754 - X))
 Depth of chromosome: 4
```

A figura 6.9 mostra o comportamento da série, somando os resultados da Programação Genética com o modelo de *Winters* para o conjunto total da série.

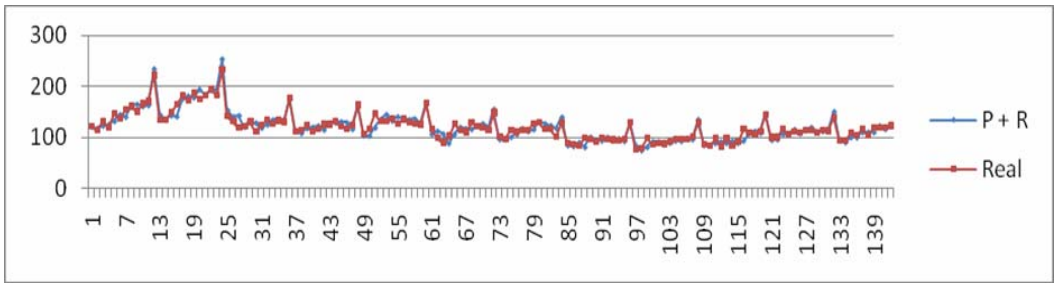

Figura 6.9 - Dados da Série de Consumo com a soma da PG de Resíduos e *Winters*

Utilizando o conjunto de testes, nota-se que usando a Programação os resultados de melhoria não são significati

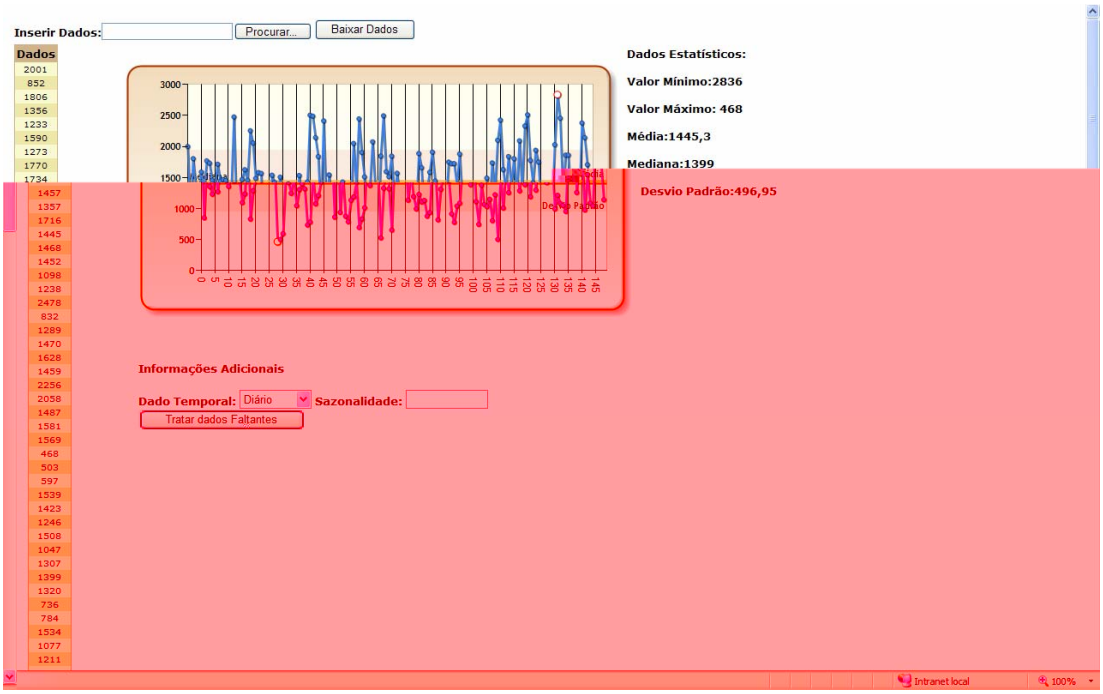

Figura 6.11 – Dados Originais da série de Fortaleza

Ao identificar a série, evidencia-se que ela não apresenta comportamento claro. Há várias oscilações dentro da série.

A Figura 6.12 confirma a assertiva, demonstrando que os resultados encontrados não são satisfatórios. Resta claro que o modelo não consegue identificar o comportamento da série. Os resultados encontrados foram: T: 0.00275; N: 0.80738; S: 0.11881; sazonalidade 12.

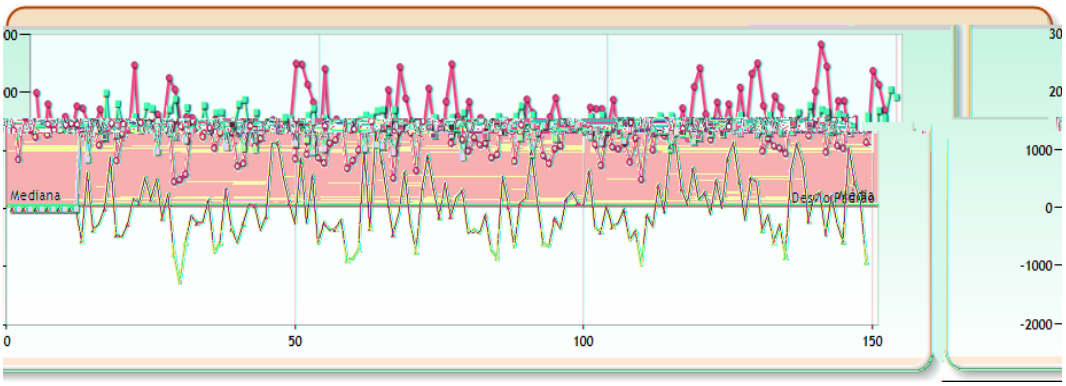

Figura 6.12 Resultado do modelo de *Winters* e resíduos da Serie Fortaleza

Ao aplicar a Programação Genética, esta também não conseguiu identificar o comportamento, bem como não conseguiu aproximar a função. A figura 6.13 mostra o exemplo:

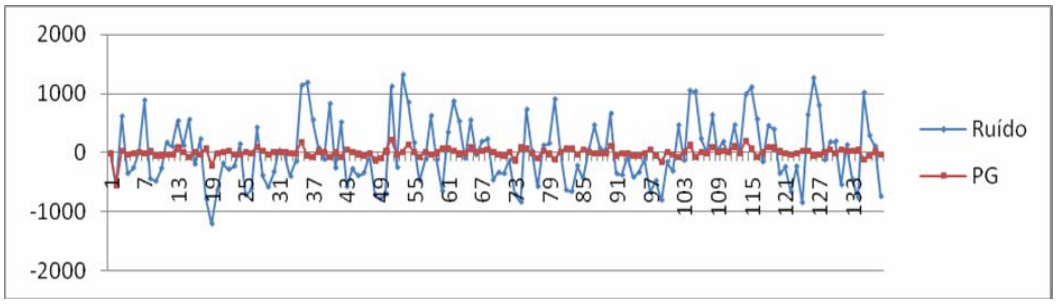

Figura 6.13 – Comparação dos Conjunto de Resíduos com a Previsão PG da Série Fortaleza

Os resultados encontrados foram:

```
Best solution fitness: 506.95257343010667
 Best solution: ((sine (66.09884 ^ 60.10205)) - ((60.10205 * 14.529264) / (X ^
81.10965))) * (((70.958206 + X) * (sine 81.10965)) * ((14.529264 / 66.09884) - (cosine
X)))
 Depth of chromosome: 4
```

Ao utilizar o conjunto de testes para comparação, pode-se constatar que a soma da previsão de *Winters* com a previsão de resíduos, não apresenta melhora, ou seja, o modelo de *Winters* encontrou o RSME de 627,971959 , já com a soma da previsão de resíduos o RSME subiu para 659,405504. Se comparado com os outros modelos, o resultado também não é satisfatório, eis que a PG tradicional informou o RSME de 592,9839, o modelo de *Boosting* gerou o RSME de 427,1147, e o ARMA 667,6005. Através desses resultados, nota-se que a série não possui um comportamento de fácil identificação.

