

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
FACULDADE DE CIÊNCIAS ECONÔMICAS  
CENTRO DE PÓS-GRADUAÇÃO E PESQUISA EM ADMINISTRAÇÃO**

**A UTILIZAÇÃO DE UM MODELO HÍBRIDO *ALGORITMO GENÉTICO* / *REDES NEURAIS*  
NO PROCESSO DE SELEÇÃO DE CARTEIRAS**

**Marcelo Soares Cartacho**

Belo Horizonte  
2001

**Marcelo Soares Cartacho**

**A UTILIZAÇÃO DE UM MODELO HÍBRIDO *ALGORITMO GENÉTICO /  
REDES NEURAS* NO PROCESSO DE SELEÇÃO DE CARTEIRAS**

Dissertação apresentada ao Curso de Mestrado do Centro de Pós-Graduação e Pesquisa em Administração da Faculdade de Ciências Econômicas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Administração.

Área de concentração: Finanças e Contabilidade

Orientador: Prof. Dr. Antônio Artur de Souza

Belo Horizonte  
Faculdade de Ciências Econômicas da UFMG  
2001

*Esta dissertação é dedicada aos meus pais.*

## **AGRADECIMENTOS**

Ao Professor Dr. Antônio Artur de Souza, pela orientação, liberdade e confiança durante todo o período do mestrado.

Aos membros da banca examinadora, Professor Dr. Hudson Fernandes Amaral e Professora Dra. Maria Elenita M. Nascimento, que muito contribuíram com suas críticas e sugestões ao trabalho.

Aos funcionários e demais professores do CEPEAD, pela oportunidade de aprendizado.

Aos amigos, pela paciência, e aos colegas de curso, por me proporcionarem um ambiente extremamente agradável em meio a um momento tão intenso. Um agradecimento especial ao amigo e colega Sander Oliveira de Freitas, pelas contribuições feitas ao trabalho e pelo incentivo nas horas mais difíceis, e a Josmária Lima Ribeiro, por toda a ajuda durante esse período.

À Adriana Corrêa Alves, pelo apoio e por estar sempre presente nos momentos de maior tensão e ansiedade.

Aos meus pais, Ivan Brescia Cartacho e Maria da Glória Soares Cartacho, e ao meu irmão, Rodrigo Soares Cartacho, que me incentivaram e souberam entender a minha ausência durante boa parte dessa etapa.

## SUMÁRIO

<b><u>1 INTRODUÇÃO</u></b> .....	<b>12</b>
<u>1.1 Objetivos</u> .....	13
<u>1.2 Justificativa</u> .....	13
<u>1.3 Problema</u> .....	14
<u>1.4 Métodos de pesquisa</u> .....	14
<u>1.5 Organização do trabalho</u> .....	14
<b><u>2 SELEÇÃO DE CARTEIRAS</u></b> .....	<b>18</b>
<u>2.1 Modelo de Markovitz</u> .....	18
<u>2.2 Outros modelos</u> .....	24
<u>2.2.1 Modelo de Markovitz com ativos sem risco</u> .....	24
<u>2.2.2 Modelo de índice único de Sharpe</u> .....	25
<u>2.2.3 Modelo de índices múltiplos</u> .....	26
<b><u>3 ALGORITMOS GENÉTICOS</u></b> .....	<b>27</b>
<u>3.1 Busca e otimização</u> .....	27
<u>3.2 Um algoritmo genético simples</u> .....	28
<u>3.3 Codificação das soluções</u> .....	30
<u>3.4 Métodos de seleção</u> .....	32
<u>3.5 Operadores genéticos</u> .....	37
<u>3.5.1 Recombinação</u> .....	37
<u>3.5.2 Mutação</u> .....	38
<u>3.6 Eficácia dos algoritmos genéticos</u> .....	39
<u>3.7 Algoritmos genéticos X outros métodos de otimização</u> .....	40
<b><u>4 REDES NEURAIS</u></b> .....	<b>44</b>

<u>4.1 Principais componentes e organização</u> .....	44
<u>4.1.1 Unidade de processamento</u> .....	44
<u>4.1.2 Função de ativação</u> .....	47
<u>4.1.3 Organização em camadas</u> .....	48
<u>4.2 Arquiteturas</u> .....	49
<u>4.2.1 Padrão de conectividade</u> .....	49
<u>4.2.2 Número de camadas</u> .....	52
<u>4.2.3 Estrutura adequada</u> .....	53
<u>4.3 Aprendizado</u> .....	54
<u>4.3.1 Paradigmas de aprendizado</u> .....	55
<u>4.3.2 Algoritmos de aprendizado</u> .....	56
<u>4.4 Metodologia para construção de uma RN</u> .....	57
<u>4.4.1 Coleta de dados</u> .....	58
<u>4.4.2 Análise e formatação de dados</u> .....	59
<u>4.4.3 Separação dos dados em conjuntos</u> .....	60
<u>4.4.4 Seleção das variáveis</u> .....	60
<u>4.4.5 Desenvolvimento e otimização do modelo</u> .....	61
<u>4.4.6 Validação</u> .....	62
<u>4.4.7 Aplicação do modelo</u> .....	62
<u>4.5 Pontos fortes e limitações</u> .....	62
<b><u>5 RNs E AGs APLICADOS À SELEÇÃO DE CARTEIRAS</u></b> .....	<b>65</b>
<b><u>6 METODOLOGIA</u></b> .....	<b>68</b>
<u>6.1 Unidade e período de análise</u> .....	68
<u>6.2 Coleta de dados</u> .....	69
<u>6.3 Aplicação do método de Markovitz</u> .....	70
<u>6.4 Aplicação do modelo híbrido</u> .....	71

<u>6.4.1 Modelagem e execução das redes neurais</u> .....	71
<u>6.4.2 Modelagem e execução do algoritmo genético</u> .....	76
<u>6.5 Comparação dos resultados</u> .....	78
<b><u>7 RESULTADOS OBTIDOS</u></b> .....	<b>79</b>
<b><u>8 CONCLUSÕES E RECOMENDAÇÕES</u></b> .....	<b>85</b>
<b><u>9 REFERÊNCIAS BIBLIOGRÁFICAS</u></b> .....	<b>89</b>
<b><u>10 GLOSSÁRIO</u></b> .....	<b>96</b>

## LISTA DE FIGURAS

Figura 1	Fronteira eficiente de ativos de risco .....	23
Figura 2	Fronteira eficiente de ativos com e sem risco .....	25
Figura 3	Unidade de processamento simples .....	45
Figura 4	Rede <i>feed-forward</i> .....	50
Figura 5	Estrutura básica das redes neurais utilizadas .....	75

## LISTA DE GRÁFICOS

Gráfico 1	Retornos mensais obtidos entre jan/99 e dez/00.....	82
Gráfico 2	Retornos acumulados obtidos.....	83

## LISTA DE TABELAS

Tabela 1	Ações utilizadas no trabalho .....	69
Tabela 2	Variável a ser prevista .....	73
Tabela 3	Entradas da rede neural .....	73
Tabela 4	Configuração final da rede neural .....	76
Tabela 5	Retornos obtidos, riscos e índices de Sharpe .....	80
Tabela 6	Resultados obtidos, separados por ano.....	82
Tabela 7	Composição das carteiras geradas pelo método de Markovitz....	84
Tabela 8	Composição das carteiras geradas pelo modelo híbrido .....	85

## RESUMO

A seleção de carteiras é o processo pelo qual se escolhem, dentre várias alternativas de investimento, os ativos que comporão a carteira com melhor relação risco-retorno, dado o perfil do investidor. A chamada Moderna Teoria de Carteiras engloba as idéias utilizadas na realização dessa seleção e tem como base o método de Markovitz. Este método, utilizado na geração de uma fronteira eficiente composta por carteiras de mínimo risco dado o nível de retorno ou de máximo retorno dado o nível de risco, necessita que certas premissas sejam válidas para sua devida utilização, além de dados de entrada que nem sempre refletem da melhor forma a realidade. O presente trabalho, desse modo, teve como objetivo verificar se o desempenho obtido no processo de seleção de carteiras, através do modelo de Markovitz, poderia ser alcançado ou superado por um modelo alternativo, capaz de captar as relações entre as variáveis do problema. Este modelo seria composto por métodos genéricos, mais especificamente algoritmos genéticos e redes neurais. As redes neurais, utilizadas no modelo, são estruturas capazes de reconhecer padrões aparentemente não detectáveis entre variáveis, permitindo que se realizem previsões sobre valores futuros com certa confiabilidade nos resultados. Já os algoritmos genéticos são métodos de busca e otimização que utilizam idéias da seleção natural e da genética na solução de problemas. Neste trabalho, criou-se um modelo que utiliza redes neurais na previsão de preços futuros de ações e um algoritmo genético que otimiza a composição de carteiras a partir da relação entre risco e retorno das ações que a compõem. Dessa forma, para se alcançar o objetivo traçado, foi feita uma simulação de investimentos durante os anos de 1999 e 2000, utilizando-se, para isso, carteiras compostas por oito ações negociadas na Bovespa nesse período. Na primeira etapa da simulação, utilizou-se o método de Markovitz na geração das carteiras a serem investidas. Na outra etapa, fez-se uso do modelo híbrido recém-criado, composto por um algoritmo genético e diversas redes neurais. Ao analisar os resultados, viu-se que os retornos obtidos pelo modelo híbrido foram superiores aos obtidos pelo método tradicional nos dois anos estudados, 1999 e 2000. Além disso, apesar da variabilidade das taxas de retorno ter se mostrado menor ao utilizarmos o método de Markovitz, a relação retorno/risco também foi melhor nas carteiras geradas pelo modelo alternativo.

# 1 INTRODUÇÃO

Os métodos atualmente utilizados na avaliação de ações e na seleção de carteiras têm tido seus resultados questionados nos últimos anos. Alguns autores, como Haugen (1995), apontam estudos que mostram que os modelos atualmente utilizados na seleção de carteiras não geram carteiras eficientes.

Uma abordagem alternativa ao método tradicionalmente utilizado seria a aplicação de um modelo composto por *métodos genéricos*, ou seja, métodos que podem ser aplicados em diversas áreas do conhecimento e não apenas em um campo específico.

Dentre esses métodos destacam-se os algoritmos genéticos – AGs – e as redes neurais – RNs. Os algoritmos genéticos são algoritmos que utilizam princípios da seleção natural e da genética na solução de problemas de busca e otimização. Funcionam gerando uma população inicial de soluções e evoluindo-as ao longo do tempo de acordo com suas respectivas adaptações ao ambiente externo, esperando que, ao final do processo, soluções ótimas ou quase-ótimas sejam encontradas.

Já as redes neurais são estruturas capazes de, dentre várias outras aptidões, captar as inter-relações entre as variáveis de um problema sem a necessidade de se conhecer profundamente sua lógica. Basicamente, funcionam definindo-se as variáveis independentes e dependentes a serem relacionadas, criando-se um modelo configurado especificamente para o problema com o qual se está lidando e atualizando-se sua estrutura de acordo com dados reais fornecidos como entrada.

Métodos como as redes neurais e os algoritmos genéticos geralmente são aplicados em áreas de conhecimento que apresentam algumas características especiais. A primeira

delas é o fato de a teoria relativa a essa área ainda não conseguir explicar adequadamente o comportamento dos fenômenos observados. A outra refere-se ao espaço de soluções do problema ser grande demais para se pesquisar todas as soluções possíveis. Em situações como essa (como é o caso da Teoria de Carteiras), torna-se viável a aplicação de métodos como os acima mencionados.

### 1.1 Objetivos

O objetivo geral deste trabalho é verificar se a utilização de um modelo híbrido composto por um algoritmo genético e diferentes redes neurais consegue se adaptar ao problema de seleção de carteiras, obtendo um desempenho melhor ou igual do que aquele visto por um modelo tradicional, mais especificamente a versão *ex-ante* do método de Markovitz.

### 1.2 Justificativa

A seleção de carteiras é uma tarefa de extrema importância, pois uma boa realização dessa seleção implicará um bom rendimento para o investidor após determinado tempo de posse da carteira.

Ao realizar um investimento em ações, o investidor espera que o seu retorno seja grande o suficiente para compensar os riscos que está correndo. A Teoria de Finanças tenta explicar essa relação entre risco e retorno, relacionando as diversas variáveis do problema.

Entretanto, nem toda teoria consegue captar todas as nuances de um universo de conhecimento. Um modelo teórico é apenas uma abstração da realidade. Sua utilização

implica a troca de parte da caracterização dessa realidade por um entendimento mais claro do problema.

Com a Teoria de Finanças, mais especificamente a Moderna Teoria de Carteiras, isso não poderia ser diferente. Por tratar-se de uma área de grande complexidade, as teorias construídas à sua volta ainda não conseguiram explicar de forma satisfatória as relações entre suas variáveis.

Por isso, é de grande relevância um estudo que compara a aplicação dessa teoria tradicional – que já vem sendo construída há várias décadas – com um diferente tipo de abordagem, formada através da integração de diversas áreas do conhecimento, como as Finanças e a Ciência da Computação, e presumivelmente capaz de detectar as especificidades do problema. Não se discute, nesse trabalho, o fato de o progressivo desenvolvimento da Teoria de Carteiras ser de extrema importância para o entendimento da realidade que a cerca. Entretanto, enquanto abordagens genéricas conseguirem, em qualquer campo de conhecimento, ajudar no entendimento da teoria específica e/ou aprimorar os resultados obtidos, elas podem e devem ser aplicadas.

### 1.3 Problema

O método de Markovitz para seleção de carteiras baseia-se em algumas premissas que não são verificadas na realidade. Além disso, para ser aplicado, este método necessita de dados de entrada que nem sempre podem ser conseguidos com um grau de confiabilidade considerado satisfatório. Vários autores, como Haugen (1995), realizaram estudos e verificaram que os modelos atualmente utilizados para selecionar as carteiras de maior rentabilidade não são adequados, já que as carteiras resultantes desse processo não se mostram eficientes.

Sendo o problema de seleção de carteiras um problema de otimização e sendo a precificação de ações um problema de detecção da inter-relação entre diferentes variáveis, surge a seguinte pergunta:

Pode uma combinação de métodos genéricos, como as redes neurais e os algoritmos genéticos, captar as especificidades do processo de seleção de carteiras, de modo a formar carteiras de melhor relação risco-retorno do que aquelas formadas pelo método de Markovitz?

#### 1.4 Métodos de pesquisa

Para que o objetivo do trabalho seja alcançado, definiram-se os seguintes métodos de pesquisa:

- simulação mensal, durante os anos de 1999 e 2000, do investimento em carteiras geradas através do método de Markovitz, utilizando-se, para isso, ações negociadas na Bovespa nesse período;
- construção de um modelo composto por algoritmos genéticos e redes neurais, capaz de captar as especificidades dessa área de conhecimento e gerar carteiras com uma boa relação risco-retorno para o investidor. Na construção desse modelo, utilizam-se redes neurais na precificação de ações e um algoritmo genético na otimização das carteiras. As precificações são feitas com base em variáveis históricas, tais como cotações (preços de abertura, máximo, mínimo e fechamento), volumes de negociação, índice Ibovespa, taxa do dólar, taxa de juros, poupança e índice Dow Jones. Já a otimização toma como base as previsões realizadas pela rede neural, gerando carteiras de acordo com a relação entre risco e retorno das ações que a compõem.

- simulação mensal, durante os anos de 1999 e 2000, do investimento nas carteiras geradas através do modelo criado na etapa anterior, utilizando-se, para isso, ações negociadas na Bovespa nesse período;
- comparação, pelo índice de Sharpe, das relações risco-retorno resultantes dos investimentos realizados pelas duas estratégias.

### 1.5 Organização do trabalho

No capítulo 2 será abordada a teoria de seleção de carteiras, dando-se ênfase ao método de Markovitz para geração da fronteira eficiente.

No capítulo 3 introduz-se o tema dos algoritmos genéticos, explicando-se para que servem, como funcionam, como são codificados e quais os métodos de seleção e operadores genéticos a serem utilizados. Por fim, explicam-se as especificidades desse método que o diferenciam de outros métodos de otimização.

O referencial teórico relativo às redes neurais é visto no capítulo 4. Nele, dá-se uma explanação sobre os principais componentes dessas redes, como podem ser estruturados, quais as formas de se realizar o treinamento e quais são as vantagens e desvantagens de se utilizar uma rede neural. Apresenta-se também uma metodologia para a construção de tais redes.

Na quinto capítulo, apresentam-se exemplos de aplicações de algoritmos genéticos e redes neurais no processo de seleção de carteiras.

A metodologia utilizada na realização do trabalho é apresentada no capítulo 6, sendo os resultados obtidos descritos no capítulo 7 e as conclusões no oitavo capítulo.

Um glossário contendo os termos mais utilizados durante a leitura do texto pode ser visto ao final do trabalho, logo após as Referências Bibliográficas.

## 2 SELEÇÃO DE CARTEIRAS

A seleção de carteiras é o processo em que se analisam as perspectivas de risco e retorno de vários títulos com o objetivo de formar um conjunto que, através da diversificação, apresente risco e retorno adequados ao perfil do investidor.

Os trabalhos de Harry Markovitz são considerados o marco inicial de pesquisas na área e contêm os princípios básicos da formação de carteiras. Segundo Brealey e Myers (1998), o primeiro artigo de Markovitz relacionado ao tema, publicado em 1952, constitui a base de tudo o que há a dizer sobre a relação entre risco e retorno e estabelece um modelo para a construção de carteiras que combina de forma eficiente diferentes níveis de risco e retorno. Em 1959, Markovitz publica *Portfolio Selection*, livro que reúne as idéias desenvolvidas pelo autor.

### 2.1 Modelo de Markovitz

O modelo de Markovitz para a seleção de carteiras tem algumas premissas que devem ser consideradas (SÁ, 1999):

- o processo de análise considera que as expectativas de retorno referem-se ao período seguinte ao atual;
- os investidores buscam a maximização da utilidade esperada do retorno e não a maximização do próprio retorno. Além disso, a utilidade marginal decresce com o aumento da riqueza;

- as estimativas de rentabilidade dos ativos são geradas a partir da distribuição de probabilidades para os retornos que podem ser alcançados;
- risco significa variabilidade das taxas de retorno. Quanto mais voláteis forem os retornos, maior será o risco do investimento;
- a análise dos investimentos só necessita das variáveis retorno esperado e risco;
- para qualquer nível de risco, os investidores preferem maiores retornos a menores retornos.

