**UNIVERSIDADE DE SÃO PAULO
FACULDADE DE ZOOTECNIA E ENGENHARIA DE ALIMENTOS**

FLÁVIA DEVECHIO PROVIDELO MEIRELLES

**Modelo computacional de um rebanho bovino de corte virtual utilizando
Simulação Monte Carlo e Redes Neurais Artificiais**

**Pirassununga
2005**

FLÁVIA DEVECHIO PROVIDELO MEIRELLES

**Modelo computacional de um rebanho bovino de corte virtual utilizando
Simulação Monte Carlo e Redes Neurais Artificiais**

Tese apresentada à Faculdade de Zootecnia e Engenharia de Alimentos da Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Doutor em Zootecnia.

Área de Concentração: Qualidade e Produtividade Animal

Orientador: *Prof. Dr. José Bento S. Ferraz*

Co-orientador: *Prof. Dr. Ernane José X. Costa*

**Pirassununga
2005**

DEDICATÓRIA

*Dedico este trabalho às duas pessoas
que mais amo nesta vida ao meu
marido Flávio e meu filhote
Henrique que formam minha amada
família!*

OFERECIMENTO

*Aos meus pais Cleyde e Arlindo por
cuidar do Henrique quando
necessitei deixando-me tranqüila....*

*Aos meus irmãos: Lucas, Gaspar e
Claudia e meus afilhados queridos
Felipe e Giovana...*

AGRADECIMENTOS

A Deus que está sempre presente em nossas vidas nos confortando.

Ao Prof. Dr. José Bento Sterman Ferraz pela orientação e a liberdade de pensamento, sempre indicando o caminho correto da pesquisa, sendo otimista em todos os momentos e, principalmente, pelo exemplo que nos mostra em sua vida.

Ao Prof. Dr. Ernane José Xavier Costa pela co-orientação deste trabalho e por me iniciar nos estudos da pesquisa em inteligência artificial e, também, pelo exemplo de trabalho e seriedade.

Em especial, ao Prof. Dr. Antonio Caliri pela colaboração na parte de Simulação Monte Carlo dedicando um horário de sua agenda lotada e estando sempre disposto a transmitir conhecimento e sabedoria em nossas reuniões em Ribeirão Preto, sou muito grata

Ao Prof. Dr. Joanir Pereira Eler pela colaboração neste trabalho na parte que tange o Melhoramento Animal.

Ao Prof. Dr. Julio Balieiro pelas sugestões e discussões após a qualificação na parte do Melhoramento Animal ajudando a enriquecer os resultados.

Ao GMA - *Grupo de Melhoramento Genético Animal*, principalmente os pesquisadores: Elisângela, Sancho, Gerson e Marcelo por me fornecerem os bancos de dados reais para contrastar com os resultados das simulações e por me acolherem neste grupo para realização deste projeto.

Ao LAFAC – *Grupo de Física Aplicada e Computacional*, ao Aldo e a Ana Carolina pelo convívio diário e as discussões do trabalho e a união e por ter feito parte deste grupo.

A FZEA-USP - *Faculdade de Zootecnia e Engenharia de Alimentos da Universidade de São Paulo* por colocar à disposição sua infraestrutura para realização desta pesquisa.

RESUMO GERAL

MEIRELLES, F.D.P. **Modelo computacional de um rebanho bovino de corte virtual utilizando Simulação Monte Carlo e Redes Neurais Artificiais.** 2005. PhD Thesis – Faculdade de Zootecnia e Engenharia de Alimentos, Universidade de São Paulo, Pirassununga, 2005.

Neste trabalho, foram utilizadas duas ferramentas computacionais para fins de auxiliar tomadas de decisões na produção de bovinos de corte, criados de maneira extensivas, em condições de manejo encontrados no Brasil. A primeira parte do trabalho visou à construção de um *software* utilizando a técnica de Simulação Monte Carlo para analisar características de produção (ganho de peso) e manejo (fertilidade, anestro pós-parto, taxa de natalidade e puberdade). Na segunda parte do trabalho foi aplicada a técnica de Redes Neurais Artificiais para classificar animais, segundo ganho de peso nas fases de crescimento (nascimento ao desmame, do desmame ao sobreano) relacionado com o valor genético do ganho de peso do desmame ao sobreano (GP345) obtidos pelo BLUP. Ambos modelos mostraram potencial para auxiliar a produção de gado de corte.

Palavras Chaves: Simulação Monte Carlo; Modelos Estocásticos; Redes Neurais Artificiais; Gado de Corte; Nelore; Reprodução; Produção.

ABSTRACT

MEIRELLES, F.D.P. **Computational model of virtual beef cattle heard applying Monte Carlo Simulation and Artificial Neural Networks.** 2005. PhD Thesis – Faculdade de Zootecnia e Engenharia de Alimentos, Universidade de São Paulo, Pirassununga, 2005.

Herein we applied two different computational techniques with the specific objective to help the decision-making at Brazilian extensive beef cattle production systems. The first part of the work was dedicated to the construction of software based on Monte Carlo Simulation. Two different models were designed for further fusion and willing the analysis of productions (weight gain) and reproduction traits (fertility, post *partum anestrus*, born rate and puberty). The second part of the work applied Artificial Neural Network techniques to classify animals related to the weight gain during growing period (Weight at Calving, Weaning Weight, Weight at 550 days) comparing data with genetic value of the daily gain from weaning to 550 days adjusted to 345 days BLUP output. The results obtained in both models showed potential to help beef cattle production.

Keywords: Monte Carlo Simulation; Stochastic Models; Artificial Neural Network; Beef Cattle; Nelore, Reproduction, Production.

SUMÁRIO

CAPÍTULO I: MODELO COMPUTACIONAL DE UM REBANHO BOVINO DE CORTE VIRTUAL UTILIZANDO SIMULAÇÃO MONTE CARLO.....	9
RESUMO.....	9
1 INTRODUÇÃO.....	11
2 OBJETIVO.....	16
3 JUSTIFICATIVA.....	17
4 MATERIAL E MÉTODOS.....	18
4.1 Descrição Detalhada Do Modelo.....	18
4.1.1. Equações do Modelo I. Ganho De Peso.....	24
4.2 Algoritmos dos Programas.....	27
4.2.1. Algoritmo do Programa Do Modelo I. Ganho De Peso.....	27
4.2.2. Algoritmo do Programa Do Modelo II. Manejo dos Animais.....	28
5. RESULTADOS DO MODELO I: GANHO DE PESO.....	29
5.1 O Efeito do Ajuste do Ganho de Peso.....	30
5.2 O Efeito da DEP de Peso ao Nascer, ao Desmame e ao Sobreano.....	33
6. RESULTADOS DO MODELO II: MANEJO DOS ANIMAIS.....	42
6.1 O Efeito de Reprodução dentro de MV.....	43
6.1.1 Comportamento Gráfico das Fases de Crescimento ao Longo do Tempo.....	44
6.1.2 Fertilidade.....	47
6.1.2.1 Taxa de Gestação	47
6.1.2.2 Puberdade.....	49
6.1.3 Anestro Pós-Parto.....	50

6.1.4 Taxa de Natalidade	52
6.1.5 Taxa de Mortalidade.....	53
7 CONCLUSÕES.....	55
Capítulo II. A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS (RNAs) NA AVALIAÇÃO E NA PREDIÇÃO DE VALORES GENÉTICOS PARA GANHO DE PESO EM BOVINO DE CORTE DE RAÇAS ZEBUÍNAS BRASILEIRAS.....	56
RESUMO.....	56
1 INTRODUÇÃO.....	57
2 OBJETIVO.....	58
3 JUSTIFICATIVA.....	58
4 MATERIAL E MÉTODOS.....	59
4.1 Algoritmo <i>Backpropagation</i>	61
4.2 Escolhas de Parâmetros Ótimos de uma RNA do Tipo MLP.....	62
4.1.1 Função de Ativação.....	63
4.1.2 O Problema XOR.....	64
4.2 Aplicação da RNA do tipo MLP em Predição de Valores Genéticos para Ganho de Peso.....	69
4.2.1 Experimento RNA-MLP para Dados dos Filhos.....	69
5. RESULTADOS.....	72
6 CONCLUSÕES.....	75
REFERÊNCIAS.....	76
APÊNDICES.....	81

CAPÍTULO I: MODELO COMPUTACIONAL DE UM REBANHO BOVINO DE CORTE VIRTUAL UTILIZANDO SIMULAÇÃO MONTE CARLO.

RESUMO

A pecuária bovina teve um grande desenvolvimento nas últimas décadas, graças à aplicação de técnicas de melhoramento genético quantitativo e ao avanço do desenvolvimento computacional.

Os estudos de melhoramento permitiram identificar diversas características produtivas no gado, predizer sua herdabilidade e a contribuição de um reprodutor para sua progênie (DEP).

Atualmente existe uma série de características que podem ser selecionadas em um rebanho, e a escolha das características a serem selecionadas dependem em grande parte da percepção do produtor ou de modelos determinísticos.

Neste trabalho, propõe-se estruturar um modelo computacional estocástico que simule rebanhos em diversas condições, com o intuito de predizer o possível efeito no longo prazo da seleção de um touro reprodutor ou um manejo específico com a finalidade de ajudar os criadores na tomada de decisão.

Para tanto foram codificados dois programas aplicando o Método Monte Carlo que permitam simular o nascimento, crescimento, engorda, maturação sexual e a reprodução de bovinos.

Foram introduzidas características de reprodução e produção, tais como: a utilização de **estação de monta (EM)** que implica em cuidados especiais com o anestro, por exemplo; **taxa de prenhes** que implica em fertilidade, **de mortalidade, de natalidade, taxa de vendas e engorda** para poder simular o desenvolvimento destas características do rebanho ao longo do tempo.

A simulação dos dados por um sistema foi separada em dois modelos:

- 1. Modelo I: Ganho de Peso:** obteve-se por meio de simulação os dados de ganho de peso nos intervalos pré-determinados pela estação de monta (EM), introduzindo os efeitos parentais da DEP (diferença esperada na progênie) em diferentes fases de crescimento (nascimento, desmame, sobreano, garrote, boi magro e boi gordo) de três touros distintos;

2. Modelo II. Manejo dos Animais: teve-se a simulação da dinâmica de produção de bovinos em sistemas extensivos.

Os resultados obtidos até o momento mostram que é possível simular o desenvolvimento de rebanhos com resultados similares aos observados em sistemas reais sob condições brasileiras.

O modelo se mostrou geral e pode ser reparametrizado nas diversas situações de mudança de fase de crescimento sob uma estação de monta (EM) e nos parâmetros que envolvem ganho de peso como as diferenças esperadas nas progênies (DEP) do ganho de peso.

1. INTRODUÇÃO

O melhoramento genético quantitativo atingiu, com o desenvolvimento dos modelos animais, uma capacidade de estimar de maneira bastante acurada o valor genético de diversas características nos animais.

A pecuária bovina teve um grande desenvolvimento nas últimas décadas, graças à aplicação de técnicas de melhoramento genético quantitativo e ao avanço no desenvolvimento computacional.

Os estudos de melhoramento permitiram identificar diversas características produtivas no gado e prever sua herdabilidade e a contribuição de um reprodutor para sua progênie (DEP).

Existe, portanto, uma série de características que podem ser selecionadas em um rebanho, e as escolhas destas dependem em grande parte da percepção do produtor ou de modelos computacionais determinísticos.

Neste trabalho, propõe-se a estruturar um modelo computacional estocástico que simule rebanhos em diversas condições para prever o possível efeito a longo prazo da seleção de um reprodutor.

A **problemática** é que os bovinos em particular têm um intervalo de geração longo e tomadas de decisões para as características a serem selecionadas precisam ser bastante discutidas e pautadas no efeito longo prazo.

Fazer este tipo de estudo experimentalmente com animais reais, requer anos e um custo altíssimo na manutenção destes bovinos, tornando inviável tomadas de decisões suficientemente rápidas.

A simulação computacional ou modelagem matemática vem ao encontro desta problemática servindo como uma ferramenta de alta performance e baixo custo e, nos oferece maior eficiência na análise do impacto nas mudanças nas diferentes estratégias de manejo utilizadas (TESS; KOSTALD, 2000 a, b).

Exemplos de simulação com respostas rápidas baseadas em modelos são:

1. Baseado na escolhas de reprodutores para um rebanho, visando seleção para ganho de peso dos animais. Como evolui a tendência genética de rebanhos no longo prazo (vinte anos)?
2. Visando aumentar a eficiência reprodutiva de um rebanho é possível induzir o primeiro cio logo no início da estação de monta e inseminar todos animais em tempo fixo com uma taxa de prenhes de aproximadamente 50%. Todavia este tratamento tem custo elevado ao passo que o cio natural não tem custo. Um experimento computacional com os dados simulados permite verificar se esta estratégia resulta em aumento de número de animais para custear o investimento com o fármaco.

Portanto, **modelos matemáticos** são ferramentas que devem ser utilizadas para apoiar a síntese de informações, a compreensão dos sistemas de produção e suas interações. O exercício exigido para o desenvolvimento de modelos, facilita a explanação de princípios de manejo reprodutivo de bovinos (PERRY; SMITH, 2004) e uma otimização no processo de aprendizagem do funcionamento de uma fazenda extensiva de gado de corte.

No Brasil, a experiência na utilização de modelos matemáticos de simulação para a análise de sistemas de cria, em bovinos de corte, é crescente (ABREU; AMSTALDEN, 2004).

CAVALHEIRO et al. (2002) investigaram, por meio de dados simulados o efeito da heterogeneidade de variância residual entre grupos de contemporâneos. O BLUP

(melhor preditor linear não viesado) pressupõe homogeneidade de variâncias, e assim concluíram que os efeitos da heterogeneidade sobre as avaliações genéticas estão associados à pressão de seleção. Concluíram que as vacas e os filhos têm seus valores genéticos preditos mais afetados e, se, as correções não forem feitas, rebanhos que praticam seleção intensa podem ter seus animais classificados de maneira incorreta, indicando avaliações genéticas ponderadas para reduzir os riscos.

CUNHA et al (2003) propuseram um modelo, com dados simulados, para avaliar diferentes tipos de acasalamentos entre animais escolhidos com base no BLUP para pais da próxima geração, quanto aos valores fenotípicos médios e de endogamia média ao longo de 50 gerações. O intuito foi o estudo da preservação da variabilidade genética, ou seja, grau de heterozigose, pois animais homozigotos são mais propensos a doenças e, conseqüentemente, este fator leva à extinção dos animais, inviabilizando o sistema de produção animal.

O estudo motrou que os tipos de acasalamentos que não permitiram acasalar irmãos, e com razão sexual menor do que 10, proporcionaram maiores ganhos fenotípicos e foram mais efetivos no controle da endogamia, atrasando seu aparecimento no curto prazo, embora não tivessem impedido seu aumento e acúmulo, ao longo das gerações.

ABREU et al. (2003) construíram um modelo determinístico de análise bioeconômica com intuito de avaliar o impacto produtivo e econômico da introdução de período ou estação de monta (EM) em sistemas de produção de rebanhos de cria na Região do Brasil Central, ao longo de 20 anos. Segundo os resultados com os dados simulados, a tecnologia EM possibilitou um acréscimo de 31% na margem bruta de lucro quando comparado ao cenário tradicional, sem a utilização EM.

A adoção de EM é altamente vantajosa para proprietários de fazendas de cria extensiva de gado de corte, pois aumenta a taxa de natalidade, e, no contexto destes sistemas, dificilmente se terá disponível algo capaz de promover tal impacto positivo (ABREU et al., 2003).

Um outro fator importante no melhoramento de um sistema de produção extensiva tem sido a introdução de pastagem cultivada. Em 1982, foi elaborado um modelo determinístico de simulação por computador que permitia experimentar diversos cultivares de pastagens, e estimar o risco devido à cultura e o preço da carne. Foram introduzidos fatores como: financiamento de custeio da cultura, investimentos fixos e semi-fixos envolvendo inflação e taxas de juros (CESAR, 1982).

O tempo, isto é, a idade ideal das novilhas para serem entouradas em um rebanho, pode otimizar a eficiência na produção, pois quanto mais rápido as novilhas tornarem-se vacas prontas para se reproduzirem, melhor será o retorno econômico. Esta característica é conhecida como idade do primeiro parto (IP).

Neste sentido, PÖTTER et al. (2000), trabalhando com modelo matemático computacional, e dados da Região Sul do Brasil, verificaram que novilhas entouradas aos 14-15 meses apresentaram maior percentual de fêmeas em reprodução, com maior taxa de desfrute e maior produção de peso vivo/ha. Porém, novilhas entouradas de 26-27 meses obtiveram maior margem de lucro bruto.

Na mesma linha de pesquisa, BERRETA et al. (2001), também com dados da região Sul do Brasil, analisaram diferentes médias IP juntamente com a característica taxa de natalidade (TN).

Concluíram que sistemas com IP aos 24 meses apresentaram maior produtividade. Entretanto, temos uma resposta dependente da TN do rebanho de cria sendo maior para TN de 60% a 70% e o efeito é mínimo quando a TN é maior que 80%. O maior efeito do manejo na produtividade geral foi observado quando da redução de IP de 48 para 36 meses, aumento da TN de 50 para 75% e diminuição da idade ao abate de 56 para 36 meses.

Finalizando, dentre a produção nacional de modelos matemáticos, citamos SAKAGUTI et al. (2003) que propuseram um modelo utilizando Simulação Monte Carlo, onde analisaram curvas de crescimento de bovino de corte zebuínos brasileiros, mediante aplicação da equação de Von Bertalanffy variando seus parâmetros, utilizaram 200 touros com 10000 vacas totalizando 25000 progênes por geração. O estudo dos dados produzidos por meio irmãos foram analisadas em 10 gerações de seleção das características utilizando o BLUP.

Concluíram, que se selecionarmos o animal para o número de dias necessários para engordar 240 kg pós-desmame implicará em um aumento do peso à maturidade e na endogamia, o que pode ser uma consequência de estimativas pequenas de herdabilidade para esta característica em gerações recentes (SAKAGUTI *et al.*, 2003).

Em termos mundiais, também são crescentes as pesquisas em modelagem matemática em produção de bovino de corte (TESS; KOLSTAD, 2000 a, b).

Um dos primeiros modelos foi desenvolvido por SANDERS e CARTWRIGHT em (1979) e forneceu suporte a outros, como KAHN e SPEDDING (1984) e BOURDON e BRINKS (1987).

TESS e KOLSTAD (2000 a, b) desenvolveram dois modelos generalizados, onde os objetivos foram testar: 1) a performance de diversos tipos genéticos com qualidade e eficiência da alimentação e estratégias de manejos relacionados ao estado fisiológico do animal, tais como metabolismo de proteína; 2) a performance econômica na produção de vacas e bezerros alternando criação e sistema de manejo.