A figura 6.14 mostra os resultados do modelo de *Winters* somado à previsão com PG.

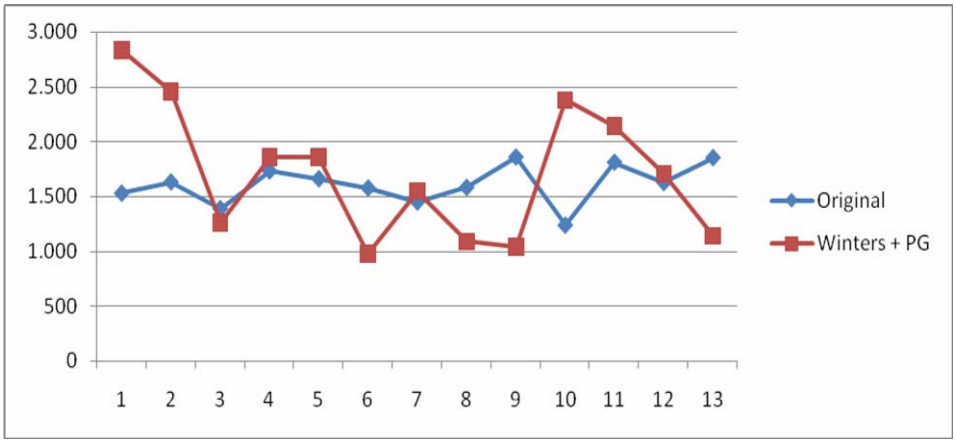

Figura 6.15 – Conjunto de Teste da Série Original e Valores Previsto

6.6 Experimento V – Conjunto de Dados IPI

A série de IPI representa a coleta dos dados no período de janeiro de 1985 a julho de 2000. A periodicidade é mensal.

A figura 6.16 mostra os dados originais da série e seus dados estatísticos.

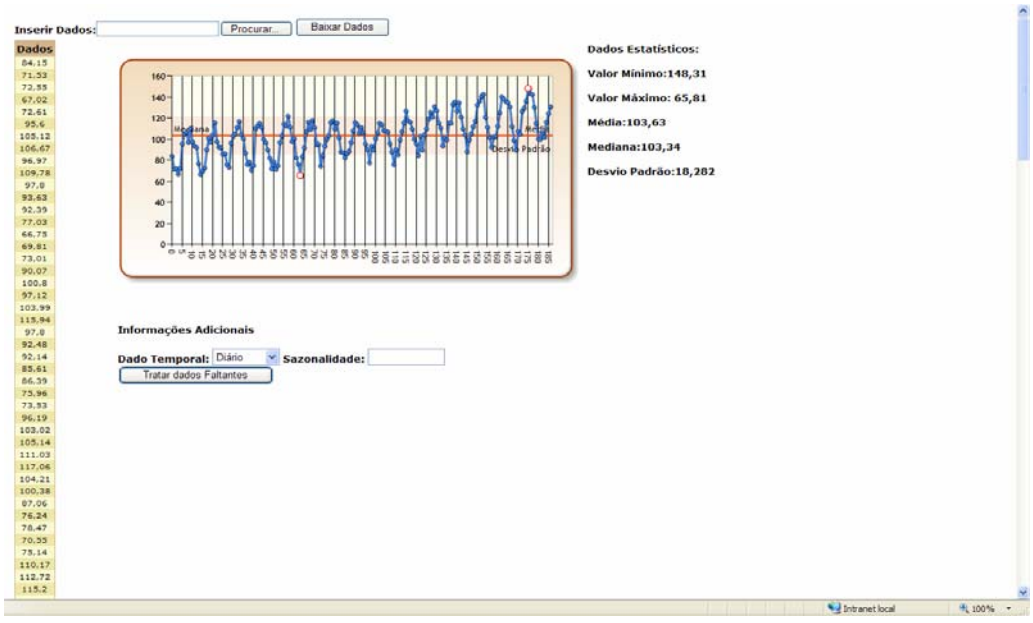

Figura 6.17 – Série Original do conjunto de dados IPI

Utilizando o modelo de *Winters* para previsão inicial os fatores encontrados foram: Fator Tendência de 0.18932; Nível de 0.0165; fator Sazonalidade de 0.40766 e a

sazonalidade é de 12. A figura 6.18 mostra que o modelo de *Winters* logrou identificar o comportamento.

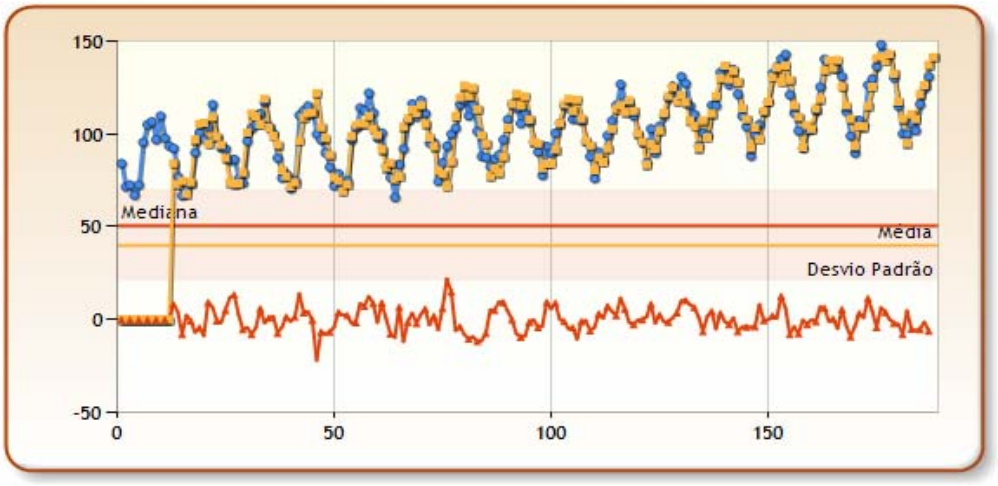

Figura 6.18 – Modelo de *Winters* e resíduos gerados na Serie IPI

Na previsão realizada com Programação Genética atenta-se, por meio da figura 6.19, que a metodologia consegue identificar o comportamento do Ruído. Os resultados foram:

```
Best solution fitness: 5.880291180314984
 Best solution: 30.34105 * (X / 47.11195)
 Depth of chromosome: 2
```

Os resultados também revelam que a árvore gerada tem apenas dois nós.