Outra premissa a ser levada em conta é o fato de o modelo considerar que os investidores são avessos ao risco, ou seja, os investidores não aceitariam participar de um jogo cujo retorno esperado é igual a zero (SHARPE et al., 1999). Com isso, pode-se dizer que o investidor prefere sempre menos risco a mais risco.

Independentemente da necessidade ou não da verificação dessas premissas, a aplicação da teoria de carteiras de Markovitz exige que alguns dados sejam fornecidos como entrada. Esses dados são os retornos esperados de cada título, as variâncias desses retornos e a covariância entre cada par de títulos (SHARPE et al., 1999).

Uma consideração importante observada pelo próprio Markovitz (1959) é que seu modelo considera que os valores fornecidos como entrada refletem bem a realidade. Ou seja, se os dados de entrada – retornos esperados, variâncias e covariâncias – estiverem fora da realidade, o modelo não gerará bons resultados.

Em sua forma original, os retornos esperados utilizados como entrada estão relacionados a um período futuro, sendo geralmente calculados a partir de distribuições

de probabilidades de possíveis retornos. A partir dessas distribuições, também é calculado o risco, dado pela variância dos retornos. Segundo Sharpe et al. (1999), o retorno esperado e o risco de cada título devem ser calculados por especialistas no setor da empresa que está sendo considerada, cabendo ao administrador de carteiras definir qual a melhor combinação possível desses títulos, dadas as previsões acerca de cada título individualmente.

O cálculo do retorno esperado e da variância desse retorno é dado pelas Equações 2.1 e 2.2, respectivamente:

$$E(R) = \sum_{t=1}^n P_i \times \text{Est}(R_i) \quad (2.1)$$

$$\sigma^2 = \sum_{t=1}^n P_i \times [\text{Est}(R_i) - E(R)]^2 \quad (2.2)$$

em que

$E(R)$  = retorno esperado do título

$P_i$  = probabilidade de cada retorno

$\text{Est}(R_i)$  = estimativa de retorno com probabilidade  $P_i$  de ocorrer

$\sigma^2$  = risco, dado pela variância da distribuição dos retornos.

Já as covariâncias são geralmente calculadas utilizando-se séries históricas. São, assim, dados que refletem o passado. Sua utilização, no modelo de Markovitz, tem como premissa a idéia de que covariâncias históricas podem representar de forma adequada as relações futuras entre os preços dos vários títulos.

O cálculo da covariância entre um título  $j$  e um título  $k$  é mostrado pela Equação 2.3:

$$\sigma_{jk} = \Sigma (R_j - E(R_j)).(R_k - E(R_k)) / N \quad (2.3)$$

em que

- $\sigma_{jk}$  = covariância entre os títulos  $j$  e  $k$
- $R_i$  = retorno de  $i$  em um período  $t$  do passado
- $E(R_i)$  = retorno esperado de  $i$  (média ponderada dos retornos)
- $N$  = número de observações.

Outra maneira de aplicar o método de Markovitz é utilizando-se apenas informações passadas sobre o comportamento dos preços dos ativos analisados, ao invés de previsões futuras sobre riscos e retornos. Esta é a versão *ex-ante* do método de Markovitz, bastante utilizada devido às dificuldades e subjetividades relacionadas à definição de cenários futuros para as variáveis de entrada. (BRUNI e FAMÁ, 1998).

O objetivo básico da formação de carteiras é permitir que o investidor tenha acesso a alternativas de investimento com níveis de risco reduzidos em comparação com os títulos, se considerados isoladamente. Isso é possível através da diversificação, processo pelo qual uma parte do risco não sistemático – aquele específico das empresas, que não se deve às condições do mercado como um todo – é eliminada ao se combinarem ativos não perfeitamente correlacionados (BREALEY e MYERS, 1998).

Para que se possam visualizar os resultados da diversificação, devem-se calcular o retorno esperado e o risco da carteira como um todo. É necessário saber como a combinação de vários títulos pode afetar o retorno e o risco do conjunto.

O cálculo do retorno da carteira é simples: é a média ponderada do retorno esperado para cada título:

$$E(R_c) = X_1 \cdot E(R_1) + X_2 \cdot E(R_2) + \dots + X_n \cdot E(R_n) \quad (2.4)$$

em que

$E(R_c)$  = retorno esperado da carteira

$E(R_i)$  = retorno esperado do título  $i$

$X_i$  = participação do título  $i$  na carteira.

O caso do risco é um pouco mais complexo. Para que esse cálculo seja possível, é necessário utilizar a covariância entre os retornos de cada par de títulos.

Assim, o risco de uma carteira composta de diferentes títulos pode ser calculado pela seguinte fórmula:

$$\sigma_c^2 = \sum_{j=1}^n \sum_{k=1}^n X_j X_k \sigma_{jk} \quad \text{para } k \neq j. \quad (2.5)$$

em que

$\sigma_c^2$  = risco da carteira, dado pela variância

$X_i$  = participação do título  $i$  na carteira

$\sigma_{jk}$  = covariância entre os títulos  $j$  e  $k$ .

O teorema da fronteira eficiente diz que um investidor deve escolher sua carteira ótima a partir de um conjunto de carteiras que oferecem (SHARPE et al., 1999):

- os maiores retornos esperados dados diferentes níveis de risco;
- os menores riscos para diferentes níveis de retorno esperado.

A FIGURA 1 ilustra a fronteira eficiente (entre A e B) num gráfico *retorno esperado X risco*.

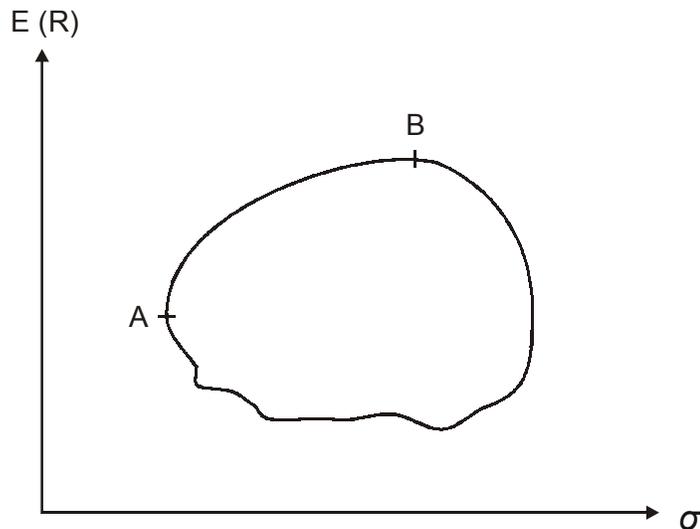


FIGURA 1 – Fronteira eficiente de ativos de risco

FONTE – Adaptado de Sá, 1999, p.61.

Para que a fronteira eficiente possa ser construída, deve-se ter em mãos o retorno esperado de cada título, a variância dos retornos e uma matriz de covariância. Essa matriz apresenta os valores das covariâncias entre cada par de títulos. Se há  $n$  títulos passíveis de serem incluídos, serão necessários  $(n^2 - n)/2$  covariâncias (ROSS et al., 1995).

O processo de formação dessa fronteira inicia-se determinando os percentuais investidos em cada título, de modo que cada nível de retorno esperado da carteira apresente o menor risco possível. Além disso, o somatório das participações dos títulos na formação da carteira deve ser igual a 1.

O processo de geração da fronteira foi resolvido por Markovitz através do método da programação quadrática. Outros métodos surgiram mais tarde, alcançando, de uma maneira mais simples, os mesmos resultados.

O método da programação quadrática forma a fronteira eficiente através da identificação de algumas carteiras de características especiais, chamadas *carteiras de canto*. Após determinar a primeira carteira de canto (a de maior retorno esperado), o método inclui um novo título e acha a próxima carteira, logo a seguir. À medida que novos títulos vão sendo incorporados, novas carteiras de canto vão sendo geradas e a fronteira eficiente vai sendo formada. O algoritmo continua até que a carteira de menor variância seja determinada. Como as combinações de carteiras de canto adjacentes também representam carteiras eficientes, a fronteira está formada.

Gerada a fronteira eficiente, o investidor deve escolher uma das carteiras que a compõem, baseado em algum critério que leve em consideração a relação entre risco e retorno. De uma forma geral, quanto mais avesso ao risco, mais ele tenderá a escolher uma carteira na parte esquerda da fronteira eficiente, em que o risco é menor. Quanto menos avesso ao risco, mais ele tenderá a escolher uma carteira na parte direita da fronteira, cujos retornos são maiores.

## 2.2 Outros modelos

### 2.2.1 Modelo de Markovitz com ativos sem risco

Markovitz construiu sua teoria considerando apenas a possibilidade de investir em títulos de risco. Em seu trabalho, os títulos sem risco não entravam na composição da carteira. Ao considerarmos as possibilidades de o investidor utilizar ativos sem risco em sua carteira e pegar emprestado para alavancar os investimentos, a fronteira eficiente de Markovitz é descaracterizada, tomando a forma da FIGURA 2.

A nova fronteira eficiente é agora representada por uma linha reta, que tem início no ponto  $R_f$  e tangencia no ponto M a antiga fronteira eficiente. Pode-se ver pela FIGURA 2 que essas carteiras apresentam, para um mesmo nível de risco, um maior retorno esperado que as carteiras da fronteira de Markovitz.

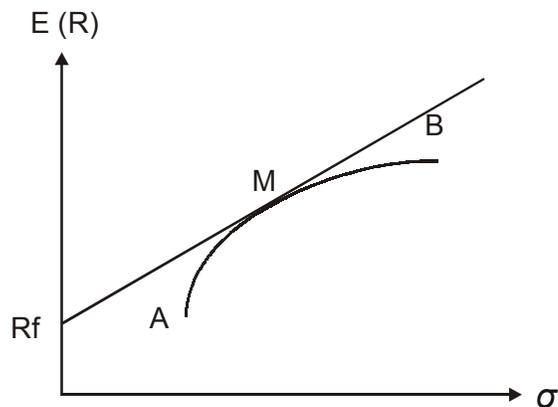


FIGURA 2 – Fronteira eficiente de ativos com e sem risco

FONTE – Adaptado de Sá, 1999, p.70.

Uma composição entre a carteira com risco do ponto M e títulos sem risco está situada entre os pontos  $R_f$  e M. Investimentos feitos apenas na carteira em M, utilizando uma combinação de recursos próprios e recursos emprestados a uma taxa  $R_f$ , estão situados à direita de M, na nova fronteira.

### 2.2.2 Modelo de índice único de Sharpe

A necessidade de estimar as covariâncias entre cada par de títulos é um grande problema para determinar, através dos métodos mencionados, a fronteira eficiente dos títulos (ELTON e GRUBER, 1995). O modelo de índice único de Sharpe acha as carteiras eficientes de uma maneira menos complexa, tornando mais fácil sua utilização na prática.

Ao invés de relacionar aos pares os retornos dos títulos, este modelo considera uma carteira que representa o mercado como um todo e relaciona o retorno de cada título com o retorno dessa carteira. Desse modo, cada título relaciona-se com os demais de forma indireta, utilizando para isso um índice único.

O modelo de índice único de Sharpe facilita a análise do investimento, já que não requer mais as covariâncias par a par entre os retornos dos títulos. A quantidade de entradas exigidas é muito menor que o exigido pelo modelo de Markovitz, e seus cálculos são muito mais simples. Entretanto, as carteiras geradas pelo modelo de índice único não são tão eficientes quanto aquelas geradas pelo modelo de Markovitz (SÁ, 1999).

### 2.2.3 Modelo de índices múltiplos

O modelo de índices múltiplos também busca as carteiras eficientes de uma maneira mais simples, utilizando mais de um índice para relacionar os retornos esperados dos títulos. Para isso, são identificados vários fatores que podem influenciar no desempenho das empresas emissoras, de seu setor de atividade ou do mercado como um todo.

Este modelo tem como premissa a idéia de que os fatores identificados não estão correlacionados entre si. O problema deste modelo, entretanto, é identificar quais são esses fatores que influenciam o retorno das ações e qual a intensidade dessas influências, o que não é tarefa simples (SÁ, 1999).

### 3 ALGORITMOS GENÉTICOS

Os algoritmos genéticos são algoritmos que utilizam idéias da seleção natural e da genética no processo de busca por soluções para um problema. Funcionam de modo similar ao processo de evolução dos organismos biológicos, em que as populações, através das gerações, evoluem de acordo com o princípio da seleção natural, proposto por Charles Darwin em 1859, em *A origem das espécies*.

A utilização das idéias da genética e da seleção natural na solução de problemas reais tem como base trabalhos de Holland (1975), que mostram como o processo evolucionário pode ser aplicado aos sistemas artificiais (KOZA, 1992).

#### 3.1 Busca e otimização

Os algoritmos genéticos são métodos adaptativos utilizados com frequência em problemas de busca e otimização (Beasley et al., 1993b). Um *problema de busca* é caracterizado pela necessidade de se vasculhar o conjunto de possíveis soluções para o problema (chamado *espaço de soluções*, ou *espaço de busca*) e achar aquelas soluções que lhe são satisfatórias. Num *problema de otimização*, busca-se, nesse mesmo espaço de soluções, aquele ponto que representa a melhor resolução (GOLDBERG, 1989).

Para realizar uma busca, o algoritmo pode recorrer a heurísticas que facilitem o seu trabalho. Se o espaço de soluções é muito grande, de forma que uma busca por todos os seus pontos não seja viável, é necessário que alguma regra (ou algum conjunto de regras) seja utilizada, permitindo que o algoritmo vasculhe esse mesmo espaço sem ter que percorrer todos os seus pontos (MITCHELL, 1996).

Segundo Beasley et al. (1993b), não se pode garantir que os algoritmos genéticos encontrarão sempre a solução ótima para o problema, mas pode-se dizer que eles se mostram eficientes em encontrar soluções boas em um tempo aceitável.

### 3.2 Um algoritmo genético simples

O objetivo básico dos algoritmos genéticos é fazer com que as soluções propostas para o problema a ser resolvido evoluam com o tempo. Esse objetivo é alcançado permitindo-se que as soluções mais adaptadas ao problema tenham maiores chances de sobreviver e, com isso, maiores oportunidades de gerar soluções descendentes, parecidas a elas mesmas e com grandes possibilidades de estarem mais adaptadas.

Em cada passo do processo de evolução, essas soluções têm suas estruturas ligeiramente alteradas, fazendo com que as soluções iniciais e aleatórias evoluam e convirjam para uma solução ótima, ou quase ótima, ao final do processo.

Essas soluções podem também ser chamadas de *indivíduos* ou *soluções candidatas* e, quando agrupadas num mesmo período do tempo, formam uma *população*. Esses indivíduos, candidatos a resolverem o problema proposto, nada mais são do que uma codificação das variáveis do problema, que é feita utilizando-se um conjunto de *cadeias de caracteres*, ou *strings* (GOLDBERG, 1989). Assim como ocorre no mundo natural, onde os cromossomos são *strings* formadas por seqüências dos símbolos A, C, G e T, os caracteres que formam essas *strings* devem pertencer a um alfabeto de tamanho finito.

Os indivíduos agrupados em uma população e em determinado período do tempo formam uma *geração*, que é sempre substituída no período seguinte por seus

descendentes. Para que a nova geração seja formada, os indivíduos da geração atual são selecionados conforme suas capacidades de sobreviver e, com isso, reproduzir e gerar descendentes, sendo que a determinação dessa capacidade é realizada pela *função de adequação*. É essa função que diz quão adaptado ao ambiente está cada indivíduo da população. Seu valor resultante também é chamado de *grau de adequação*.

Os indivíduos que se desempenham melhor que os outros na competição pelos recursos existentes em seu ambiente propagam mais do seu material genético do que a média da população (GOLDBERG, 1989). Com isso, a geração seguinte será sempre composta por uma porcentagem maior de descendentes dos indivíduos melhor adaptados. Esses descendentes, por sofrerem pequenas mudanças em relação à estrutura de seus pais, poderão ter uma adaptação ainda maior ao problema, que seria propagada outra vez à geração seguinte e dessa à seguinte e assim por diante, até que a população consiga convergir para uma solução satisfatória ou algum outro critério de parada seja satisfeito.

Dependendo da situação específica em que for aplicado, o algoritmo genético pode dispor de estrutura e mecanismo de funcionamento diferentes. Um algoritmo genético básico, entretanto, é sugerido por Koza (1992):

1. Crie aleatoriamente uma população inicial de indivíduos com *strings* de caracteres de tamanho fixo.
2. Siga os passos abaixo até que um critério de término tenha sido satisfeito:
  - a) avalie a adequação de cada indivíduo na população;
  - b) crie uma nova população de *strings* aplicando, pelo menos, as duas primeiras das três operações seguintes. As operações são aplicadas a *strings* escolhidas

proporcionalmente à sua adequação ao ambiente.

- I copie as *strings* existentes na nova população;
- II crie duas novas *strings* recombinação geneticamente duas *strings* existentes;
- III crie uma nova *string* através da mutação de um dos caracteres de uma *string* existente.

3. A *string* gerada nesse processo que apresentar melhor adaptação ao ambiente é o resultado do algoritmo genético para essa execução.

O algoritmo proposto acima é um algoritmo simples, sendo diversas as suas variações. Segundo Koza (1992), o que vai definir a seqüência adequada de passos a serem seguidos e os parâmetros específicos para o algoritmo é a especificidade do problema com o qual se está tratando.

### 3.3 Codificação das soluções

Junto com a determinação da função de adequação, a codificação das soluções é o fator de maior importância na determinação do sucesso ou o fracasso de um algoritmo genético (BEASLEY et al., 1993b). Para perceber isso, basta ver que estes algoritmos não trabalham com as soluções do problema diretamente, mas, sim, com representações dessas soluções (GOLDBERG, 1998).

Há diversas maneiras de realizar essa codificação: utilizando-se *strings* binárias, grafos, listas ordenadas, expressões em Lisp ou vetores de caracteres reais, por exemplo (SPEARS, 1998).

A maioria dos algoritmos genéticos codifica as soluções candidatas representando-as através de *strings* de tamanho e ordem fixos, geralmente utilizando apenas os símbolos 0 e 1. Nesse último caso, diz-se que a codificação usada foi a binária. Mas como a forma de codificar as variáveis depende muito da aplicação em que se vai trabalhar, um alfabeto de apenas dois símbolos pode se mostrar insuficiente em vários tipos de problemas. Nesses casos, seria desejável a utilização de um alfabeto com mais símbolos.