Os modelos desenvolvidos por eles utilizaram o procedimento de animais simulados, e contêm elementos determinísticos tais como crescimento, composição de carcaça além de elementos estocásticos que são as características reprodutivas. Estes modelos são esforços extremamente válidos em pesquisas básicas.

A técnica de Monte Carlo em modelos para estratégias de seleção e predição de características genéticas (FAMULA; VAN VLECK, 1981) vem sendo muito utilizada, tanto para efeitos macroscópicos, como para estimar a habilidade do efeito aditivo de fêmeas bovinas em grupos genéticos e tipos de produção, (MULLIN; PARK, 1995) quanto para efeitos microscópicos, por exemplo, simular a frequência de mutações em novos alelos em populações de *Drosófila* (FOURNET et al., 1995).

PLEASANTS (1997) construiu um modelo estocástico utilizando o Método de Simulação Monte Carlo com o objetivo de estudar estratégias de acasalamentos, mais precisamente, o momento do acasalamento, o que vem a ser a estação de monta (EM).

Concluíram que animais que iniciam o acasalamento mais cedo, obtiveram vantagens em termos de número de bezerros e taxa de concepção, e portanto a proporção de rebanhos que parem mais cedo, aumentou (PLEASANTS, 1997).

WERTH, et al. (1991) propuseram dois modelos: um de dinâmica estocástica para reprodução e um outro determinístico para estudar o efeito econômico. As características de estudo foram: intervalo de anestro (IAN) (tempo em que a vaca fica sem ciclar ou intervalo pós-parto), taxa de concepção no primeiro serviço (TPS) e EM. Este trabalho resultou nas seguintes implicações: quando IAN é longo, (90 dias ou mais) a renda líquida é maior com EM de 120 dias porque a TPS é alta (cerca de 80%)devido a própria duração do anestro.

E em geral, a renda líquida foi maior quando as vacas foram manejadas por ter TPS médio de 60% e 70 %.

PERRY e SMITH (2004) desenharam um software educativo utilizando simulação, visando dar subsídios para que os alunos da Universidade de Missouri-Columbia compreendessem o manejo reprodutivo de bovinos, tais como: estação de monta, prenhes, intervalo de anestro dentre outras e concluíram, mediante estudos das respostas dos alunos, que se tratava de uma maneira de melhorar o entendimento dos alunos sobre o efeito das técnicas de manejo.

TEDESCHI et al. (2004) trabalharam com Simulação Monte Carlo visando estudar índices de eficiência energética em bovinos leiteiros, sugeriram que selecionar animais com eficiência energética de 13,9 Mcal/libra teria um grande impacto na produção animal.

O método de Simulação Monte Carlo tem-se mostrado promissor comparado a outros métodos que são baseados em modelos determinísticos (OIKAWA e SATO, 1994), devido à variedade de suposições que podem ser consideradas e analisadas no modelo (PIERTESMAN et al, 2003) e os efeitos de estocasticidade e dinamismo que causam nos resultados e na simulação (KIRCHNER et al., 2004).

2 OBJETIVO

O objetivo geral do trabalho foi codificar dois programas utilizando o método de Simulação Monte Carlo que permitiram simular crescimento, reprodução, ganho de peso em bovinos da raça Nelore em propriedades extensivas.

3 JUSTIFICATIVA

É possível estimar o valor genético de um determinado animal e selecionar reprodutores baseados nesta estimativa para as características que se deseja selecionar.

Entretanto, não há um sistema que permita avaliar os efeitos de decisões tomadas ao longo do tempo. A simulação é uma ferramenta que pode colaborar na previsão ao longo do tempo e auxiliar nas decisões, como por exemplo:

1. Escolha de reprodutores;
2. Duração da estação de monta;
3. Máxima pressão de seleção;
4. Escolha de momentos de vendas;
5. Permanência do animal no rebanho.

Segundo ABREU e AMSTALDEN (2004), as vantagens que a modelagem nos possibilitam são:

1. Tratar-se de uma ferramenta de alta performance e baixo custo;
2. Ser utilizada para compreender e otimizar o desempenho de sistemas reais de produção;
3. Fornecer maior eficiência na análise de impacto das mudanças nas diferentes estratégias de manejo utilizadas;

Trata-se de um esforço multidisciplinar de pesquisadores, envolvendo também, produtores extensionistas.

4 MATERIAL E MÉTODO

4.1 DESCRIÇÃO DETALHADA DO MODELO

Neste trabalho, utilizou-se o método de Simulação Monte Carlo (SMC), no qual constrói-se uma cadeia de Markov (GAMERMAN, 1997). Esta, por sua vez, é uma regra para gerar sucessivas configurações sendo que uma nova configuração do sistema é obtida a partir unicamente da configuração presente, sendo muito apropriada para sistemas finitos e discretos. Como o interesse deste trabalho foi simular uma população de bovinos e, portanto, finita (um rebanho de 10^7), torna-se viável a utilização desta metodologia (PROVIDELO, 2000).

O método de SMC está sendo utilizado de forma crescente em sistemas de produção de bovinos de corte (FAMULA; VAN VLECK, 1981; PLEASANTS, 1997; SAKAGUTI et al, 2003; TEDESCHI et al, 2004). Seguindo esta vertente foram construídos dois modelos matemáticos descritos mais à posteriori.

Estes dois programas foram codificados (ver Apêndice A e B) em linguagem FORTRAN 90, utilizando a técnica de SMC, com duas sub-rotinas próprias de geradores de números aleatórios próprias, tendo em vista que o gerador de números aleatórios é a base de uma SMC.

Nos modelos apresentados neste trabalho, foram aplicados dois geradores de números aleatórios: o primeiro, utilizado no Modelo I e II, gera uma distribuição Uniforme de valores entre zero e um sendo que a qualidade deste gerador foi testada previamente (PROVIDELO, 2000).

A definição de qualidade de um gerador de números pseudo-aleatórios é determinada mediante avaliação de como e quão é aleatória a sequência gerada. O segundo gerador, utilizado no Modelo I e II, gera resultados de uma sequência (MULLIN; PARK, 1995) sob uma distribuição Normal de média zero e variância um (PRESS et al., 2002).

Para simular um rebanho real, o rebanho virtual total foi subdividido em 2 rebanhos (machos e fêmeas), enquanto o modelo geral foi subdividido em dois sub-modelos:

1. Modelo I: Ganho de Peso: os dados de ganho de peso são produzidos por meio de simulação nos intervalos pré-determinados pela estação de monta (EM) introduzindo os

efeitos parentais da DEP (diferença esperada na progênie) em diferentes fases de crescimento (nascimento, desmame e sobreano) de três touros distintos.

Neste modelo, foi aplicado o gerador de números aleatórios da distribuição Normal de média zero e variância um, ($N(0,1)$), pois segundo o Teorema Central do Limite, a soma de quaisquer números elevados de variáveis independentes, igualmente distribuídas, tende a uma distribuição Normal. Este resultado explica o aparecimento freqüente desta distribuição na natureza (SOBOL, 1983; PRESS et al, 2002).

2. Modelo II: Manejo dos Animais: neste modelo, foi simulada a dinâmica de um sistema de produção de bovinos criados de maneira extensiva, ou seja, a pasto. Para o Modelo II, foram aplicados dois geradores de números aleatórios para tomadas de decisões, o que gera uma distribuição Uniforme (PROVIDELO, 2000) e o outro que gera uma distribuição $N(0,1)$.

A esquematização do Modelo Geral pode ser observada na figura 1 abaixo:

Figura 1. Esquematização do Modelo Geral dividido em dois sub-modelos: **Modelo I:** Ganho de Peso: o ganho de peso nas fases de crescimento são analisadas neste sub-modelo; **Modelo II.** Manejo dos Animais: características como fertilidade das vacas e estação de monta são avaliadas neste sub-modelo.

Na figura 1, é possível observar que o manejo dos animais ocorre dentro de cada rebanho, gerando assim, uma evolução de “estados”; a cada mudança de estado é gerado um número aleatório, com uma certa probabilidade e um intervalo de tempo que está condicionado a uma estação de monta (EM, determinada pelo número de Passos Monte Carlo - onde um PMC equivale a um dia), permitindo alteração de um estado para outro ou não.

Os rebanhos de machos e fêmeas estão distribuídos em uma matriz $150 \times 150 \times 10$ ($MV(i,j,k)$), a qual (tendo em vista o número de animais simulados) é bastante esparsa para a evolução dos estados.

A evolução dos estados das fêmeas ocorrem em uma camada k distinta da evolução dos estados dos machos.

A separação dos rebanhos (machos e fêmeas) fez-se devido aos machos não contribuírem com o ciclo do rebanho.

No rebanho das fêmeas, tem-se os seguintes estados: vaca ciclando, animal fêmea que está pronto para reprodução; este animal pode morrer, ou mudar para o estado prenhe, mediante visita da matriz de reprodutores no período de estação de monta pré-estabelecida.

As vacas ciclando que não mudaram para o estado prenhe, se mantêm no estado vaca ciclando até a próxima estação de monta com uma certa probabilidade, ou são descartadas.

A vaca prenhe é mantida neste estado durante a gestação de 293 PMC com uma variabilidade segundo uma distribuição Normal de média 293 com uma variação de ± 12 dias.

Após o período de gestação, ocorre a mudança para o estado parida (momento onde ocorrem os nascimentos de machos e fêmeas). A parida vaca após um período de anestro pós-parto de 75 dias e variação de ± 21 dias, pode tornar-se prenhe desde que satisfaça uma probabilidade e que esteja dentro da estação de monta.

Caso a vaca parida não satisfaça nenhuma destas condições, o animal, por sua vez, retorna (com uma certa probabilidade) ao estado vaca ciclando, é descartado ou morre. A taxa de mortalidade é verificada pelo histórico de mortes ao longo dos anos.

Pode-se observar na figura 2 o modelo geral.

Figura 2. Evolução dos compartimentos do rebanho total $MV(i,j,k)$.

Pode-se observar na figura 2, a evolução dos estados de **MV**. A varredura desta matriz é seqüencial e a cada mudança de estado, é regida por probabilidades fixas e distintas p_i onde $i = 1, \dots, 25$, e cada p_i correspondente a uma conexão entre os estados: vaca ciclando, prenhe, parida, bezerra, desmamada, púbere, bezerro, desmamado, sobreano. Por exemplo, a novilha púbere pode tornar-se prenhe com uma certa probabilidade.

Nota-se que não foram citados os estados abate, descarte e mortes porque são acumuladores em nosso sistema, pois o número de óbitos em uma população é acumulativo. A conexão pontilhada entre mortes e pasto vazio corresponde apenas à simultaneidade em que ocorre os dois eventos, ou seja, à medida que o animal morre, é abatido ou é descartado temos um sítio vazio na matriz, “um espaço vazio no pasto” para introdução de mais um animal no rebanho total. Isto ocorre tanto na matriz de estados das fêmeas quanto dos machos.

A reposição do estado vaca ciclando é feita por uma parcela das vacas paridas e pelas novilhas públicas que não ficaram em tempo para a EM.

A fertilidade do rebanho pode ser analisada mediante o estudo e variação das probabilidades que interferem nas seguintes mudanças de estados: vaca ciclando tornar-se prenhe, a vaca parida voltar a emprenhar e a novilha púbere, emprenhar.

Já a taxa de reconcepção pode ser analisada mediante estudo da população de vaca parida que conclui o anestro pós-parto antes da EM e ficar prenhe.

A taxa de natalidade é proporcional ao número de vacas paridas no ano.

A taxa de mortalidade é o número de animais mortos no ano dividido pelo número total de animais.

Com o número de animais abatidos (machos) adicionado aos descartados (fêmeas) tem-se um histórico da geração de recursos da fazenda.

A dinâmica de mudança de estado depende de probabilidades distintas e passíveis de alteração, como também do período gerado pela EM. A EM está representando as condições das regiões sudeste e centro oeste do Brasil, e são distintas dentre os dois modelos, mostrando que pode ser experimentada.

O rebanho dos touros reprodutores interfere apenas no Modelo I e é composto por n animais que pretendemos testar distribuídos em uma matriz fictícia **MT**.

A introdução da **DEP** (diferença esperada na progênie) para touros distintos pode ser feita através do arquivo de entrada do programa do Modelo I. Ela interfere no ganho de peso de cada animal i nas seguintes fases: nascimento, desmame e sobreano. Além da DEP, dois outros fatores causam a variabilidade para ganho de peso para cada animal. São fatores multiplicativos *harvest* do gerador de números aleatórios de uma distribuição $N(0,1)$ e um desvio padrão escolhido, para cada mudança de fase. Estes fatores são explicados nas equações (i à iv).

4.1.1. EQUAÇÕES DO MODELO I. GANHO DE PESO

Neste modelo, introduzimos três estados (E_i) ou fases de crescimento. São eles:

1. **Nascimento:** os animais nascem em dias diferentes correspondentes ao intervalo [201,261], ou seja, o equivalente à 60 dias, de 21 julho até 18 de novembro este nascimento é aleatório regido pela seguinte equação:

$$PNB = 0,5 / \Delta \quad (i)$$

Onde,

$\Delta = 261 - 201$ (intervalo de nascimento dos bezerros).

$PNB = 0,5 = 50\%$ de probabilidade de nascer macho.

Como condição inicial do programa, aplicou-se o peso distinto ao nascer, como uma variabilidade aleatória para cada animal. Esta estocasticidade é regida por uma distribuição Normal de média zero e variância um. Conforme a equação abaixo:

$$PE_{ki} = \mu + DEPE_{ji} + harvest * 3,7 \quad (ii)$$

Onde :

PE_{ki} = peso em kg do animal k no estado i = nascido;

$\mu = 31,4$ = valor médio ao nascer da população total;

$DEPE_{ji}$ = diferença esperada na progênie do touro j ao nascer;

harvest = número aleatório entre $[-1,1]$ regido por uma distribuição $\sim N(0,1)$;

3,7 = constante que altera a amplitude da gaussiana.

O valor 31,4 de média, ao nascer, foi extraído do Sumário de Touros da CFM do ano de 2003.

2. **Desmame:** animais correspondentes à faixa de $I_1 = 201$ até $I_2 = 441$, com aproximadamente 240 dias dependendo da data de nascimento;

3. **Sobreano:** animais correspondentes à faixa de $I_3 = 441$ até $I_4 = 786$, com aproximadamente 345 dias dependendo da data de nascimento;

$$PE_{ki}(t+1) = PE_{ki}(t) + \mu_i + DEPE_{ji}/n + harvest * \Delta_i \quad (iii)$$

t = tempo t ;

$PE_{ki}(t+1)$ = peso em Kg do animal k no estado i e no tempo t ;

$\mu_{desmame} = 0.737$ = ganho médio de peso em gramas do nascimento ao desmame;

$\mu_{sobreano} = 0.345$ = ganho médio de peso em gramas do desmame ao sobreano;

$DEPE_{ji}$ = diferença esperada na progênie do touro j no estado i ;

$harvest$ = número aleatório entre $[-1,1]$ regido por uma distribuição $\sim N(0,1)$;

$\Delta_{desmame} = 1,78$ constante que altera a amplitude da gaussiana ocasionando um DP $\cong 28$;

$\Delta_{sobreano} = 1,68$ constante que altera a amplitude da gaussiana ocasionando um DP $\cong 40$;

n = número de dias do estado.

Como os animais nascem em dias diferentes, correspondente ao intervalo [201,261], ou seja, em 60 dias, de 21 julho até 18 de setembro houve a necessidade de corrigir o ganho de peso para cada animal i em cada estado E_i .

Sejam;

P_{aj} = peso ajustado;

P_s = peso simulado;

$\Delta = I_{i+1} - I_i$;

DaE_i = data de mudança do estado i ;

DaN = data de nascimento.

$$P_{aj} = \frac{\Delta * P_s}{DaE - DaN} \quad (iv)$$

Esta introdução permitirá que nascimentos derivados de um acasalamento específico possuam, por exemplo, evoluções de ganhos de pesos diferentes causando variabilidade no ganho de peso no rebanho.

É mostrado na figura 1 o armazenamento dos dados na matriz tridimensional correspondente ao **Modelo I: Ganho de Peso** é simulado da seguinte maneira em MRebM (i,j,k) onde, (i,j) representa cada animal e k os dados que queremos estudar:

- $MRebM(i,j,0)$ = armazena a data de nascimento de cada animal (i,j) ;

- $MRebM(i,j,1)$ = armazena o estado do animal; este pode ser ocupado onde ocorreu nascimento ou não ocupado;
- $MRebM(i,j,2)$ = peso ao nascer;
- $MRebM(i,j,4)$ = peso ao desmame, animal com aproximadamente 205 dias;
- $MRebM(i,j,6)$ = peso ao sobreano, animal com aproximadamente 550 dias;

O ganho de peso k de cada animal (i,j,k) é estocástico em cada estado, segundo as equações de ganho de peso já descritas, mas a varredura dos animais na matriz é sequencial.

Observa-se também pela figura 1 a adição de dados na matriz tridimensional $MV(i,j,k)$ para **Modelo II: Manejo** que é feito da seguinte maneira (i,j,k) onde, (i,j) representa cada animal e k os dados que queremos estudar:

- $MV(i,j,0)$ = armazena o número de sítios vazios na matriz para poder ocorrer os nascimentos;
- $MV(i,j,1)$ = armazena a data de nascimento de cada animal (i,j) do sexo feminino;
- $MV(i,j,2)$ = armazena o estado do animal que pode ser: vazio, bezerra, desmamada, púbere, vaca ciclando, prenhe e parida;
- $MV(i,j,3)$ = armazena o estado do animal que pode ser: vazio, bezerro, desmamado e sobreano;
- $MV(i,j,4)$ = armazena a data de nascimento de cada animal (i,j) do sexo masculino;
- $MV(i,j,5)$ = armazena a data de puberdade de cada animal (i,j) do sexo feminino;

Determinadas probabilidades podem ser alteradas a qualquer momento pelo usuário nos arquivos de entrada dos programas do Modelo I e II gerando experimentos distintos.

4.2 ALGORITMOS DOS PROGRAMAS

4.2.1. ALGORITMO DO PROGRAMA DO MODELO I. GANHO DE PESO.

Nesta seção introduzimos o algoritmo do programa que representa o Modelo I.

Figura 3. Algoritmo do modelo 1

4.2.2 ALGORITMO DO PROGRAMA DO MODELO II. MANEJO DOS ANIMAIS

Nesta seção introduzimos o algoritmo do programa que representa o Modelo II.