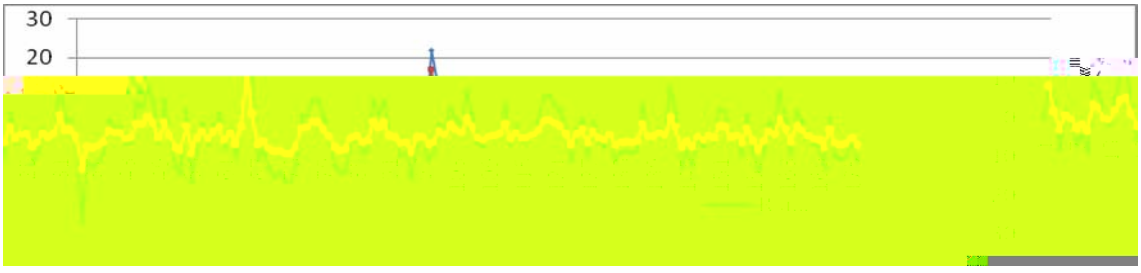

Figura 6.19 – Comparação dos Resíduos com a Previsão da Programação Genética da Série IPI

Tendo a PG identificado o comportamento, na previsão do conjunto de testes, houve também, a melhora do resultado final do modelo de *Winters*, onde o RMSE passou de 5.314434 para 3.237822. Comparado com outras metodologias, o resultado

desse trabalho evidenciou-se mais eficiente, já que a PG tradicional conseguiu 9.875241, o ARMA 20.4493 e o metodologia de *Boosting* conseguiu 3.5541.

A figura 6.29 mostra o comportamento da série usada no conjunto de testes.

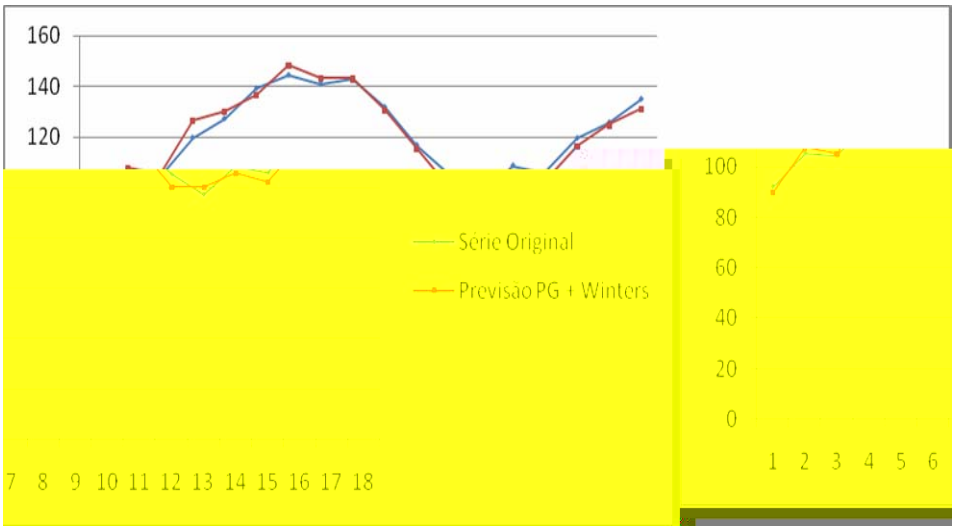

Figura 6.29 – Previsão do conjunto de Testes da Série IPI

6.7 Experimento VI – Série de Manchas

A série de dados de Manchas, corresponde ao número de manchas solares de Wolfer, tendo sido coletada no período de 1724 a 1924, os dados são anuais. A figura 6.21 aponta o comportamento da série de manchas.

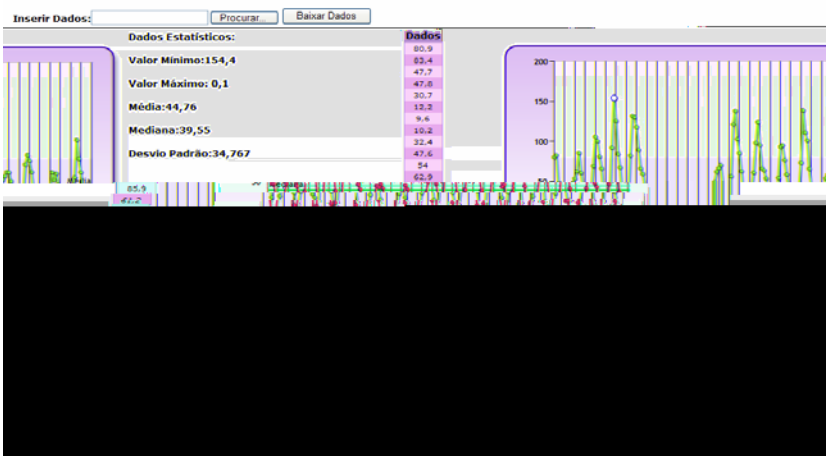

Figura 6.21 – Dados Originais da Série de Manchas

Utilizando a previsão inicial com o modelo de *Winters* os resultados foram: T: 0.00259, o N: 0.8464 e S: 1.25585. Sazonalidade igual a 1. A figura 6.22 revela os dados encontrados pelo modelo de *Winters* e os resíduos que serão avaliados pela Programação Genética.

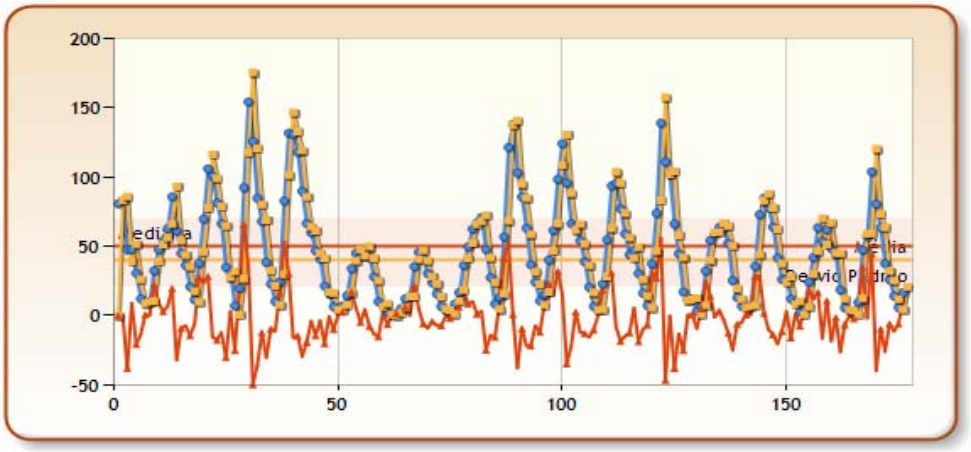

Figura 6.22 – Previsão de *Winters* + resíduos da Série de Mancha.

Extrai-se, ainda, da figura 6.22 que no momento da realização da previsão, os picos foram gerados um passo a frente. Referido comportamento deve-se à sazonalidade encontrada, com valor igual a 1.

O resultado obtido da previsão de resíduos com Programação Genética, compreende o exposto na lista abaixo:

```
Best solution fitness: 18.40521177717897
 Best solution: (Exp((sine (Exp(61.56456))))) * ((63.33647 * (sine (61.56456 ^
61.951836))) * (((cosine 63.33647) ^ (61.951836 + 61.56456)) + (cosine (49.46995 ^
X))))
 Depth of chromosome: 5
```

A figura 6.23 demonstra que, apesar de identificado o comportamento, não foram previstos os picos.