Além disso, um problema pode ter parte de suas especificidades perdidas caso suas variáveis sejam codificadas como números inteiros. Para casos como esse, deve-se realizar uma codificação que reflita a possibilidade de os valores serem números reais.

Alguns estudos já foram feitos no intuito de comparar o desempenho dos algoritmos genéticos quanto ao alfabeto utilizado (por exemplo, Yao, 1993), confirmando que a utilização do alfabeto mínimo, com apenas dois símbolos, geralmente não é a melhor opção.

Outra forma possível de realizar a codificação é utilizando-se uma estrutura em árvore, em que as variáveis do problema são representadas por nodos conectados entre si, de forma a criar uma hierarquia. Nessa hierarquia, cada nodo é conectado apenas ao seu pai e aos nodos filhos, caso os possua, o que quer dizer que, na estrutura final, não é possível a presença de qualquer ciclo. Segundo Mitchell (1996), uma codificação em árvore apresenta várias vantagens sobre as outras maneiras de codificar as soluções em um algoritmo genético, sendo uma dessas vantagens o fato de que qualquer tamanho de árvore pode ser gerado através do operador que realiza a recombinação genética, o que dá maior flexibilidade ao algoritmo. Mas a escolha dessa estrutura também implica aspectos negativos. Um deles é que as árvores podem crescer de forma descontrolada, o que evitaria a formação de soluções candidatas com estruturas bem definidas.

Além das formas de codificação mencionadas, há várias outras maneiras de codificar o problema. Conforme já mencionado, não existe uma definição de qual seja o melhor método a ser utilizado. Segundo Davis (1991), a melhor codificação é aquela que parece a mais natural para o problema a ser resolvido.

Uma possível solução para a tarefa de encontrar a codificação adequada para o problema pode ser a utilização de um outro algoritmo genético. Nesse caso, o algoritmo original teria a sua estrutura codificada de uma forma qualquer, inicialmente, e um processo de evolução de sua própria codificação seria realizado, de modo que o melhor esquema de codificação para o problema pudesse ser encontrado.

### 3.4 Métodos de seleção

Outra importante decisão a ser tomada ao construir um algoritmo genético refere-se ao modo de selecionar os indivíduos da população que irão reproduzir-se e gerar descendentes.

Segundo Goldberg (1994), a seleção nada mais é do que a sobrevivência e reprodução dos mais adaptados ao ambiente. Nos algoritmos genéticos, esses indivíduos mais adaptados têm maior probabilidade de gerarem descendentes do que a média da população. A geração seguinte, com um maior número de descendentes de bons pais, contará com uma adequação ao ambiente pelo menos tão boa quanto a anterior. A tendência, dessa forma, é que com o tempo as novas gerações vão melhorando esse grau de adequação.

Um ponto que vale ser destacado é que o método de seleção utilizado no algoritmo genético deve estar em harmonia com os métodos utilizados para realizar-se a

recombinação genética e a mutação. Caso a seleção dê uma importância além da necessária aos indivíduos mais adaptados, por exemplo, o algoritmo genético pode convergir muito rapidamente para uma solução final, o que significa que uma boa solução provavelmente não tenha sido encontrada. No extremo oposto, se não for dada muita importância aos indivíduos mais adaptados no processo de seleção, provavelmente o algoritmo genético demorará muito para convergir para uma solução, o que também não seria interessante.

Nesse sentido, a determinação de qual função de adequação utilizar é essencial ao sucesso do algoritmo, já que é essa função que determinará quanto cada indivíduo avaliado está adaptado ao ambiente. Beasley et al. (1993b) propõem que se construam funções de adequação que não tenham muitos máximos locais, ou um máximo global muito isolado.

Segundo Mitchell (1996), analogamente ao que ocorre com os métodos de codificação, há várias maneiras de realizar a seleção dos indivíduos que gerarão descendentes para a próxima geração da população, ainda não havendo consenso entre os autores sobre a melhor forma de se realizar a seleção, dado o tipo de problema com o qual se depara. Algumas técnicas para realizar a seleção desses indivíduos, descritas por Mitchell (1996), são relacionadas a seguir:

#### Adequação proporcional

Segundo a autora, essa é a forma mais comum de realizar a seleção. Sua idéia básica é que um indivíduo é escolhido para reproduzir e gerar descendentes proporcionalmente ao seu grau de adequação ao ambiente. Ou seja, a quantidade de vezes em que isso ocorre é igual ao grau de adequação do indivíduo dividido pela adequação média da população.

O método mais comum de implementar esse mecanismo é o da roleta. Nesse método, cada indivíduo da população é representado por uma *fatia* da roleta, proporcional ao seu grau de adequação ao ambiente. A roleta é *girada* (obtendo-se um número aleatório representado na roleta) tantas vezes quantos indivíduos existirem na população. Os indivíduos que tiverem as fatias da roleta onde a roleta *parou* são os escolhidos para produzirem descendentes. Então, quanto maior a adequação do indivíduo, maior é a sua fatia na roleta, e, conseqüentemente, maior a chance de essa parar em sua área.

Este método resulta, estatisticamente, no número esperado de descendentes para cada indivíduo. Segundo Mitchell (1996), deve-se apenas tomar cuidado com o fato de que a utilização de pequenas populações pelo algoritmo genético pode tornar esse número esperado estatisticamente inválido, fazendo com que o número de descendentes gerados para cada indivíduo situe-se longe do valor esperado.

Outro fator negativo na utilização desse método é a possibilidade de convergência prematura da população. Nesse caso, aqueles indivíduos mais adaptados seriam escolhidos com uma frequência muito maior do que os outros, fazendo com que a evolução se desse de maneira muito rápida. Assim, não daria tempo para que todas as regiões do espaço de busca fossem exploradas. A causa dessa tendência à convergência prematura, então, é o fato de a taxa de evolução da população depender exclusivamente da variância dos graus de adequação dos indivíduos ao ambiente. O método da roleta, entretanto, já apresenta variações que contornam os problemas mencionados.

### Torneio

Nesse método, vários indivíduos são escolhidos aleatoriamente na população. Os indivíduos pertencentes ao grupo gerado irão competir entre si para ver quais serão

selecionados para reprodução (MITCHELL, 1996). Nessa competição, quanto mais adaptado for o indivíduo, maiores as chances de ele ser escolhido para integrar a geração seguinte.

### Escala

Para que o problema da convergência prematura pudesse ser resolvido, foram criados vários métodos que fazem um mapeamento entre os valores de adequação e os valores esperados, sendo a escala sigma um desses métodos. Segundo a autora, o que esse método faz é manter constante a taxa com que os indivíduos mais adaptados são selecionados, durante toda a execução do algoritmo. Assim, essa taxa não dependerá mais das variâncias das adequações na população. A adequação do indivíduo, a média da população e o desvio padrão da adequação na população são os fatores que influenciam o valor esperado de um indivíduo nessa escala.

### Ranking

Esse método também tem como propósito principal evitar uma rápida convergência do algoritmo genético. Aqui, os indivíduos são ordenados inicialmente de acordo com o seu grau de adequação ao ambiente. O valor esperado para a sua probabilidade de seleção é, então, definido a partir da sua posição nesse *ranking* e não mais a partir do valor absoluto da sua adequação. Com isso, as diferenças entre as adequações não são mais tão significativas quanto antes. O maior mérito desse método, portanto, é diminuir a velocidade de evolução quando a variância nos graus de adequação é muito grande.

Segundo a autora, entretanto, essa abordagem tem a desvantagem de fazer com que o algoritmo genético possa demorar muito para encontrar a solução final para o problema. O que se vê, então, é uma troca entre velocidade de execução do algoritmo e precisão na solução.

### Steady-State

Na grande maioria dos métodos de seleção, as populações geradas consistem exclusivamente de indivíduos descendentes da geração anterior. O *steady-state*, entretanto, utiliza uma abordagem diferente: geralmente, indivíduos da geração atual sobrevivem e permanecem na geração seguinte. Apenas uma minoria – os menos adaptados ao ambiente – é substituída, dando lugar a novos indivíduos.

### Elitismo

O elitismo pode ser considerado um adendo aos métodos tradicionais existentes. Sua ideia básica, segundo a autora, é garantir que os indivíduos com maiores graus de adequação sejam mantidos na população seguinte. Essa tática é válida, já que esses indivíduos podem ser perdidos caso não sejam selecionados para a reprodução (uma possibilidade remota) ou caso eles sejam modificados pela recombinação ou pela mutação.

### Seleção Boltzmann

O propósito deste método de seleção é fazer com que a velocidade de evolução das soluções seja diferente ao longo da execução do algoritmo. No início, esse método imprime um menor ritmo na evolução, permitindo que os indivíduos menos adaptados sejam selecionados quase na mesma frequência que os mais adaptados. Com isso, a variabilidade das soluções propostas cresce durante esse período. À medida que o tempo vai passando, a velocidade da evolução vai aumentando, até que a convergência do algoritmo seja atingida. Para a autora, esse método considera que, durante a fase inicial do processo, quando se vê um menor ritmo no processo de evolução, a população consegue encontrar a região correta do espaço de busca a ser explorada. A

partir desse ponto, então, a velocidade de evolução já pode ser mais rápida, pois apenas uma pequena parte do espaço de soluções necessitará ser vasculhada.

### 3.5 Operadores genéticos

Os operadores genéticos mais comumente encontrados nos trabalhos ligados ao tema são a recombinação e a mutação. Apesar de outros operadores também existirem, esses dois são, de longe, os mais utilizados.

#### 3.5.1 Recombinação

Segundo Goldberg (1994), a recombinação, ou *crossover*, é o principal operador utilizado pelos algoritmos genéticos. Sua tarefa é misturar o material genético dos pais na geração de seus descendentes.

A recombinação geralmente é realizada escolhendo-se aleatoriamente um ponto de corte na *string* que representa o indivíduo. Se realizado dessa forma, o primeiro filho gerado teria os genes de um dos pais à esquerda desse ponto e os genes do outro pai à direita. O segundo filho teria tratamento semelhante, mas utilizando as partes contrárias dos cromossomos dos pais.

Às vezes, mais de um ponto de corte podem ser definidos. DeJong (1975) observa que a utilização de dois pontos de corte gera uma melhora nos resultados, mas a inserção de quaisquer novos pontos geralmente diminui o desempenho do algoritmo genético.

Outra forma de realizar a recombinação é utilizando-se uma máscara (BEASLEY et al., 1993a). A máscara é uma seqüência de caracteres do mesmo tamanho dos indivíduos da população e com apenas dois símbolos como alfabeto. Ela funciona da seguinte

maneira: se o primeiro caractere da máscara for um 0, o descendente n° 1 receberá seu primeiro caractere do primeiro pai. Se ele for um 1, receberá do outro. O raciocínio pode ser seguido para cada um dos caracteres desse descendente. O outro descendente, dessa forma, teria seus caracteres herdados sempre do pai que não replicou seu caractere no primeiro filho.

Esse processo faz com que a chance de que um caractere dos filhos tivesse como origem um pai ou o outro fosse a mesma, não sendo mais tão freqüente ver-se caracteres adjacentes do filho apresentando a mesma origem.

### 3.5.2 Mutação

O outro operador genético freqüentemente utilizado é a mutação. A mutação consiste, basicamente, na troca do valor de um dos caracteres de um indivíduo (GOLDBERG, 1994). Se o alfabeto utilizado na codificação for binário, um 0 seria trocado por um 1, e vice-versa. Geralmente, após realizada a reprodução, cada novo indivíduo da população teria uma certa probabilidade de ter um de seus caracteres modificados. Essa probabilidade pode variar de problema para problema, mas esse valor deve ser sempre muito baixo, algo em torno de 0,001.

Segundo Beasley et al. (1993b), a visão tradicional na área é que a recombinação é o principal agente responsável pela variabilidade das soluções, por explorar o espaço completo de busca de uma forma extremamente rápida.

A mutação, dessa forma, ficaria em segundo plano no processo de otimização realizado pelos algoritmos genéticos. Mas é a mutação, entretanto, que garante que todos os pontos do espaço de busca tenham alguma chance de serem considerados como solução para o problema. Com isso, a convergência do algoritmo para um mínimo local

se torna mais difícil, já que soluções candidatas não tenderão tanto a permanecerem fixas em uma mesma região do espaço.

Mas devido ao seu caráter de busca aleatória, a utilização da mutação não pode ser feita de uma forma exagerada, pois isso poderia descaracterizar o algoritmo genético e torná-lo um método quase aleatório, o que seria prejudicial à resolução do problema.

### 3.6 Eficácia dos algoritmos genéticos

Os algoritmos genéticos fazem parte do conjunto de métodos chamados *métodos genéricos*, por poderem ser aplicados em diversas áreas do conhecimento humano. Sua utilização, entretanto, só faz sentido em problemas com determinadas características definidas:

- o espaço de soluções é demasiado grande para a realização de uma busca exaustiva;
- a teoria relativa ao campo do conhecimento específico ainda não consegue explicar de forma exata o comportamento dos fenômenos observados.

Se o primeiro caso não fosse verdadeiro (ou seja, se o espaço de soluções fosse pequeno), não faria muito sentido utilizarmos um método heurístico como os algoritmos genéticos, já que uma busca por todos os pontos do espaço seria simples, viável e acharia a solução ótima num curto espaço de tempo.

O segundo ponto diz que os métodos genéricos de busca e otimização podem ser utilizados nos casos em que os modelos teóricos construídos para explicar os fenômenos observados no mundo real e realizar projeções sobre o futuro ainda não são

eficientes em seus propósitos. Nesses casos, enquanto a teoria específica para esse campo do conhecimento não conseguir dar tais explicações, os métodos genéricos podem ser utilizados, já que conseguem, algumas vezes, obter resultados melhores que os métodos específicos.

Mas para os casos em que o espaço de soluções é bem conhecido e já existam técnicas específicas que resolvam o problema, essas técnicas provavelmente terão um desempenho muito melhor que um método como os algoritmos genéticos, considerando-se tanto o tempo de processamento quanto a precisão nas respostas (BEASLEY et al., 1993b).

Os algoritmos genéticos têm demonstrado, e com sucesso, sua aplicabilidade em um grande número de aplicações práticas (KHURI et al., 1994). Por serem simples computacionalmente e extremamente poderosos na busca por melhorias nas soluções, os algoritmos genéticos vêm sendo a cada dia utilizados em novas aplicações. Seu desempenho, entretanto, vai depender muito de como foi feita a codificação das variáveis e dos indivíduos, de quais são os operadores utilizados e de como foram definidos os parâmetros.

### 3.7 Algoritmos genéticos X outros métodos de otimização

Além dos algoritmos genéticos, há vários outros métodos genéricos que também podem ser utilizados em problemas de busca e otimização. Todos têm em comum o fato de percorrerem o espaço de soluções utilizando heurísticas próprias. Essas heurísticas lhes permitem percorrerem apenas uma fração desse espaço, sem que a precisão da resposta seja muito comprometida.

Há alguns aspectos, entretanto, que diferenciam os algoritmos genéticos dos métodos tradicionais de busca e otimização, fazendo com que os mesmos sejam preferidos em várias situações.

Para Goldberg (1989), os algoritmos genéticos são diferentes dos outros métodos de busca e otimização devido a quatro fatores principais:

- os algoritmos genéticos trabalham com a codificação dos parâmetros do problema e não com os parâmetros em si;
- os algoritmos genéticos realizam a busca através de uma população de pontos do espaço e não de apenas um ponto;
- a única informação externa de que precisam os algoritmos genéticos é o grau de adequação do indivíduo ao ambiente, dado pela função de adequação;
- os algoritmos genéticos utilizam regras probabilísticas de transição, e não regras determinísticas.

O primeiro tópico diz respeito ao fato de os algoritmos genéticos não trabalharem diretamente sobre as variáveis do problema; ou seja, eles não mudam os valores dessas variáveis com o objetivo de maximizar a função de adequação. Os algoritmos genéticos, na verdade, trabalham sobre as codificações dessas variáveis, o que faz com que várias das limitações encontradas nos métodos tradicionais de otimização não sejam problemas para os algoritmos genéticos. Os algoritmos genéticos necessitam apenas que o conjunto das variáveis do problema de otimização seja codificado como uma *string* de tamanho finito sobre um alfabeto também finito de símbolos.

O segundo fator refere-se à técnica utilizada para se determinar o caminho a ser percorrido por estes métodos no espaço total de soluções. A busca, na grande maioria desses métodos, consiste em mover-se de ponto em ponto no espaço de soluções, partindo-se sempre do ponto atual e chegando-se a um ponto próximo. Para isso, essas técnicas utilizam alguma regra que determina em que direção seguir e onde parar. A utilização desses métodos ponto-a-ponto pode não ser muito adequada, dependendo da necessidade de precisão do problema em questão, já que o método pode dar como resposta o ponto ótimo de uma determinada sub-região do espaço. Ou seja, o método pode achar um ótimo local em um espaço de busca em que há vários máximos locais.

Os algoritmos genéticos, ao contrário, não utilizam essa abordagem ponto-a-ponto. Sua técnica consiste em trabalhar sobre vários pontos simultaneamente, *escalando vários picos* ao mesmo tempo. Dessa forma, a probabilidade de este método resultar em um máximo local é reduzida de forma drástica.

Outro fator que diferencia os algoritmos genéticos é que eles não precisam de muitas informações externas para realizarem as suas tarefas. Para fazerem a busca por melhores soluções, os algoritmos genéticos requerem apenas os valores resultantes de uma função que mede a adequação dos indivíduos ao ambiente.

O último fator apontado por Goldberg (1989) é o fato de os algoritmos genéticos utilizarem regras probabilísticas na transição entre os pontos no espaço e, não, regras determinísticas. Com isso, os algoritmos genéticos recebem imensa ajuda em sua tarefa de procurar soluções por todas as regiões do espaço.

O uso de probabilidades, entretanto, não implica que o método tenha um caráter meramente aleatório: os algoritmos genéticos não são tão simples como um *random walk* e nem tampouco funcionam da mesma maneira. O que eles fazem é explorar de um modo eficiente as informações históricas de sua busca para *saltarem* para novos

pontos do espaço com um desempenho melhorado. Os algoritmos genéticos utilizam processos estocásticos, mas seus resultados não são aleatórios.