Figura 4. Algoritmo do modelo 2

5. RESULTADOS DO MODELO I: GANHO de PESO

O Modelo I teve como objetivo principal testar o efeito das DEPs parental (diferença esperada na progênie) estimada em um rebanho bovino real (Sumário de Touros Nelore CFM, 2003), sobre **ganho de peso** simulado em três fases distintas de crescimento (nascimento, desmame e sobreano).

Para testar a robustez do programa de simulação e a influência da DEP na população, fez-se o experimento aplicando três touros distintos de número de progênie elevada e com DEPs conhecidas. Dentre os possíveis touros com um número elevado de progênie, foram selecionados o touro **Regato**: que representa o animal com a melhor DEP ao nascer; o touro **Diamante**: melhor DEP ao sobreano e melhor DEP ao desmame e, finalmente, o touro **Tietê**: maior progênie e portador de DEPs nas três fases de crescimento com valores moderados. Os touros, números de descendentes na base de dados de grupos contemporâneos safra 2003 e suas estimativas de DEPs podem ser observados na tabela 1.

Nome do Touro	Nº de filhos no ano de 2003	DEP ao Nascer	DEP ao Desmame (205 dias)	DEP ao Sobreano (550 dias)	DEP do Ganho de Peso ao Sobreano (em 345 dias)
Diamante	1117	1,19	15,04	26,67	11,63
Tietê	3925	1,39	8,10	14,29	6,19
Regato	1049	0,84	6,24	7,45	1,21

Tabela1. Dados fornecidos pelo GMA - Grupo de Melhoramento Animal -USP-Pirassununga

Acredita-se que as simulações possam gerar uma base de dados de ganho de peso bovino semelhante ao real, e que os valores médios simulados possam auxiliar na decisão de compra de sêmen de um determinado touro do Sumário de Touro CFM 2003, entre outras aplicações discutidas posteriormente.

A aplicação de metodologias estocásticas no crescimento e engorda de bovinos de corte de raças zebuínas brasileiras, vem sendo trabalhada de diferentes maneiras, via de regra com a aplicação de equações como Von Bertalanffy (SAKAGUTI, 2003; DUARTE, 1975) ou de Brody (DUARTE, 1975). Estas duas equações originam-se da seguinte equação geral:

$$y_t = A(1 - Be^{-kt})^M \quad (i)$$

Onde:

y_t = o peso no tempo t ;

A = estimativa assintótica de ;

B = intercepto com o eixo y_t ;

y_0 = peso inicial;

K = taxa de variação da função exponencial;

M = constante que exhibe a forma da curva;

t = tempo.

Na equação de Brody, os parâmetros $B=M=1$ e segundo DUARTE (1975), ela é mais precisa em determinar o intercepto observado em dados reais. Já a equação de Von Bertalanffy (SAKAGUTI, 2003; DUARTE, 1975) com $M = 3$ é mais eficiente no geral em curvas de crescimento.

A proposta deste trabalho é simular os dados de ganho de peso a partir de dados reais de médias de peso do rebanho e efeitos genéticos dos reprodutores, diferente dos citados anteriormente, que visavam ajustar ou determinar a melhor curva que aproxima os dados de ganho de peso em bovinos de corte de raças zebuínas brasileiras (SAKAGUTI, 2003; DUARTE, 1975).

A metodologia empregada foi Simulação Monte Carlo (SMC), onde o ganho de peso diário (GPD) é variável e estocástico para cada animal i em cada PMC e em cada fase de crescimento (nascimento, desmame e sobreano) como descrito no Material e Método (ver equações (i) a (iv)).

O GPD, como previsto, aconteceu a cada passo Monte Carlo (PMC) onde um PMC equivale a um dia; portanto, o animal ganhou (ou eventualmente perdeu) peso a cada PMC.

5.1 O EFEITO DO AJUSTE DO GANHO DE PESO

Para o efeito de auxílio no entendimento dos resultados, foi introduzido um esquema do compartimento de ganho de peso (figura 5) do rebanho.

Figura 5. Evolução das fases de crescimento do rebanho.

Conforme já mencionado, os animais nascem em dias diferentes, correspondentes ao intervalo [201,261], ou seja, durante 60 dias e equivalente aos dias 21 julho até 18 de setembro de cada ano. Sendo assim de maneira análoga ao que acontece no rebanho real, foi necessário corrigir o ganho de peso para cada animal i em cada estado E_i , como descrito na equação abaixo:

$$P_{aj} = \frac{\Delta * P_s}{DatE_i - DatN} \quad (i)$$

Onde;

P_{aj} = peso ajustado;

P_s = peso simulado;

$\Delta = I_{i+1} - I_i$;

$DatE_i$ = data de mudança do estado i ;

$Dat N$ = data de nascimento.

Figura 6. Efeito do ajuste para ganho de peso no estado desmame aos 240 dias aproximadamente.

Pode-se observar na figura 6 que o ganho de peso simulado é regido por uma distribuição Normal como ocorre na natureza. A variação no ganho de peso, observada nos histogramas, do ponto de vista biológico, pode ser explicada pela variação fenotípica que inclui o efeito genético (genótipo) mais o efeito do ambiente. No modelo, esta variação é gerada por uma perturbação no ganho de peso diário, produzido por uma variável aleatória de distribuição normal (*harvest*), multiplicado por uma constante Δ_i que gera uma amplitude grande no ganho de peso diário (até seis vezes o valor do ganho de peso diário médio). Esta variação permite que um animal ganhe peso e eventualmente perca peso em um determinado dia (PMC) assim como ocorre na natureza. Como a variável aleatória tem média zero, as perturbações não afetam o ganho de peso médio da população e como os efeitos são diários, a variação do peso final do animal em uma determinada fase (desmama ou sobreano) tem uma amplitude bastante reduzida em comparação ao ganho de peso diário.

Durante a construção do modelo matemático, havia a possibilidade de promover a mudança de estado quando os animais atingissem um certo número de dias (PMC); entretanto, optamos pela mudança em um único dia determinado para cada estado E_i como ocorre em fazendas de criação extensiva de bovinos de corte (desmame em lotes). Esta estratégia de desmame tem a função específica de facilitar o manejo. Daí, à semelhança do que ocorre nas propriedades que fazem controle do rebanho, a importância do ajuste dos dados.

Na figura 6 é também exposto que o ajuste dos pesos levou a uma diminuição do peso médio ao desmame, comparado com os dados não ajustados. Houve também uma modificação importante na distribuição dos pesos, observada quando o desvio padrão da média (DP) é analisado. O ajuste dos dados resulta em um DP = 26,7 e média = 189,8 para a progênie simulada do touro Regato. Os dados ajustados são, portanto, mais precisos do que os gerados sem correção (DP = 30,61 e média = 195,58) e os histogramas dos dados ajustados se aproximam melhor da gaussiana.

Os dados ajustados obtidos nesta simulação foram validados posteriormente neste trabalho com o cálculo da média e DP reais da prole deste mesmo animal (touro Regato; média = 189,96 e DP = 26,94).

O DP (30,61) dos dados simulados não ajustados é maior devido à variação da idade no desmame de cada animal i decorrente da variação no PMC do nascimento que pode variar no intervalo de [201, 261] dias.

5.2 O EFEITO DA DEP DE PESO AO NASCER, AO DESMAME E AO SOBREANO.

Os animais nascem em um período de 60 dias, sendo que o momento do nascimento é aleatório e esta variação, determinada pela probabilidade de nascimento diária (determinada pela taxa de nascimento no período/ tempo de duração da EM).

Como condição inicial do programa, foi introduzido o peso médio ao nascer da população, com uma variação aleatória para cada animal. Esta variação foi regida de maneira estocástica por uma distribuição Normal de média zero e variância um, multiplicada pelo valor médio estimado do DP da população real na fase de nascimento.

Tendo em vista que o peso ao nascimento é uma medida determinada imediatamente após o nascimento, não há necessidade de ajuste.

Foram avaliadas diferenças entre o peso ao nascer dos três touros mencionados anteriormente: **Regato**, **Tietê** e **Diamante**. Os dados simulados foram, então, contrastados com os dados de uma população real de filhos de cada touro. Na população real da CFM em 2003, o touro Regato possuía 1049 filhos; o Tietê 3925 filhos e o touro Diamante, 1117 filhos.

Figura 7. Dados de peso ao nascer, ao desmame e ao sobreano de uma população real (A, C, E) e simulada (B,D,F) de filhos do touro Regato aplicando o efeito DEP nas três fases de crescimento.

Para fins de comparação, foi selecionado um número de dados simulados equivalente ao número de filhos dos touros cadastrados na base de dados reais.

A simulação aplicando as DEPs do touro Regato gerou a melhor média ao nascer (31,55; figura 7B), ou seja, o menor valor quando comparado aos demais touros. A simulação com os demais touros, Tietê e Diamante, possibilitou médias de peso ao nascer da suas proles de 32,52 (figura 8 H) e 31,89 (figura 9 O), respectivamente. Em todos os casos, os desvios

padrões foram próximos tanto para os dados reais quanto para os dados simulados na fase peso ao nascer.

Para validar o modelo foi contrastado os dados reais de média e DP da população de 1049 filhos do touro Regato com os dados simulados. Os resultados utilizados para comparação são apresentados nas figuras: 7A, 7C e 7E para os dados reais e 7B, 7D e 7F para dados simulados.

Em geral, a simulação gerou valores de pesos médios próximos aos esperados nas três fases: nascimento, desmame e sobreano quando comparado à população real. A medida do DP, no entanto, foi semelhante a somente parte dos valores esperados de DP real. Dentre os valores simulados com as DEPs do touro Regato considerados próximos aos reais, estão o DP do peso ao nascer e peso ao desmame. O peso ao sobreano, ajustado para peso aos 550 dias, apresentou um DP mais elevado (44,76) nos dados reais quando comparado aos dados simulados (41,17). Esta medida permite o cálculo do coeficiente de variação (cv) que foi de 15% para os dados simulados e 21% para os dados reais. Portanto, uma diferença de 6% entre as duas populações. Observando o histograma (figura 7 E) é patente uma maior frequência dos dados em torno da média (figura 7E) e uma maior amplitude da gaussiana.

Figura 8. Dados de peso ao nascer, desmame e ao sobreano de uma população real (G, I, L) e uma simulada (H, J, M) de filhos do touro Tietê aplicando o efeito DEP nas três fases de crescimento.

Assim como o caso do touro Regato, a simulação aplicando as DEPs do touro Tietê gerou valores semelhantes à média, comparando dados simulados e reais nas fases: ao nascer, desmame e sobreano. O DP também foi bastante semelhante quando comparado os dados de peso ao nascer e peso ao sobreano entre os dois grupos. O peso ao desmame embora tenha os valores médios aproximados (figuras 4I e 4J) teve uma grande variação no valor do DP dos dados reais (21,84) e simulados (31,03). Os dados anteriores permitiram calcular um cv de 11% para os dados reais e de 15% para os dados simulados, portanto, uma diferença de 4%.

Nos histogramas (figuras 8I e 8J) é possível observar que os dados reais formam uma gaussiana mais alta e estreita e o contrário é observado nos dados simulados.

A explicação para esta variação depende de uma maior compreensão da natureza dos dados reais e também de um exercício de testes dos dados simulados. Obviamente que a variação dos dados simulados é regida pela constante Δ_i que multiplica o número aleatório, sendo assim, a variação da amplitude dos dados gerados é fixa.

Por outro lado, os dados reais sofrem variações diversas, incluindo desde variações de medidas (balança e operador) até do ambiente como temperatura, regime de chuvas, qualidade da pastagem, suplementação mineral ou protéica entre outras e, sendo assim, é mais suscetível a modificações.

Uma característica digna de nota é que, embora não tenha gerado uma variação importante no desvio padrão, os animais descendentes do touro Tietê assim como os do touro Diamante (ver figura 8), apresentaram uma gaussiana mais alta e de menor amplitude. Também é possível observar pelos histogramas (figuras 8 G e I; figura 8 N e P) que os dados simulados se aproximam mais da gaussiana.

Figura 9. Dados de peso ao nascer, desmame e ao sobreano de uma população real (N, P, R) e simulada (O, Q, S) de filhos do Diamante aplicando o efeito DEP nas três fases de crescimento.

A comparação entre os resultados simulados e reais da prole do touro Diamante, assim como dos descendentes do touro Tietê, foram semelhantes com relação à média nas três fases de crescimento. Porém, com relação ao DP, temos variações nas fases desmame e sobreano (figuras 9: P, Q, R e S).

Na desmama, os dados reais apresentaram um cv menor comparado aos dados simulados (11% e 15% respectivamente). O mesmo foi observado na fase ao sobreano onde

os dados reais e os dados simulados apresentaram cv pouco menores que nas medidas anteriores (10,84% e 12,89%, respectivamente). A mesma característica descrita nos gráficos anteriores é observada nos histogramas dos dados simulados (mais baixos e mais largos).

O touro Diamante nos dados simulados apresentou a melhor média ao desmame (202,28), ou seja, o maior valor quando comparado com os outros dois touros, o que indica que os bezerros são mais pesados, portanto, de melhor valor venal para indivíduos que exploram a criação de bezerros (fazendas de cria).

Os filhos do touro Diamante na fase sobreano são em média 6 kg mais pesados do que os filhos do touro Tietê. Quando comparados aos filhos do touro Regato, os filhos do Diamante são 19kg mais pesados. Portanto, o touro Diamante é o melhor para peso ao desmame e ao sobreano, para dados simulados o mesmo ocorre para os dados reais. Em uma população de 1049 animais (número de descendentes do touro Regato na população estudada) esta diferença aos 550 dias representa o total de 18882 kg de peso vivo, equivalentes a 9441 kg de carcaça e de aproximadamente \$ 12.500,00 dólares americanos.

Figura 10. Representação dos dados simulados e reais para as três fases de crescimento: nascer, desmama e sobreano. Observar falhas na distribuição dos dados reais (setas).

Os dados reais apresentam um padrão de distribuição visível na fase de nascimento (possivelmente gerado pela aproximação dos dados ou falta de precisão no equipamento utilizado para tomar as medidas) e, algumas falhas de distribuição podem ser observadas nas fases de crescimento (figura 10). Os dados simulados aparentemente não apresentam este “viés” sugerindo que dados simulados sejam utilizados para a validação de estimativas geradas pelos dados reais.

O ganho de peso para as fases ao desmame e sobreano é regido pela seguinte equação:

$$PE_{ki}(t+1) = PE_{ki}(t) + \mu_i + DEPE_{ji}/n + harvest * \Delta_i \quad (iii)$$

t = tempo t ;

$PE_{ki}(t+1)$ = peso em kg do animal k no estado i e no tempo t ;

$\mu_{desmame} = 0.737$ = ganho médio de peso em gramas do nascimento ao desmame;

$\mu_{sobreano} = 0.345$ = ganho médio de peso em gramas do desmame ao sobreano;

$DEPE_{ji}$ = diferença esperada na progênie do touro j no estado i ;

$harvest$ = número aleatório entre $[-1,1]$ regido por uma distribuição $\sim N(0,1)$;

$\Delta_{desmame} = 1,78$ constante que altera a amplitude da gaussiana ocasionando um DP $\cong 26$;

$\Delta_{sobreano} = 1,68$ constante que altera a amplitude da gaussiana ocasionando um DP $\cong 40$;

n = número de dias do estado

TOURO	PESO AO NASCER SIMULADO	PESO AO DESMAME SIMULADO	PESO AO SOBREANO SIMULADO
REGATO	31,65	189,8	295,6
TIETÊ	32,52	199,44	309,1
DIAMANTE	31,89	202,28	315,7

Tabela 2. Representação das médias obtidas pelos dados simulados nas três fases de crescimento: nascimento, desmame e sobreano (550 dias).

Na tabela 2 estão resumidos os resultados obtidos no modelo 1. Conforme demonstrado na tabela, o modelo matemático proposto foi robusto para as fases de crescimento, nascimento, desmame e sobreano para os três touros: Regato, Tietê e Diamante, gerando resultados aproximados aos reais e funcionando como uma ferramenta de experimentar touros com diferentes DEP nas três fases de crescimento.

Finalmente, o “*software*” proposto no Modelo I pode ser aplicado como “*software*” educativo (PERRY e SMITH, 2004), onde os alunos de cursos como: Zootecnia, Veterinária, Agronomia e outras áreas da Biologia, assim como técnicos e produtores rurais podem gerar dados para análises ou testar o efeito da aplicação do melhoramento genético na bovinocultura.

6. RESULTADOS DO MODELO II: MANEJO dos ANIMAIS

O Programa de manejo de animais foi constituído de maneira que permite simular todo o ciclo de produção e reprodução de uma propriedade de manejo extensivo de cria e recria de bovinos da raça Nelore (condição mais freqüente no Brasil tropical).

O ciclo de produção de bovinos envolve várias categorias de animais (denominados de estados neste trabalho). A partir de bezerras, por exemplo, esta categoria evolui até atingir uma condição cíclica, como o caso de vacas ciclando que se tornam e em seguida paridas (geram um bezerro ou bezerra) e podem novamente retornar a uma das condições anteriores.

Tendo em vista todas estas categorias (estados) e as variáveis condicionais que regulam a passagem de um estado para o outro, este estudo foi analisado de maneira sistemática para a princípio testar a robustez do modelo.

O número total de animais, no primeiro dia (passo Monte Carlo) apresentou a seguinte condição inicial: 4500 vacas ciclando e, com o passar do tempo e segundo a EM, foi distribuído aproximadamente dentre os outros estados: vaca ciclando, prenhe e parida.

Inicialmente foi analisado o comportamento dos animais de todas as categorias presentes na simulação. Para tanto foram simulados 3650 passos Monte Carlo, o equivalente a 10 anos e 7300 passos, o equivalente a 20 anos.

Nesta fase inicial, foi constituída empiricamente uma população que atinge o equilíbrio e utilizando como variáveis (parâmetros biológicos) condições que são encontradas em propriedades que possuem bom manejo sanitário, reprodutivo e nutricional na cria e recria de bovinos de corte.

Na seqüência, para fins de testar o modelo, foram experimentados mediante alteração, probabilidades de mudança de estados e de outras medidas ligadas ao tempo e ciclo da estação de monta fatores como: fertilidade, anestro pós-parto, taxa de mortalidade, puberdade e taxa de natalidade. Os resultados discutidos são relativos às alterações da evolução dos estados ocorridas dentro de cada rebanho: rebanho das fêmeas e machos na matriz $MV(i,j,k)$ todos sob efeito da estação de monta (EM) pré-estabelecida embora passível de alteração no modelo.

O número de animais de cada estado esteve vinculado à probabilidade de mudança de estado, às variações periódicas do estoque de cada grupo e sob controle cíclico estabelecido pela EM. Esta última de grande valor para controle de diferentes aspectos biológicos na fase

de cria de bovinos de corte. “A concentração das atividades de manejo otimiza o desempenho geral e facilita a administração da atividade pecuária” (AMSTALDEN; ABREU, 2004).