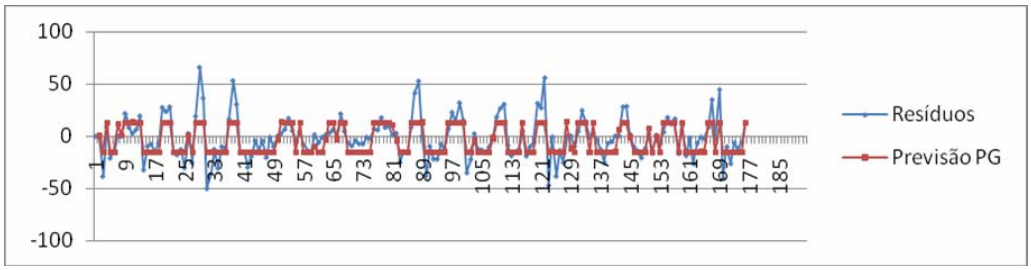

Figura 6.23 – Comparação dos resíduos com Previsão usando PG da Série de Manchas

O objetivo final consiste em somar a previsão de *Winters* com a previsão dos Resíduos por PG no conjunto de teste e verificar o quanto a previsão dos resíduos ajuda na melhoria do modelo. Para o caso da série de manchas, a união dos dois modelos gerou um resultado satisfatório, já que no *Winters* o RMSE foi de 20.4367 e com a união caiu para 14.2529. O modelo de ARMA atingiu 29.77, o modelo de *Boosting* 14.571732 e o modelo de PG Tradicional 14,697371, revelando que a metodologia dessa dissertação também foi melhor.

A figura 6.24 apresenta os resultados encontrados no conjunto de teste.

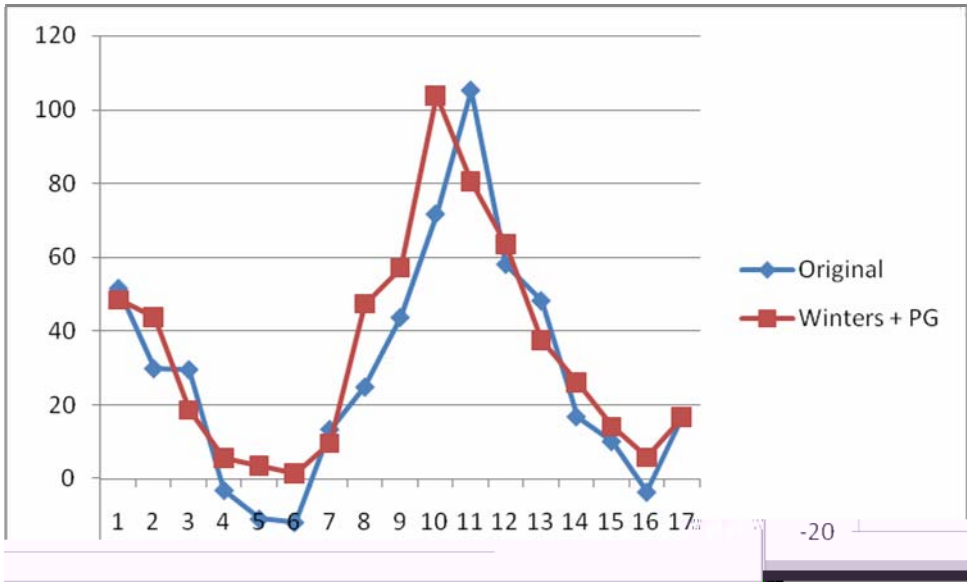

Figura 6.24 – Comparação entre a série original e o modelo proposto do conjunto de teste

6.8 Análise dos Resultados

Os experimentos expostos anteriormente, comprovam que a metodologia proposta atingiu seu objetivo já que comprando os seis experimentos apenas um não superou os outros modelos. A tabela 6.1 exibe o comparativo entre os modelos.

Tabela 6.1 – Comparação do RMSE no conjunto de Testes para cada método

Série	***Winters* + PG**	**PG Tradicional**	**BCIGP**	**ARMA**
Atmosfera	**1,64415124**	5,938722	2,4677	6,244
Bebida	**2,24590241**	14,794	6,613493	29,7477
Consumo	**5,2246**	10,3239	7,8964	11,7583
Fortaleza	659,405504	592,9839	**427,1147**	667,6005
IPI	**3,237822**	9,8752	3,5541	20,4493
Manchas	**14,25298488**	14,6973	14,5717	29,7763

CAPÍTULO 7 CONCLUSÃO

As Séries Temporais foram o ponto principal deste trabalho.

Destacou-se a importância do tratamento de dados faltantes e de uma metodologia para previsão dos resíduos gerados pelo modelo de *Winters,* utilizando Programação Genética.

Para explorar o tratamento de dados faltantes, estudou-se a técnica dos vizinhos mais próximos no tempo, razão pela qual, criou-se uma ferramenta onde todos os algoritmos abordados foram implementados.

Nesta mesma ferramenta foi implementado o modelo de *Winters* e um algoritmo genético que otimizasse os parâmetros desse modelo.

Como a visualização dos dados é de grande importância para o tomador de decisão, esta ferramenta mostra, graficamente, a série original, destacando os dados faltantes. Ao executar a substituição exibe um comparativo de todos os algoritmos, para que seja tomada a decisão mais correta. Ao gerar o modelo de *Winters,* mostra-se um comparativo da série original com o modelo previsto.

O modelo de *Winters*, foi escolhido por basear-se na idéia de que observações passadas contêm informações sobre o padrão de comportamento da série temporal. O propósito desse método é distinguir o padrão de qualquer ruído que possa estar contido nas observações e então usá-lo para prever valores futuros da série.

Abordou-se o tema computação evolucionista, especificamente, algoritmos genéticos e programação genética.

Baseado nos resíduos gerados pelo modelo de *Winters*, ou seja, os erros de cada ponto da série comparado com a original, estudou-se a viabilidade de se obter uma

equação matemática que identifique o comportamento dos resíduos. Para isso, utilizou-se a Programação Genética.

Foram utilizados diversos experimentos para verificar a viabilidade da metodologia estudada. Os testes realizados com séries reais foram retirados da literatura de previsão de séries temporais.

A partir da realização dos experimentos pode-se concluir que a metodologia foi bastante eficiente na tarefa de previsão. Deve-se ressaltar que esta estratégia é adequada quando a predição efetuada com o Método de *Winters* consegue capturar adequadamente a resposta do sinal. Assim a correção dos resíduos fica extremamente precisa. Os resultados obtidos foram comparados com outros modelos de previsão como a metodologia de Box & Jenkins, metodologia de BCIGP e a Programação Genética usando a série original. O protótipo de software implementado pode ser considerado uma ferramenta de uso prático amigável e confiável.

Como trabalhos futuros pode-se destacar os seguintes itens:

- Utilização da metodologia em outras séries, como por exemplo, séries financeiras;
- Comparação do modelo com outras técnicas, como Redes Neurais;
- Utilização da Programação Genética na previsão de resíduos gerados por outros modelos estatísticos.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] ALLEMÃO, Antonio Freire, Redes Neurais Aplicadas à Previsão de Demanda de Numerário em Agências Bancarias, DCC/IM/UFRJ. Rio de Janeiro 2004.

[2] ANDRADE, Lucio Pereira, Procedimento Interativo de Agrupamento de Dados. COPPE/UFRJ. Rio de Janeiro 2004.

[3] BANZHAF; W. NORDIN, P.; KELLER, R. E. & FRANCONE, F. D., Genetic Programming an introduction. Morgan Kaufmann, 1998.

[4] BICKLE, Y. & THIELE, L., A Mathematical Analysis of Tournament Selection. In: ICGA95. L. J. Eshelman. Ed. San Francisco: Morgan Kaufmann Publishers, p. 9-16, 1995.

[5] BARRETO, 3snB.H13i TsH13 CKL. L. J.46[(BARREelection.)]TJ0 4195i do S6[(

[11] DARWIN, C., A origem das Espécies e a Seleção Natural. Ed. Hemus, 5ª Ed., 2000.