Juntos, esses quatro fatores diferenciadores – uso de codificação, busca sobre vários pontos, pouca necessidade de informações externas e uso de operadores aleatórios – contribuem para a robustez dos algoritmos genéticos e resultam em uma vantagem sobre os outros métodos tradicionalmente utilizados.

## 4 REDES NEURAIS

As redes neurais são sistemas paralelos distribuídos compostos por unidades de processamento simples – os nodos – que calculam determinadas funções matemáticas, geralmente não-lineares (BRAGA et al., 2000).

Elas têm como base estudos realizados sobre a estrutura e funcionamento do cérebro humano, sofrendo uma grande simplificação em relação ao modelo original (Osório e Vieira, 1999).

Sua principal característica é a capacidade de aprender, através de um treinamento exaustivo, campos específicos de conhecimento. Em grande parte das áreas em que são aplicadas, as redes neurais têm mostrado um excelente desempenho quando comparadas com as abordagens convencionais.

### 4.1 Principais componentes e organização

#### 4.1.1 Unidade de processamento

O componente básico das redes neurais é a unidade de processamento ou nodo. Essa unidade tem papel equivalente ao desempenhado por um neurônio no cérebro humano, sendo, por isso, usualmente chamada de neurônio. É essa unidade a responsável pela computação dos sinais dentro da rede. A FIGURA 3 ilustra uma unidade de processamento simples e sua inter-relação com os nodos da camada anterior.

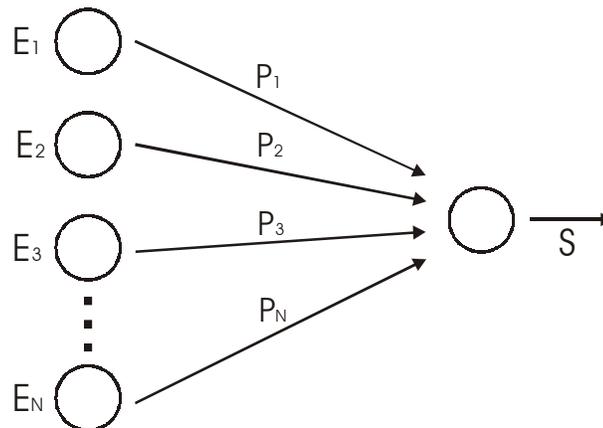


FIGURA 3 – Unidade de processamento simples

FONTE – Adaptado de Hawley, 1996, p. 30.

Haykin (1999) identifica alguns elementos básicos na estrutura e funcionamento de uma unidade de processamento:

- um conjunto de conexões, ou sinapses, cada uma com um peso específico. Um sinal  $E_j$  na entrada da sinapse  $j$  conectada à unidade de processamento é multiplicada pelo peso sináptico  $P_j$ ;
- um somador, para somar todos os sinais de entrada já multiplicados pelos respectivos pesos;
- uma função de ativação, para limitar a amplitude da saída da unidade de processamento;
- um viés (*bias*), que é um elemento externo que tem o efeito de aumentar ou diminuir a entrada da função de ativação.

Segundo o autor, a unidade de processamento  $k$  pode ser descrita pelas seguintes equações, em termos matemáticos:

$$u_k = \sum_{j=1}^m P_{kj} E_j \quad (4.1)$$

$$v_k = u_k + b_k \quad (4.2)$$

$$S_k = \varphi(v_k) \quad (4.3)$$

em que

$E_1, E_2, \dots, E_m$	= sinais de entrada
$P_{k1}, P_{k2}, \dots, P_{km}$	= pesos das conexões à unidade $k$
$u_k,$	= combinação linear das entradas, já pesadas
$b_k,$	= viés
$v_k,$	= soma das entradas pesadas mais o viés
$\varphi()$	= função de ativação
$S_k,$	= saída final da unidade de processamento.

O funcionamento das unidades de processamento se dá da seguinte forma: cada uma recebe um ou mais sinais de entrada, provenientes de unidades a ela conectadas. Esses sinais são multiplicados pelos pesos de suas respectivas conexões e somados, produzindo o chamado *nível de atividade*. O valor resultante será usado como entrada da função de ativação, que o processará e gerará a saída do neurônio. Essa saída, então, será repassada às unidades de processamento seguintes.

#### 4.1.2 Função de ativação

Segundo Osório e Vieira (1999), a *função de ativação* determina a ativação de um neurônio em função da influência vinda de suas entradas, ponderadas pelos seus respectivos pesos.

Haykin (1999) destaca, dentre os diversos tipos de funções de ativação, as funções degrau, linear e sigmoidal:

##### Função degrau

Esse é o tipo mais simples de função de ativação. A saída dessa função só pode ser 0 ou 1, e é determinada por um valor que, se ultrapassado, implica a propagação de um sinal 1. Em caso contrário, propaga-se um sinal 0. Um neurônio com a estrutura mostrada anteriormente e com essa função de ativação também é chamado de McCulloch-Pitts.

A função degrau, para um limite igual a 0, é descrita como:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (4.4)$$

##### Função linear

Essa função define duas regiões cujas saídas são os limites máximo e mínimo e uma em que a saída varia linearmente com o  $v$ . A função Piecewise-linear é um exemplo, definindo a região  $[-\frac{1}{2}$  e  $\frac{1}{2}]$  como linear.

A função linear pode ser descrita como:

$$\begin{aligned} \Phi(v) &= 1, & \text{se } v > \frac{1}{2} \\ &v, & \text{se } \frac{1}{2} > v > -\frac{1}{2} \\ &0, & \text{se } v < -\frac{1}{2} \end{aligned} \quad (4.5)$$

### Função sigmoidal

É a função mais comum em trabalhos que envolvem redes neurais e tem um gráfico em forma de S. Exemplos de funções sigmoidais são a tangente hiperbólica e a função logística.

A função tangente hiperbólica é descrita pela seguinte equação:

$$\Phi(v) = \tanh(v) \quad (4.6)$$

Já a função logística é definida como:

$$\Phi(v) = \frac{1}{1 + \exp(-av)} \quad (4.7)$$

em que  $a$  = inclinação da função sigmoidal.

#### 4.1.3 Organização em camadas

As redes neurais são, de um modo geral, estruturadas em camadas, sendo que cada camada apresenta uma ou mais unidades de processamento. Toda rede tem pelo menos uma camada de entrada e uma de saída. A de entrada é aquela que recebe os

sinais do ambiente externo. A camada de saída é a última, aquela que guarda o resultado final da rede.

Além dessas duas, uma rede neural pode apresentar uma ou mais camadas intermediárias, que se situam entre as camadas de entrada e de saída. Segundo Russel e Norvig (1995), a utilização de uma única camada intermediária já permite que a saída da rede neural seja gerada como uma função contínua e não-linear das entradas.

Geralmente, as unidades de processamento das camadas intermediárias estão conectadas apenas às unidades das camadas adjacentes, embora também seja possível a existência de conexões entre elementos dentro de uma mesma camada ou de camadas mais distantes, como é o caso das redes recorrentes.

## 4.2 Arquiteturas

As redes neurais podem ter várias estruturas diferentes. Quanto ao número de camadas, podem ser classificadas em redes de camada única ou em redes multicamadas. Quanto ao padrão de conectividade entre suas unidades de processamento, são classificadas em redes *feed-forward* ou redes recorrentes.

### 4.2.1 Padrão de conectividade

A principal classificação das redes neurais, quando levamos em consideração sua estrutura, refere-se ao modo como as conexões entre as unidades podem ser realizadas. Nesse sentido, distinguem-se as redes neurais em redes *feed-forward* e redes recorrentes (RUSSEL e NORVIG, 1995).

### Redes *feed-forward*

As redes *feed-forward* são aquelas que não apresentam ciclos em sua estrutura. Numa rede como esta, um sinal que sai de um nodo nunca volta ao mesmo ponto. Além disso, as ligações entre os nodos são sempre unidirecionais; ou seja, os sinais vão de um nodo a outro em uma só direção, sem poderem voltar.

A estrutura das redes *feed-forward* é, na maioria das vezes, arranjada em camadas, como pode ser observado na FIGURA 4. A saída de um nodo na  $i$ -ésima camada da rede não pode ser usada como entrada de nodos em camadas de índice menor ou igual a  $i$  (BRAGA et al., 2000). Dessa forma, as unidades emitem sinais apenas para as unidades das camadas seguintes. Não é possível transmitir estímulos para a camada anterior e não existem ligações entre as unidades de processamento de uma mesma camada.

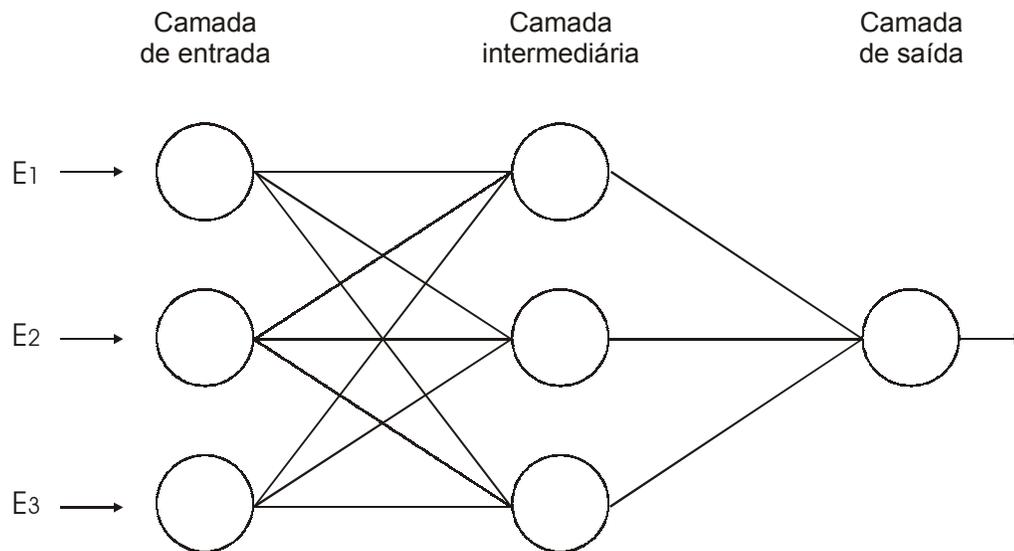


FIGURA 4 – Rede *feed-forward*

FONTE – Adaptado de RAHIMIAN et al., 1996, p. 247.

## Redes recorrentes

O que caracteriza as redes recorrentes é a presença de conexões que saem de uma unidade de processamento e vão em direção a outras unidades da mesma camada ou de camadas anteriores (OSÓRIO e VIEIRA, 1999).

Uma característica dessas redes, justamente por apresentar essa estrutura, é a possibilidade de se tornarem instáveis, exibindo um comportamento caótico (RUSSELL e NORVIG, 1995). Dependendo dos valores dados como entrada para o treinamento, a rede recorrente pode necessitar de um grande tempo de processamento para que a saída gerada seja estável. Segundo Osório e Vieira (1999), devido a essa instabilidade, os algoritmos de aprendizado utilizados pelas redes recorrentes devem ser específicos a suas estruturas, sendo geralmente bem mais complexos.

Por outro lado, essas redes podem ser mais adequadas que as *feed-forward* em problemas cujas funções que realizam o mapeamento entre entradas e saídas são mais complexas.

A forma mais conhecida de rede recorrente é a rede de Hopfield, que é caracterizada pela utilização de conexões bidirecionais entre as unidades e pesos simétricos. Uma característica da rede de Hopfield é o fato de que, depois de realizado o treinamento, qualquer novo estímulo levará à associação desse estímulo com o exemplo utilizado no treinamento que mais a ele se assemelha.

Segundo Russel e Norvig (1995), as redes de Hopfield têm a capacidade de armazenar com segurança informações referentes a uma quantidade de até  $0,138 \times n$  exemplos de treinamento, em que  $n$  é o número de unidades da rede.

#### 4.2.2 Número de camadas

##### Rede de única camada

Esta é a forma mais simples de rede neural, caracterizada pela ausência de camadas intermediárias e pela existência de conexões apenas da camada de entrada para a camada de saída. A denominação *rede de única camada* se dá porque apenas a camada de saída realiza alguma computação de sinais: as saídas geradas por essa camada já serão o resultado final da rede.

Um modelo muito conhecido de rede de camada única é o Perceptron. Segundo Portugal e Fernandes (1996), o Perceptron foi o primeiro modelo conexionista desenvolvido e se caracteriza por apresentar entradas binárias, saídas que assumem os valores +1 ou -1, função degrau como função de ativação, ausência de ciclos e aprendizado supervisionado.

Para Portugal e Fernandes (1996), uma rede neural com estrutura semelhante à do Perceptron pode ser facilmente relacionada ao modelo estatístico de regressão linear. Dessa forma, esses modelos só podem ser utilizados em problemas em que a função de mapeamento entrada/saída também seja linear, pois o processamento realizado se limita ao cálculo da saída como uma combinação linear das entradas.

##### Rede multicamadas

As redes multicamadas são caracterizadas pela presença de pelo menos uma camada intermediária. Segundo Portugal e Fernandes (1996), ao introduzirmos uma camada intermediária, estamos introduzindo não-linearidade na relação entre as entradas e a saída. Com isso, os modelos que utilizam redes multicamadas poderão ser aplicados à

resolução de problemas mais complexos, em que a relação entre as variáveis de entrada e as de saída não seja linear.

#### 4.2.3 Estrutura adequada

Segundo Balakrishnan e Honavar (1995), apesar das intensas pesquisas realizadas na área, o processo de desenvolvimento de uma rede neural ainda é, de uma forma geral, um processo de tentativa e erro, baseado especialmente em experiências do passado e modelagens de aplicações semelhantes.

Pelo fato de cada problema apresentar suas próprias características, pela freqüente presença de ruídos nos dados de entrada e pelas características específicas dos algoritmos de aprendizado, não há como dizer antes da realização dos testes qual é a melhor configuração da rede neural para determinado tipo de problema (KLIMASAUSKAS, 1994). O que deve ser feito é utilizarem-se heurísticas que dão algumas *dicas* de qual caminho deve ser seguido na busca da estrutura ideal de uma rede neural.

Assim, durante o desenvolvimento de qualquer modelo de rede neural torna-se necessária a realização de testes com várias configurações de rede diferentes, com o objetivo de encontrar aquela que mais se adapta ao problema. A estrutura adequada da rede neural, na verdade, é aquela que, após comparada com várias outras configurações, alcança os melhores resultados possíveis, usando os dados de testes.

Dessa forma, a estratégia a ser adotada na construção de uma rede neural deve ser utilizarem-se inicialmente as heurísticas existentes e, a partir daí, melhorar seu desempenho através de mudanças em sua configuração.

Uma possível solução para o problema da definição da estrutura adequada de uma rede neural pode ser encontrada através da utilização de um algoritmo genético (MITCHELL, 1996). Em situações em que o problema pode ser classificado como um problema de busca (nesse caso, busca-se a configuração adequada para a rede), os algoritmos genéticos têm-se mostrado extremamente eficientes.

### 4.3 Aprendizado

A principal característica de uma rede neural é a sua capacidade de aprender. Essa capacidade é adquirida exclusiva e exaustivamente através de treinamento e se dá através de um processo contínuo e iterativo em que os pesos de suas conexões são ajustados de acordo com os dados fornecidos. Segundo Osório e Vieira (1999), o aprendizado ocorre quando a rede neural atinge uma solução genérica para o problema, entendendo-se por generalização a capacidade de responder corretamente a exemplos não utilizados durante o treinamento.

Para que o aprendizado da rede neural possa ocorrer, é necessário que haja a disponibilidade de uma grande quantidade de dados a serem utilizados como entrada. Nem todos os dados disponíveis, entretanto, devem ser fornecidos ao treinamento da rede: uma parte deve ser separada para a realização de testes.

A realização de testes com a estrutura final da rede é de grande importância, pois é nessa fase que se faz a avaliação da adequação da rede neural ao problema que se quer tratar. Os dados utilizados nesse processo de testes devem ser reais e diferentes daqueles utilizados no processo de treinamento. Com isso, garante-se que a rede dará, durante os testes, respostas compatíveis com aquilo que aprendeu e não respostas decoradas.

Uma vez que a estrutura da rede já é a final e os testes foram realizados com sucesso, a rede já estará pronta para ser utilizada. Mas, ao contrário do que ocorre durante o treinamento, quando cada entrada de dados implica várias e várias iterações do algoritmo de aprendizado, a rede treinada pode ser facilmente utilizada em tempo real, já que a computação é direta e realizada apenas uma vez (TRIPPI e LEE, 1996).

Um problema bastante comum encontrado durante o aprendizado das redes neurais é que o erro verificado durante cada fase do treinamento não diminui monotonicamente, necessariamente (TRIPPI e LEE, 1996). Com isso, nem sempre é fácil saber quando parar o processo de treinamento. Além disso, apesar de a rede ter seu erro minimizado durante esse processo, isso não quer dizer que ela terá o erro mínimo quando utilizada nos dados fora do treinamento. Quando isso ocorre, diz-se que houve *overfitting*: a rede acabou decorando casos específicos apresentados durante o treinamento e não aprendeu a função generalizada intrínseca ao problema, que era o que deveria ocorrer.

#### 4.3.1 Paradigmas de aprendizado

Há várias formas em que a rede neural pode realizar o seu aprendizado. Todas elas, entretanto, devem ocorrer através de um relacionamento da rede com o ambiente (BRAGA et al., 2000). Segundo Kröse e Smagt (1993), essas diferentes formas de aprendizado podem ser categorizadas em dois grupos distintos: aprendizado supervisionado e não-supervisionado.

##### Aprendizado supervisionado

Ao utilizar-se esse paradigma de aprendizado, para cada padrão de entrada recebido pela rede, deve-se fornecer também a saída esperada, sendo que os pares *{entrada, saída}* podem ser providos tanto por um professor externo quanto pelo próprio sistema

que contém a rede (KRÖSE e SMAGT, 1993). Nesse último caso, o aprendizado é auto-supervisionado.

O objetivo, nesse sentido, é ir fornecendo os pares  $\{entrada, saída\}$  até que uma ligação adequada entre os dados de entrada e sua saída esperada seja encontrada. A cada par fornecido, os parâmetros da rede vão sendo ajustados.