A EM foi introduzida no modelo como um fator fixo, pois, segundo Amstalden e Abreu (2004) o manejo controlado é chave principal para a produtividade e desempenho do rebanho. Eles afirmam que o ideal é que a vaca de corte emprenhe, leve a gestação a termo, crie e desmame um bezerro dentro de um intervalo de 12 meses. Para que isto ocorra, o intervalo entre a concepção e reconcepção (tempo de anestro) não pode exceder 80 a 90 dias.

Utilizamos o método Simulação Monte Carlo (SMC) para simular a variabilidades de mudanças de estados dos animais dentro de cada rebanho MV.

6.1 O EFEITO DE REPRODUÇÃO DENTRO DE MV

Para efeito de um melhor entendimento foi introduzida a figura 2 que representa a evolução dos estados do rebanho das fêmeas e machos.

Figura 2. Evolução compartimental do Sistema Geral de Manejo dos Animais.

6.1.1 COMPORTAMENTO GRÁFICO DAS FASES DE CRESCIMENTO AO LONGO DO TEMPO

Ao introduzir a estação de monta no rebanho é desencadeado no sistema, um efeito de ciclicidade nas fases de crescimento dos machos e fêmeas assim como na reprodução que são as mudanças de fase de vaca ciclando para prenhe e de prenhe para parida. Estes efeitos podem ser observados na figura 11 e 12 ao longo de 10 anos.

Figura 11. Simulação Monte Carlo ao longo de 10 anos das fases de crescimento do rebanho dos machos (bezerro tornar-se desmamado que torna-se sobreano).

O efeito cíclico da estação de monta é observado nas fases de crescimento dos machos: bezerro, desmamado e sobreano.

O estado desmamado possui um ruído na ciclicidade. Este ruído acontece devido a desmama em duas ocasiões diferentes (aos dias 76 e aos dias 120 de cada ano) correspondendo a 16 de março e ao 1^o de maio.

Os patamares observados no estado desmamado correspondem ao tempo em que o animal permanece neste estado. Estes patamares(Δ s) são também observados nas demais fases de crescimento: bezerra e sobreano. No último gráfico da figura 11, embora não seja o objetivo do teste demonstra o acumulado do número de animais vendidos nos 10 primeiros anos da propriedade.

Figura 12. Simulação Monte Carlo ao longo de 10 anos das fases de crescimento do rebanho das fêmeas (bezerra ao tornar-se desmamada que torna-se púbere e torna-se vaca ciclando ou prenhe) e dos efeitos de reprodução (vaca ciclando ao tornar-se prenhe que torna-se parida)

A ciclicidade é caracterizada nas fases: vaca ciclando, prenhe, parida, bezerra, desmamada e púbere. Ela se manteve conforme esperado em 10 (figura 12) ou 20 anos (dados não apresentados) de simulação.

Os patamares (Δ s), assim como no rebanho macho, são freqüentes nos gráficos do rebanho fêmea. Picos (ϕ s) também são observados na evolução de crescimento das fêmeas.

No estado parida, embora, os animais permaneçam por 75 ± 21 dias nesta condição não é observado Δ . A presença de ϕ s, neste caso, ocorre devido ao intervalo de re-concepção que acontece na estação de monta. Tendo em vista que a gestação tem uma duração de 293 ± 12 dias e que a estação de monta está com 80 dias, quando a última vaca dá a luz, as primeiras já estão no estado prenhe. Na figura 13 pode-se observar a comparação dos gráficos de frequência diária e de um contador acumulador zerado anualmente.

Sítios vazios são um artifício computacional, mas em simulações que desejam estudar densidade populacional e distribuição da população em regiões específicas, podem ser úteis. Nesta fase do programa, não há nenhuma simulação destas características e, sendo assim elas podem ser implementadas em outras fases de desenvolvimento do programa.

Também são estimados os efeitos acumulativos das mortes, descartes e vendas observados na figura 2 e 3, obtendo daí um histórico destes fatores que geram prejuízo (descapitalização) e renda (líquida) ao longo de 10 anos.

As demais informações contidas nas figuras 11 e 12 serão discutidas oportunamente no momento da discussão das características de fertilidade.

Figura 13. Simulação Monte Carlo ao longo de 20 anos do estado de vaca parida. Na linha (vazada) estão os dados derivados do contador acumulador zerado anualmente e, em negro (área cheia), estão os números de animais estimados diariamente (contador zerado a cada PMC).

6.1.2 FERTILIDADE

6.1.2.1 TAXA DE GESTAÇÃO ou CONCEPÇÃO (p_1 p_2 e p_3)

A fertilidade é reconhecidamente um dos componentes de maior importância para produção de gado de corte (AMSTALDEN; ABREU, 2004). Durante a execução deste trabalho, foi testado o efeito de se manipular fatores ligados à reprodução dentro do rebanho MV, variando as taxas de probabilidades p_1 p_2 e p_3 como se observa na figura 14 abaixo. Obviamente, não se trata de um estudo aprofundado sobre efeito de manipular as características reprodutivas ou variar os parâmetros, mas de uma maneira de validar o modelo.

Figura 14. Efeito da fertilidade (p_1 p_2 e p_3) do Rebanho das Vacas (MV).

Para testar o efeito de fertilidade variamos as probabilidades p_1 , p_2 e p_3 representadas na figura 15; as demais probabilidades foram mantidas constantes nos três experimentos.

Figura 15. Efeito de manipulação das taxas de gestação dos compartimentos vaca ciclando, parida e novilha púbere do rebanho das fêmeas.

Nos três casos descritos na figura 15, o número de vacas paridas é igual ao número de vacas prenhez, menos as mortes (dados dos contadores acumuladores anuais não mostrados). Para os estados, prenhe e parida, à medida que cresce o número de animais de um estado decresce, o outro e a razão prenhe/ciclando é indicativa da fertilidade do rebanho (ou seja, quanto maior, melhor).

Contrastou-se a diferença entre o melhor resultado, o de fertilidade alta onde $p_1 = 85\%$, $p_2 = 70\%$ e $p_3 = 50\%$ com o pior $p_1 = 60\%$, $p_2 = 50\%$ e $p_3 = 30\%$, observou-se o número de vacas prenhez, no pior caso, representado pela figura 15. A é em torno de 500/animais prenhe ano e aumentou o para 1700 animais/ prenhe ano representado pela figura 15 B.

6.1.2.2 PUBERDADE

Dado os vultosos investimentos iniciais para estabelecer um sistema de produção de bovinos (compra de terra e matrizes) e o aproveitamento da mão-de-obra, um criatório tem por meta manter um número máximo de animais (manejo ótimo) comportados pela propriedade. Sendo assim, a quantidade de animais após o equilíbrio no sistema deve estar próximo de uma constante. Para fazendas que realizam a cria, todos os anos há o nascimento de um grande número de animais o que leva a um conseqüente descarte de novilhas. A maneira como este descarte e esta reposição é realizada, determina a velocidade como um determinado rebanho pode ser substituído durante os anos. Esta taxa depende da reposição pelas novilhas púberes.

Neste modelo foram estabelecidos uma série de critérios sobre os descartes de animais, que, para efeitos de simulação, estão ligados à reprodução onde parte dos animais é descartada com fim de manter a taxa de desfrute na propriedade. Para manter relativamente constante o número de matrizes, foram introduzidas duas variáveis que permitem selecionar somente parte dos animais que não conceberam em uma determinada EM (Figura 2). Estes animais seriam, então, poupados do descarte por razões ligadas, por exemplo, à sua qualidade fenotípica ou de sua prole. Para reposição do rebanho das vacas são utilizadas as novilhas púberes.

A figura 16 representa a dinâmica da população de novilhas púberes. Trata-se, como mencionado anteriormente, de uma população importante. A idade à puberdade que foi imposta por uma distribuição Normal (TESS; KOSTALD, 2000a ; TESS; KOSTALD, 2000b) de média 540 e DP = 40 representado pelo histograma da figura 16. O tempo necessário para

as novilhas passarem ao estado púbere é hoje muito estudado (característica conhecida como precocidade sexual; ELER et al., 2002), especialmente no zebu, conhecido por apresentar um tempo prolongado de anestro (NOGUEIRA, 2004). O modelo descrito neste trabalho se presta para os estudos destas características já que permite modelar os o momento da puberdade.

Figura 16. Dinâmica da puberdade no modelo. O histograma representa a idade dos animais no momento da puberdade. Do lado direito estão descritos os dados de média, DP máximo e mínimo. O gráfico superior esquerdo mostra o acumulado de novilhas que atingem a puberdade no período de um ano e o superior direito os estoques diários de novilhas púberes.

6.1.3 ANESTRO PÓS-PARTO

Dentre as características reprodutivas mais importantes em propriedades rurais que desenvolvem cria, o tempo de anestro pós-parto é determinante. Esta característica é especialmente importante em vacas zebras, que apresentam um tempo prolongado de anestro na fase de lactação. Para efeito de compreender os eventos da variação do tempo foram simuladas algumas variações do tempo médio de retorno ao cio. Tendo em vista a duração da gestação e o intervalo entre as estações de monta, a vaca parida, para poder engravidar no ano seguinte, tem que voltar ao ciclo estral antes do final da estação de monta. No caso de animais

que conceberam no início da EM, o parto ocorre com certa antecedência em relação à EM subsequente; porém animais que conceberam no fim da EM devem ciclar rapidamente para ter a re-concepção no mesmo ano. Segundo Amstalden e Abreu (2004) o período da concepção e re-concepção não pode exceder 90 dias.

Para testar a capacidade do modelo em simular estas características foram selecionados 3 períodos médios distintos de anestro pós-parto: 75, 96 e 117 com uma variação de 21 dias distribuídos de maneira normal. Tendo em vista que as vacas paridas que não voltam a ciclar a tempo da EM são descartadas nesta configuração. O efeito da vaca que não retornou ao estro pode ser medido pelos estoques de paridas ao longo dos anos. Os resultados indicam que existe um grande efeito do tempo de anestro pós-parto ao longo dos anos (figura 17). E que este efeito tem início no segundo ano do modelo quando o período de anestro médio é de 96 dias, mas é mais importante quando este anestro se prolonga por outros 21 dias. A partir de 117 dias não há diferença desde que não ultrapasse o período de 293 dias.

Figura 17. Efeito de tempo de anestro pós-parto nos compartimento, parida do rebanho das fêmeas. As medidas são o total de vacas paridas acumuladas em cada ano.

6.1.4 TAXA DE NATALIDADE (p_1 ; p_2 ; p_3 e p_7)

Os nascimentos no modelo ocorrem nas camadas $k = 2$ e 4 , ou seja, no rebanho das fêmeas e machos com uma probabilidade p_7 que via de regra é 50% e indica a relação macho/fêmea. Embora esta seja uma regra na natureza durante a estruturação deste programa esta variável foi instituída para fins de testar as novas tecnologias de sexagem de sêmen ou embriões disponíveis no mercado.

Do ponto de vista prático as variáveis que estão alterando a natalidade estão ligadas à fertilidade do rebanho e como mencionado anteriormente as variáveis p_1 , p_2 e p_3 (figura 2 e 14) são as mais importantes.

Na figura 18 estão representados os estados de bezerros e bezerras com variação nas taxas de fertilidade do rebanho. Conforme esperado não houve diferença entre o número de bezerros e bezerras. Rebanhos com alta fertilidade geram um número de animais maior comparado ao rebanho de menor fertilidade.

Figura 18. Efeito da fertilidade sobre a natalidade nos rebanhos de machos e fêmeas.

6.1.5 TAXA DE MORTALIDADE

A taxa mortalidade no modelo proposto ocorre em cada estado de ambas as matrizes e sendo assim, pode ser alterada no rebanho das fêmeas individualmente pela simples alteração da probabilidade. Para testar o efeito da mortalidade sobre a população simulada, foram estabelecidas duas condições: uma considerando uma taxa baixa de mortalidade (2% anual para as categorias desmamados em diante e 5% para a fase de desenvolvimento pré-desmama) e outra com uma taxa elevada (5% para as fases pós desmama e 8% para a fase pré desmama). A figura 19 ilustra o efeito da alteração das probabilidades de morte em cada uma das fases. No modelo de baixa mortalidade as perdas totais ao longo de 10 anos foram de 1050 animais na matriz das fêmeas e 339 animais na matriz dos machos o que representou uma mortalidade média de 3,6% para as fêmeas e 3,3% para os machos. Trata-se obviamente uma mortalidade considerada muito baixa e característica de fazendas de manejo muito bom. Quando foram determinadas probabilidades mais altas de mortes houve uma mudança significativa do número de animais mortos. As probabilidades selecionadas foram de 5% para os animais pós-desmama e 8% para os animais mamando. Trata-se de uma taxa de mortalidade elevada, mas freqüentemente observada em propriedades de manejo pouco controlado. Em média 7,7% das fêmeas e 5,1% dos machos morreram no período de um ano somando 2040 fêmeas e 441 machos.

Figura 19. Efeito da taxa de mortalidade (p_5 , p_6 e p_7) do Rebanho das Vacas (MV).

7 CONCLUSÕES

O modelo computacional favorece estudos de variações em diversas características produtivas e reprodutivas, como estação de monta, fertilidade, tempo de anestro, taxa de mortalidade e taxa de natalidade entre outras.

O programa, com o método de MC permite o estudo da dinâmica de produção de animais de maneira similar ao que acontece na realidade.

Com a adição de dados na matriz tridimensional é possível simular animais introduzindo variáveis como idade, e outras características produtivas como o peso nas diversas idades submetido a variação gerada pelas diferenças esperadas na progênie DEPs.

Estes modelos, embora bastante simples, mostraram a possibilidade de simular de maneira bastante efetiva uma população de animais de corte nas condições de criação brasileiras.

CAPÍTULO II. A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS (RNAs) NA AVALIAÇÃO E NA PREDIÇÃO DE VALORES GENÉTICOS PARA GANHO DE PESO EM BOVINO DE CORTE DE RAÇAS ZEBUÍNAS BRASILEIRAS

RESUMO

As Redes Neurais Artificiais (RNAs) surgiram das pesquisas na área de inteligência artificial, onde o objetivo central era desenvolver ferramentas capazes de desempenhar funções complexas usando métodos baseados em aprendizado de máquinas.

Na agricultura RNAs vem sendo utilizadas em aplicações como predição, classificação automática e identificação de características genéticas em bovinos: como ganho de peso e lactação, por exemplo.

Este trabalho teve como objetivo testar a viabilidade de aplicar RNAs do tipo perceptron multi-camadas (MLP) para o estudo de avaliação genética para a característica ganho de peso em bovinos de corte e comparar esta metodologia com o método de melhor preditor linear não viesado (BLUP – do inglês *Best Linear Unbiased Predictor*) gerado por meios do *software* MTDFREML.

As características estudadas para este experimento foram: peso ao nascimento (PESNAS); peso ao desmame (PESDES); peso ao sobreano (PESSOB) e ganho de peso do desmame ao sobreano (GP345). Para os pesos ao desmame foram considerados registros na faixa de valores de idade entre 140 e 270 dias, enquanto para os pesos ao sobreano foram considerados registros na faixa de valores de idade entre 450 e 650 dias.

Estes dados de pesos foram utilizados como entradas na RNA-MLP. Como se trata de uma rede supervisionada utilizamos como *target's* o valor genético obtido pelo BLUP-MTDFREML da característica GP345 para cada animal.

O algoritmo utilizado para o treinamento da RNA-MLP foi o *Backpropagation*. A performance do RNA foi comparada com a do modelo BLUP e resultou em uma boa correlação entre as duas. Em geral, os resultados apresentados neste trabalho indicam o potencial das RNAs na avaliação genética e justificam a continuidade desta linha de pesquisa.

1. INTRODUÇÃO

Muitas técnicas foram desenvolvidas para lidar com grandes quantidades de dados na tentativa de produzir processos preditivos. Redes Neurais Artificiais (RNAs) representam um exemplo de uma classe destes classificadores, que pretendem estimar valores baseados em aprendizado prévio, utilizando uma base de conhecimento.

Na área de modelagem e melhoramento animal, o modelo animal desenvolvido por Henderson em 1949 foi a princípio dispensado. Não porque tratasse de um modelo ruim, mas porque não havia, na época, recursos computacionais disponíveis para desempenhar as funções computacionais necessárias para, por exemplo, inverter grandes matrizes (HENDERSON, 1949).

A partir dos anos 80 iniciou-se a utilização do método de máxima verossimilhança restrita (REML) que é o mais utilizado atualmente (VAN MELLIS, 2001).

E, mesmo com a implementação de uma metodologia para computar o inverso da matriz de parentesco (HENDERSON, 1976), ainda é complexo implementar estas técnicas em aplicações práticas, devido em grande parte às grandes demandas computacionais e o tamanho do banco de dados.

Ultimamente as Redes Neurais Artificiais (RNAs) vêm sendo aplicadas de forma crescente em modelagem para predições de períodos de lactações em bovino leiteiro, mostrando melhoras significativas nestas predições comparado a outros métodos utilizados correntemente (WADE et al, 1994; LACROIX et al, 1997, SALEHI et al, 1998, SALEHI et al, 2000, KOMINAKS et al., 2002).

Neste trabalho utilizamos uma rede neural artificial (MLP) Perceptron de Múltiplas Camadas (do inglês *MultiLayer Perceptron*) (FAUSETT, 1994; HERTZ et al, 1995; HAYKIN, 2001; LIPPMANN, 1987), cuja característica básica reside no fato da rede possuir várias camadas de neurônios conectados entre si. O algoritmo utilizado para o treinamento da RNA-MLP foi o *Backpropagation* (RUMELHART, 1986).

O estudo utilizando RNA-MLP com algoritmo de *Backpropagation* foi tratado por SALEHI et al em 1998 com o objetivo de investigar diferentes valores para os seguintes parâmetros: taxa de aprendizado, número de neurônios nas camadas escondidas da rede,

dentre outras, para visando a predição de período de lactação em bovino leiteiro. Os resultados mostraram que ao alterar os parâmetros da rede obtem-se uma performance considerável no aprendizado da rede.

Seguindo estes estudos foi decidido testar a capacidade de classificação de padrões da RNA-MLP através de variáveis genéticas com algoritmo de *Backpropagation* em bovinos de corte da raça Nelore.

2 OBJETIVO

O objetivo principal deste trabalho foi usar uma rede neural do tipo MLP para classificar ou rankiar bovinos de corte da raça Nelore sob condições brasileiras através de variáveis genéticas.

Utilizou-se as seguintes variáveis genéticas: ganho de peso dentre as fases de crescimento (nascimento ao desmame, do desmame ao sobreano), somente dos filhos relacionandos com o valor genético do ganho de peso do desmame ao sobreano (GP345) dos filhos obtidos pelo BLUP.

3 JUSTIFICATIVA

Apresentar uma alternativa baseada na metodologia RNA-MLP treinada com algoritmo de *Backpropagation* para classificar valores genéticos.

Diminuir o tempo computacional gasto em análises genéticas feitas por modelos como o BLUP pelo “*software*” MTDFREML com o banco de dados suficientemente grandes.

4 MATERIAL E MÉTODOS

Neste trabalho, utilizou-se uma rede neural artificial (RNA) do tipo perceptron em múltiplas camadas (do inglês *Multilayer Perceptron* - MLP) (FAUSETT, 1994; HERTZ *et al*, 1995; HAYKIN, 2001; LIPPMANN, 1987). O algoritmo utilizado para o treinamento da RNA-MLP foi o *Backpropagation* (RUMELHART, 1986).

No primeiro experimento foi aplicado como entrada na rede RNA-MLP as seguintes variáveis genéticas: ganho de peso dentre as fases de crescimento (nascimento ao desmame, do desmame ao sobreano) somente dos filhos e, utilizado como *target* o valor genético do ganho de peso do desmame ao sobreano (GP345) dos filhos obtidos pelo BLUP; este experimento é dependente ou acoplado à saída do BLUP;

Existem dois tipos de RNA: as supervisionadas e as não supervisionadas. A MLP é uma rede supervisionada. Na fase de treinamento, a rede aproxima uma saída a um valor desejado e realiza o algoritmo de treinamento escolhido. Aqui, neste trabalho o escolhido foi o *Backpropagation*.

O algoritmo de *Backpropagation* utiliza o erro quadrático médio ou MSE (do inglês *Mean Squared Error*) para corrigir os pesos da rede quando sua saída for diferente da saída desejada, denominadas pelos *target's*.

$$MSE = \frac{1}{p \times c} \sum_{i=1}^p \|t_i - y_i\|^2$$

Onde:

p = número de dados ou padrões de entrada na rede;

c = número de unidades de saída, descrito mais à frente;

t_i = *target's*, os valores genéticos da característica GP345 ;

y_i = os dados de pesos dos animais ao nascer ao desmame e ao sobreano e GP345 dos filhos.

As características ou variáveis genéticas estudadas foram no primeiro experimento: peso ao nascimento (PESNAS); peso ao desmame (PESDES); peso ao sobreano (PESSOB) e ganho de peso do desmame ao sobreano (GP345). Para os pesos ao desmame foram

considerados registros na faixa de valores de idade entre 140 e 270 dias, enquanto para os pesos ao sobreano foram considerados registros na faixa de valores de idade entre 450 e 650 dias (VAN MELLIS, 2001). O GP345 foi obtido pela função:

$$GP345 = \left(\frac{PESSOB - PESDES}{IDASOB - IDADES} \right) * 345$$

Onde:

GP345 = ganho de peso do desmame ao sobreano em 345 dias;

PESSOB = peso ao sobreano;

PESDES = peso ao desmame;

IDASOB = idade medida ao sobreano;

IDADES = idade medida à desmama.

O treinamento das RNAs é realizado por meio de um método gradiente descendente que consiste na identificação de um mínimo global de uma função complexa (RUMELHART et al., 1986).

Os neurônios recebem uma entrada de valores, aplicam a função de ativação associada com os pesos, e transmitem os valores para a próxima camada até final da rede, ou seja, sua última camada. A função de ativação utilizada neste trabalho pela rede MLP é:

$$f(x_i) = \frac{a}{1 - e^{-bx_i}} + c \quad (i)$$

Onde a, b e c são parâmetros da função, e têm influência no aprendizado da rede (SALEHI et al., 1998) e x_i são os valores de pesos de cada animal i. Esta influência foi estudada na seção 4.1, assim como outros parâmetros como a taxa de aprendizado que compõem o algoritmo de Backpropagation para uma MLP (FAUSETT, 1994; HERTZ et al., 1995; HAYKIN, 2001; LIPPMANN, 1987).

Codificamos a rede MLP com o algoritmo de *Backpropagation* em linguagem Fortran 90.

4.1 ALGORITMO BACKPROPAGATION

Figura1. Fases da rede MLP com o algoritmo de Backpropagation no treinamento.

A figura 1 representa o diagrama de uma rede neural artificial MLP; as setas indicando as fases para frente (*Forward*) ou para trás (*Backward*) representam o procedimento do algoritmo *Backpropagation*, descrito a seguir:

Fase *Forward*:

1. As entradas x_i são apresentadas à primeira camada da rede C^0 ;
2. Para camada C^i a partir da camada de entrada;
 - 2.1 Após os neurônios da camada C^i ($i > 0$) calcularem seus sinais de saída, estes servem como entrada para a definição das saídas produzidas pelos neurônios da camada C^{i+1} ;
3. As saídas produzidas pelos neurônios da última camada são comparadas às saídas desejadas.

Fase *Backward*:

1. A partir da última camada, até chegar na camada de entrada;
 - 1.1 Os neurônios da camada atual ajustam seus pesos de forma a reduzir seus erros;

1.2 O erro de um neurônio é calculado utilizando os erros dos neurônios da camada seguinte conectados a ele, ponderados pelos pesos das conexões entre eles.

Algoritmo *Backpropagation*

1. Iniciar pesos (aleatoriamente pequenos entre $[-1,1]$) e parâmetros (taxa de aprendizado, e os coeficientes da função de ativação);
2. Repita até um erro mínimo desejado (menor ou igual à 0.00001, por exemplo);

2.1 Para cada padrão de treinamento x_i ;

2.1.1 Definir a saída da rede através da fase *forward*;

2.1.2 Comparar as saídas produzidas com as saídas desejadas;

2.1.3 Atualizar os pesos dos neurônios através da fase *backward*.

A taxa de aprendizado é um fator multiplicativo do erro implícito dentro do algoritmo *Backpropagation* (FAUSETT, 1994; HERTZ et al, 1995; HAYKIN, 2001; LIPPMANN, 1987).

4.2 ESCOLHAS DE PARÂMETROS ÓTIMOS DE UMA RNA DO TIPO MLP (MULTILAYER PERCEPTRON)

Foi codificada a rede MLP com o algoritmo de *Backpropagation* em linguagem fortran 90.

Para verificar se a implementação da MLP com algoritmo de *Backpropagation* estava correta, o problema XOR foi testado na rede. Conforme sugere a literatura (FAUSETT, 1994; HERTZ et al, 1995; HAYKIN, 2001; LIPPMANN, 1987), este tipo de rede foi utilizado para solucionar tal problema.

O problema XOR foi testado com diversos coeficientes distintos da função de ativação e também com taxas de aprendizados diferentes. Os resultados dos parâmetros coincidiram com o trabalho de Lippmann em 1987, onde o **alfa** = **0.3** que é a taxa de aprendizado que obteve melhor resultado assim como os coeficientes da função de ativação, **a** = 3, **b** = 1.5 e **c** = 1 (HAYKIN, 2001; LIPPMANN, 1987).

4.1.1 FUNÇÃO DE ATIVAÇÃO

A função sigmóide $f(x_i)$ onde $i=1,...,n$ e n é a dimensão de entrada da rede ou o tamanho da amostra, foi a função de ativação escolhida para o treinamento da rede MLP, utilizada neste trabalho. Ela possui a importante característica de ativar ou inibir, que é uma motivação biológica (HAYKIN, 2001).

Além disso a $f(x_i)$ é não linear, caso contrário, a relação de entrada e saída da rede se reduziria a um perceptron de uma única camada, finalmente para a utilização do algoritmo de treinamento da rede, o *Backpropagation*, $f(x_i)$ deve ser diferenciável em qualquer ponto de seu domínio devido a própria construção do algoritmo (RUMELHART et al., 1986).

Foi feita a seguinte pergunta. As alterações dos parâmetros a , b , e c influem no aprendizado ou não, da rede?

O primeiro estudo é variar os parâmetros: a , b e c da função de ativação mostrado na figura 2. A função de ativação é representada pela seguinte equação (i) já descrita.

Figura 2. Cinco representações da função sigmóide. Observe que a função que obteve melhor resultado segundo a quadro 2 foi a $f1$ que está representado em amarelo no gráfico, já $f2$ ocasionou o não aprendizado da rede.

A figura 2 mostra que para os parâmetros $a = 1$, $b = 1$ e $c = 1$, representados pela função f_3 torna-se $f(x)$ totalmente positiva, o contrário acontece com f_2 para $c = -1$. Já, a função f_1 com os parâmetros $a = 1$, $b = 1.5$ e $c = -1$, têm as propriedades de ser anti-simétrica e de ser ímpar.

Nos problemas em geral (HAYKIN, 2001) a função de ativação que possui as propriedades de ser anti-simétrica e de ser ímpar leva o aprendizado da rede neural do tipo MLP treinada com o algoritmo de Backpropagation. Este resultado é constatado na seção 4.1.2 aplicando o problema XOR.

Portanto, é visto a seguir que a melhor escolha de f é para a função f_1 , pois ocasionou o aprendizado da rede para o problema XOR.

4.1.2 O PROBLEMA DO OU EXCLUSIVO OU XOR

Aplicamos a rede neural MLP treinada com o algoritmo de Backpropagation para o problema da lógica XOR.

O problema XOR é um problema não linear de classificar dois conjuntos representado (0 e 1) pelo quadro 1.

Portanto, por se tratar de um problema não linear, não é separável por um perceptron simples, daí o surgimento da rede MLP

x	y	z
X₁	X₂	Alvo
1	1	0
0	1	1
1	0	1
0	0	0

Quadro 1: Entrada e saída da RNA, para XOR .

No quadro 1 o alvo é conhecido como *target* da rede, ou seja o valor que se espera que a rede aprenda e X_1 e X_2 são as entradas da rede. Os valores do quadro 1 representam o lógica XOR

Testamos o código da rede MLP proposto por este trabalho com o problema XOR, para verificar o programa e experimentar as mudanças de parâmetros na função de ativação e na taxa de aprendizado alfa que compõem o algoritmo de Backpropagation.

Nos quadros 2 e 3 a seguir, temos a seguinte nomenclatura: os coeficientes **a, b** e **c** são compostos na função de ativação, já explicada; **alfa** = taxa de aprendizado; **N. Int.** = número de interações pedidas na condição inicial; **N. Int Alc.** = número de interações que precisamos rodar a rede para alcançar o erro pedido (**Erro Ped.**) ou desejado e se a rede aprendeu (**Erro Alc. < Erro Ped.**) ou não (**Erro Alc. > Erro Ped.**).

a	b	c	alfa	Nº inter.	Nº int. alc.	Erro Ped.	Erro Alc.	Aprendeu?
1	1	-1	0.3	10000	10⁴	0.00001	0.25	não
1	1	1	0.3	10000	10⁴	0.00001	0.25	não
3	1.5	1	0.3	10000	98	0.00001	8.810⁻⁶	sim
3	1.5	0	0.3	10000	10⁴	0.00001	0.027	não
2	0.5	0.25	0.3	10000	10⁴	0.00001	0.4	não

Quadro 2. Variações das constantes a,b e c da função de ativação.

Observa-se pelo quadro 2, a primeira análise, que foram variar as constantes a,b e c, onde verificou-se que a variação: a = 3, b = 1.5 e c = 1 foi a que obteve o melhor resultado, pois as variações restantes implicaram no não aprendizado da rede.

Seguindo este experimento, fixou-se a = 3, b = 1.5 e c = 1 e nestes valores e variou-se o alfa próximo de 0.3, cujos resultados são mostrados no quadro 3.

a	b	c	alfa	Nº inter.	Nº int. alc.	Erro Ped.	Erro Alc.	Aprendeu ?
3	1.5	1	0.3	10000	98	0.00001	8.810^{-6}	sim
3	1.5	1	0.35	10000	10^4	0.00001	0.153	não
3	1.5	1	0.25	10000	108	0.00001	7.310^{-6}	sim
3	1.5	1	0.28	10000	306	0.00001	9.610^{-6}	sim
3	1.5	1	0.32	10000	102	0.00001	4.910^{-6}	sim
3	1.5	1	0.29	10000	106	0.00001	8.310^{-6}	sim
3	1.5	1	0.31	10000	10^4	0.00001	2.310^{-2}	não
3	1.5	1	0.299	10000	84	0.00001	7.410^{-6}	sim

Quadro 3. Variações do parâmetro alfa que compõem à atualização do peso da RNA.

É observado no quadro 3, que o melhor resultado para este experimento foi alfa = 0,299, pois obteve menor número de interações alcançadas (84) juntamente com o menor erro quadrático médio ($7,4 \times 10^{-6}$). Outra observação é que uma pequena mudança no alfa = 0,31 implica no não aprendizado da rede.

Foi analisado também o comportamento do gráfico do erro quadrático médio, características como: se existe ciclicidade ou não, no decaimento do erro, como é a forma do erro de um treinamento que aprende e de um treinamento que não aprende.

Figura 3. Representação do decaimento do erro quadrático médio para dois alfas distintos que implicaram no aprendizado da rede.

É observado na figura 3, que as convergências das duas representações são distintas, as duas implicaram no aprendizado da rede com erro na ordem de 10^{-5} . Porém, quando são comparadas outras características como a forma de decaimento percebe-se que a primeira representação o erro fica “preso” ao redor 0,2 e na 300^a interação ele decai progressivamente atingindo o erro de $9,49 \times 10^{-5}$, já a segunda representação oscilou no início do treinamento em torno da 10^a interação e obteve a convergência bem mais rápida na 80^a interação.

Pode-se afirmar que a representação, com o parâmetro $c = 0,5$ levou uma convergência mais lenta porque localizou um mínimo local mais profundo (HAYKIN, 2001).

Figura 4. Representações de mínimos locais (1,2,3,4) e mínimo global (4) de uma função qualquer $f(x)$.

Os mínimos locais de uma função são representados na figura 4, pelos números: 1,2,3 e 4, porém temos o mínimo global representado somente pelo número 3.

Figura 5. Representação do decaimento do erro quadrático médio para dois alfas distintos que implicaram no não aprendizado da rede.

Como a rede MLP foi treinada para 10000 interações, observa-se pela figura 5, que a escolha de tais parâmetros ocasiona o não aprendizado da rede, pois a intenção é obter o erro na ordem ou menor de 10^{-3} , na primeira representação temos erro = 0,027 e na segunda temos o erro = 0,25.

4.2 APLICAÇÃO DA RNA DO TIPO MLP EM PREDIÇÃO DE VALORES GENÉTICOS PARA GANHO DE PESO

4.2.1 EXPERIMENTO RNA-MLP PARA DADOS DOS FILHOS

Neste experimento cada unidade (ou neurônio) de entrada recebe um sinal que representa o valor da variável de entrada. O algoritmo de treinamento de *Backpropagation* consiste em encontrar uma série de pesos que minimizem o erro global da rede, baseada na comparação da saída e os seus valores alvos correspondentes. Neste trabalho, a entrada de valores é representada pelos dados de pesos: ao nascer (PESNAS); peso ao desmame (PESDES); peso ao sobreano (PESSOB) e ganho de peso do desmame ao sobreano (GP345).

Uma representação esquemática do sistema de treinamento pode ser observada na figura 6. A lei de aprendizado do *Gradiente Descendente* (HAYKIN, 2001) foi aplicada no ajuste dos pesos para a rede. Com esta regra, o termo de correção que ajustou cada peso na RNA foi uma função de ativação de entrada para cada erro e a taxa de treinamento correspondentes.

Figura 6 – Diagrama esquemático das estratégias de treinamento. Base de dados com 5000 animais foram utilizados no modelo BLUP e 100 dados selecionados da base de dados utilizados no modelos de entrada do RNA.

O conjunto de dados de peso do animal foi obtido da base de dados do Programa de Melhoramento Genético da Agro-pecuária CFM Ltda – Brasil. Esta base armazena dados de animais Nelore criados no noroeste de São Paulo e Mato Grosso do Sul. Dados de pesos de 5000 animais foram utilizados para estimar o valor genético do animal aplicando o modelo animal BLUP.

Para o treinamento da rede, um conjunto de dados de 600 animais foi aleatoriamente selecionado da base de dados. Os dados dos 4400 animais restantes foram utilizados para validar o teste. O experimento foi dividido em duas partes:

- No experimento 1 os alvos foram o valores genéticos para PESNAS; PESDES e GP345;
- No experimento 2 usou o valor genético para o GP345 como alvo.

As figuras 7a e 7b mostram o esquema da estrutura utilizada que é a arquitetura da rede.

Figura 7a - Arquitetura da RNA utilizando os quatro valores genéticos PESNAS, PESDES, PESSOB e GP345 como alvos durante o processo de treinamento.

Figura 7b - Arquitetura da RNA utilizando GP345 parâmetro genético como alvo durante o processo de treinamento.

A convergência (no estágio de treinamento) da arquitetura da RNA mostrada na figura 7a não foi atingida. Estes resultados mostram que a configuração da RNA é uma variável importante. A incapacidade de otimizar pode ter sido originada por diferentes problemas como, por exemplo, número de camadas escondidas insuficientes, elementos de processamentos, e calendário de treinamento. No caso de RNAs serem adotadas para uso na prática todas estas variáveis têm que ser avaliadas.

Portanto, utilizamos a arquitetura representada pela figura 7b.

5. RESULTADOS

Figura 8. Valores genéticos preditos utilizando 40 dados de validação.

A figura 8 mostra a comparação dos resultados obtidos com o BLUP (círculos abertos) versus os valores da RNA (círculos fechados).

Os círculos fechados representam as saídas da RNA aplicando a arquitetura da figura 7b depois do processo de treinamento. Na figura 8, a abscissa representa a identificação de animais escolhidos aleatoriamente da base de dados e identificados de 1 à 40. A correlação entre as saídas da RNA e os valores do BLUP estão discriminados na ordenada da figura 8.

Observa-se que há um acompanhamento do padrão gerado pela rede MLP e do resultado do BLUP que consideramos como nosso valor controle, ou seja, os valores verdadeiros.

A diferença entre a RNA e o BLUP pode ter sido gerada porque a RNA aprendeu as variâncias dos dados de entrada que são desprezados no BLUP.

Figura 9 – Correlação entre Saídas da RNA-MLP e do BLUP.

Foi calculado o coeficiente de correlação de Pearson entre as saídas da RNA-MLP e os valores do BLUP representada na figura 9, observa-se que obteve-se uma correlação $R = 0,877$ com um desvio padrão de $DP = 0,024$.

A união dos resultados destas duas figuras mostra um estudo preliminar e viabilidade da utilização da RNA-MLP em predições de ganho de peso nas fases de crescimento descritas, mostrando que pode ter uma correlação entre o ganho de peso de uma fase para outra.

Mas por que comparar o modelo de RNA com o BLUP? O BLUP é um método estatístico geral para avaliação genética e sendo assim constitui uma família inteira de modelos estatísticos.

Ele possui as seguintes características:

1. Produz comparações válidas entre animais;
2. Adiciona níveis de competição entre touros bons ou ruins;
3. Tem bons resultados para tendências genéticas;
4. Tem baixa eficiência em re-ranquiamento de touros em ambientes diferentes;
5. Lida razoavelmente bem com dados de má qualidade;
6. Otimiza razoavelmente o tempo computacional.

As características acima são suficientes para justificar a aplicação de um modelo de RNA. A RNA é mais robusta para variância dos dados e pode ser usada toda a informação disponível para classificar um animal por vez.

É importante ressaltar que este trabalho foi representado um estudo preliminar na área. Muitas opções de melhorias devem ser examinadas uma vez que a habilidade preditiva da RNA foi estabelecida para este processo particular.

Enquanto o método BLUP leva em consideração um número limitado de fatores, é também possível colocar informação nas RNAs que permitam maior representatividade por animal como pedigree, efeitos ambientes, manejo, alimentação, doença, dentre outros ou ainda para cada rebanho (posição geográfica, idade, fazenda, etc...).

6 CONCLUSÕES

A RNA tem um potencial de auxiliar na predição de valor genético em bovinos de corte.

De acordo com o resultado deste estudo, pôde ser possível obter os valores genéticos com a aplicação de RNA e descartando operações de inversão de matrizes grandes. A metodologia de RNA apresentou resultados similares ao BLUP e pode apresentar vantagens de velocidade de processamento quando for necessária.

No futuro, as pesquisas em RNA para predições de características em bovino de corte devem focar a análise da arquitetura ideal, otimização da configuração dos neurônios e aplicação de outros modelos de rede assim como está sendo feito na bovinocultura leiteira.

Apesar da RNA demandarem tempo e esforço para a manipulação dos dados e treinamento das redes, a classificação é rápida e a metodologia pode ser usada como ferramenta valiosa para avaliação genética em bovinos de corte.

REFERÊNCIAS

ABREU, U.G.P.; AMSTALDEN, M. Uso de modelagem para análise de eficiência reprodutiva animal. In: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE ZOOTECNIA, 41., 2004, Campo Grande. **Anais...** Campo Grande: SBZ, 2004. CD-ROM.

ABREU, U.G.P.; CEZAR, I.M.; TORRES, R.A. Análise bioeconômica da introdução de período de monta em sistemas de produção de rebanhos de cria na região do Brasil Central. **R. Bras. Zootec.**, Viçosa, MG, v.32, n.5, p.1198-1206, 2003.

BERRETA, V.; LOBATO, L.F.P.; MIELITZ NETTO, C.G. Produtividade e eficiência biológica de sistemas pecuários de cria diferindo na idade das novilhas ao primeiro parto e na taxa de natalidade do rebanho do Rio Grande do Sul. **R. Bras. Zootec.**, Viçosa, MG, v.30, p.1278-1286, 2001.

BOLDMAN, K.G. et al. **A manual for use of MTDFREML**: a set of programs to obtain estimates of variances and covariances. Lincoln: USDA-Agricultural Research Service, 1993, 120 p.

BOURDON, R.M.; BRINKS, J.S. Simulated efficiency of range beef production. II. Fertility traits. **J. Anim. Sci.**, Savoy, v.65, p.956-962, 1987.

CAVALHEIRO, R. et al. Efeitos de heterogeneidade de variância residual entre grupos de contemporâneos na avaliação genética de bovinos de corte. **R. Bras. Zootec.**, Viçosa, MG, v.31, n.4, p.1690-1688, 2002.

CEZAR, I. Modelo bioeconômico de produção de bovinos de corte: I. descrição do modelo. **Pesq. Agropec. Bras.**, Brasília, v.17, n.6, p.941-949, 1982.

CUNHA, E.E. et al. Efeitos de tipos de acasalamentos e razões sexuais na seleção baseada no BLUP. **R. Bras. Zootec.**, Viçosa, MG, v.32, n.6, p.1297-1303, 2003.

DODENHOFF, J.; VAN VLECK, L.D; GREGORY, K.E. Estimation of direct, maternal, and grandmaternal genetic effects for weaning weight in several breeds of beef cattle. **J. Anim. Sci.**, Savoy, v.77, p.840-845, 1999.

DUARTE, F.A.M. **Estudo da curva de crescimento de animais da raça Nelore (Bos taurus indicus) através de cinco modelos estocásticos**. 1975. 284 f. Tese (Livre-Docência) Faculdade de Medicina, Universidade de São Paulo, Ribeirão Preto, 1975.

ELER, J.P.; SILVA, J.A.; FERRAZ, J.B.; DIAS, F.; OLIVEIRA, H.N.; EVANS, J.L.; GOLDEN, B.L. Genetic evaluation of the probability of pregnancy at 14 months for Nelore heifers. **J. Anim Sci.**, Champaign, v. 80, 951-954, 2002.

FAMULA, T.R; VAN VLECK, L.D. Monte Carlo study of genetic groups in sire evaluation. **J. Dairy Sci.**, Champaign, v.65, n., p.1286-1293, 1981.

FAUSETT, L. **Fundamentals of neural networks:** architectures algorithms and applications. Englewood Cliffs: Prentice Hall International Inc., 1994.

FERRAZ, J.BS; ELER, J.P. **Sumário de touros Nelore 2001**. São José do Rio Preto: Agro- Pecuária CFM, 2003.

FOURNET, F.; HOSPITAL, F.; ELSSEN, J.M. A FORTRAN program to simulate the evolution of genetic variability in a small population. **CABIOS**, v.11, n.5, p.469-475, 1995.

GAMERMAN, D. **Markov chain Monte Carlo:** stochastic simulation for Bayesian inference. London: Chapman & Hall, 1997.

HAYKIN, S. **Redes neurais princípios e prática**. Tradução de Paulo Martins Engel. 2.ed. Porto Alegre: Bookman, 2001.

HEERMANN, D.W; BINDER, K. **Monte Carlo Simulation in Statistical Physics: an introduction**. New York: Springer Verlag, 1997.

HERTZ, J.; KROGH, A.; PALMER, R.G. **Introduction to the theory of neural computation**. Rewood City: Addisom-Wesley, 1995.

KAHN, H.E.; LEHRER, A.R. A dynamic model for the simulation of cattle herd production systems: part 3- reproductive performance of beef cows. **Agricultural Systems**, England, v.13, p.143-159, 1984.

KIRCHNER, K.; TÖLLE, K.H.; KRIETER, J. The analysis of simulated sow herd datasets using decision tree technique. **Computers and Electronics in Agriculture**, Amsterdam, v.42, p.111-127, 2004.

KOMINAKS, A.P. et al. A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep. **Computers and Electronics in Agriculture**, Amsterdam, v.35, p.35-48, 2002.

LACROIX, R. et al. Effects of data preprocessing on the performance of artificial neural networks for dairy yield prediction and cow Culling Classification. **American Society of Agricultural Engineers**, v.40, p.839-846, 1997.

_____. Prediction of cow performance with a connectionist model. **American Society of Agricultural Engineers**, v.38, p.1573-1579, 1995.

LIPPMANN, R.P. An introduction to computing with neural nets. **IEEE ASSP Magazine**, New York, p.1-21, 1987.

MULLIN, T.J.; PARK, S. Stochastic simulation of population management strategies for tree breeding; a new decision-support tool for personal computers. **Silvae Genetica**, Frankfurt, v.44, p.2-3, 1995.

NOGUEIRA, G.P. Puberty in South American *bos indicus* (Zebu) cattle. **Animal Reproduction Science**, Netherlands, v.82-83, p.361-372, 2004.

OIKAWA, T.; SATO, K. Comparison of prediction methodology for binary traits through Monte-Carlo simulation. **J. Anim. Breed.**, Berlin, v.111, p.337-345, 1994.

PERRY, G.A.; SMITH, M.F. A simulation exercise to teach principles of bovine reproductive management. **J. Anim. Sci.**, Savoy, v.82, p.1543-1549, 2004.

PLEASANTS, A.B. Use of a stochastic model of a calving distribution for beef cows for formulating optimal natural mating strategies. **Animal Science**, Scotland, v.64, p.413-421, 1997.

PÖTTER, L. et al. Análises econômicas de modelos de produção com novilhas de corte Primíparas aos dois, três, e quatro anos de idade. **Rev. Bras. de Zootec.**, Viçosa, MG, v.28, p.861-870, 2000.

PRESS, W.H. et al. **Numerical recipes I Fortran 90**: the art of parallel scientific computing. 2.ed. Cambridge: Cambridge University Press, 2002. Disponível em: <<http://www.library.cornell.edu/nr/book90pdf.html>>. Acesso em: 12 nov. 2004.

PROVIDELO, F.D. **Modelo computacional da dinâmica de transmissão da tuberculose**. 2000. 82 p. Dissertação (Mestrado em Ciências) – Faculdade de Filosofia e Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2000.

RUMELHART, D.E.; HILTON, G.E.; WILLIAMS, R.J. Learning representations by back-propagation errors. **Nature**, London, v.323, p.533-536, 1986.

SAKAGUTI, E.S.; MARTINS, E.N.; SILVA, L.O.C. Comparison of different selection criteria in populations simulated under growth curve parameters of Brazilian Zebu cattle. **J. Anim. Sci.**, Savoy, v.81, 2003. Supplement 1.

SALEHI, F.; LACROIX, R.; WADE, K.M. Development of neuro-fuzzifiers for qualitative analyses of milk yield. **Computers and Electronics Agriculture**, v.28, p.171-186, 2000.

SALEHI, F.; LACROIX, R.; WADE, K.M. Effects of learning parameters and data presentation on the performance of backpropagation networks for milk yield prediction. **American Society of Agricultural Engineers**, v.41, p.253-259, 1998a.

_____. Improving dairy yield predictions through combined records classifiers and specialized artificial neural network. **Computers and Electronics Agriculture**, v.20, p.199-213, 1998b.

SANDERS, J.O.; CARTWRIGHT, T.C. A general cattle production system model. I: Structure of the model. **Agricultural Systems**, v.4, p.217-227, 1979.

SOBOL, I. **Método de Monte Carlo**. Moscou: Mir, 1983.

TEDESCHI, L.O.; FOX, D.G; BAKER, M.J.; SAINZ, D.; BARIONI, L.G.; BOIN; C.. Using mathematical models in ruminant nutrition. In: REUNIÃO ANUAL DA SOCIEDADE BRASILEIRA DE ZOOTECNIA, 41., 2004, Campo Grande. **Anais...** Campo Grande: SBZ, 2004. CD-ROM.

TESS, M.W.; KOLSTAD, B.W. Simulation of cow-calf production systems in a range environment: I. model development. **J. Anim. Sci.**, Savoy, v.78, p.1159-1169, 2000a.

_____. Simulation of cow-calf production systems in a range environment: II. model evaluation. **J. Anim. Sci.**, Savoy, v.78, p.1170-1180, 2000b.

VAN MELLIS, M.H. **Utilização do método R na estimação de parâmetros genéticos em gado de corte**. 2002. 85 f. Dissertação (Mestrado em Zootecnia) – Faculdade de Zootecnia e Engenharia de Alimentos, Universidade de São Paulo, Pirassununga, 2002.

VAN VLECK, L.D. Contemporary groups for genetic evaluations. **Journal of Dairy Science**, Champaign, v.70, p.2456-2464, 1987.

WADE, K.M.; LACROIX, R. The role of artificial neural networks. In: WORLD CONGRESS ON GENETICS APPLIED TO LIVESTOCK PRODUCTION GUELPH, 5., 1994, Ottawa. **Proceedings...** Ottawa: [s.n.], 1994. v.22, p.31-34.

WERTH, L.A. et al. Use of a simulation model to evaluate the influence of reproductive performance and management decisions on net income in beef production. **J. Anim. Sci.**, Savoy, v.69, p.4710-4721, 1991.
fzea

APÊNDICES

APÊNDICE A. Código fonte do Modelo I: Ganho de Peso.

```

!-----
!   Last change: FVM 13 Jan 2005   2:20 pm
!-----
!MODELO I. PROGRAMA DE GANHO DE PESO - 2004
!UTILIZANDO O MÉTODO DE SIMULAÇÃO MONTE CARLO
!Programadora: Flávia Deveschio Providelo
!-----
IMPLICIT NONE      !OBRIGA DELARAÇÃO DAS VAR/S
integer, parameter :: LV = 1, LX = 150
integer, parameter :: M = 150, N = 150 !Parâmetros da Matriz do Rebanho
!-----
!Matrizes dos Rebanhos
!MRebM = matriz dos machos que nao sao reprodutores
!MT = matriz dos touros
!-----
      real :: MRebM(LV:LX,LV:LX,0:20)!Matrizes de Interação local
      integer :: i,j,k,sement,passos,PMCT
      real :: RAND,harvest,DEPPN,DEPPD,DEPSob
      character*20 :: nome
!-----
!                                     INICIO DE TODOS OS COMMONS
!-----

!VARIÁVEIS COMUNS PROGRAM PRINCIPAL
      common/arg0/ sement
      common/npassos/ passos,PMCT
!VARIÁVEIS DA SUBROTINA 0 _ que gera números aleatórios
      common/arg1/ rand
!VARIÁVEIS DA SUBROTINA 2 _ que gera números normais
      common/arg3/ harvest
!VARIÁVEIS DA SUBROTINA - MACHO
      common/arg4/MRebM,DEPPD,DEPSob
!-----
!                                     ARQUIVOS DE ENTRADA
!*****
! Semente e o Número de Passos Monte Carlo e
! Probabilidades = Parâmetros de Entrada
!-----
      sement = 11131311
print*, '----- MENU: PROGRAMA DE TENDENCIA GENETICA PARA GANHO DE PESO-----'
print*, 'Digite o nome do TOURO'
read*, nome
print*, 'Digite o numero de dias'
read(*,*)passos
print*, 'Digite a DEP peso ao nascer'
read*, DEPPN
print*, 'Digite a DEP peso ao desmame'
read*, DEPPD
print*, 'Digite a DEP GP ao sobreano'
read*, DEPSob
print*, '-----Fim do Menu-----'
!-----
!                                     ARQUIVOS DE SAÍDA
!*****
open(0,file='pesnascertxt')
open(1,file='pesdesm205.txt')
open(2,file='pessobr550.txt')
open(3,file='pesboigord.txt')

```

```

!-----
! NUMERO DE SÍTIOS NAS MATRIZES
!numvis = M*N  !Número de Vizinhos    M=150 e N=150
! * * * * *
!-----
!
! ZERANDO AS MATRIZES ANTES DE INICIAR A SIMULAÇÃO
!-----
! Matriz do Rebanho dos Machos
do i = LV, LX
do j = LV, LX
do k = 0,8
  MRebM(i,j,k) = 0
enddo
  call gasdev_s()
  MRebM(i,j,2) = 31.4 + DEPPN + harvest*3.7    !31.4=média do peso ao nascer CFM 2001
enddo

enddo
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!PROGRAMA PRINCIPAL
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

!-----LOOPING GERAL DE MONTE CARLO-----
DO 5000 PMCT = 1, passos
!-----SUBROTINA DO REBANHO DOS MACHOS-----
      CALL REBMACHO()
!-----FINAL DO LOOPING GERAL DE MONTE CARLO-----

!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!FIM DO PROGRAMA PRINCIPAL
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!-----
!-----
5000 CONTINUE
!-----AMOSTRA DE 1000 GANHO DE PESO DO REBANHO MACHO-----
!-----
!Indice
!MRebM(i,j,0) = data de nascimento
!MRebM(i,j,1) = estado enquanto é bezerro 0 = VAZIO OU 1 = OCUPADO = NASCIDO
!MRebM(i,j,2) = peso ao nascer
!-----
DO K = 1, N*M
CALL RANDOM()
I = INT(RAND*148.9)+1
CALL RANDOM()
J = INT(RAND*148.9)+1
IF (MRebM(i,j,1).NE.0) THEN
!-----
!      DATA DE NASCIMENTO, ESTADO E O PESO AJUSTADO AO NASCER
!-----
      WRITE(0,*) MRebM(i,j,0),MRebM(I,J,1),MRebM(i,j,2) ! registro=datanasc e estado ocupado
!      WRITE(*,*) MRebM(i,j,0),MRebM(I,J,1),MRebM(i,j,2) ! registro=datanasc e estado ocupado
!-----
!      DATA DE NASCIMENTO, PESO NÃO AJUSTADO AO DESMAME, PESO AJUSTADO
!-----
      WRITE(1,*) MRebM(i,j,0),MRebM(I,J,4),(205*MRebM(I,J,4))/(440-MRebM(i,j,0))
!DESMAME 180<=I<=385 ->205 dias
      WRITE(*,*) MRebM(i,j,0),MRebM(I,J,4),(205*MRebM(I,J,4))/(440-MRebM(i,j,0))
!DESMAME 180<=I<=385 ->205 dias
!-----
!      DATA DE NASCIMENTO, PESO NÃO AJUSTADO AO SOBREANO, PESO AJUSTADO
!-----

```

[illegible]

```
gaus_stored=.false. !and unset the flag.  
else !We don't have an extra deviate handy, so  
do  
call RANDOM() !pick two uniform numbers in the square extending  
!from -1 to +1 in each direction,  
 $v1 = 2.0 * rand - 1.0$   
call RANDOM()  
 $v2 = 2.0 * rand - 1.0$   
 $rsq = v1^2 + v2^2$  !see if they are in the unit circle,  
if ( $rsq > 0.0$  .and.  $rsq < 1.0$ ) exit  
end do !otherwise try again.  
 $rsq = \sqrt{-2.0 * \log(rsq) / rsq}$  !Now make the Box-Muller transformation to  
!get two normal deviates. Return one and  
!save the other for next time.  
harvest= $v1 * rsq$   
 $g = v2 * rsq$   
gaus_stored=.true. !Set flag.  
end if  
END SUBROUTINE gasdev_s  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!  
! INICIO DA SUBROTINA 6 - REBANHO MACHO  
!  
! MRBm(i,j)  
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&  
!-----  
SUBROUTINE REBMACHO()  
!DECLARAÇÃO DAS VARIÁVEIS  
IMPLICIT NONE !OBRIGA DELARACÃO DAS VAR//S  
!-----PARÂMETROS-----  
integer, parameter :: LV = 1, LX = 150  
integer, parameter :: M = 150, N = 150 !Parâmetros da Matriz do Rebanho  
!-----VARIÁVEIS-----  
!-----  
!Matrizes dos Rebanhos  
!MRebF = matriz das femeas jovens que nao reproduzem  
!MRebM = matriz dos machos que nao sao reprodutores  
!MV = matriz das vacas prontas p/ se acasalarem  
!MT = matriz dos touros  
!-----  
real :: MRebM(LV:LX,LV:LX,0:20) !Matrizes de Interação local  
integer :: i, j, index,passos,PMCT  
real :: PNB,DEPPD,DEPSob ! pro's dos machos  
real :: rand,harvest  
!VARIÁVEIS COMUNS - COMMON'S  
!Programa Principal  
common/npassos/passos,PMCT  
!VARIÁVEIS DA SUBROTINA 0 _ que gera números aleatórios  
common/arg1/rand  
!VARIÁVEIS DA SUBROTINA 2 _ que gera números normais  
common/arg3/ harvest  
!VARIÁVEIS DA SUBROTINA - MACHO  
common/arg4/MRebM,DEPPD,DEPSob  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!  
! VARIAÇÃO LOCAL DO SISTEMA DO REBANHO DOS MACHOS  
!  
! INICIO DA SIMULAÇÃO MONTE CARLO NO REBANHO dos MACHOS  
!Analizando a rebanho  
!-----  
!SIMULACAO COMPARTIMENTAL DO REBANHO DE MACHOS  
!-----  
!
```

```

! "TRANSIÇÃO DE ESTADO LOCAL - 1"
!-----
! Sorteio da linha (i) e coluna (j)
!-----
index = int(PMCT/365.0)
do i = 1,150
do j = 1,150
!-----
select case(int(MRebM(i,j,1)))
!-----
!Compartimento 0 - MACHOS --> Nascidos (1 - 150 dias)
!-----
!Sítios Vazios "0" ----> Nascidos "1"
!Nascem machos na proporção p1 = 50%/365.
!MECANISMO 0 Vazio --> Bezerro NASCE NO DIA 180
!-----
case (0)!FAÇO NASCE 50% VIRA ESTADO 1 = OCUPADO EM 180-240
! PNB = natalidade de machos na matriz do rebanho
CALL RANDOM()
PNB = 0.5/60
if(rand.le.PNB.and.PMCT.ge.201.and.PMCT.le.261) then
    MRebM(i,j,1) = 1
    MRebM(i,j,0) = PMCT
endif
!-----
!Compartimento 1 - MACHOS
!-----
! Ganho de peso do estado enquanto o animal e BEZERRO
MRebM(i,j,4) = MRebM(i,j,2)
!-----
case(1) ! Estado Ocupado
if(PMCT.ge.201.and.PMCT.lt.442) then ! [180,240] bezerro
    call gasdev_s()
    MRebM(i,j,4) = MRebM(i,j,4) + (0.74 + DEPPD/205 + harvest*1.85) !bezerro
else
!Muda de estado : DESMAMADO
if(PMCT.EQ.442) then !DESMAMO TODOS NO DIA 441
    MRebM(i,j,6) = MRebM(i,j,4)
    MRebM(i,j,6) = (205*MRebM(i,j,4))/(440-MRebM(i,j,0))
else
    call gasdev_s()
    if(PMCT.gt.442.and.PMCT.lt.789) then! aos 580 dias (sobreano)
        MRebM(i,j,6) = MRebM(i,j,6) + (0.305 + DEPSob/345 + harvest*1.57)! ! SOBREANO
    else
!-----
!Muda de estado : SOBREANO (550)
if(PMCT.eq.789) then
    MRebM(i,j,8) = MRebM(i,j,6) !recebe o peso anterior
! Ganho de peso do estado enquanto o animal e boigordo
else
    if(PMCT.gt.789.and.PMCT.le.1200) then !550
        call gasdev_s()
        MRebM(i,j,8) = MRebM(i,j,8) + (0.5 + DEPSob/100 + harvest*3)! boigordo
    endif
endif
endif
endif
end select
!-----

```

```
end do
end do
!*****
END
!-----FIM DA SUBROTINA - 6 - macho
!-----
```


APÊNDICE B. Código fonte do Modelo II: Manejo.

```

!-----
!   Last change: FVM 12 Jan 2005   4:11 pm
!-----
!
!               MODELO II. MANEJO DOS ANIMAIS
! Programadora: Flávia Devechio Providelo Meirelles
!-----
IMPLICIT NONE      !OBRIGA DELARAÇÃO DAS VAR//S
integer, parameter :: LV = -2, LX = 153
integer, parameter :: M = 149, N = 149 !Parâmetros da Matriz do Rebanho
!-----
!Matrizes dos Rebanhos
!MV = matriz VACAS-MÃES PRONTAS PARA->CICLAR->EMPRENHAREM->PARIREM-
>LACTANTES
!-----
real :: MV(LV:LX,LV:LX,1:5)!Matrizes de Interação local
integer :: i,j,sement,passos,PMCT,ano
INTEGER :: vazio2,bezerro,desmamado,sobreano,abate,rem2,indexano
INTEGER :: vazio,bezerra,desmamada,vacaciclado,prenhe,parida,rem,pubere,descarte,somaV,parida2
REAL :: RAND,harvest
!-----
!
!               INICIO DE TODOS OS COMMONS
!-----
!VARIÁVEIS COMUNS PROGRAM PRINCIPAL
common/arg0/ sement
common/npassos/ passos,PMCT
!VARIÁVEIS DA SUBROTINA _ que gera números aleatórios UNIFORMES
common/arg1/ rand
!VARIÁVEIS DA SUBROTINA _ que gera números aleatórios NORMAIS
common/arg3/ harvest
!VARIÁVEIS DA SUBROTINA - REBANHO TOTAL
common/arg40/ MV,parida2
common/arg41/ vazio2,bezerro,desmamado,sobreano,abate,rem2
common/arg42/ vazio,bezerra,desmamada,pubere,vacaciclado,prenhe,parida,descarte,rem
!-----
!
!               ARQUIVOS DE ENTRADA
!*****
! Semente e o Número de Passos Monte Carlo e
! Probabilidades = Parâmetros de Entrada
!-----
sement = 11131311
print*, '----- MENU: PROGRAMA DE MANEJO DO REBANHO VIRTUAL-----'
print*, 'Digite o numero de dias'
READ*, passos
print*, '-----Fim do Menu-----'
!-----
!
!               ARQUIVOS DE SAÍDA
!*****
open(1,file='estadmacho.txt')
open(2,file='estadfemea.txt')
!-----
! NUMERO DE SÍTIOS NAS MATRIZES
!numvis = M*N !Número de Vizinhos M=120 e N=120
! * * * * *
!
! ZERANDO AS MATRIZES ANTES DE INICIAR A SIMULAÇÃO
!-----
do i = LV, LX !EFEITO DE AJUSTE DE CONTORNO

```

```

do j = LV, LX
  MV(i,j,1) = 100
  MV(i,j,2) = 100
  MV(i,j,3) = 100
  MV(i,j,4) = 100
ENDDO
ENDDO
!CONDIÇÃO INICIAL
DO i = 0, N
DO j = 0, M
  call random()
  if(rand.le.0.2)then
    MV(i,j,2) = 10
    vacaciclando = vacaciclando + 1 !contador de vacas ciclando da cond. inicial 20%
    call random()
    MV(i,j,1) = INT( 285 + 79*rand)
  else
    MV(i,j,1) = 0  !DATA
    MV(i,j,2) = 0  !ESTADO:10,11,12,3,4,6
    vazio = vazio + 1 !contador de vazios das femeas (maes e filhas)
  endif
!100% da matriz dos machos é sitio vazio
  MV(i,j,3) = 0  !data
  MV(i,j,4) = 0  !estado:0,1,2,5
  vazio2 = vazio2 + 1 !contador de vazios dos machos
ENDDO
ENDDO
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!PROGRAMA                                PRINCIPAL
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
DO 234 PMCT = 1, passos
!-----SUBROTINA DO REBANHO DAS VACAS-----
  CALL MVACA()
!-----
!-----looping geral de visita aos sitios das matriz MV rebanho geral-----
!-----zerando os vetores-----
!Zerando os vetores
!fêmeas
vacaciclando = 0  ! vaca ciclando
  prenhe = 0
  parida = 0
  ! rem = 0 ! removido
  bezerra = 0 ! bezerro
  desmamada = 0
  pubere = 0
  vazio = 0 ! vazio da femeas
  ! descarte = 0
!machos
  bezerro = 0 ! bezerra
  desmamado = 0
  sobreano = 0
  ! rem2 = 0 ! removido
  vazio2 = 0 ! vazio dos machos
  ! abate = 0
!-----SELECT CASE DAS FEMEAS-----
do i = 0, M
do j = 0, N
  select case(INT(MV(i,j,2)))
    case(10)!contador vacas ciclando
      vacaciclando = vacaciclando + 1
    case(11)!contador gestante

```

```

    prenhe = prenhe + 1
    case(12)!contador lactante
    parida = parida + 1
    case(0) !contador vazios
    vazio = vazio + 1
case(3) !contador bezerras
    bezerra = bezerra + 1
    case(4) !contador bezerras desmamadas
    desmamada = desmamada + 1
    case(6) !contador bezerras puberes
    pubere = pubere + 1
end select
enddo
enddo
!-----SELECT CASE DOS MACHOS-----

do i = 0, M
do j = 0, N
select case(INT(MV(i,j,4)))
    case(0) !contador vazios dos machos
    vazio2 = vazio2 + 1
    case(1) !contador bezerros
    bezerro = bezerro + 1
    case(2) !contador bezerros desmamados
    desmamado = desmamado + 1
    case(5) !contador bezerros sobreanos
    sobreano = sobreano + 1
end select
enddo
enddo
!-----

somaV
      (PMCT+vazio+bezerro+desmamado+bezerra+desmamada+vacacicl原因ando+prenhe+parida+rem+sobreano+pubere)
write (*,501)PMCT,vazio2,bezerro,desmamado,sobreano,abate,rem2
write (1,501)PMCT,vazio2,bezerro,desmamado,sobreano,abate,rem2
!write (*,500)PMCT,vazio,bezerra,desmamada,pubere,vacacicl原因ando, prenhe, parida,discarde,rem,parida2
write (2,500)PMCT,vazio,bezerra,desmamada,pubere,vacacicl原因ando, prenhe, parida,discarde,rem,parida2
    !Anotar a soma final de cada ano de vacas paridas
    indexano = mod(PMCT,365) !pmct É VARIAVEL
    ano =INT(PMCT/365)
    IF(indexano.eq.0) parida2 = 0
234 CONTINUE
500 format(11i7)
501 format(7i7)

!-----FINAL DO LOOPING GERAL DE MONTE CARLO-----
!print*,'------Fim da Simulacao do MANEJO MODELO II-----'
!print*,'----- Obrigada e teste de novo na proxima ESTACAO de Monta (EM) !!!----'
END
!* * * * *
!--- FIM DO PROGRAMA PRINCIPAL -----
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!          INICIO DA SUBROTINA 0
!          GERADOR DE NÚMEROS ALEATÓRIOS
!&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
SUBROUTINE RANDOM()
!          VARIÁVEIS
IMPLICIT NONE           !OBRIGA DECLARAÇÃO DAS VAR//S

```

```

integer :: sement
      real :: rand
! VARIÁVEIS COMUNS PROGRAM PRINCIPAL
common/arg0/ sement
!VARIÁVEIS DA SUBROTINA 0 _ que gera números aleatórios
common/arg1/ rand
!-----
!          Subrotina que gera o número RANDÔMICOS = ALEATÓRIOS
!-----
      sement = sement*1103515245 + 453816693
      if (sement.eq.-2147483648) then
        rand = 0.999999996
      return
    endif
    rand = dble(sement)/(-2147483648.0d0)
    rand = ABS(rand)
    return
!*****
END
!          FINAL SUBROTINA 0
!*****
SUBROUTINE gasdev_s()
IMPLICIT NONE
REAL :: harvest,rand
!VARIÁVEIS DA SUBROTINA 0 _ que gera números aleatórios
      common/arg1/ rand
!VARIÁVEIS DA SUBROTINA 2 _ que gera números normais
      common/arg3/ harvest
!Returns in harvest a normally distributed deviate with zero mean and unit variance, using
!ran1 as the source of uniform deviates.
REAL:: rsq,v1,v2
REAL, SAVE :: g
LOGICAL, SAVE :: gaus_stored=.false.
if (gaus_stored) then !We have an extra deviate handy,
  harvest=g !so return it,
  gaus_stored=.false. !and unset the flag.
else !We don't have an extra deviate handy, so
  do
    call RANDOM() !pick two uniform numbers in the square extending
!from -1 to +1 in each direction,
    v1 = 2.0*rand - 1.0
    call RANDOM()
    v2= 2.0*rand - 1.0
    rsq=v1**2+v2**2 !see if they are in the unit circle,
    if (rsq > 0.0 .and. rsq < 1.0) exit
  end do !otherwise try again.
  rsq=sqrt(-2.0*log(rsq)/rsq) !Now make the Box-Muller transformation to
!get two normal deviates. Return one and
!save the other for next time.
  harvest=v1*rsq
  g=v2*rsq
  gaus_stored=.true. !Set flag.
end if
END SUBROUTINE gasdev_s
!-----
!-----
SUBROUTINE MVACA()
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!-----
IMPLICIT NONE      !OBRIGA DELARAÇÃO DAS VAR//S

```

```

integer, parameter :: LV = -2, LX = 153
integer, parameter :: M = 150, N = 150 !Parâmetros da Matriz do Rebanho
!-----
!Matrizes dos Rebanhos
!MV = matriz VACAS-MÃES PRONTAS PARA->CICLAR->EMPRENHAREM->PARIREM-
>LACTANTES
!-----
real :: MV(LV:LX,LV:LX,1:5)!Matrizes de Interação local
integer :: i1,j1,i,j,passos,PMCT
REAL :: RAND,harvest
REAL :: p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p90,p91,p92,p93,p94,p95
INTEGER :: vazio2,bezerro,desmamado,sobreano,abate,rem2
INTEGER :: vazio,bezerra,desmamada,pubere,vacaciclando,prenhe,parida,descarte,rem
INTEGER :: tempdesm,tempdesm2,tempabat,temp sob,temp pub,parida2
!-----
!                               INICIO DE TODOS OS COMMONS
!-----
!VARIÁVEIS COMUNS PROGRAM PRINCIPAL
common/npassos/ passos,PMCT
!VARIÁVEIS DA SUBROTINA 0 _ que gera números aleatórios UNIFORMES
common/arg1/ rand
!VARIÁVEIS DA SUBROTINA 2 _ que gera números aleatórios NORMAIS
common/arg3/ harvest
!VARIÁVEIS DA SUBROTINA - rebanho total
common/arg40/ MV,parida2
common/arg41/ vazio2,bezerro,desmamado,sobreano,abate,rem2
common/arg42/ vazio,bezerra,desmamada,pubere,vacaciclando,prenhe,parida,descarte,rem
!INICIO DA VARREDURA
do i = 0, M
do j = 0, N
!-----
select case(int(MV(i,j,2))) !ESTADO DAS FEMEAS
!-----
! Rebanho Geral - ESTADO OCUPADO
!-----
! COMPARTIMENTO 1: ESTADO VACACICLANDO
!-----
case(10) ! É vacaciclando entao
!PODE VIRAR REMOVIDA: MORRER
CALL RANDOM()
p1 = 0.02/365
IF(RAND.LE.p1) THEN
MV(i,j,2) = 0 ! MORRE FICA VAZIO NA MATRIZ DE ESTADOS
rem = rem + 1 ! MORRE: AUMENTO O N DE MORTES
vazio = vazio + 1 ! LIBERO ESPAÇO: AUMENTO O N DE VAZIOS
vacaciclando = vacaciclando - 1 ! DIMINUI O NÚMERO DE VACACICLANDO
MV(i,j,1) = 0 ! MUDO a Data ESTADO: data 0 : VAZIO
ELSE
!Tornar-se Prenhe;
!EM QUE DIA FICOU PRENHE? !85%
p4 = 0.85
p8 = MOD(PMCT,365) !estacao de monta
IF(p8.eq.MV(i,j,1)) THEN !15 OUT a 31 dez
call random()
if(rand.le.p4) THEN
MV(i,j,2) = 11
call gasdev_s()
MV(i,j,1) = int(PMCT + 293 + 4*harvest) ! duração da gestação
prenhe = prenhe + 1

```

```

    vacaciclando = vacaciclando - 1
else
P14 = 0.7
    call random()
    if(rand.le.p14) THEN
        MV(i,j,1) = INT( 285 + 79*rand)          !vaca volta a ser ciclando
    else
        MV(i,j,2) = 0
        descarte = descarte + 1          ! se nao ficou prenhe descarto
        vazio = vazio + 1          ! LIBERO ESPAÇO: AUMENTO O N DE VAZIOS
        vacaciclando = vacaciclando - 1
    endif
endif
ENDIF
ENDIF
!-----
!  COMPARTIMENTO 2:  ESTADO PRENHE
!-----
case(11) ! Estado ocupado PRENHE
CALL RANDOM()
p3 = 0.02/365    !PROB. MORRE
IF(RAND.LE.p3) THEN
    MV(i,j,2) = 0  !ESTADO
    vazio = vazio + 1
    MV(i,j,1) = 0  !DATA
    rem = rem + 1  !MORRE
    prenhe = prenhe - 1
ELSE
!VIRAR Parida
IF(PMCT.EQ.MV(i,j,1))THEN
    MV(i,j,2) = 12      !
    MV(i,j,1) = INT(PMCT + 54 + 7*harvest) !tempo de anestro pos parto
    parida = parida + 1
    parida2 = parida2 + 1
    prenhe = prenhe - 1
!-----!
!      NASCIMENTOS DOS BEZERROS E BEZERRAS
!-----!
    call random()
    p7 = 0.5
    IF(RAND.le.p7)then    !NASCE FEMEA
        ! Procura em volta se tem espaço para nascer
        do i1 = -2,2
            do j1 = -2,2
                IF(MV(i+i1,j+j1,2).EQ.0)then  ! Procuro ao redor se tenho sitio vazio
                    MV(i+i1,j+j1,2) = 3
                    MV(i+i1,j+j1,1) = PMCT  !data de nascimento da fêmea
                    MV(i+i1,j+j1,5) = INT(PMCT + 540 + 40*harvest) !data da bezerra ficar pubere
                    bezerra = bezerra + 1
                    vazio = vazio - 1
                    GOTO 339
                ENDIF
            end do
        end do
339  continue
    ELSE  !SENÃO NASCE MACHO 50% restantes
        do i1 = -2,2
            do j1 = -2,2
                IF(MV(i+i1,j+j1,4).EQ.0)then  ! Procuro ao redor se tenho sitio vazio
                    MV(i+i1,j+j1,4) = 1 !ESTADO