[12] DAVIS, M. M.; AQUILANO, N. J.; CHASE, R. B. Fundamentos da Administração da Produção. 3ª ed. Porto Alegre: Bookman, 1997.

[13] C. Darwin, The Origin of Species, Jonh Murray, 1859 (Penguin Classics 1985)

[14] CHELLAPILLA, K., Evolutionary programming with tree mutations: Evolving computer programs without sub-tree crossover. Genetic Programming. In: Proc. Second Annual Conference of San Francisco, CA. Morgan Kauffmann, p. 432-438, 1997.

[15] CORREIA, Davi, Algoritmos Genéticos e Elementos Finitos na Síntese de Dispositivos Fotônicos, DMO/FEEC/UNICAMP, marco de 2002.

[16] DAIDA, J. M. Challenges with verification, repeatability and meaningful comparisions in genetic programming. Proceedings of the 4th Annual Conference in Genetic Programming (GECCO'99). ISBN 1558606114. pp. 1069-1076. Morgan Kaufmann, 1999.

[17] GECCO, Genetic and Evolutionary Computation Conference. 2002, 2006.

[18] GIL, M. G. B. Análise de regressão.<http://dequim.ist.utl.pt/doe/ACETATOS/REGRES-ace.pdf > Acesso em: 20 jan. 2007.

[19] HOLLAND J. H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. Ann Arbor, MI: University of Michigan Press. 1975.

[20] KABOUDAN, M. A. (2000). Genetic Programming Prediction of Stock Prices. Journal Computational Economics, 16:207–236.

[21] KOZA, J. R. Hierarquical genetic algorithms operating on populations of computer programs. Proceedings of the 11th International Joint Conference on Artificial Intelligent (IJCAI-89). Detroit, MI. Pp 768-774. Morgan Kaufmann, 1989.

[22] KRAMER, M. D. & ZHANG, D. A Genetic Programming System. In: The 24th Annual International Computer Software and Applications Conference, p. 614-619, IEEE Press, 2000.

[23] KOZA, J. R., Genetic Programming: On the Programming of Computers by Means of Natural Selection. WIT Press, 1992.

[24] LATINI G. & PASSERINI G., Handling Missing Data – Applications to Environmental Analysis, Ed. WitPress.Boston, 2004.

[25] LUKE, S. & PAINAT, L., A survey and comparison of tree generation algorithms. Proceedings of the 6th Annual Conference in Genetic Programming (GECCO 2001). Springer-Verlag, 2001.

[26] MAKRIDAKIS, S.; WHEELWRIGHT, S. C; McGEE, V. E. Forecasting: methods and applications. 2ª ed., New York: John Wiley & Sons, 1983.

[27] MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; YNDMAN, R. J. Forecasting: methods and applications. 3ª ed., New York: John Wiley & Sons, 1998.

[28] MICHALEWICZ, Z. Genetic Algorithms + Data Structures = Evolution Programs. ESpringer- Verlag, 3rd Ed. New York, 1997.

[29] MONTGOMERY, Douglas C., JOHNSON, Lynwood A.; GARDINER, John S. Forecasting & Time Series Analysis 2. Ed. New York: McGraw-Hill Inc., 1990. 381p.

[30] MORETTIN; TOLOI, P. A. & TOLOI, C. M. C. Análise de séries temporais. Ed. Edgard Blucher LTDA. São Paulo, 2004.

[31] MUHLENBEIN, H. & SCHIERKAMP-VOOSEN, D. Predictive models for the breeder genetic algorithms. Evolutionary Computation, v.1, n.1, p. 25-49. MIT Press, 1993.

[32] PELLEGRINI, F. R. Metodologia para implementação de sistemas de previsão de demanda. Porto Alegre, 2000. 146 f. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul.

[33] PINHO, Flávia e BORGES, Silva. Otimização via Algoritmo Genético do Processo Construtivos de Estruturas de Concreto Submetidos à Retração Restringida Tendo em Vista a Fissuração nas Primeiras Idades. Rio de Janeiro, RJ, abril de 2002.

[34] RODRIGUES, E. L. M. Evolução de funções em programação genética orientada a gramáticas. Dissertação (Mestrado em Informática). Universidade Federal do Paraná, 2002.

[35] SCHWEFEL, H., Evolution and optimum seeking. Sixth-Generation Computer Technology Series. John Wiley & Sons. New York, 1995.

[36] SOUZA, L. V.; COSTA, E. O. & POZO, A. T. R., Previsão de Séries Temporais utilizando Programação Genética. In: XXXVII Simpósio Brasileiro de Pesquisa Operacional.Gramado, RS, Brasil, setembro 2005 b.

[37] SOUZA, Luzia Vidal, Programação Genética e Combinação de Preditores Para Previsão de Séries Temporais. Tese de Doutorado. UFPR. Curitiba 2006.

[38] STEVENSON, William J. Estatística Aplicada à Administração. São Paulo: Harbra Ltda., 1986.

[39] TERADA, P. D., Compilers and Compiler Generators, an introduction with C++. McGraw-Hill, Makron, 1991

[40] GREFENSTETTE, J. J. & BAKER, J. E., How genetic algorithms work: A critical look at implicit parallelism. In Proc. 3rd International Conference on Genetic Algorithms, p. 20-27. San Mateo. CA. Morgan Kaufmann. San Francisco, CA, 1989.

[41] WHITLEY, D., The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trial is best. In: Schaffer, J. D., editor, Proc. 3rd Int. Conference on Genetic Algorithm, pp 116-121, San Mateo, CA. Morgan Kaufmann, San Francisco, CA, 1989.