### Aprendizado não-supervisionado

Como o próprio nome diz, nesse paradigma de aprendizado não há um *supervisor* acompanhando cada etapa de treinamento. Assim, apenas os padrões de entrada são fornecidos. Segundo Kröse e Smagt (1993), os modelos que utilizam esse paradigma são capazes de descobrir sozinhos padrões específicos da população de entrada, não havendo a necessidade de estabelecer previamente as categorias em que as entradas devem se encaixar.

#### 4.3.2 Algoritmos de aprendizado

Apesar de existirem diversos tipos de algoritmos de aprendizado, o *back-propagation* é o mais utilizado na modelagem de redes neurais. Esse método é do tipo supervisionado; ou seja, para cada padrão de entrada fornece-se também o que deverá ser a saída.

Segundo Trippi e Lee (1996), a descoberta desse algoritmo foi em grande parte responsável pela retomada das pesquisas relacionadas às redes neurais, em meados da década de 80, após uma década de pouco progresso.

Segundo Braga et al. (2000), o *back-propagation* utiliza pares  $\{entrada, saída desejada\}$  para ajustar os pesos da rede, fazendo isso por meio de um mecanismo de correção de

erros. Inicialmente, as entradas e as saídas desejadas são apresentadas à rede. Se, baseado nos valores das entradas e nos pesos das ligações, o valor calculado para a saída for igual ao valor desejado, nada precisa ser feito. Se existir uma diferença entre o valor desejado e o valor calculado, os pesos terão de ser reajustados até que a diferença entre eles chegue a zero, ou a um valor bem próximo de zero.

O cálculo da diferença entre o valor de saída calculado pela rede e o valor fornecido pelo exemplo de treinamento deve ser realizado em seguidas iterações, até que a disposição dos pesos seja tal que essa diferença seja minimizada.

Pelo fato de os erros serem propagados de camada em camada, de trás para frente, desde a camada de saída até a inicial, o método foi denominado *back-propagation* (KRÖSE e SMAGT, 1993).

#### 4.4 Metodologia para construção de uma RN

Klimasauskas (1994) sugere uma metodologia para o desenvolvimento de um modelo de rede neural a ser utilizado em sistemas financeiros reais. A metodologia consiste em sete etapas, que não devem, necessariamente, ser executadas apenas uma vez, uma após a outra. Ao contrário, o autor sugere que essas etapas façam parte de um processo iterativo, em que, sempre que necessário, se deve voltar às etapas anteriores e realizar alterações. As etapas propostas são: coleta de dados; análise e formatação de dados; separação dos dados em conjuntos; seleção das variáveis; desenvolvimento e otimização do modelo; validação; aplicação do modelo.

#### 4.4.1 Coleta de dados

Segundo Klimasauskas (1994), deve-se sempre ter em mente três fatores ao coletar os dados que serão introduzidos em uma rede neural: sua disponibilidade, limpeza e o tratamento a ser realizado em caso de dados incompletos.

A disponibilidade dos dados é fundamental, já que sem os dados para realizar o treinamento não é possível gerar um modelo que resolva o problema proposto.

Já a limpeza dos dados tem a função de ajudar a reduzir o ruído que será dado como entrada durante o processo de aprendizado da rede. Klein e Rossin (1999) investigam o efeito da má qualidade dos dados de entrada para redes neurais do tipo *back-propagation*. Segundo os autores, no caso dos dados de testes, à medida que aumenta a presença de erros, diminui-se a acuracidade das previsões. Para os dados de treinamento, entretanto, viu-se que existe um certo limite até o qual não vale a pena despender-se recursos na tentativa de diminuir a taxa de erros das bases de dados, já que a modelagem da rede neural consegue abstrair-se desses problemas.

É comum, entretanto, deparar-se com dados incompletos ao fazer a coleta. Nesse caso, Klimasauskas (1994) sugere a utilização de uma das seguintes abordagens: não utilizar os dados caso qualquer parte esteja faltando; utilizar os dados de um período anterior em seu lugar; ou, caso o algoritmo de treinamento seja o *back-propagation*, colocar os dados incompletos como zero, já que a fórmula de atualização dos pesos no *back-propagation* é:

$$\Delta P_{ij} = \alpha \delta_i E_j \quad (4.8)$$

em que  $\alpha$  é a taxa de aprendizado,  $\delta_i$  o erro interno da unidade de processamento atual e  $E_i$  a entrada para a conexão. Então, se  $E_i$  for zero, nenhuma mudança ocorrerá a  $\Delta P_{ij}$ .

#### 4.4.2 Análise e formatação de dados

Para o autor, a etapa de análise e formatação dos dados é a mais importante no processo de desenvolvimento da rede neural.

A análise é importante porque só com a compreensão dos dados é possível tomar a decisão de como formatá-los e estruturá-los para que sejam utilizados como entrada da rede neural. E é durante a formatação que os dados são colocados no formato adequado à entrada na rede, pois geralmente não podem ser utilizados na forma em que aparecem logo após a coleta.

Um problema geralmente encontrado – e que exemplifica a necessidade de formatar os dados antes de utilizá-los – é o fato de a rede geralmente não conseguir lidar com intervalos de valores muito amplos. Nesses casos, deve-se estabelecer um intervalo adequado e criar uma escala que insira os dados dentro desse novo intervalo.

Além disso, vê-se com bastante freqüência a concentração dos dados em uma região muito pequena do intervalo de entrada. Em casos como esse, as pequenas variações nessa entrada terão pouco ou nenhum efeito sobre a saída do modelo, o que não é desejável. A solução, mais uma vez, é criar uma escala e estabelecer um novo intervalo válido, utilizando apenas a região que tem os dados concentrados e não o espaço completo.

#### 4.4.3 Separação dos dados em conjuntos

Segundo Klimasauskas (1994), os conjuntos de dados coletados e formatados nas etapas anteriores devem ser divididos em três grupos: (1) dados de treinamento, que são aqueles utilizados durante o processo de aprendizado da rede; (2) dados de teste, que serão usados na verificação de quão bem o modelo gerado interpola soluções para dados nunca antes vistos pela rede; (3) dados de validação, que verificarão o desempenho do modelo gerado quando este já estiver integrado a um sistema financeiro, pronto para ser utilizado.

Um ponto interessante abordado pelo autor é que, ao contrário do que pode parecer à primeira vista, há situações em que é melhor utilizar menos dados durante o treinamento do que todos aqueles que estão disponíveis. Esse é o caso de quando os dados de treinamento estiverem acumulados em pequenas regiões do intervalo de entrada. Como os algoritmos numéricos usados para fazer a interpolação otimizam o desempenho do modelo em que os dados são densos (às custas de regiões em que há menos ocorrências, que podem ser até mais importantes), é possível construir melhores modelos com menos dados, eliminando-se dados semelhantes da área mais densa.

#### 4.4.4 Seleção das variáveis

A seleção de variáveis é o processo de redução do número de variáveis a serem usadas como entrada de uma rede neural. Para Klimasauskas (1994), modelos com poucas entradas podem ser mais eficientes que outros com muitas entradas. Afinal, quando um bom grupo de variáveis de entrada foi encontrado para determinado problema, esse grupo dificilmente mudará com o tempo.

Para fazer essa seleção, uma opção é utilizar a própria rede neural, que, durante o treinamento, pode fazer uma análise de sensibilidade e verificar a importância de cada

variável. Nesse caso, basta eliminar as variáveis que não atinjam determinado patamar de influência no modelo para que se tenha um conjunto mais reduzido de variáveis de entrada. Assim, além de melhorar o desempenho da rede, a redução de variáveis diminuirá o tempo necessário para o treinamento da rede.

#### 4.4.5 Desenvolvimento e otimização do modelo

A dificuldade de modelar o problema, a presença de ruídos nos dados e as características específicas de cada algoritmo de aprendizado fazem com que sejam necessárias a utilização e a comparação de vários modelos diferentes, segundo o autor.

Para avaliar quão bem a rede interpola, o processo de treinamento deve ser periodicamente interrompido e a rede testada. Se o desempenho da configuração atual com os dados de teste for melhor que o da rede anterior, a atual configuração deverá ser guardada. Se após várias tentativas não se conseguir melhorar seu desempenho, o treinamento estará terminado.

Existem, entretanto, algumas técnicas para reduzir a complexidade do desenvolvimento do modelo de rede neural. Os pesos iniciais utilizados, por exemplo, devem ser pequenos ( $1/n$ , onde  $n$  é o número de entradas), ter média zero e distribuição gaussiana ao invés de valores maiores distribuídos uniformemente. Já a função de ativação utilizada deve ser a tangente hiperbólica, que, salvo algumas poucas exceções, funciona melhor do que as outras.

Quanto ao número de camadas intermediárias, isso depende muito da complexidade do problema a ser resolvido. Caso a rede escolhida seja do tipo *feed-forward*, a utilização de uma única camada intermediária já permitirá a geração de uma função contínua aproximada das entradas, quaisquer que sejam elas (RUSSEL e NORVIG, 1995). A

utilização de uma rede com duas camadas intermediárias iria ainda mais longe e permitiria que qualquer tipo de função pudesse ser gerada. Assim, dependendo da complexidade do problema a ser tratado, já se teria uma boa idéia do número de camadas a serem utilizadas.

#### 4.4.6 Validação

Com o acréscimo de um terceiro conjunto de dados, o modelo desenvolvido pode ser testado após sua integração a um sistema financeiro real. Com isso, torna-se possível simular o desempenho real do modelo, já que o ambiente em que será executado será o mesmo de quando for aplicado.

#### 4.4.7 Aplicação do modelo

Nos casos em que o modelo desenvolvido mostrar bom desempenho e for integrado a algum sistema financeiro real, Klimasauskas (1994) diz que é de extrema importância ter a preocupação de entender muito bem o seu funcionamento, mesmo que o modelo tenha se mostrado confiável nas etapas anteriores.

Além disso, deve-se ter em mente que esses sistemas provavelmente não funcionarão em casos excepcionais, como o assassinato de um presidente, uma declaração de guerra ou outros eventos desse tipo, que são impossíveis de se preverem. Nesses casos, a intervenção humana torna-se imprescindível.

### 4.5 Pontos fortes e limitações

As redes neurais têm algumas características próprias que a colocam entre as técnicas de Inteligência Artificial mais utilizadas atualmente, principalmente na área financeira.

Segundo Medsker et al. (1996), as redes neurais não requerem que o conhecimento sobre o qual atuam seja formalizado, sendo, por isso, apropriadas para domínios em que esse conhecimento é escasso.

Além disso, as redes neurais podem mapear entradas em saída através de funções não-lineares, bastando para isso introduzir pelo menos uma camada intermediária na estrutura da rede. Dessa forma, diversas classes de problemas podem ser resolvidas através da utilização de redes neurais.

Um ponto importante indicado por Trippi e Lee (1996) é que as entradas a serem consideradas pelas redes neurais não precisam, necessariamente, ser quantitativas. Entradas qualitativas também são aplicáveis.

Até mesmo a multicolinearidade não é apontada como um problema tão sério para as redes neurais, segundo os autores, apesar do fato de que a eliminação de variáveis de entrada altamente correlacionadas deixaria o modelo um pouco mais robusto.

Já Medsker et al. (1996) ressaltam o fato de que, mesmo quando os dados utilizados como entrada pela rede neural apresentarem ruído, estiverem incompletos ou forem diferentes daqueles usados no treinamento, a rede poderá apresentar uma boa resposta.

Entretanto, há alguns pontos em que as redes neurais deixam a desejar. Um desses pontos e provavelmente o principal, é que as redes neurais não têm capacidade de explicar o que generalizam. Com isso, a saída de uma rede neural dificilmente contribuirá para a teoria à qual o problema está relacionado, apesar de conseguir resolvê-lo.

Segundo Trippi e Lee (1996), ainda, as redes neurais podem identificar certos fatores como extremamente importantes na geração do resultado, sendo que esses fatores podem ser irrelevantes e/ou confrontarem-se com o que diz a teoria tradicional do domínio específico de conhecimento. Além disso, pode ser também que a rede não consiga generalizar os resultados e funcionar bem apenas nos dados de treinamento.

As redes neurais também não podem garantir a solução ótima para um problema e nem mesmo gerar a mesma solução quando os mesmos dados de entrada são providos (TRIPPI e LEE, 1996).

## 5 RNS E AGs APLICADOS À SELEÇÃO DE CARTEIRAS

Pode-se ver com bastante frequência, na literatura científica, trabalhos utilizando redes neurais no processo de análise de séries temporais e previsão de retornos de ativos financeiros, sejam esses ativos ações, índices de ações, opções, etc.

Abhyankar et al. (1997), por exemplo, utilizam redes neurais para verificar se as séries temporais de retornos dos índices S&P 500, DAX, Nikkei e FTSE seguem um caminho aleatório ou se são, pelo menos em parte, explicáveis por um processo determinístico. Os autores concluem, a partir da análise dos resultados, que todas as séries pesquisadas apresentam padrões não lineares em sua estrutura. Viu-se também que, apesar de existir um processo determinístico não-linear nessas séries de dados, também há a presença de um processo estocástico que não pode ser desprezado.

Donaldson e Kamstra (1996) mostram, na predição combinada da volatilidade dos índices S&P 500, Nikkei, FTSE e TSEC (Toronto Stock Exchange Composite Index), um exemplo em que a utilização de redes neurais conseguiu melhores resultados do que as abordagens lineares tradicionalmente utilizadas. Segundo os autores, isso se deu pelo fato de elas conseguirem captar relações mais complexas entre as variáveis do problema - relações estas que nem sempre podem ser captadas por modelos lineares.

Leung et al. (2000), ao realizarem a previsão dos mesmos índices, utilizam também redes neurais para comparar o uso de modelos de classificação (que classificam a direção tomada pelos preços entre *alta* ou *baixa*) e modelos de estimação (que tentam prever o nível do retorno propriamente dito), concluindo que geralmente se obtém melhores resultados ao utilizar-se a primeira abordagem.

Desai e Bharati (1998), Leung et al. (2000), Qi (1999) também utilizam redes neurais, entre elas redes *back-propagation* e probabilísticas, dentre outros modelos, para preverem a movimentação dos retornos de ações e índices de ações. A eficácia das redes neurais na previsão de retornos é verificada nesses trabalhos, sendo tal habilidade atribuída à capacidade dessas redes em detectar padrões não-lineares nas séries temporais. De forma geral, obtiveram-se nesses trabalhos resultados melhores do que aqueles alcançados por suas contrapartes lineares.

Mas não apenas ações ou índices de ações têm sido precificados através da ajuda das redes neurais. Outros ativos financeiros, como as opções, também têm sido analisadas sob a ótica dessas redes, como demonstram Burgess e Refenes (1999) e Yao et al. (2000).

Com menor frequência, vê-se também a utilização de algoritmos genéticos no processo de seleção de carteiras. Arifovic e Gençay (1999), por exemplo, utilizam algoritmos genéticos na simulação do processo de gerenciamento de uma carteira cambial.

Já Rabatin (1997, 1998a) descreve um modelo inteligente auto-adaptativo capaz de gerenciar carteiras cambiais sem a necessidade de intervenção humana. O modelo, baseado em algoritmos genéticos, é treinado para desenvolver um padrão de negociações que lhe permite obter um desempenho consistente e previsível. O algoritmo genético utilizado é modelado de forma a selecionar as melhores alternativas dentre as opções de investimento disponíveis, alocá-las dentro da carteira, realizar compras e vendas e analisar o risco dos ativos individualmente e/ou em conjunto.

Allen e Karjalainen (1999) usam algoritmos genéticos na tentativa de encontrar padrões na série do índice de ações S&P 500, de modo que fosse possível a criação de regras técnicas de negociação.

Rabatin (1998b) mostra como o processo de gerenciamento do risco em carteiras de investimentos pode ser aprimorado através da utilização de algoritmos evolutivos<sup>1</sup>.

Além destas aplicações, vê-se a utilização de algoritmos genéticos em várias outras áreas da administração financeira. Varetto (1998), por exemplo, utiliza um algoritmo genético na avaliação do risco de insolvência de empresas italianas, obtendo bons resultados.

LeBaron et al. (1999) utilizam algoritmos genéticos no processo de aprendizado dos agentes atuantes num mercado de capitais simulado por computador. Através de regras aprendidas e desenvolvidas pelos algoritmos genéticos, esses agentes fazem previsões sobre o futuro, negociando ações com base em suas expectativas de risco e retorno.

---

<sup>1</sup> *Algoritmos evolutivos* são um conjunto de técnicas computacionais que têm como base o processo evolutivo visto na natureza. Os algoritmos genéticos são o principal representante dessa família, que também conta com a programação evolutiva, as estratégias de evolução, os sistemas classificadores e a programação genética.

## 6 METODOLOGIA

Procurou-se, neste trabalho, comparar as aplicações do método de Markovitz e de um método alternativo, composto por redes neurais e um algoritmo genético, no processo de seleção de carteiras.

A comparação foi realizada da seguinte forma: para cada mês analisado, executaram-se os dois modelos separadamente, gerando-se uma carteira de ações para cada um. Simulando-se o investimento nas carteiras geradas mensalmente durante 24 meses consecutivos, verificou-se qual das duas abordagens obteve melhores resultados, utilizando-se para isso o Índice de Sharpe.

### 6.1 Unidade e período de análise

As ações a serem utilizadas nas composições das carteiras foram escolhidas com base em sua representatividade no volume total de ações negociadas na Bovespa em janeiro de 1995. Foram escolhidas tantas ações quanto necessárias para que se trabalhasse com pelo menos 80% do volume total negociado nesse mês. As oito ações selecionadas podem ser visualizadas na TABELA 1.

O período em que se realizou a simulação dos investimentos (baseados nas carteiras geradas pelos modelos) vai de jan/1999 a dez/2000. Considerando-se o horizonte de investimentos como de um mês, realizou-se o processo de seleção das carteiras e simulação dos resultados 24 vezes para cada modelo – uma vez para cada mês.