```

```

        MV(i+1,j+1,3) = PMCT !data de nascimento do macho
        bezerro = bezerro + 1
        vazio2 = vazio2 - 1
        GOTO 340
    ENDIF
enddo
enddo
340 continue
ENDIF
ENDIF
ENDIF
!-----
! COMPARTIMENTO 2: ESTADO PARIDA
!-----
case(12) ! Estado ocupado PARIDA
CALL RANDOM() ! pode morrer
p5 = 0.02/365
IF(RAND.LE.p5) THEN
    MV(i,j,2) = 0
    rem = rem + 1
    parida = parida - 1
    vazio = vazio + 1
    MV(i,j,1) = 0
ELSE
    ! Ficar prenhe +ou-60%
    IF(PMCT.ge.MV(i,j,1)) THEN !periodo fertil
        p9 = MOD(PMCT,365)
        IF(p9.ge.285) THEN !213i365 estacao de monta
            CALL RANDOM()
            p2 = 0.7
            if(RAND.LE.p2) THEN
                MV(i,j,2) = 11
                prenhe = prenhe + 1
                parida = parida - 1
                call gasdev_s()
                MV(i,j,1) = int(PMCT + 293 + 12*harvest) ! TEMPO DE GESTACAO
            else !descarte se nao entrou na estacao de monta
                MV(i,j,2) = 0 !estado
                MV(i,j,1) = 0 !data
                parida = parida - 1
                vazio = vazio + 1
                descarte = descarte + 1
            endif
        ELSE
            P15 = 0.7
            call random()
            if(rand.le.p15) THEN
                MV(i,j,1) = INT( 285 + 79*rand) !volta a ser ciclando
                MV(i,j,2) = 10
                parida = parida - 1
                vacaciclando = vacaciclando +1
            else
                MV(i,j,2) = 0
                MV(i,j,1) = 0
                descarte = descarte + 1 ! se nao ficou prenhe descarto
                vazio = vazio + 1 ! LIBERO ESPAÇO: AUMENTO O N DE VAZIOS
                parida = parida - 1
            endif
        ENDIF
    ENDIF
ENDIF

```

```

ENDIF
!*****
!
!           FILHAS FEMEAS
!-----
!  COMPARTIMENTO :   ESTADO BEZERRA DESMAMADA
!-----
CASE(3)
CALL RANDOM() ! pode morrer
p93 = 0.05/365
IF(RAND.LE.p93) THEN
  MV(i,j,2) = 0
  rem = rem + 1
  bezerra = bezerra - 1
  MV(i,j,1) = 0
  vazio = vazio + 1
ELSE
  !Virar desmamada
  tempdesm2 = (PMCT-MV(i,j,1))
  p11=mod(PMCT,365)
  !-----
  IF(tempdesm2.ge.180)THEN   !2 DESMAMA
    if(p11.eq.76) then
      MV(i,j,2) = 4
      desmamada = desmamada + 1
      bezerra = bezerra - 1
    endif
  IF(tempdesm2.ge.180)THEN  !1 DESMAMA
    if(p11.eq.120) then
      MV(i,j,2) = 4
      desmamada = desmamada + 1
      bezerra = bezerra - 1
    endif
  ENDIF
ENDIF
ENDIF
!-----
!  COMPARTIMENTO :   ESTADO BEZERRA pubere
!-----
CASE(4)
CALL RANDOM() ! pode morrer
p94 = 0.02/365
IF(RAND.LE.p94) THEN
  MV(i,j,2) = 0
  rem = rem + 1
  desmamada = desmamada - 1
  MV(i,j,1) = 0
  vazio = vazio + 1
else !vira pubere
IF(PMCT.eq.MV(i,j,5))THEN  !TEMPO DE PUBERDADE
  MV(i,j,2) = 6
  pubere = pubere + 1
  desmamada = desmamada - 1
endif
endif
!-----
!  COMPARTIMENTO :   ESTADO BEZERRA pubere morrer
!-----
CASE(6)
CALL RANDOM() ! pode morrer
p95 = 0.03/365

```



```

IF(RAND.LE,p95) THEN
  MV(i,j,2) = 0
  rem = rem + 1
  pubere = pubere - 1
  MV(i,j,1) = 0
  vazio = vazio + 1
else !A novilha pubere PRENHE
call random()
p6 = 0.5
p12 = MOD(PMCT,365)
IF(p12.ge.285.or.p12.le.30) THEN ! 15 de outubro a 30 janeiro
IF(RAND.LE,p6) THEN
  MV(i,j,2) = 11
  prenhe = prenhe + 1
  pubere = pubere - 1
  MV(i,j,1) = int(PMCT + 293 + 12*harvest) ! 9 MESES
ELSE
  P16 = 0.7
  call random()
  if(rand.le.p16) THEN
    MV(i,j,1) = INT( 285 + 79*rand) !volta a ser ciclando
    MV(i,j,2) = 10
    pubere = pubere - 1
    vacaciclando = vacaciclando +1
  else
    MV(i,j,2) = 0
    MV(i,j,1) = 0
    descarte = descarte + 1 ! se nao ficou prenhe descarto
    vazio = vazio + 1 ! LIBERO ESPAÇO: AUMENTO O N DE VAZIOS
    pubere = pubere - 1
  endif
endif
endif
Endif
!-----
END SELECT
!-----
! Select Case dos FILHOS MACHOS
!-----
select case(int(MV(i,j,4))) !ESTADO DAS FEMEAS
!-----
!*****
! MACHOS
!-----
! COMPARTIMENTO : ESTADO BEZERRO
!-----
CASE(1)
CALL RANDOM() ! pode morrer
p90 = 0.05/365
IF(RAND.LE,p90) THEN
  MV(i,j,3) = 0
  rem2 = rem2 + 1
  bezerro = bezerro - 1
  MV(i,j,4) = 0
  vazio2 = vazio2 + 1
ELSE
  !Virar desmamada
  tempdesm = (PMCT-MV(i,j,3))
  p10 = mod(PMCT,365) !tempo de desmamar
!-----

```

```

IF(tempdesm.ge.180)THEN  !2 DESMAMA :idade minima de desmame 180 dias
  if(p10.eq.76) then
    MV(i,j,4) = 2
    desmamado = desmamado + 1
    bezerro = bezerro - 1
  endif
IF(tempdesm.ge.180)THEN  !1 DESMAMA
  if(p10.eq.120) then
    MV(i,j,4) = 2
    desmamado = desmamado + 1
    bezerro = bezerro - 1
  endif
ENDIF
ENDIF
ENDIF
!-----
!  COMPARTIMENTO : Bezerro  Desmamado
!-----
CASE(2)
CALL RANDOM() ! pode morrer
p91 = 0.02/365
IF(RAND.LE.p91) THEN
  MV(i,j,3) = 0
  rem = rem + 1
  desmamado = desmamado - 1
  MV(i,j,4) = 0  !morre
  vazio2 = vazio2 + 1
ELSE
  !Virar sobreano
  tempsob = (PMCT-MV(i,j,3))
  p13 =mod(PMCT,365)
  !-----
IF(tempsob.ge.440)THEN  !idade maxima para sobreano e 640  (a mínima é 440)
  if(p13.eq.21)then
    MV(i,j,4) = 5  !sobreano
    sobreano = sobreano + 1
    desmamado = desmamado - 1
  endif
ENDIF
ENDIF
!-----
!  COMPARTIMENTO :  ESTADO SOBREANO MORRE ou é abatido
!-----
CASE(5)
CALL RANDOM() ! pode morrer
p92 = 0.01/365
IF(RAND.LE.p92) THEN
  MV(i,j,3) = 0
  rem = rem + 1
  sobreano = sobreano - 1
  MV(i,j,4) = 0
  vazio = vazio + 1
else
  tempabat = (PMCT-MV(i,j,3))
IF(tempabat.gt.640)THEN
  MV(i,j,3) = 0
  sobreano = sobreano - 1
  vazio = vazio + 1
  MV(i,j,4) = 0
  abate = abate + 1

```

```
    ENDIF
endif
END SELECT
!-----
!*****
end do
end do  !Varredura da Matriz  MV do Rebanho Total
!*****
END
!-----FIM DA SUBROTINA GERAL-----
```



```

print *, REDE NEURAL MLP - BACK PROPAGATION
print *,
print *, OBJETIVO: PREVER VALOR GENETICO
print *, EM BOVINO DE CORTE
print *, CFM - NELORE BRAZIL
print *,
print *, Tecnica: ALGORITMO - BACKPROPAGTION
print *, LAFAC - LAB. FIS. APL. COMPUCIONAL
print *, Version 1.0
print *, 2004
print *,
print *, Orientadores: Ernane J. X. Costa e
print *, Jose B. S. Ferraz
print *, Doutoranda: Flavia Providelo
print *,
print *, ZAB - FZEA - PIRASSUNUNGA
print *, USP - Sao Paulo University
print *, *****
print *
!-----
! * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
!-----
! FASE DE TREINAMENTO DA REDE
!-----
print *, *****
print *, DIGITE: MAX. N.INTERACOES
read(*,*)maxnint
print *, ENTRE COM O TAXA DE APRENDIZADO=SIGMA (0-1)=..... E MOMENTO =....
read(*,*)sigma,umy
print *, DIGITE: SEMENTE=.... DO NÚMERO ALEATORIO
read(*,*)sement
print *,REDE Neural: Np=n padroes =linhas(1000), Ni=n colunas na entrada
read(*,*) Np,Ni
print *,Nh = n colunas na camada escondida,No = n colunas na saida = TARGET
read(*,*)Nh,No
!--Inicializar a taxa de aprendizagem
print *,DIGITE As ctes das BIAS ---> b0,b1 Ctes da fç de ativacao: a,b,c
read(*,*)b0,b1,a,b,c
print *, DIGITE O LIMAR DO ERRO QUADRATICO MEDIO -> EXEMPLOS = 0.001 OU 0.00...1
read(*,*)LIMIAR
print *,Abrindo o arquido dos animais (original)de TREINO 2/3 dos dados
print *, DIGITE O ARQUIVO DE DADOS DE ENTRADA:.....txt
open(300,file='',form='formatted')
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!-----
!-----LER OS DADOS DE ENTRAD X(I,J) E T (I,J)-----
do 59 i = 1, Np !N1 = tamanho do arquivo = quantas LINHAS tem o arquivo
read(300,*)(MX(I,J),J=1,Ni+1)
! write(*,*)(MX(I,J),J=1,Ni+1)! TESTE DE LER O ARQUIVO DE ENTRADA
59 continue
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!-----
! INICIAR OS PESOS ALEATORIAMENTE DA REDE MV = 1 camada e MW = 2camada
!-----
DO J = 1, Nh
CALL RANDOM()
MVb0(j) = rand !MATRIZ DOS PESOS
! print*, rand
END DO

```

```

!-----peso da 2 camada-----
DO K = 1, No
    CALL RANDOM()
    MWb1(k) = rand !MATRIZ DOS PESOS bias
END DO
!-----
!-Iniciar os pesos--da 1 camada = MVb0 ---> 2 camadas = MVb1 -
!-----
DO i = 1, Ni
    DO J = 1, Nh
        CALL RANDOM()
        MV(i,j) = rand !MATRIZ DOS PESOS
! write(*,*)      i,j,MV(i,j)
    END DO
END DO
!-----2 camada-----
DO J = 1, Nh
    DO K = 1, No
        CALL RANDOM()
        MW(j,k)= rand !MATRIZ DOS PESOS
    END DO
END DO
!-----erquad2=10000 -----
nint = 0
p=0
i=0
k=0
j=0
k1=0
erquad = 0
erquad2 = 10000
!-----
DO k = 1, No
    DO j = 1, Nh
        NET(j)=0
        NETOUT(K)=0
        erhidden1(j)=0
    END DO
END DO
!-----
DO 10 WHILE(erquad2.ge.limiar.and.nint.le.maxnint)
!-----INICIO DO FEED-FORWARD-----
!-----
DO 100 p = 1, Np !Ni de padrões = LINHAS (ni)
! Colunas da saída da rede (calculo do target)
! 1 CAMADA ESCONDIDA
DO j = 1, Nh ! Nh de colunas na Camada ESCONDIDAS (Nj)
    NET(j) = 0
    DO i = 1, Ni ! Ni de colunas na Entrada MX(Np,i)
        NET(j) = NET(j) + MX(p,i)*MV(i,j)
    END DO ! fecha i
    NET(j) = NET(j) + b0*MVb0(j)
    Z(j) = a/(1+exp(-(b*NET(j)))) - c !FUNÇÃO DE ATIVAÇÃO
    derivada1(j) = a*b*exp(-b*NET(j))/(1+exp(-b*NET(j)))**2
END DO ! fecha jteste
!write(*,*)'@@@--nint-j--i--p---net(j)-----'
!write(*,*)nint,j,i,p,net(j)
!-----
! 2 CAMADA ESCONDIDA
DO k = 1, No ! Nh de colunas na Camada ESCONDIDAS (Nj)

```

```

NETOUT(K) = 0
DO j = 1, Nh ! No de colunas na SAIDA = TARGET
    NETOUT(K) = NETOUT(K) + Z(j)*MW(j,k) !FUNÇÃO DE ATIVAÇÃO na saida
END DO ! fecha j
    NETOUT(K) = NETOUT(K) + b1*MWb1(k)
    Y(K) = a/(1+exp(-b*NETOUT(K))) - c ! FUNÇÃO DE ATIVAÇÃO
    derivada2(k) = a*b*exp(-b*NETOUT(k))/(1+exp(-b*netout(k)))**2 !derivada da fç sigmoide
!write(*,*)'***--nint--y(k)---NETOUT(K)---derivada2(k)-----'
! write(*,*)nint,Y(K),NETOUT(K),derivada2(k)
END DO ! fecha k
!-----FIM DO FEED-FORWARD-----
!*****
!-----INICIO DO FEED-BACKWARD-----
!-----Atualizacao da 2 camada-----
DO k = 1, No ! No de colunas na SAIDA = TARGET
!---Calculo a fcs de Erro da Camada K = saida MX(p,ni)= TARGET-----
    errout(k) = (MX(p,ni+1) - Y(K))*derivada2(k)
    DO j = 1, Nh
        deltaw(j,k) = sigma*errout(k)*Z(j)
        deltab1(k) = sigma*errout(k)
        erhidden1(j) = 0
    END DO
END DO ! fecha
!-----atualizacao da 1 camada-----
DO k = 1, No ! No de colunas na SAIDA = TARGET
    DO j = 1, Nh ! Nh de colunas na Camada ESCONDIDAS (Nj)
!---Atualizar pesos da 1 camada Vij ---> somatorio
        erhidden1(j) = erhidden1(j) + errout(k)*MW(j,k)
    END DO ! fecha k
END DO ! fecha j
!-----
DO j = 1, Nh ! Nh de colunas na Camada ESCONDIDAS (Nj)
    erhidden2(j) = erhidden1(j)*derivada1(j)! derivada da saida final
    deltab0(j) = sigma*erhidden2(j)
DO i = 1, Ni ! Ni de colunas na Entrada MX(Np,i)
    deltav(i,j) = sigma*erhidden2(j)*MX(p,i)
!-----
    END DO ! fecha i
END DO ! fecha j
!---Atualizar pesos da 2 camada Wij
MW(0,0) = MW(1,1)
temp1=0
DO k = 1, No
    DO j = 1, Nh
        MW(j,k) = MW(j,k) + deltaw(j,k)
        MWb1(k) = MWb1(k) + deltab1(k)
    END DO
END DO
!---Atualizar pesos da 1 camada vij-----
MV(0,0)=MV(1,1)
DO j = 1, Nh
    DO i = 1, Ni
        MV(i,j) = MV(i,j) + deltav(i,j)
        MVb0(j) = MVb0(j) + deltab0(j)
    END DO
END DO
!---somatorio do ERRO QUADRÁT---MX(p,ni+1)= TARGET e Saida -->Y(p)-----
DO k1 = 1, No
    erquad = erquad + 0.5*(MX(p,ni+1) - Y(k1))**2
END DO

```

```

100 CONTINUE ! fecha p -- looping geral do padrao Np
!-----
nint = nint + 1
!---ERRO QUADRÁTICO MÉDIO
erquad2 = erquad/(Np*No)
!-----
write(*,*)nint,erquad,erquad2
write(4,*)nint,erquad,erquad2
!-----
!-----ZERAR AS VARIÁVEIS APOS AS INTERAÇÕES-----
erquad=0
!ZERAR OS SOMATORIOS
DO k = 1, No
  DO j = 1, Nh
    !      MW(j,k) = 0
    !      NET(j)=0
    !      NETOUT(K)=0
    !      erhidden1(j)=0
  END DO
END DO
DO j = 1, Nh
  DO i = 1, Ni
    !      MV(i,j) = 0
  enddo
enddo
!-----
10 CONTINUE ! fecha while
!-----FIM DO FEED-BACKWARD-----
!-----
!* * * * * ARMAZENAR OS DADOS* * * * *
!-----INICIO DA VALIDAÇÃO-----
!-----INICIO DO FEED-FORWARD-----
print *, 'REDE Neural: Npval, Ni=n colunas na entrada '
read(*,*) Npval,Nvc
print *, 'Abrindo o arquivo dos animais (original)de VALIDAÇÃO 1/3 dos dados'
print *, ' DIGITE O ARQUIVO DE DADOS DE ENTRADA:.....txt '
!open(301,file='',form='formatted')
!-----LER OS DADOS DE ENTRADA MVALID(I,J) E T (I,J)-----
do 70 i = 1, Npval !Npval quantas LINHAS tem o arquivo
  read(301,*)(MVALID(I,J),J=1,Nvc+1)
  ! write(*,*)(MVALID(I,J),J=1,Nvc+1)! TESTE DE LER O ARQUIVO DE ENTRADA
70 continue
!@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
!-----
DO 1000 p = 1, Npval ! LINHAS
! Colunas da saida da rede (calculo do target)
! 1 CAMADA ESCONDIDA
DO j = 1, Nh ! Nh de colunas na Camada ESCONDIDAS
  NET(j) = 0
  DO i = 1, Nvc ! Ni de linhas na Entrada
    NET(j) = NET(j) + MVALID(p,i)*MV(i,j)
  END DO ! fecha i
  NET(j) = NET(j) + b0*MVb0(j)
  Z(j) = a/(1+exp(-(b*NET(j)))) - c !FUNÇÃO DE ATIVAÇÃO
END DO ! fecha jteste
! write(*,*)'***--nint--y(k)--MVALID(p,3)-----'
! write(*,*)nint,Z(j),MVALID(p,3)
!-----
! 2 CAMADA ESCONDIDA
DO k = 1, No ! Nh de colunas na Camada ESCONDIDAS (Nj)

```


[illegible]