ANEXO I

Data	Cons
jan/84	114,13
fev/84	110,79
mar/84	116,46
abr/84	111,57
mai/84	120,66
jun/84	121,15
jul/84	121,27
ago/84	127,02
set/84	129,04
out/84	133,3
nov/84	130,6
dez/84	179,39
jan/85	120,64
fev/85	114,05
mar/85	130,6
abr/85	118,26
mai/85	145,54
jun/85	135,13
jul/85	153,35
ago/85	159,95
set/85	150,01
out/85	164,93
nov/85	170,37
dez/85	220,96
jan/86	134,26
fev/86	133,11
mar/86	147,84
abr/86	164,46
mai/86	181,86
jun/86	170,44
jul/86	186,64
ago/86	174,21
set/86	181,62
out/86	194,16
nov/86	181,9
dez/86	232,01
jan/87	140,16
fev/87	130,78
mar/87	119,04
abr/87	120,73
mai/87	129,81
jun/87	111,04
jul/87	122,75
ago/97	133,95

Data	Temp
1/jan/97	18,4
2/jan/97	17,4
3/jan/97	17,1
4/jan/97	16,6
5/jan/97	15,8
6/jan/97	18,5
7/jan/97	17,3
8/jan/97	18,7
9/jan/97	19,6
10/jan/97	19,1
11/jan/97	18,9
12/jan/97	19,6
13/jan/97	19,5
14/jan/97	19,7
15/jan/97	20,8
16/jan/97	19,5
17/jan/97	18,5
18/jan/97	18,6
19/jan/97	17,8
20/jan/97	19,5
21/jan/97	19,4
22/jan/97	19,7
23/jan/97	19,4
24/jan/97	18,9
25/jan/97	18,9
26/jan/97	19
27/jan/97	18,6
28/jan/97	18,8
29/jan/97	19,6
30/jan/97	19,5
31/jan/97	19,5
1/fev/97	20
2/fev/97	19,4
3/fev/97	18,9
4/fev/97	18,2
5/fev/97	19
6/fev/97	17,3
7/fev/97	18,7
8/fev/97	18,1
9/fev/97	18,5
10/fev/97	17,1
11/fev/97	17,4
12/fev/97	18
13/fev/97	17

Bebida
56.92
49.67
49.63
53.84
57.49
59.94
54.51
58.28
63.46
68.68
69.3
69.72
69.88
60.99
57.48
64.84
72.92
67.25
77.81
74.96
81.57
87.45
81.78
79.45
76.84
72.69
73.31
71.74
64.27
55.71
59.83
66.17
71.38
77.84
77.88
78.67
77.4
66.73
73
69.85
64.22
72.22
64.64
71.84

Data	ipi
jan/85	84,15
fev/85	71,53
mar/85	72,55
abr/85	67,02
mai/85	72,61
jun/85	95,6
jul/85	105,1
ago/85	106,7
set/85	96,97
out/85	109,8
nov/85	97,8
dez/85	93,63
jan/86	92,39
fev/86	77,03
mar/86	66,75
abr/86	69,81
mai/86	73,01
jun/86	90,07
jul/86	100,8
ago/86	97,12
set/86	104
out/86	115,9
nov/86	97,8
dez/86	92,48
jan/87	92,14
fev/87	85,61
mar/87	86,39
abr/87	75,96
mai/87	73,53
jun/87	96,19
jul/87	103
ago/87	105,1
set/87	111
out/87	117,1
nov/87	104,2
dez/87	100,4
jan/88	87,06
fev/88	76,24
mar/88	78,47
abr/88	70,55
mai/88	75,14
jun/88	110,2
jul/88	112,7
ago/88	115,2

Data	Manchas
1749	80,9
1750	83,4
1751	47,7
1752	47,8
1753	30,7
1754	12,2
1755	9,6
1756	10,2
1757	32,4
1758	47,6
1759	54
1760	62,9
1761	85,9
1762	61,2
1763	45,1
1764	36,4
1765	20,9
1766	11,4
1767	37,8
1768	69,8
1769	106,1
1770	100,8
1771	81,6
1772	66,5
1773	34,8
1774	30,6
1775	7
1776	19,8
1777	92,5
1778	154,4
1779	125,9
1780	84,8
1781	68,1
1782	38,5
1783	22,8
1784	10,2
1785	24,1
1786	82,9
1787	132
1788	130,9
1789	118,1
1790	89,9
1791	66,6
1792	60

Data	Fort.
1849	2001
1850	852
1851	1806
1852	1356
1853	1233
1854	1590
1855	1273
1856	1770
1857	1734
1858	1457
1859	1357
1860	1716
1861	1445
1862	1468
1863	1452
1864	1098
1865	1238
1866	2478
1867	832
1868	1289
1869	1470
1870	1628
1871	1459
1872	2256
1873	2058
1874	1487
1875	1581
1876	1569
1877	468
1878	503
1879	597
1880	1539
1881	1423
1882	1246
1883	1508
1884	1047
1885	1307
1886	1399
1887	1320
1888	736
1889	784
1890	1534
1891	1077
1892	1211

set/87	125,41	14/fev/97	18,1	75.73	**set/88**	111,2	1793	46,9	1893	1430
out/87	132,05	15/fev/97	19	74.7	**out/88**	99,91	1794	41	1894	2505
nov/87	129,54	16/fev/97	18,6	75.32	**nov/88**	96,85	1795	21,3	1895	2491
dez/87	176,37	17/fev/97	17,2	81.01	**dez/88**	89,9	1796	16	1896	2144
jan/88	110,09	18/fev/97	19,2	73.99	**jan/89**	82,1	1797	6,4	1897	1839
fev/88	113,25	19/fev/97	18,3	65.1	**fev/89**	72,06	1798	4,1	1898	863
mar/88	124,03	20/fev/97	21,1	75.69	**mar/89**	78,65	1799	6,8	1899	2414
abr/88	110,63	21/fev/97	19,1	77.47	**abr/89**	71,45	1800	14,5	1900	940
mai/88	116,72	22/fev/97	20,1	85.84	**mai/89**	74,93	1801	34	1901	1545
jun/88	124,63	23/fev/97	19	89.84	**jun/89**	96,85	1802	45	1902	878
jul/88	124,38	24/fev/97	19,3	80.53	**jul/89**	103,2	1803	43,1	1903	789
ago/88	130,27	25/fev/97	19	87.12	**ago/89**	114,2	1804	47,5	1904	1136
set/88	119,87	26/fev/97	17,3	86.68	**set/89**	112,7	1805	42,2	1905	1189
out/88	115,75	27/fev/97	18,4	93.66	**out/89**	122,2	1806	28,1	1906	1430
nov/88	122,44	28/fev/97	17,3	91.55	**nov/89**	111,6	1807	10,1	1907	697
dez/88	162,43	1/mar/97	17,1	86.61	**dez/89**	97,83	1808	8,1	1908	834
jan/89	105,89	2/mar/97	15,4	89.89	**jan/90**	100,6	1809	2,5	1909	1015
fev/89	115,59	3/mar/97	17,4	80.13	**fev/90**	82,5	1810	0,1	1910	2051
mar/89	147	4/mar/97	16,1	73.99	**mar/90**	76,56	1811	1,4	1911	1373
abr/89	131,7	5/mar/97	18,2	69.5	**abr/90**	65,81	1812	5	1912	2446
mai/89	131,32	6/mar/97	18,2	90.27	**mai/90**	83,59	1813	12,2	1913	1905
jun/89	136,66	7/mar/97	18,9	82.57	**jun/90**	92,06	1814	13,9	1914	1512
jul/89	126,43	8/mar/97	18	80.37	**jul/90**	107,2	1815	35,4	1915	530
ago/89	134,88	9/mar/97	19,3	83.45	**ago/90**	116,3	1816	45,8	1916	1328
set/89	128,26	10/mar/97	17,8	83.13	**set/90**	109,3	1817	41,1	1917	2077
out/89	125,32	11/mar/97	18,8	92.47	**out/90**	118,2	1818	30,4	1918	1319
nov/89	124,61	12/mar/97	18,3	97.07	**nov/90**	111,2	1819	23,9	1919	656