TABELA 1

## Ações utilizadas no trabalho

<b>Ação</b>	<b>Vol. Jan/95 (R\$)</b>	<b>%</b>
Telebrás RCTB PN	1.647.521.346	42,15
Petrobrás PN	350.352.563	8,96
Eletrobrás ON	329.368.445	8,43
Eletrobrás PNB	310.271.818	7,94
Vale Rio Doce PNA	225.164.739	5,76
Usiminas PNA	116.387.117	2,98
Cemig PN	109.532.161	2,80
Sid. Tubarão PN	50.108.836	1,28
		<b>80,29</b>

## 6.2 Coleta de dados

O trabalho realizado foi experimental e os dados utilizados, secundários. Foram coletados, para cada uma das oito ações, o volume de negociação e as cotações mensais referentes aos preços de abertura, mínimo, máximo e de fechamento. Além disso, coletaram-se também valores mensais do índice Ibovespa, taxa do dólar, taxa da poupança, taxa de juros (Selic 30 dias) e índice Dow Jones, todos eles ajustados aos dividendos e deflacionados. Esses dados foram obtidos no Economática<sup>®</sup>, programa largamente utilizado para a obtenção de dados financeiros e contábeis relativos às maiores empresas da América Latina<sup>2</sup>.

Conforme já mencionado, trabalhou-se nesta pesquisa com um horizonte de investimento de um mês. Mais especificamente, supôs-se que o investidor aplique no início de um determinado mês para obter seu retorno no mês subsequente, tomando suas decisões com base em informações publicamente disponíveis e atualizadas com, pelo menos, a mesma frequência que o tamanho do horizonte de investimentos. Não se puderam utilizar, dessa forma, informações que não são publicadas pelo menos

<sup>2</sup> [www.economica.com.br](http://www.economica.com.br)

mensalmente, tais como as informações presentes nos demonstrativos financeiros das empresas.

### 6.3 Aplicação do método de Markovitz

Utilizou-se, nessa parte do trabalho, a mesma abordagem utilizada em trabalhos empíricos na área, como os de Bruni e Famá (1999) e Figueiredo et al. (2000), em que se calcularam os dados de entrada ao modelo a partir de séries históricas. Segundo Bruni e Famá (1999), para evitar o forte grau de subjetividade inerente à previsão dos retornos das ações e devido à dificuldade relativa à previsão de cenários e comportamento futuros, a prática, em pesquisas como esta, é elaborar carteiras com base nos dados do passado (abordagem *ex-ante*), ao invés de previsões para o futuro.

Para aplicar, então, o modelo de Markovitz, foram utilizados os históricos dos preços de fechamento das oito ações que compõem a unidade de análise. Esses históricos foram retirados do Economática<sup>®</sup>. Já os dados necessários como entrada ao modelo – os retornos esperados das ações e suas covariâncias – foram geradas a partir dessa série de preços, utilizando-se uma janela de 12 meses, conforme sugerem Bruni e Famá (1999).

Utilizando-se a *Financial Toolbox*<sup>3</sup> do programa Matlab<sup>®</sup>, criou-se um *script* que gerou a fronteira eficiente para cada um dos 24 meses a serem analisados. Na geração dessas fronteiras, não se permitiram investimentos em ativos livre de risco e/ou empréstimos de capital. Seguindo ainda a metodologia de Bruni e Famá (1998), escolheu-se no início do primeiro mês (Jan/1999) a carteira de menor risco dentro da fronteira eficiente, investindo-se na mesma. No mês seguinte, calcula-se o retorno obtido no mês anterior

---

<sup>3</sup> [www.mathworks.com](http://www.mathworks.com)

e repete-se o processo, gerando-se uma nova carteira, investindo-se na mesma e calculando-se o novo retorno. Adotou-se tal procedimento até o último mês, dez/2000.

Ao final do processo, obtiveram-se 24 retornos mensais, decorrentes da simulação de investimento nas carteiras geradas pelo método de Markovitz ao longo de cada mês. Calculando-se o retorno acumulado nesse período e a variabilidade das taxas de retorno mensais (risco), chegou-se ao Índice de Sharpe a ser utilizado na comparação entre os métodos.

#### 6.4 Aplicação do modelo híbrido

Na geração das carteiras pelo modelo híbrido, utilizaram-se redes neurais (uma para cada ação) para se realizar as previsões dos preços futuros das ações e um algoritmo genético para otimizar as carteiras. A escolha das redes neurais na primeira etapa do processo teve como principal razão a reconhecida capacidade dessas redes de descobrir padrões existentes em séries temporais e de prever valores futuros. Já o algoritmo genético foi utilizado na etapa final por ser um método de otimização adequado a problemas em que o espaço de soluções é demasiado grande para uma busca exaustiva e em que a teoria específica do campo de conhecimento não consegue explicar de forma satisfatória o comportamento dos fenômenos observados.

##### 6.4.1 Modelagem e execução das redes neurais

Tomou-se como base, durante a modelagem e execução das redes neurais, a metodologia proposta por Klimasauskas (1994), descrita em 4.4.

A modelagem foi feita em várias etapas, iniciando-se com a coleta de dados e terminando com a otimização do modelo. Essas etapas, apesar de apresentarem uma

seqüência previamente definida, tiveram de ser realizadas diversas vezes até que a rede mostrasse uma estrutura considerada satisfatória. A cada vez que se percebia que o processo poderia ser aprimorado ao retornar-se à etapa anterior e alterar alguns dos procedimentos realizados, voltava-se e fazia-se as mudanças necessárias.

A modelagem e execução da rede neural foram feitas usando-se a *Neural Network Toolbox*<sup>4</sup> do programa Matlab<sup>®</sup>. O Matlab<sup>®</sup> é uma ferramenta extremamente utilizada em pesquisas científicas e para os mais diversos propósitos, como o desenvolvimento de modelos, criação de algoritmos, simulação, análise de dados e desenvolvimento de aplicações, tendo ampla aceitação como ferramenta para modelagem de redes neurais. Sua utilização neste trabalho se deu através da criação de *scripts* contendo o código necessário para a criação da rede neural, a definição de seus parâmetros, a transformação dos dados de entrada, o treinamento da rede e a execução dos dados de teste.

Na modelagem da rede neural, definiu-se como variável de saída (a variável dependente do problema, aquela a ser prevista pela rede) o preço de fechamento da ação para o período seguinte  $t+1$ . Como entrada, utilizaram-se os preços de abertura, mínimo, máximo e de fechamento, além do volume de negociação, do índice Ibovespa, da taxa do dólar, taxa de juros (Selic 30 dias), taxa da poupança e o índice Dow Jones, todos no período  $t$ . Utilizou-se também o preço de fechamento nos períodos  $(t-1)$ ,  $(t-2)$  e  $(t-3)$  e as média móveis de 5, 10 e 15 meses das cotações do preço de fechamento. As saídas e entradas da rede podem ser vistas nas TABELAS 2 e 3, respectivamente.

As entradas relativas aos períodos  $t$ ,  $(t-1)$ ,  $(t-2)$  e  $(t-3)$  foram utilizadas com o intuito de prover à rede uma memória de curto e médio prazo, ficando com as médias móveis de 5, 10 e 15 meses a tarefa de dar à rede a capacidade de reconhecer mudanças de padrões iniciadas num prazo um pouco mais longo. As médias móveis, além de

---

<sup>4</sup> [www.mathworks.com](http://www.mathworks.com)

ajudarem na detecção de tendências de longo prazo, têm a característica de eliminar grande parte do ruído presente nos dados, que, por serem pontuais, apresentam maior possibilidade de revelarem alguma distorção.

TABELA 2

Variável a ser prevista

<b>Saída</b>	<b>Descrição</b>
$Pf_{(t+1)}$	Preço de fechamento em $t+1$

TABELA 3

Entradas da rede neural

<b>Entrada</b>	<b>Descrição</b>
$Pa_{(t)}$	Preço de abertura em $t$
$Pmi_{(t)}$	Preço mínimo em $t$
$Pma_{(t)}$	Preço máximo em $t$
$Pf_{(t)}$	Preço de fechamento em $t$
$V_{(t)}$	Volume negociado em $t$
$Ibo_{(t)}$	Índice Bovespa em $t$
$D_{(t)}$	Cotação do dólar em $t$
$TJ_{(t)}$	Taxa de juros em $t$
$TP_{(t)}$	Taxa da poupança em $t$
$DJ_{(t)}$	Índice Dow Jones em $t$
$Pf_{(t-1)}$	Preço de fechamento em $(t-1)$
$Pf_{(t-2)}$	Preço de fechamento em $(t-2)$
$Pf_{(t-3)}$	Preço de fechamento em $(t-3)$
$Mm_{(5)}$	Média móvel de 5 meses
$Mm_{(10)}$	Média móvel de 10 meses
$Mm_{(15)}$	Média móvel de 15 meses

Os dados utilizados como entrada para a rede neural foram calculados a partir das cotações mensais coletadas da base Económica<sup>®</sup>, sendo organizados no Microsoft Excel<sup>®</sup> e transformados em arquivo texto para uso no Matlab<sup>®</sup>. As médias móveis foram

calculadas utilizando-se os preços de fechamento imediatamente anteriores, ponderadas de acordo com a seguinte fórmula:

$$Mm(n) = (P[t]*n + P[t-1]*(n-1) + P[t-2]*(n-2) + \dots + P[t-n+1]*1) / (n + (n-1) + (n-2) + \dots + 1)$$

Como exemplo, a média móvel ponderada de cinco meses teria a forma:

$$Mm(5) = (P[t]*5 + P[t-1]*4 + P[t-2]*3 + P[t-3]*2 + P[t-4]*1) / 5 + 4 + 3 + 2 + 1$$

Para que os dados de entrada se apresentassem na forma pedida pela rede neural, tiveram de submeter-se antes a uma transformação, após a qual passaram a apresentar média zero e desvio padrão unitário. O objetivo dessa transformação foi atenuar possíveis problemas de ruído, diminuir as distâncias entre os valores de variáveis muito espaçadas (por exemplo, volumes de negociação) e diminuir a influência causada por valores que se destacavam excessivamente em relação aos demais, os chamados *outliers*.

Na modelagem da rede neural, utilizaram-se dois conjuntos de dados: o de treinamento e o de testes. Os dados referentes aos anos de 1995 a 1998 foram usados no processo de aprendizado da rede, sendo alocados no conjunto de treinamento. Já os dados referentes aos dois anos seguintes (1999 e 2000, anos em que se geraram as carteiras) foram destinados ao conjunto de testes.

Para conseguir uma boa interpolação da rede, o processo de treinamento, que foi do tipo supervisionado, teve de ser periodicamente interrompido e a rede testada. Para cada um dos parâmetros a serem variados (por exemplo, número de épocas, algoritmo de treinamento, quantidade de camadas) iniciou-se o treinamento com um valor, testou-se a rede e verificou-se o resultado, fazendo-se o mesmo com diversos valores para esse mesmo parâmetro, até que melhores resultados já não pudessem ser obtidos.

Quando isso ocorria, mantia-se o melhor valor na configuração e passava-se a trabalhar com outro parâmetro, reiniciando-se o processo de refinamento.

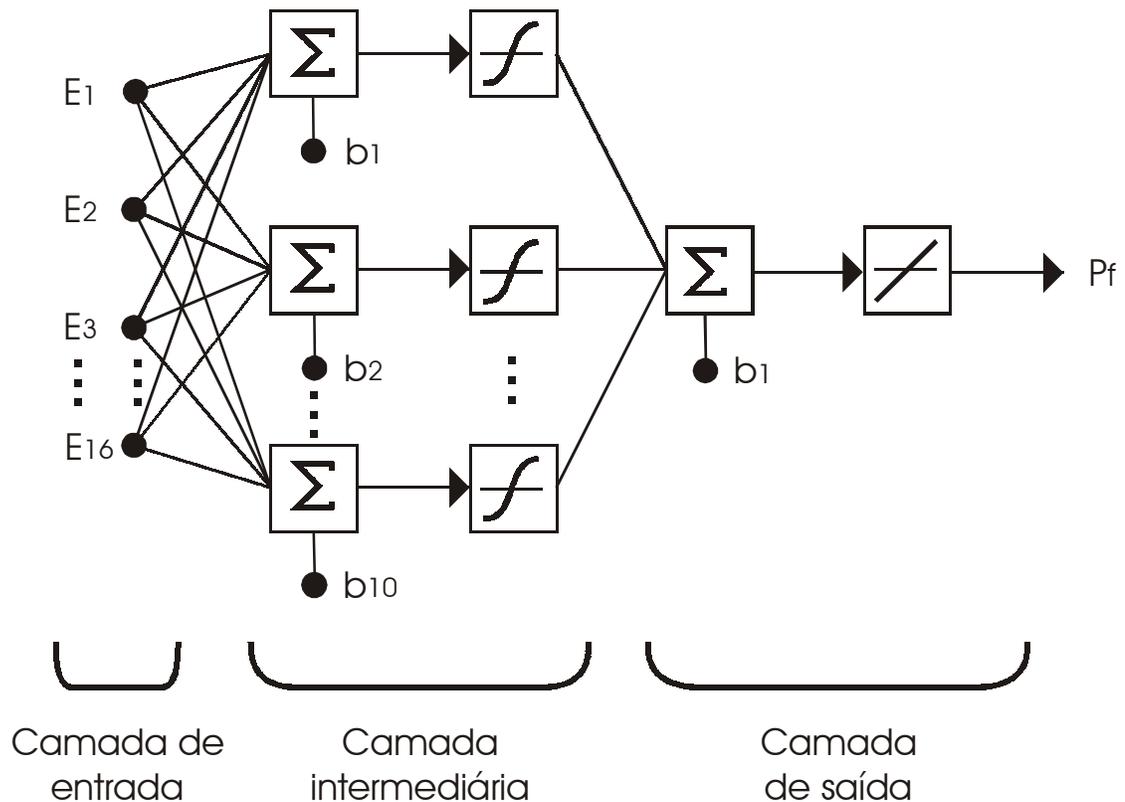


FIGURA 5 – Estrutura básica das redes neurais utilizadas

Testadas diversas combinações entre os parâmetros, foi estabelecida a melhor configuração para as redes neurais, que pode ser vista na FIGURA 5: redes *feed-forward* com uma camada intermediária de dez nodos, 16 variáveis de entrada e uma de saída. Como função de ativação, usou-se a tangente hiperbólica para os nodos da camada intermediária e a função linear para a camada de saída. No treinamento, utilizou-se o algoritmo Levenberg-Marquardt, uma variação do *back-propagation*, com

taxa de aprendizado 0,10 e erro quadrático médio como função de desempenho. Um resumo da configuração final da rede neural pode ser visto na TABELA 4.

TABELA 4  
Configuração final da rede neural

Arquitetura	Rede <i>feed-forward</i> Uma camada intermediária
Número de nodos por Camada	16 na camada de entrada 10 na camada intermediária 1 na camada de saída
Função de transferência	Tangente hiperbólica na camada intermediária Linear na camada de saída
Algoritmo de treinamento	Levenberg-Marquardt
Taxa de aprendizado	0,10
Função de desempenho	Erro quadrático médio

#### 6.4.2 Modelagem e execução do algoritmo genético

Terminada a modelagem da rede neural, foram executados (com um tempo de processamento da ordem de segundos) os dados referentes ao período de aprendizado, obtendo-se as previsões para os preços de fechamento de cada uma das ações em cada um dos 24 meses analisados. Com esses dados em mãos, passou-se à etapa de modelagem do algoritmo genético. Para isso, utilizou-se o programa Evolver<sup>®</sup>, da Palisade<sup>5</sup>. O Evolver<sup>®</sup> é um aplicativo que usa algoritmos genéticos na resolução de problemas de otimização, trabalhando de forma integrada ao Microsoft Excel<sup>®</sup>.

Nessa fase, realizou-se a otimização da composição das carteiras de ações para cada um dos 24 períodos. Para isso, fez-se uso do método *recipe* interno ao Evolver<sup>®</sup>, definindo-se uma codificação real para os indivíduos, tamanho da população igual a 200, taxa de mutação 0,01, taxa de recombinação (*crossover*) 0,5 e 3.000 iterações na busca pela melhor solução.

<sup>5</sup> [www.palisade.com](http://www.palisade.com)

O método utilizado realiza a seleção dos indivíduos através de uma abordagem derivada dos métodos *ranking* e *steady-state*. Desse modo, a cada iteração os organismos eram escolhidos com base em sua ordem em um *ranking* de adequações ao ambiente, evitando-se, assim, que os melhores indivíduos dominassem rapidamente a população. Além disso, apenas um indivíduo era substituído a cada iteração, ao invés de toda a população. Já a recombinação era realizada sem a utilização de pontos fixos de corte. Ao invés disso, distribuía-se aleatoriamente cada gene do indivíduo pai para um ou outro filho. Com isso, estimulou-se a busca de soluções em todas as regiões do espaço.

Algumas restrições também foram definidas em relação à alocação das ações dentro das carteiras. Definiu-se que o somatório das alocações deveria ser menor ou igual a 1 e que cada ação poderia representar no mínimo 0 e no máximo 100% da carteira gerada, evitando-se, assim, o investimento alavancado com recursos emprestados.

A função de adequação do algoritmo genético foi definida como:

$$\text{Adequação} = E(R_1) / \sigma(R_1) + E(R_2) / \sigma(R_2) + \dots + E(R_8) / \sigma(R_8) \quad , \quad (6.1)$$

em que

*Adequação* = Grau de adequação do indivíduo ao ambiente

$E(R_i)$  = Previsão de retorno para a ação  $i$  durante o período

$\sigma(R_i)$  = Risco da ação  $i$ , medido pelo desvio-padrão.

Para a realização desse cálculo, usaram-se os dados de retornos previstos pelas redes neurais, sendo os riscos definidos como os desvios-padrão dos retornos nos 12 meses anteriores.

A otimização das carteiras, aqui, funcionou da seguinte forma: para cada período, várias carteiras foram geradas aleatoriamente e definidas como a população inicial de soluções do algoritmo genético. Em algumas iterações, as soluções candidatas evoluem, fazendo com que as composições sugeridas da carteira se modifiquem. Continua-se o processo de evolução até que um critério de convergência seja atendido, ou que uma composição satisfatória seja criada. Ao final do processo, tem-se uma solução que representa a melhor composição encontrada para a carteira a ser investida naquele período.

### 6.5 Comparação dos resultados

Realizadas as gerações das carteiras ótimas e as simulações de investimento, obtiveram-se 48 retornos mensais, 24 para cada método. Com esses dados, calcularam-se os retornos acumulados no período e seus respectivos riscos, medidos pela variabilidade das taxas mensais de retorno.

A comparação entre os métodos, que considerou a relação risco-retorno, foi feita usando-se o Índice de Sharpe, definido como (Figueiredo et al., 2000):

$$I_{\text{Sharpe}} = E(R_i) / \sigma(R_i), \quad (6.2)$$

em que

$I_{\text{Sharpe}}$  = Índice de Sharpe

$E(R_i)$  = Média dos retornos em excesso da carteira  $i$  durante o período

$\sigma(R_i)$  = Desvio-padrão dos retornos em excesso carteira  $i$  durante o período.

Considera-se como *retorno em excesso* a parcela do retorno que ultrapassa a taxa livre de risco, definida aqui como a taxa da poupança, deflacionada.

## 7 RESULTADOS OBTIDOS

Os resultados obtidos através da simulação de investimento nas carteiras mensais geradas por cada método podem ser vistos na TABELA 5.

TABELA 5

Retornos obtidos (%), riscos (%) e índices de Sharpe

	Sem descontar poupança			Descontando poupança			
	Ibovespa	Método Markovitz	Modelo Híbrido	Poupança	Ibovespa	Método Markovitz	Modelo Híbrido
Jan/1999	17,9202	60,8962	-1,0631	-0,1296	18,0498	61,0258	-0,9334
Fev/1999	4,4084	-9,6828	10,7330	-2,9740	7,3824	-6,7088	13,7070
Mar/1999	17,7142	20,1309	48,4437	-0,3066	18,0208	20,4375	48,7503
Abr/1999	6,0826	56,7941	65,0136	1,0819	5,0007	55,7122	63,9317
Mai/1999	-1,9662	-3,6244	-5,5011	1,4239	-3,3902	-5,0483	-6,9250
Jun/1999	3,7840	11,7224	8,3436	-0,2055	3,9896	11,9280	8,5491
Jul/1999	-11,5983	-13,7265	0,1320	-0,7828	-10,8155	-12,9437	0,9148
Ago/1999	-0,2700	3,0856	16,2273	-0,6477	0,3777	3,7333	16,8750
Set/1999	3,6076	4,6627	9,6478	-0,6870	4,2946	5,3497	10,3349
Out/1999	3,3943	5,0608	1,1174	-1,1408	4,5351	6,2016	2,2583
Nov/1999	14,8549	16,8923	60,1116	-1,7840	16,6389	18,6763	61,8956
Dez/1999	24,0457	27,9990	8,3718	-0,4235	24,4691	28,4225	8,7952
Jan/2000	-6,2347	-13,8734	-18,4213	-0,3009	-5,9338	-13,5724	-18,1204
Fev/2000	7,5574	-1,6204	-4,5796	0,5430	7,0144	-2,1634	-5,1225
Mar/2000	0,7247	0,6904	0,8659	0,4731	0,2517	0,2173	0,3929
Abr/2000	-12,9246	-12,6719	-13,9607	0,5002	-13,4248	-13,1720	-14,4608
Mai/2000	-4,3801	2,0751	8,0857	0,0799	-4,4600	1,9953	8,0058
Jun/2000	10,8108	20,1101	27,0088	-0,2129	11,0238	20,3230	27,2217
Jul/2000	-3,8061	0,5524	4,4465	-1,5690	-2,2370	2,1214	6,0155
Ago/2000	3,5368	2,0549	-14,5408	-1,0965	4,6333	3,1514	-13,4443
Set/2000	-8,8041	-4,9273	0,6178	-0,0851	-8,7189	-4,8422	0,7029
Out/2000	-7,0053	-5,8308	-5,3631	0,6323	-7,6376	-6,4632	-5,9954
Nov/2000	-10,9747	-12,5075	-10,4293	-0,1401	-10,8346	-12,3674	-10,2892
Dez/2000	13,9753	12,8241	19,5910	-0,1592	14,1345	12,9833	19,7502
<b>Ret. Acumulado</b>	<b>69,06</b>	<b>257,24</b>	<b>419,41</b>	<b>-7,72</b>	<b>81,89</b>	<b>286,49</b>	<b>456,44</b>
<b>Retorno Médio</b>	<b>2,21</b>	<b>5,45</b>	<b>7,11</b>	<b>-0,33</b>	<b>2,52</b>	<b>5,79</b>	<b>7,41</b>
<b>Desvio Padrão</b>	<b>10,14</b>	<b>19,65</b>	<b>21,88</b>	<b>0,95</b>	<b>10,33</b>	<b>19,53</b>	<b>22,00</b>
<b>Índice de Sharpe</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>0,2443</b>	<b>0,2968</b>	<b>0,3370</b>

Vê-se, pela TABELA 5, que o modelo híbrido composto pelo algoritmo genético e pelas redes neurais obteve um desempenho superior ao método de Markovitz. Enquanto o modelo de Markovitz conseguiu um retorno acumulado de 257,24% ao longo dos dois anos analisados, a aplicação do modelo híbrido gerou 419,41% de retorno. Se analisarmos o risco das duas aplicações, dado pela variabilidade das taxas mensais de retorno, vê-se que houve uma pequena diferença favorável ao método de Markovitz: 19,65% para este e 21,88% para o modelo híbrido. Entretanto, ao utilizarmos o índice de Sharpe na verificação da relação retorno/risco de ambas as abordagens, vê-se que os modelos híbrido e de Markovitz obtiveram índices de 0,3370 e 0,2968, respectivamente, o que mostra a obtenção de mais retorno por unidade de risco ao se utilizar o modelo composto por métodos genéricos.

Vê-se também, pela TABELA 5, que ambos os métodos conseguiram resultados melhores que o índice de mercado, dado pelo Ibovespa, que obteve nos dois anos um retorno acumulado de 69,06%, risco de 10,14% e índice de Sharpe de 0,2443.

A TABELA 6 mostra os resultados separados por ano. Pode-se ver que os resultados se mantiveram, mesmo ao analisarmos separadamente anos com características tão distintas (enquanto, em 1999, o índice de mercado obteve um retorno acumulado de 110,48%, em 2000, esse retorno foi negativo, de -19,68%).

Nesse caso, os retornos médios para os métodos de Markovitz e o modelo híbrido foram, respectivamente, de 12,95% e 16,35% para 1999 e -1,56% e -1,40% para 2000. Já os riscos foram 23,73% e 24,75%, para 1999, e 10,15% e 13,82%, para 2000.

TABELA 6

## Resultados obtidos, separados por ano

	Ibovespa		Markovitz		Híbrido	
	1999	2000	1999	2000	1999	2000
Ret Acumulado (%)	110,48	-19,68	331,23	-17,16	515,17	-15,57
Retorno Médio (%)	6,40	-1,81	12,95	-1,56	16,35	-1,40
Desvio Padrão (%)	10,03	8,76	23,73	10,15	24,75	13,82

Os retornos mensais e acumulados obtidos durante os dois anos analisados podem ser visualizados nos GRÁFICOS 1 e 2, respectivamente.

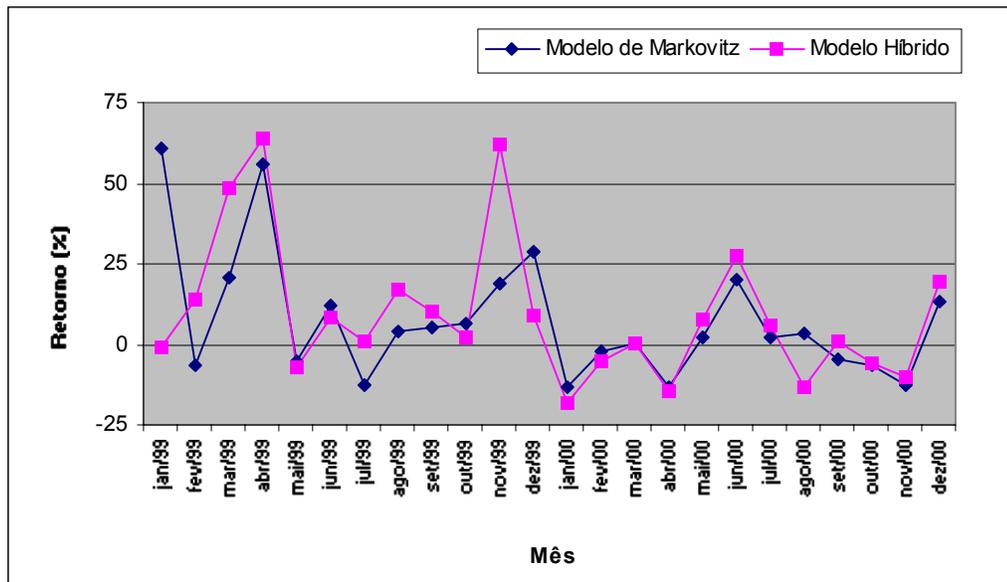


GRÁFICO 1 – Retornos mensais obtidos

É possível observar, pelo GRÁFICO 1, que as diferenças entre os retornos obtidos por ambas as estratégias foram relativamente grandes em 1999, ano em que o índice de mercado obteve um crescimento de 110,48%. Já em 2000, quando o Ibovespa caiu 19,68%, estes retornos pareceram estar, de certa forma, correlacionados, com exceção do mês de agosto. Viu-se, assim, que a diferença nos resultados obtidos pelas

diferentes estratégias ocorreu em 1999, enquanto o mercado estava em alta. Percebe-se, também, pelo GRÁFICO 2, que o período exato em que ocorreu tal diferença foi no segundo semestre desse mesmo ano. Enquanto em jun/99 os retornos acumulados ainda mostravam-se bastante semelhantes, em dez/99 a diferença entre os retornos já era de 56%.

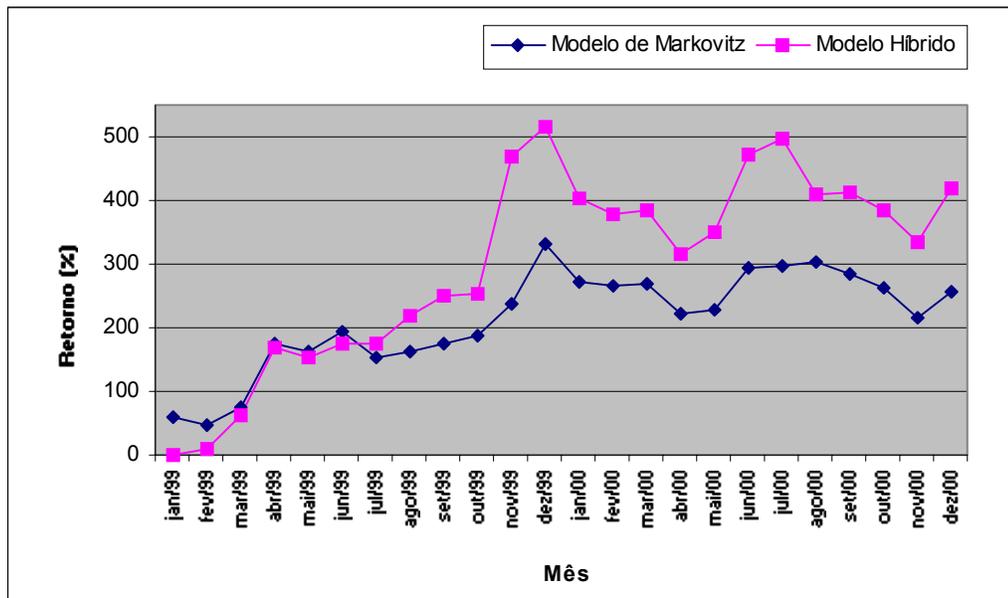


GRÁFICO 2 – Retornos acumulados obtidos entre jan/99 e dez/00

Uma possível explicação para esta diferença é a maneira como os métodos alocaram as ações em suas respectivas carteiras. Vê-se, pelas TABELAS 7 e 8, que, enquanto o modelo de Markovitz visava a diversificação de seus investimentos (tomando como base a matriz de correlações históricas entre as ações), o modelo alternativo investiu a maior parte de cada carteira naquele ativo com melhor relação risco-retorno.

TABELA 7

Composição das carteiras geradas pelo método de Markovitz (%)

	Telebrás PN	Petrobrás PN	Eletrobrás ON	Eletrobrás PNB	Vale PNA	Usiminas PNA	Cemig PN	Tubarão PN
Jan/1999	0	0	0	0	0,6324	0,3676	0	0
Fev/1999	0	0	0,0878	0	0	0,7183	0,1939	0
Mar/1999	0	0	0,0924	0	0	0,7542	0,1534	0
Abr/1999	0	0	0,119	0	0,0305	0,8505	0	0
Mai/1999	0,4218	0	0,0392	0	0	0,3637	0,1753	0
Jun/1999	0,3982	0	0,0405	0,1178	0	0,3823	0,0612	0
Jul/1999	0,3818	0	0	0,1958	0	0,3641	0,0582	0
Ago/1999	0,3862	0	0,1771	0	0	0,3673	0,0694	0
Set/1999	0,3274	0,296	0,1759	0	0,0589	0,1418	0	0
Out/1999	0,3744	0,3203	0,1089	0	0,0535	0,1429	0	0
Nov/1999	0,4356	0,3962	0	0	0,0497	0,1185	0	0
Dez/1999	0,6052	0,3073	0	0,0143	0	0,0732	0	0
Jan/2000	0,1054	0	0,7486	0	0	0,1459	0	0
Fev/2000	0,1985	0	0,4857	0	0,3158	0	0	0
Mar/2000	0,2131	0	0,4378	0	0,3491	0	0	0
Abr/2000	0,1523	0	0,5885	0	0,2592	0	0	0
Mai/2000	0	0,0195	0,012	0,5096	0,4589	0	0	0
Jun/2000	0	0,0291	0	0,5116	0,4593	0	0	0
Jul/2000	0	0,1136	0,0386	0	0,5121	0,3356	0	0
Ago/2000	0	0,0801	0,1256	0	0,3776	0,4167	0	0
Set/2000	0	0	0,0084	0	0,6827	0,309	0	0
Out/2000	0	0	0,0445	0	0,6683	0,2872	0	0
Nov/2000	0	0	0,0198	0	0,7701	0,2101	0	0
Dez/2000	0	0	0,0856	0	0,3362	0	0	0,5782

TABELA 8

Composição das carteiras geradas pelo modelo híbrido (%)

	Telebrás PN	Petrobrás PN	Eletróbrás ON	Eletróbrás PNB	Vale PNA	Usiminas PNA	Cemig PN	Tubarão PN
Jan/1999	0,0011	0,9743	0,0084	0,0008	0,0154	0	0	0
Fev/1999	0	0,8357	0,0208	0,0966	0,0443	0	0,0025	0
Mar/1999	0,0001	0,9596	0,0226	0,0128	0	0,0050	0	0
Abr/1999	0,0001	0,0074	0,0010	0	0	0,9915	0	0
Mai/1999	0,0053	0,9925	0	0	0,0022	0	0	0
Jun/1999	0,0010	0,9585	0	0	0,0385	0	0	0,0021
Jul/1999	0	0	0	0	0,9997	0	0	0
Ago/1999	0,0063	0,0075	0	0,0013	0,0178	0	0	0,9670
Set/1999	0,0356	0,9567	0,0036	0,0040	0	0	0	0
Out/1999	0,0075	0	0,0042	0,0052	0,0139	0	0	0,9692
Nov/1999	0,0011	0,0001	0	0,0010	0,0023	0	0,0008	0,9947
Dez/1999	0,0002	0	0	0,9933	0,0003	0,0062	0	0
Jan/2000	0	0	0,9531	0,0070	0	0	0	0,0399
Fev/2000	0	0,0016	0,9819	0,0107	0	0	0,0049	0,0009
Mar/2000	0	0,0102	0,9879	0,0009	0	0	0,0002	0,0007
Abr/2000	0	0	0,9951	0,0026	0,0004	0	0,0019	0
Mai/2000	0	0,0026	0,9863	0,0027	0,0081	0	0,0004	0
Jun/2000	0	0,0001	0	0,9458	0,0538	0,0003	0,0001	0
Jul/2000	0	0	0,9903	0,0033	0	0,0049	0,0009	0,0007
Ago/2000	0	0	0,0579	0,9413	0,0001	0,0006	0,0001	0,0001
Set/2000	0	0	0,0174	0,9717	0	0	0,0001	0,0108
Out/2000	0,0005	0	0,0080	0,0088	0,9827	0	0	0
Nov/2000	0	0	0,0817	0,0077	0,9106	0	0	0
Dez/2000	0,0229	0	0,0091	0,0073	0,9543	0	0	0,0065

## 8 CONCLUSÕES E RECOMENDAÇÕES

Uma das tarefas de maior importância na administração financeira é a seleção de carteiras, processo pelo qual se selecionam, dentre as várias alternativas existentes, os ativos que farão parte de uma carteira de investimentos. A forma como essa tarefa é realizada atualmente baseia-se nas idéias da Moderna Teoria de Carteiras, conjunto de teorias que tem como pilar básico o método de Markovitz.

Devido à complexidade do processo de avaliação de ações e seleção de carteiras, tais teorias, apesar de brilhantes e amplamente aceitas pela comunidade científica internacional, ainda não conseguem captar de forma satisfatória as relações entre as variáveis desse problema, de modo que pudessem explicar os eventos ocorridos no passado e realizar previsões sobre eventos futuros.

O objetivo do trabalho, dessa forma, foi verificar a possibilidade de se utilizar um modelo composto por métodos genéricos, que pudessem captar as especificidades existentes no processo de seleção de carteiras e obtivessem um desempenho igual ou melhor do que aquele visto pela utilização do modelo tradicional, representado pelo método de Markovitz.

Para isso, construiu-se um modelo composto por um algoritmo genético e diversas redes neurais e simulou-se o investimento mensal em carteiras de ações durante um período de dois anos, utilizando-se o modelo recém-criado e a versão *ex-ante* do método de Markovitz. A partir dos retornos obtidos pela simulação das duas estratégias de investimento, calcularam-se seus riscos (dados pelas variabilidades das taxas de retorno) e os respectivos índices de Sharpe, utilizados como critério de comparação entre os métodos.

Viu-se, após a execução dos dados, que o modelo híbrido composto pelo algoritmo genético e pelas redes neurais obteve um desempenho superior ao método de Markovitz. Os retornos acumulados durante os dois anos de simulação foram de 257,24% para o método tradicional e 419,41% para o modelo alternativo.

Além disso, apesar de as variabilidades das taxas de retorno mostrarem-se um pouco maiores ao utilizarmos o modelo híbrido, viu-se que a relação retorno/risco – representada pelo índice de Sharpe – foi melhor no modelo composto por métodos genéricos.

Viu-se também que ambas as abordagens obtiveram resultados melhores que o retorno médio do mercado, representado pelo índice Ibovespa. Vale ressaltar que os dois métodos utilizaram apenas informações históricas em suas projeções, o que é mais um indício da possibilidade de prever retornos futuros com base em informações passadas.

A principal contribuição desse trabalho, dessa forma, foi a verificação de que é possível, sim, que um modelo composto por métodos genéricos possa obter resultados iguais ou melhores que os modelos tradicionais de seleção de carteiras, representados aqui pelo método de Markovitz. Mostrou-se que, ao integrarmos diferentes áreas de conhecimento, foi possível captar melhor as inter-relações entre as variáveis do problema de modo a obtermos melhores resultados.

O melhor desempenho do modelo criado tem como possíveis razões dois fatores principais. O primeiro deles, e mais importante, seria a melhor capacidade das redes neurais de realizar previsões quanto ao comportamento futuro dos preços das ações, quando comparada à utilização de uma regressão linear a partir de dados históricos das cotações, como foi feito para o método de Markovitz. Quanto melhores forem as

previsões, mais acertadas serão as alocações das ações dentro das carteiras escolhidas e melhores serão os investimentos realizados.

O outro fator seria a não utilização, no modelo criado, de covariâncias históricas na tentativa de eliminar parte do risco através da diversificação. Se as matrizes de covariância geradas para o método de Markovitz não refletirem bem a realidade, o que é possível, estaríamos eliminando risco onde não deveríamos, fazendo com que o retorno fosse comprometido sem nenhuma troca em diminuição de risco.

De qualquer maneira, pelo fato de o período de análise ser de apenas dois anos, não se podem generalizar os resultados obtidos e afirmar que tal modelo teria sempre desempenho superior ao método tradicional. Para isso, seria necessário que se ampliasse o período analisado, afim de se obterem resultados mais robustos e confiáveis.

Entre as sugestões para trabalhos futuros estariam a validação do modelo híbrido utilizado na comparação das estratégias e a utilização de horizontes de investimento de menor tamanho, que poderiam ser semanais ou até mesmo diários. Ao diminuirmos o horizonte de investimentos, resolveria-se uma das principais dificuldades encontradas nesse trabalho, que foi a pequena quantidade de dados disponíveis para o treinamento das redes neurais. Utilizando-se mais dados, as previsões de retornos seriam, com grandes possibilidades, mais acuradas que as atuais, o que faria com que as carteiras geradas tivessem relações risco-retorno ainda melhores. Deve-se apenas ter em mente o fato de que, ao diminuir o horizonte de investimentos, não se está apenas aumentando a quantidade de dados disponível para o treinamento da rede neural. Ao fazer isso, também se está diminuindo a quantidade de variáveis que podem ser utilizadas como entrada para o modelo, pois tais variáveis necessitam ser atualizadas com, no mínimo, a mesma periodicidade do tamanho do horizonte de investimentos.

Outra sugestão seria a utilização de outras funções de adequação durante o processo de otimização, através do algoritmo genético. Um exemplo seria o uso de uma função que tire proveito da diversificação, como faz o método de Markovitz, eliminando parte do risco.

Também como trabalho futuro estaria a utilização de outro critério de escolha da carteira a ser investida, durante a execução do método de Markovitz. Dessa forma, não se escolheria a carteira de menor risco, dentre aquelas presentes na fronteira eficiente, mas sim aquela que apresente melhor relação entre risco e retorno.

## 9 REFERÊNCIAS BIBLIOGRÁFICAS

- 1 ABHYANKAR A.; COPELAND L. e WONG W. Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. **Journal Of Business & Economic Statistics**, v. 15, n. 1, p. 1-14, Jan. 1997.
- 2 ALLEN F. e KARJALAINEN R. Using genetic algorithms to find technical trading rules. **Journal Of Financial Economics**, v. 51, n. 2, p. 245-271, Feb. 1999.
- 3 ARIFOVIC, J. e GENÇAY, R. Statistical properties of genetic learning in a model of exchange rate. **Journal Of Economic Dynamics & Control**, v. 24, n. 5-7, p. 981-1005, June 2000.
- 4 BALAKRISHNAN, K. e HONAVAR, V. Evolutionary design of neural architectures – a preliminary taxonomy and guide to literature. **TR95-01**, Ames, 1995. Acesso em 05/08/2001. Disponível em <http://citeseer.nj.nec.com/balakrishnan95evolutionary.html>.
- 5 BEASLEY, D.; BULL, D. e MARTIN, R. An overview of genetic algorithms: Part 2, Research topics. **University Computing**, v. 15, n. 4, p.170-181, 1993a.
- 6 BEASLEY, D.; BULL, D. e MARTIN, R. An overview of genetic algorithms: Part 1, Fundamentals. **University Computing**, v. 15, n. 2, p. 58-69, 1993b.

- 7 BRAGA, A.; LUDEMIR, T. e CARVALHO, A. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: Livros Técnicos e Científicos, 2000.
- 8 BREALEY, R. e MYERS, S. **Princípios de finanças empresariais**. 5. ed. Lisboa: McGraw-Hill, 1998.
- 9 BRUNI, A. e FAMÁ, R. Moderna teoria de portfólios: é possível captar, na prática, os benefícios decorrentes da sua utilização? **Resenha BM&F**, n. 128, 1998.
- 10 BURGESS, A. e REFENES, A. Modeling non-linear moving average process using neural networks with error feedback: an application to implied volatility forecasting. **Signal Processing**, v. 74, p. 89-99, 1999.
- 11 DAVIS, L. **Handbook of genetic algorithms**. Van Nostrand Reinhold, 1991 apud MITCHELL, M. **An introduction to genetic algorithms**. Cambridge: MIT Press, 1996.
- 12 DeJONG, K. **An analysis of the behavior of a class of genetic adaptive systems**. Ph.D. Dissertation, University of Michigan, 1975 apud: BEASLEY, D.; BULL, D. e MARTIN, R. An overview of genetic algorithms: Part2, Research topics. **University Computing**, v. 15, n. 4, p.170-181, 1993a.
- 13 DESAI, V. e BHARATI, R. The efficacy of neural networks in predicting returns on stock and bond indices. **Decision Sciences**, v. 29, n. 2, p. 405-425, 1998.
- 14 DONALDSON R. e KAMSTRA M. Forecast combining with neural networks. **Journal of Forecasting**, v. 15, n. 1, p. 49-61, Jan. 1996.

- 15 ELTON, E. e GRUBER, M. **Modern portfolio theory and investment analysis**. 5. ed. New York: John Wiley & Sons, 1995.
- 16 FIGUEIREDO, A.; DRESCH, A; ZANINI, F.; BROCHMANN, L e FRANZ, P. A utilização da teoria de carteiras de Markovitz e do modelo de índice único de sharpe no mercado de ações brasileiro em 1999. **Resenha BM&F**, São Paulo, n. 141, set/out 2000.
- 17 GOLDBERG, D. The design of innovation: lessons from genetic algorithms, lessons from the real world. **Illinois Genetic Algorithms Laboratory Report**, n. 98004, 1998.
- 18 GOLDBERG, D. Genetic and evolutionary algorithms come of age. **Communications of the ACM**, v. 37, n. 3, p.113-119, Mar. 1994.
- 19 GOLDBERG, D. **Genetic algorithms in search, optimization, and machine learning**. Reading, Mass.: Addison-Wesley, 1989.
- 20 HAUGEN, R. **The new finance: the case against efficient markets**. Englewood Cliffs: Prentice Hall, 1995.
- 21 HAUGEN, R. **Modern investment theory**. 5. ed. Englewood Cliffs: Prentice Hall, 2001.
- 22 HAWLEY, D; JOHNSON, J. e RAINA, D. **Artificial neural systems: a new tool for financial decision making**. In: TRIPPI, R. e TURBAN, E. (Ed.). **Neural networks in finance and investing: using artificial intelligence to improve real-world performance**. New York: McGraw Hill, 1996.

- 23 HAYKIN, S. **Neural networks**: a comprehensive foundation. 2. ed. Englewood Cliffs: Prentice Hall, 1999.
- 24 HOLLAND, J. **Adaptation in natural and artificial systems**. Ann Arbor, University of Michigan Press, 1975.
- 25 KHURI, S.; BÄCK, T. e HEITKÖTTER, J. An evolutionary approach to combinatorial optimization problems. **Proceedings of CSC'94**, Phoenix, p.66-73, Mar. 1994.
- 26 KLEIN B. e ROSSIN D. Data quality in neural network models: effect of error rate and magnitude of error on predictive accuracy. **Omega-International Journal Of Management Science**, v. 27, n. 5, p. 569-582, Oct. 1999.
- 27 KLIMASAUSKAS, C. **Neural network techniques**. In: DEBOECK, G. (Ed.). **Trading on the edge**: neural, genetic, and fuzzy systems for chaotic financial markets. New York: John Wiley & Sons, 1994.
- 28 KOUCK, C.; JOINES, J. e KAY, M. A genetic algorithm for function optimization: a Matlab implementation. **NCSU-IE TR 95-09**, Raleigh, 14p. 1995. Disponível em <http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/>. Acesso em 05/10/2001.
- 29 KOZA, J. **Genetic programming**: on the programming of computers by means of natural selection. Cambridge: MIT Press, 1992.
- 30 KRÖSE, B. e SMAGT, P. **An introduction to neural networks**. Amsterdam: University of Amsterdam, 1993, (Ph.D. Thesis).

- 31 LeBARON B.; ARTHUR W. e PALMER R. Time series properties of an artificial stock market. **Journal Of Economic Dynamics & Control**, v. 23, n. 9-10, p. 1487-1516, Sept. 1999.
- 32 LEUNG M.; DAOUK H. e CHEN A. Forecasting stock indices: a comparison of classification and level estimation models. **International Journal Of Forecasting**, v. 16, n. 2, p. 173-190, Apr-June 2000.
- 33 MARKOVITZ, H. **Portfolio selection**: efficient diversification of investments. New York: John Wiley & Sons, 1959.
- 34 MEDSKER, L.; TURBAN, E. e TRIPPI, R. **Neural network fundamentals for financial analysts**. In: TRIPPI, R. e TURBAN, E. (Ed.). **Neural networks in finance and investing**: using artificial intelligence to improve real-world performance. New York: McGraw Hill, 1996.
- 35 MITCHELL, M. **An introduction to genetic algorithms**. Cambridge: MIT Press, 1996.
- 36 OSÓRIO, F. e VIEIRA, R. Sistemas híbridos inteligentes. XIX Congresso da SBC. Tutorial do Encontro Nacional de Inteligência Artificial. Rio de Janeiro, 60 p., jul.1999.
- 37 PORTUGAL, M. e FERNANDES, L. G. Redes neurais artificiais e previsão de séries econômicas: uma introdução. **Nova Economia**, v. 6, n. 1, p. 51-74, 1996.
- 38 Qi, M. Nonlinear predictability of stock returns using financial and economic variables. **Journal Of Business & Economic Statistics**, v. 17, n. 4, p. 419-429, Oct. 1999.

- 39 RABATIN, A. Adaptive portfolio trading using genetic algorithms. Fifth International Conference on Forecasting of Financial Markets. London, May 1998a.
- 40 RABATIN, A. Artificial Intelligence based risk management strategies for trading portfolios. **Unicom Seminars**, London, Apr. 1998b.
- 41 RABATIN, A. Adaptive portfolio trading strategies for foreign exchange portfolios. **BNP Global Markets Research**, Dec. 1997.
- 42 RAHIMIAN, E.; SINGH, S.; THAMMACHOTE, T. e VIRMANI, R. **Bankruptcy prediction by neural network**. In: TRIPPI, R. e TURBAN, E. (Ed.). **Neural networks in finance and investing: using artificial intelligence to improve real-world performance**. New York: McGraw Hill, 1996.
- 43 ROSS, S.; WESTERFIELD, R. e JAFFE, J. **Administração financeira**. São Paulo: Atlas, 1995.
- 44 RUSSEL, S. e NORVIG, P. **Artificial intelligence: a modern approach**. Englewood Cliffs: Prentice Hall, 1995.
- 45 SÁ, G. **Administração de investimentos, teoria de carteiras e gerenciamento do risco**. Rio de Janeiro: Qualitymark, 1999.
- 46 SHARPE, W. **Portfolio theory and capital markets**. New York: McGraw-Hill, 1970.

- 47 SHARPE, W.; ALEXANDER, G. e BAILEY, J. **Investments**. 6. ed. Upper Saddle River: Prentice Hall, 1999.
- 48 SPEARS, W. **The role of mutation and recombination in evolutionary algorithms**. Washington: George Mason University, Ph.D. Thesis, 1998.
- 49 TRIPPI, R. e LEE, J. **Artificial intelligence in finance and investing**. New York: Irwin, 1996.
- 50 VARETTO F. Genetic algorithms applications in the analysis of insolvency risk. **Journal Of Banking & Finance**, v. 22, n. 10-11, p. 1421-1439, Oct. 1998.
- 51 YAO J.; LI Y. e TAN C. Option price forecasting using neural networks. **Omega – The International Journal Of Management Science**, v. 28, n. 4, p. 455-466, Aug. 2000.
- 52 YAO, X. Evolutionary artificial neural networks. **International Journal of Neural Systems**, v. 4, n. 3, p. 203-222, 1993.

## 10 GLOSSÁRIO

**Adequação ao ambiente:** grau em que uma *solução candidata* consegue se adaptar ao ambiente que a cerca. Quanto maior a sua adequação, maiores são as chances de essa solução sobreviver e se reproduzir, gerando descendentes. É definida por uma *função de adequação*.

**Algoritmos genéticos (AGs):** algoritmos que utilizam as idéias da *seleção natural* e da genética em problemas de busca e otimização. O princípio básico desses algoritmos é a evolução das soluções do problema de acordo com seu grau de *adequação ao ambiente*.

**Back-propagation:** algoritmo mais comumente utilizado no aprendizado das RNs. É um algoritmo do tipo supervisionado: para cada padrão de entrada, fornece-se também a saída. Ajusta os pesos da rede através de um mecanismo que propaga os erros pelas camadas, começando da última e terminando na primeira.

**Codificação:** forma pela qual as soluções candidatas são estruturadas em um AG. Depende principalmente do alfabeto de símbolos usados na representação. Na codificação binária, por exemplo, apenas dois símbolos são utilizados.

**Crossover:** o mesmo que *Recombinação*.

**Espaço de busca:** conjunto de todos os pontos que representam soluções para o problema. Quanto maior e menos conhecido o *espaço de busca*, maiores são as chances de que seja necessária uma *heurística* para vasculhá-lo.

**Evolução:** processo pelo qual os indivíduos se tornam mais adaptados ao ambiente com o decorrer do tempo.

**Feed-forward:** ver *Rede feed-forward*.

**Fronteira eficiente:** fronteira constituída pelas chamadas carteiras eficientes, ou seja, carteiras que apresentam o menor risco possível, dado o nível de retorno, ou o maior retorno possível, dado o nível de risco. É gerada pelo método de Markovitz.

**Função de adequação:** função que avalia a *adequação ao ambiente* de determinado indivíduo, num algoritmo genético.

**Função de ativação:** função que utiliza os sinais de entrada e os pesos das conexões de uma *unidade de processamento* de uma rede neural, gerando valores de saída e propagando-os para as unidades seguintes.

**Geração:** ponto específico do tempo no processo de *evolução*. Em cada geração, um conjunto de indivíduos provindos da geração anterior representa as *soluções candidatas* do problema.

**Heurística:** técnica que permite que se percorra apenas uma fração do *espaço de busca* de um problema, mesmo que isso implique em uma pequena perda na precisão da resposta.

**Método genérico:** método que não é específico de uma ou outra área do conhecimento. São aplicáveis em problemas cujo espaço de soluções é extremamente grande (onde uma busca exaustiva não é possível) e quando a teoria relativa ao campo de conhecimento em questão não consegue explicar o comportamento dos fenômenos observados.

**Modelo híbrido:** modelo que une idéias de duas ou mais teorias para representar uma área do conhecimento.

**Mutação:** troca de um ou mais caracteres de um indivíduo. É utilizada num algoritmo genético para garantir que todos os pontos do *espaço de busca* tenham alguma chance de serem alcançados.

**Nodo:** o mesmo que *unidade de processamento*, numa rede neural.

**Operadores genéticos:** operadores inspirados em idéias da genética, utilizados num AG durante a fase de reprodução dos indivíduos. Os mais utilizados são a *recombinação* e a *mutação*.

**População:** conjunto de indivíduos pertencentes a uma mesma *geração*, representando as diferentes soluções de um problema.

**Problema de busca:** problema em que se procura uma solução pertencente a um espaço de soluções predefinido.

**Problema de otimização:** problema em que se busca a melhor solução dentre todas as possíveis.

**Programação quadrática:** método utilizado por Markovitz para a geração da *fronteira eficiente*. É caracterizado pela identificação das carteiras de canto e pelo cálculo das carteiras formadas por combinações de carteiras de canto adjacentes.

**Recombinação:** mistura do material genético dos pais na geração de seus descendentes. É utilizada pelos algoritmos genéticos, durante o processo de

reprodução, e geralmente é realizada através da escolha aleatória de um ponto de corte na *string* que representa o indivíduo.

**Redes neurais (RNs):** sistemas paralelos compostos por *unidades de processamento* simples que calculam determinadas funções matemáticas. Buscam inspiração na estrutura neuronal de organismos inteligentes, em que o conhecimento é adquirido através da experiência.

**Rede *feed-forward*:** uma das possíveis estruturas de uma rede neural, em que as conexões são sempre unidirecionais. Nessas redes, um sinal que sai de um nodo nunca volta ao mesmo ponto.

**Rede recorrente:** estrutura de uma rede neural em que nem todas as conexões são unidirecionais. Ao contrário das *redes feed-forward*, nas *redes recorrentes*, vê-se a presença de pelo menos um ciclo.

**Seleção natural:** processo pelo qual os indivíduos mais adaptados ao ambiente têm maiores chances de sobreviverem e gerarem descendentes.

**Solução candidata:** uma das possíveis soluções para um problema, representada num algoritmo genético através de uma *string*.

**String:** cadeia de caracteres de tamanho finito, utilizada num algoritmo genético para representar as *soluções candidatas* de um problema.

**Unidades de processamento:** unidades encontradas nas redes neurais que recebem sinais como entrada, ponderam tais sinais de acordo com o peso da conexão proveniente, somam esses resultados e enviam uma saída às *unidades de processamento* seguintes caso o valor encontrado exceda determinado limite.