dez/89	166,11	13/mar/97	18,4	93.9	**dez/90**	95,1	1820	15,7	1920	1847
jan/90	116,25	14/mar/97	18,4	90.99	**jan/91**	95,41	1821	6,6	1921	2496
fev/90	96,93	15/mar/97	18,3	77.52	**fev/91**	74,4	1822	4	1922	1595
mar/90	89,27	16/mar/97	17,5	86.95	**mar/91**	84,5	1823	1,8	1923	1513
abr/90	101,87	17/mar/97	16,8	109.13	**abr/91**	93,51	1824	8,5	1924	1847
mai/90	125,57	18/mar/97	15,6	103.65	**mai/91**	99,95	1825	16,6	1925	1137
jun/90	113,31	19/mar/97	16	109.75	**jun/91**	103,3	1826	36,3	1926	1571
jul/90	109,39	20/mar/97	17,3	110.25	**jul/91**	115,9	1827	49,7	1927	1195
ago/90	127,33	21/mar/97	16,9	107.59	**ago/91**	118,1	1828	62,5	1928	995
set/90	120,56	22/mar/97	14	103.5	**set/91**	109,8	1829	67	1929	1230
out/90	117,73	23/mar/97	14,4	115.16	**out/91**	115,7	1830	71	1930	1107
nov/90	113,81	24/mar/97	15,6	97.86	**nov/91**	101,5	1831	47,8	1931	1133
dez/90	147,25	25/mar/97	16,9	87.65	**dez/91**	87,93	1832	27,5	1932	879
jan/91	100,15	26/mar/97	17,2	83.97	**jan/92**	87,44	1833	8,5	1933	937
fev/91	95,11	27/mar/97	16,1	74.02	**fev/92**	82,39	1834	13,2	1934	1888
mar/91	112,26	28/mar/97	13,6	77.01	**mar/92**	86,54	1835	56,9	1935	1661
abr/91	109,39	29/mar/97	14,6	74.54	**abr/92**	88,74	1836	121,5	1936	820
mai/91	114,2	30/mar/97	13	93.9	**mai/92**	96,92	1837	138,3	1937	1313
jun/91	113,8	31/mar/97	16,3	83.92	**jun/92**	107,9	1838	103,2	1938	1586
jul/91	126,47	1/abr/97	17,6	79.09	**jul/92**	116,1	1839	85,8	1939	1911
ago/91	128,36	2/abr/97	18	77.59	**ago/92**	114	1840	63,2	1940	1447
set/91	115,71	3/abr/97	18,1	79.39	**set/92**	105,7	1841	36,8	1941	916
out/91	116,09	4/abr/97	17,6	92.79	**out/92**	111,3	1842	24,2	1942	780
nov/91	99,53	5/abr/97	14,8	92.03	**nov/92**	106,5	1843	10,7	1943	1042
dez/91	127,27	6/abr/97	14,9	91.92	**dez/92**	95,59	1844	15	1944	1090

jan/92	87,08	7/abr/97	14,4	82.42	**jan/93**	90,36	1845	40,1	1945	1750
fev/92	85,67	8/abr/97	14,2	77.49	**fev/93**	77,73	1846	61,5	1946	1724
mar/92	82,02	9/abr/97	15	91.54	**mar/93**	93,38	1847	98,5	1947	1726
abr/92	98,2	10/abr/97	11,8	80.3	**abr/93**	89,65	1848	124,3	1948	1384
mai/92	96,44	11/abr/97	12,2	95.91	**mai/93**	100,5	1849	95,9	1949	1881
jun/92	90,23	12/abr/97	14,3	90.8	**jun/93**	105,8	1850	66,5	1950	1114
jul/92	97,15	13/abr/97	13,2	92.46	**jul/93**	115,2	1851	64,5	1951	747
ago/92	95,08	14/abr/97	14,4	90.25	**ago/93**	114	1852	54,2	1952	1378
set/92	94	15/abr/97	13,3	90.58	**set/93**	108	1853	39	1953	1068
out/92	93	16/abr/97	14,4	99.87	**out/93**	107,9	1854	20,6	1954	1032
nov/92	96,09	17/abr/97	17,6	102.06	**nov/93**	107,1	1855	6,7	1955	1152
dez/92	129,21	18/abr/97	17	93.45	**dez/93**	95,89	1856	4,3	1956	806
jan/93	75,39	19/abr/97	16	79.19	**jan/94**	87,9	1857	22,8	1957	1225
fev/93	77,7	20/abr/97	16,2	80.65	**fev/94**	76,12	1858	54,8	1958	504
mar/93	97,34	21/abr/97	16,3	91.97	**mar/94**	90,34	1859	93,8	1959	1493
abr/93	84,97	22/abr/97	16,4	89.91	**abr/94**	85,33	1860	95,7	1960	1011
mai/93	87,55	23/abr/97	15,2	93.98	**mai/94**	99,31	1861	77,2	1961	1737
jun/93	86,64	24/abr/97	15,7	95.14	**jun/94**	107,2	1862	59,1	1962	1258
jul/93	90,52	25/abr/97	14,3	82.88	**jul/94**	115,5	1863	44	1963	2102
ago/93	95,4	26/abr/97	14,6	107.12	**ago/94**	126,9	1864	47	1964	2428
set/93	95,2	27/abr/97	13,3	115.08	**set/94**	117,7	1865	30,5	1965	1630
out/93	95,8	28/abr/97	15,8	117.13	**out/94**	116,5	1866	16,3	1966	1288
nov/93	101,23	29/abr/97	11,8	124.46	**nov/94**	109,8	1867	7,3	1967	1839
dez/93	128,49	30/abr/97	14,3	122.79	**dez/94**	99,9	1868	37,3	1968	1385
jan/94	85,63	1/mai/97	14,4	115.55	**jan/95**	95,68	1869	73,9	1969	1805
fev/94	82,77	2/mai/97	14,4	106.41	**fev/95**	84,69	1870	139,1	1970	1192
mar/94	96,55	3/mai/97	14,4	132.93	**mar/95**	102,8	1871	111,2	1971	2093
abr/94	81,33	4/mai/97	16,8	120.24	**abr/95**	89,73	1872	101,7	1972	1299
mai/94	96,91	5/mai/97	15,4	107.2	**mai/95**	104,3	1873	66,3	1973	2331
jun/94	83,76	6/mai/97	13,8	109.97	**jun/95**	109,9	1874	44,7	1974	2512
jul/94	90,19	7/mai/97	11,2	106.86	**jul/95**	119,3	1875	17,1	1975	1778
ago/94	114,84	8/mai/97	14,6	116.11	**ago/95**	125,9	1876	11,3	1976	1417
set/94	108,4	9/mai/97	13,3	119.69	**set/95**	121,3	1877	12,3	1977	1941
out/94	106,05	10/mai/97	14,8	120.43	**out/95**	131,1	1878	3,4	1978	1752
nov/94	109,71	11/mai/97	14	128.39	**nov/95**	127,4	1879	6	1979	996
dez/94	143,86	12/mai/97	13,7	122.48	**dez/95**	115,4	1880	32,3	1980	1216
jan/95	99,12	13/mai/97	12,6	113.13	**jan/96**	110,7	1881	54,3	1981	1086
fev/95	99,28	14/mai/97	13	109.35	**fev/96**	93,78	1882	58,7	1982	1051
mar/95	114,75	15/mai/97	12,6	107.59	**mar/96**	100,8	1883	63,7	1983	955
abr/95	106,13	16/mai/97	12,3	112.54	**abr/96**	99,53	1884	63,5	1984	2029
mai/95	110,02	17/mai/97	13,1	109.17	**mai/96**	115,6	1885	52,2	1985	2836
jun/95	108,07	18/mai/97	13	95.92	**jun/96**	115,3	1886	25,4	1986	2457
jul/95	112,52	19/mai/97	13,5	114.53	**jul/96**	133,3	1887	13,1	1987	1260
ago/95	113,87	20/mai/97	12,8	111.83	**ago/96**	135,6	1888	6,8	1988	1862
set/95	107,84	21/mai/97	13,3	113.36	**set/96**	126,6	1889	6,3	1989	1863
out/95	112,12	22/mai/97	15	123.96	**out/96**	134,9	1890	7,1	1990	978
nov/95	112,03	23/mai/97	11,8	123.47	**nov/96**	121,5	1891	35,6	1991	1549
dez/95	139,37	24/mai/97	13,8	124.58	**dez/96**	109,8	1892	73	1992	1089
jan/96	92,24	25/mai/97	11,2	113.65	**jan/97**	104	1893	84,9	1993	1043
fev/96	93,56	26/mai/97	10,6	95.87	**fev/97**	88,2	1894	78	1994	2380
mar/96	107,37	27/mai/97	11,4	100.02	**mar/97**	99,46	1895	64	1995	2144
abr/96	102,89	28/mai/97	12,3	131.62	**abr/97**	105,2	1896	41,8	1996	1708

mai/96	114,78
jun/96	102,88
jul/96	118,41
ago/96	119,23
set/96	117,36
out/96	122,06

29/mai/97	11
30/mai/97	10,6
31/mai/97	10,8
1/jun/97	8,6
2/jun/97	6,5
3/jun/97	8,2
4/jun/97	12,4
5/jun/97	14,4
6/jun/97	14,1
7/jun/97	11,5
8/jun/97	5,6
9/jun/97	7,2
10/jun/97	9,5
11/jun/97	7,9
12/jun/97	14,5
13/jun/97	11,6
14/jun/97	14,4
15/jun/97	14,5
16/jun/97	12
17/jun/97	10,9
18/jun/97	10,2
19/jun/97	13,2
20/jun/97	12,4
21/jun/97	13,1
22/jun/97	14,6
23/jun/97	13,8
24/jun/97	14,7
25/jun/97	12,7
26/jun/97	14,7
27/jun/97	13,3
28/jun/97	12,6
29/jun/97	13,5
30/jun/97	12,1
1/jul/97	11,8
2/jul/97	13
3/jul/97	15,4
4/jul/97	14,6
5/jul/97	12
6/jul/97	13,2
7/jul/97	14,9
8/jul/97	14,8
9/jul/97	13,2
10/jul/97	12,2
11/jul/97	11,5
12/jul/97	12,8
13/jul/97	13,9
14/jul/97	12
15/jul/97	10,4
16/jul/97	11,8
17/jul/97	13,4
18/jul/97	12,6
19/jul/97	14,8

106.44
102.91
112.5
110.47
117.69
121.4
119.52
123.06
109.85
94.07
124.66
101.02
102.78
102.38
110.99
105.05
109.98
117.34
118.53
128.84
94.29
88.54
113.61
129.35
97.76
97.27
100.76
107.58
114.39
123.33
126.08
132.26
95.26
93.55
134.62
117.45
115.11
99.17
101.1

mai/97	112
jun/97	117,3
jul/97	132,7
ago/97	136,4
set/97	140,7
out/97	143,1
nov/97	121,2
dez/97	111,4
jan/98	101,7
fev/98	92,32
mar/98	102,1
abr/98	102,6
mai/98	112,6
jun/98	125,5
jul/98	140,5
ago/98	138,8
set/98	135,9
out/98	135,2
nov/98	131,1
dez/98	112,1
jan/99	98,9
fev/99	89,77
mar/99	107,5
abr/99	105
mai/99	126,5
jun/99	129,8
jul/99	136,5
ago/99	148,3
set/99	143,4
out/99	143
nov/99	130,4
dez/99	115,1
jan/00	100,1
fev/00	99,9
mar/00	105,4
abr/00	102
mai/00	116,2
jun/00	124,7
jul/00	131,1

1897	26,2
1898	26,7
1899	12,1
1900	9,5
1901	2,7
1902	5
1903	24,4
1904	42
1905	63,5
1906	53,8
1907	62
1908	48,5
1909	43,9
1910	18,6
1911	5,7
1912	3,6
1913	1,4
1914	9,6
1915	47,4
1916	57,1
1917	103,9
1918	80,6
1919	63,6
1920	37,6
1921	26,1
1922	14,2
1923	5,8
1924	16,7

1997	1143

20/jul/97	12,6
21/jul/97	12,8
22/jul/97	11,8
23/jul/97	12,8
24/jul/97	13
25/jul/97	12,5
26/jul/97	10,2
27/jul/97	11,6
28/jul/97	8,6
29/jul/97	9,3
30/jul/97	10
31/jul/97	13,2
1/ago/97	11,5
2/ago/97	11,6
3/ago/97	13,4
4/ago/97	15,1
5/ago/97	11,9
6/ago/97	12,8
7/ago/97	14,2
8/ago/97	10,4
9/ago/97	9,4
10/ago/97	6,6
11/ago/97	8,3
12/ago/97	7,8
13/ago/97	9,6
14/ago/97	11
15/ago/97	10,6
16/ago/97	11,6
17/ago/97	11
18/ago/97	11,4
19/ago/97	12,6
20/ago/97	12,4
21/ago/97	9,5
22/ago/97	14,8
23/ago/97	15,5
24/ago/97	14,3
25/ago/97	11,8
26/ago/97	11,2
27/ago/97	12,4
28/ago/97	10,4
29/ago/97	11
30/ago/97	11,6
31/ago/97	12,2
1/set/97	10,6
2/set/97	11,3
3/set/97	12,2
4/set/97	14,4
5/set/97	16,8
6/set/97	13,9
7/set/97	14,5
8/set/97	13,4
9/set/97	15,9

10/set/97	14
11/set/97	12,3
12/set/97	10,8
13/set/97	14,2
14/set/97	18,2
15/set/97	13,6
16/set/97	12,8
17/set/97	13,2
18/set/97	11,8
19/set/97	13,5
20/set/97	15
21/set/97	15
22/set/97	13,6
23/set/97	13,2
24/set/97	14,8
25/set/97	14,5
26/set/97	14,3
27/set/97	14
28/set/97	15,5
29/set/97	13,9
30/set/97	17
1/out/97	17,9
2/out/97	14
3/out/97	12,4
4/out/97	12,5
5/out/97	15,6
6/out/97	14,8
7/out/97	14,5
8/out/97	15,1
9/out/97	15,6
10/out/97	16,4
11/out/97	18
12/out/97	16,1
13/out/97	17,1
14/out/97	18,3
15/out/97	18,1
16/out/97	15,1
17/out/97	13,4
18/out/97	13
19/out/97	13,4
20/out/97	15,4
21/out/97	15,8
22/out/97	15,2
23/out/97	15,7
24/out/97	16,3
25/out/97	16,9
26/out/97	15,9
27/out/97	15,9
28/out/97	16,9
29/out/97	18,9
30/out/97	19
31/out/97	18,8

1/nov/97	18,2
2/nov/97	18,2
3/nov/97	19,6
4/nov/97	18
5/nov/97	18,9
6/nov/97	19
7/nov/97	19,7
8/nov/97	18,7
9/nov/97	16,9
10/nov/97	18,1
11/nov/97	20,1
12/nov/97	20,2
13/nov/97	20,8
14/nov/97	16,3
15/nov/97	16,1
16/nov/97	17,2
17/nov/97	19
18/nov/97	16,2
19/nov/97	16,1
20/nov/97	15,8
21/nov/97	18
22/nov/97	19
23/nov/97	19,8
24/nov/97	18,7
25/nov/97	18,6
26/nov/97	18,4
27/nov/97	20,1
28/nov/97	15,1
29/nov/97	13,5
30/nov/97	16,1
1/dez/97	17,7
2/dez/97	18
3/dez/97	16,6
4/dez/97	17,4
5/dez/97	16,2
6/dez/97	16,6
7/dez/97	18,8
8/dez/97	18,5
9/dez/97	16,2
10/dez/97	15,4
11/dez/97	16
12/dez/97	18,2
13/dez/97	19,7
14/dez/97	19,4
15/dez/97	20
16/dez/97	16,2
17/dez/97	16,2
18/dez/97	16,5
19/dez/97	17,9
20/dez/97	18,6
21/dez/97	20,1
22/dez/97	19,1

23/dez/97	21,1
24/dez/97	21,1
25/dez/97	20,8
26/dez/97	20,7
27/dez/97	21,2
28/dez/97	20,3
29/dez/97	19,2
30/dez/97	21,6
31/dez/97	19,8

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo