



UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE ESPECIALIZAÇÃO EM AUTOMAÇÃO INDUSTRIAL

LÁZARO COELHO PEREIRA

**Utilização de Algoritmo Genético para a
Solução do Problema de Proporção de
Carvões para Produção de Coque
Metalúrgico**

Trabalho de Conclusão apresentado como
requisito parcial para a obtenção do grau de
Especialista em Automação Industrial

Prof. Dr. Felipe Campelo
Orientador

Prof. Dr. Eduardo Mazoni A. Marçal Mendes
Coordenador do Curso

Belo Horizonte, junho de 2014.

UNIVERSIDADE FEDERAL DE MINAS GERAIS

Reitor: Prof. Jaime Arturo Ramírez

Vice-Reitora: Prof. Sandra Regina Goulart Almeida

Pró-Reitor de Pós-Graduação: Prof. Rodrigo Antonio de Paiva Duarte

Coordenador do PPGEE: Prof. Dr. Rodney R. Saldanha

Agradecimentos

Agradeço primeiramente a Deus por me dar a oportunidade de crescimento profissional e de realizar esta monografia. Agradeço a minha esposa Alessandra por ter sempre me apoiado em todos os momentos de minha vida, principalmente no período de realização desta pós-graduação e desta monografia. Agradeço a todos os professores que fizeram parte disto e principalmente a Prof. Felipe Campelo por me aceitar como orientando e por confiar no trabalho proposto.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	10
LISTA DE TABELAS	11
RESUMO	12
ABSTRACT	13
1 INTRODUÇÃO	14
2 REFERENCIAL TEÓRICO	17
2.1 Redes Neurais	17
2.1.1 Neurônio biológico x Neurônio Artificial.....	17
2.1.2 Redes de camada única.....	19
2.1.3 Redes de múltiplas camadas.....	19
2.2 Otimização	20
2.2.1 Algoritmo Genético.....	21
2.2.1.1 Inicializar a população.....	23
2.2.1.2 Avaliar a população.....	24
2.2.1.3 Selecionar a população.....	24
2.2.1.4 Modificar a população.....	25
2.2.1.5 Critérios de convergência.....	27
3 O PROBLEMA DE MISTURA DE CARVÕES	29
3.1 Mistura de carvões	29
3.2 Introdução de RNAs como restrições no problema	30
3.3 O otimizador de carvão	34
3.3.1 O problema da mistura de carvões e a RNA do CSR.....	34
3.3.2 A montagem do problema.....	35
3.4 Resultados	37
3.4.1 Resultados da RNA.....	37
3.4.2 Resultados do otimizador.....	39

4 CONCLUSÃO.....	49
REFERÊNCIAS.....	51

LISTA DE ABREVIATURAS E SIGLAS

CSR	<i>Coke Strength after Reaction of CO₂</i>
CRI	<i>Coke Reactivity Index</i>
Cz	Cinza
DI	<i>Drum Index</i>
GA	Algoritmo genético
IAB	Instituto Aço Brasil
LB	<i>Lower Bound</i>
MLP	<i>Multilayer Perceptron</i>
PCI	<i>Pulverized Coal Injection</i>
RNA	Rede Neural Artificial
S	Enxofre
UB	<i>Upper Bound</i>
RMSE	<i>Root Mean Square Error</i>

LISTA DE FIGURAS

<i>Figura 2.1: Representação de um neurônio biológico (BRAGA, 2007).....</i>	<i>17</i>
<i>Figura 2.4: Exemplo de uma MLP (BOCCO, 2010).</i>	<i>20</i>
<i>Figura 2.5: Exemplo de uma roleta (Mognon, 2004).....</i>	<i>25</i>
<i>Figura 2.6: Exemplo de um torneio (Mognon, 2004).....</i>	<i>25</i>
<i>Figura 2.7: Exemplo de cruzamento por ponto duplo (Mognon, 2004).....</i>	<i>26</i>
<i>Figura 2.8: Exemplo de cruzamento por ponto duplo (Mognon, 2004).....</i>	<i>26</i>
<i>Figura 2.9: Exemplo de cruzamento aleatório (Mognon, 2004).....</i>	<i>26</i>
<i>Figura 2.11: Exemplo de mutação (Mognon, 2004).</i>	<i>27</i>
<i>Figura 2.12: Exemplo de um critério de parada escolhido de forma errada (Soares, 1997).....</i>	<i>27</i>
<i>Figura 3.1: Processo Geológico de formação do carvão mineral (trufa, sub-betuminoso, betuminoso e antracito) (Casagrande, p.7).</i>	<i>29</i>
<i>Figura 3.2: RNA para predição do CSR.</i>	<i>33</i>
<i>Figura 3.3: Exemplo da montagem de matrizes para solução do problema.....</i>	<i>36</i>
<i>Figura 3.4: Exemplo da montagem da matriz Aeq.....</i>	<i>36</i>
<i>Figura 3.6: Exemplo da montagem das matrizes LB e UB</i>	<i>37</i>
<i>Figura 3.7: Comparação entre CSR real e CSR previsto pela rede neural para massa de dados de treinamento e validação.</i>	<i>38</i>
<i>Figura 3.8: Comparação entre CSR real e CSR previsto pela rede neural para massa de dados de teste.</i>	<i>39</i>
<i>Figura 3.9: Comportamento do GA e do Fmincon – população GA = 300.....</i>	<i>44</i>
<i>Figura 3.10: Comportamento do GA e do Fmincon – população GA = 200.....</i>	<i>45</i>
<i>Figura 3.11: Comportamento do GA e do Fmincon – população GA = 150.....</i>	<i>45</i>
<i>Figura 3.12: Comportamento do GA e do Fmincon – população GA = 100.....</i>	<i>46</i>
<i>Figura 3.13: Comportamento do GA e do Fmincon – população GA = 80.....</i>	<i>47</i>
<i>Figura 3.14: Tamanho da população e o tempo para solução.</i>	<i>47</i>

LISTA DE TABELAS

<i>Tabela 2.1: comparação entre genética artificial e genética natural</i>	22
<i>Tabela 3.1: Correlação entre parâmetros de qualidade da mistura de carvões e o coque</i>	32
<i>Tabela 3.2: Métricas para validação do modelo</i>	38
<i>Tabela 3.3: Métricas para validação do modelo</i>	38
<i>Tabela 3.3: Tempo decorrido para solução do problema – população do GA = 300</i>	43
<i>Tabela 3.4: Tempo decorrido para solução do problema – população do GA = 200</i>	44
<i>Tabela 3.5: Tempo decorrido para solução do problema – população do GA = 150</i>	45
<i>Tabela 3.6: Tempo decorrido para solução do problema – população do GA = 100</i>	46
<i>Tabela 3.7: Tempo decorrido para solução do problema – população do GA = 80</i>	46
<i>Tabela 4.1: Variação entre o valor da mistura encontrada pelos metalurgistas e encontrada pelo GA. ..</i>	50

RESUMO

Em busca de aumentos de competitividade em um cenário de concorrência cada vez mais acirrada, as empresas procuram sempre maximizar seus lucros e/ou minimizar seus custos. Para alcançar estes objetivos uma metodologia muito utilizada é a implementação e utilização de ferramentas de otimização integradas ao processo produtivo. Desta forma espera-se obter a máxima eficiência do processo com o menor custo possível.

Um exemplo de utilização dessa metodologia é a solução do problema da mistura de carvões para a produção do coque metalúrgico. O coque metalúrgico é um insumo muito importante em um alto forno, pois é o combustível de custo mais elevado e de extrema importância na estabilização do processo do alto forno.

Este trabalho mostra o desenvolvimento de uma ferramenta visando à geração da melhor mistura de carvões metalúrgicos para a produção de coque. Normalmente os otimizadores de carvão metalúrgico são limitados por utilizarem apenas programação linear para solucionar este problema. Contudo, no presente trabalho tal formulação não seria eficiente, devido à existência de uma restrição relacionada ao valor de uma propriedade do coque, calculado através de uma rede neural. Desta forma tem-se um problema não linear, que no presente trabalho é resolvido através do uso de um algoritmo genético.

Palavras-Chave: Carvão metalúrgico, otimização, algoritmo genético.

ABSTRACT

In a highly competitiveness market environment companies are always trying reduce their costs and/or increase their profits. To reach these objectives a methodology used is the implementation of optimization tools integrated to the production process. Thus, it is expected to get maximum process efficiency with the minimum cost.

An example of utilization this methodology is the solution of the problems of blending of coals to produce coke. The metallurgical coke is the most important raw material in a blast furnace, because it is the fuel that has highest cost and it is extremely important to stabilization process of the blast furnace.

This study shows the development of a tool which is the propose of finding the best blending of metallurgical coals to produce coke. Normally the metallurgical coal optimizers are limited because they utilize only linear programming to solve this problem. However in the present study such formulation would not be efficient due to the existence of a restriction to the value of a feature of the coke, calculated using a neural network. Thus it is found a no linear problem which is solved through the use genetic algorithm.

Keywords: Metallurgical coals, optimization, genetic algorithm.

1 INTRODUÇÃO

O carvão metalúrgico, também chamado de carvão betuminoso, tem características que permitem, a partir da utilização de uma mistura de tais carvões em uma coqueria, a produção do coque metalúrgico. O coque é muito importante em usinas siderúrgicas integradas, pois é o principal combustível utilizado nos altos-fornos.

A coqueria é uma unidade industrial formada por uma ou mais baterias de coque, onde ocorre a transformação da mistura de carvões metalúrgicos em coque. As baterias de uma coqueria são compostas por vários fornos verticais que são aquecidos pelo calor resultante da combustão de um gás e onde ocorre a transformação da mistura de carvões metalúrgicos em coque.

Durante o processo de coqueificação a mistura de carvões é aquecida a uma temperatura de aproximadamente 1100 °C. No interior dos fornos não há circulação de ar, o que provoca a liberação de gases provenientes das transformações ocorridas na mistura de carvão – material volátil – transformando a matéria-prima em coque.

O Brasil ocupa lugar de destaque mundial na produção de aço com mais de 34 milhões de toneladas produzidas no ano passado (IAB, 2014). Os carvões metalúrgicos utilizados nas coquerias têm uma importância muito grande nas usinas siderúrgicas integradas a coque, devido a sua grande influência no custo do aço produzido que impacta diretamente no custo total da empresa.

Quando se fala da mistura de carvões metalúrgicos na a produção de coque é necessário atentar para várias situações como custo dos carvões, transporte marítimo, descarregamento nos portos, custos de frete dentre outros (Silva, 2011), já que normalmente as empresas adquirem milhares de toneladas de carvões por ano.

PRODUÇÃO SIDERÚRGICA BRASILEIRA

PRODUTOS	JAN/DEZ		12/11 (%)	OUT 2012	NOV 2012	DEZEMBRO		12/11 (%)	ÚLTIMOS 12 MESES
	2012(*)	2011				2012(*)	2011		
	AÇO BRUTO	34.681,9				35.220,1	(1,5)		
LAMINADOS	26.238,6	25.239,9	4,0	2.277,7	2.177,6	1.941,5	1.877,2	3,4	26.238,6
PLANOS	15.444,0	14.264,9	8,3	1.344,1	1.326,9	1.271,4	1.121,6	13,4	15.444,0
LONGOS	10.794,6	10.975,0	(1,6)	933,6	850,7	670,1	755,6	(11,3)	10.794,6
SEMI-ACABADOS P/VENDAS	7.210,4	8.037,8	(10,3)	619,7	595,1	599,4	696,1	(13,9)	7.210,4
PLACAS	6.043,3	6.745,2	(10,4)	512,5	473,6	492,2	565,0	(12,9)	6.043,3
LINGOTES, BLOCOS E TARUGOS	1.167,1	1.292,6	(9,7)	107,2	121,5	107,2	131,1	(18,2)	1.167,1
FERRO-GUSA (Usinas Integradas)	27.045,4	27.466,8	(1,5)	2.442,0	2.180,2	2.304,3	2.254,9	2,2	27.045,4

(*) Dados Preliminares.

Fonte: Aço Brasil

Tabela 1.1: Produção Siderúrgica Brasileira

Segundo Silva (2011), hoje em dia, apesar de toda tecnologia que nos cerca, ainda existe a necessidade de se realizar vários trabalhos voltados para a otimização nos processos produtivos na maioria das indústrias. Dentre os muitos problemas nos processos produtivos, pode-se destacar o problema de mistura de matérias-primas e levando este problema para o cenário de uma empresa siderúrgica integrada, temos o problema da mistura dos carvões metalúrgicos.

Silva (2011) também fornece, de forma resumida, uma definição do problema a ser resolvido: "a formulação de misturas para coqueificação consiste na definição dos carvões e de suas proporções em cada mistura, para se obter um coque na qualidade desejada, a partir dos carvões disponíveis".

Normalmente um problema como o de mistura dos carvões pode ser modelado como um problema clássico de programação linear chamado de *O Problema da Dieta*, onde o objetivo é encontrar a proporção correta de carvões metalúrgicos que formarão a melhor mistura com o menor custo, sempre respeitando as restrições de processo e qualidade impostas. Estas restrições estão ligadas a exigências de qualidade do coque colocadas pelo controle térmico e metalúrgico do alto-forno e também por necessidades operacionais da coqueria.

Ludenberg e Ye (2008) descrevem o problema da dieta da seguinte forma: considere n alimentos com m ingredientes nutricionais e um preço c_j para cada alimento. Além disso, considere que cada pessoa precisa de certa quantidade de ingredientes nutricionais por dia. Assumindo x_j o número de unidades de cada alimento na dieta, o problema consiste em obter a quantidade adequada de cada alimento de forma a minimizar o custo e atender a todas as restrições nutricionais.

Sendo assim, pode-se descrever o problema da mistura de carvões metalúrgicos como um problema onde cada carvão j possui m_j propriedades físicas e metalúrgicas e cada carvão tem um custo c_j . Sendo assim, como dividir os j carvões de forma a se obter a mistura de melhor qualidade e com o menor custo possível.

No presente trabalho, o problema da mistura de carvões não será abordado somente como um problema básico como o problema da dieta. A razão para a necessidade de uma formulação diferenciada reside na adição de uma restrição não linear que será chamada de restrição especial, e que está ligada ao valor de uma dada característica desejada para o coque.

A qualidade do coque depende da mistura utilizada na coqueria e algumas características são definidas pelo ritmo de produção do alto forno e por restrições operacionais da coqueria. Desta forma, é importante que se desenvolva a mistura de forma a se obter os melhores valores de qualidade do coque, além das restrições de qualidade e custo da mistura.

Uma característica de qualidade do coque que é muito importante na operação do Alto Forno é o parâmetro CSR - *coke strenght after reaction of CO₂* (Resistência do coque após a reação). Neste caso este parâmetro será considerado como uma restrição especial no problema de mistura de carvões.

Normalmente se usa um modelo linear para predição do valor do CSR do coque durante a criação da mistura, mas como o número de variáveis da mistura já é muito grande e as variáveis ligadas a operação da coqueria nem sempre tem um comportamento que pode ser representado por um modelo linear decidiu-se predizer o valor do CSR através de um modelo não-linear.

Como o CSR é um parâmetro de qualidade do coque que é o produto das coquerias, será utilizada uma rede neural para encontrar o valor do CSR utilizando alguns parâmetros de qualidade da mistura e parâmetros de operação da coqueria.

As RNAs (redes neurais artificiais) são uma família de modelos não-lineares capazes de gerar representações ou aproximações do comportamento de diversos processos. RNAs são estruturas matemáticas que tem seu funcionamento baseado em simplificações de redes formadas por neurônios biológicos, reproduzindo características humanas como aprendizado, associação, generalização e abstração. Desta forma, após aprender os padrões de reação de um dado processo estes modelos são capazes de prover respostas para valores de entradas baseados no comportamento em situações conhecidas (Pacheco, 1999).

Desta forma, as entradas de uma RNA utilizada na predição do valor do CSR do coque serão alguns parâmetros de qualidade da mistura de carvões metalúrgicos e alguns parâmetros de operação da coqueria e a saída será a restrição especial, ou seja, o $CSR_{ESTIMADO}$.

A partir deste ponto o problema da mistura de carvões passa a ser um problema de programação não-linear, pois uma das restrições deste problema é uma restrição possivelmente não-linear, já que é o resultado de uma rede neural.

Para solucionar problemas de programação não linear existem vários tipos de técnicas e algoritmos. Em especial neste trabalho, o algoritmo escolhido para a solução deste problema é o Algoritmo Genético.

O algoritmo genético é baseado em uma metáfora simplificada do processo de evolução biológica. Ele fornece um mecanismo de busca ao mínimo global da função avaliada através da geração de uma população de soluções-candidatas e reprodução diferenciada de indivíduos com melhores valores de função objetivo, através da utilização de operadores conhecidos como cruzamento (ou recombinação) e mutação. Desta forma, o algoritmo genético consegue se aproximar progressivamente do melhor valor de função objetivo (Pacheco, 1999).

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentadas as fundamentações teóricas utilizadas na criação da ferramenta para solução do problema. Serão revisados conceitos sobre neurônios biológicos e artificiais, algoritmo perceptron e RNAs do tipo multi-layer perceptron na solução do problema de estimação do valor de CSR do coque. Já para a solução do problema de otimização serão abordados assuntos como programação linear, programação não linear e algoritmos genéticos.

2.1 Redes Neurais

2.1.1 Neurônio biológico x Neurônio Artificial

O cérebro humano é constituído de aproximadamente cem bilhões de neurônios e centenas de trilhões de conexões entre estes. Os neurônios biológicos podem ser divididos em três partes básicas: Dendritos, corpo celular e axônio que, juntos, são responsáveis pelo funcionamento destas células. Os dendritos tem a função de receber os impulsos nervosos de outros neurônios e levar esta informação ao corpo celular que ao receber informação de vários neurônios diferentes pode enviar uma informação para os neurônios seguintes ou não, isso irá depender do nível de excitação do corpo celular.

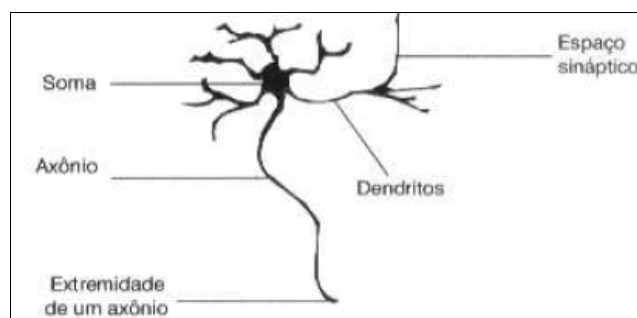


Figura 2.1: Representação de um neurônio biológico (BRAGA, 2007).

O modelo de um neurônio artificial originalmente proposto por McCulloch e Pitts (BRAGA, 2007) é uma simples interpretação do que se conhecia sobre um neurônio biológico, ou seja, um neurônio artificial contendo as principais partes de um neurônio biológico.

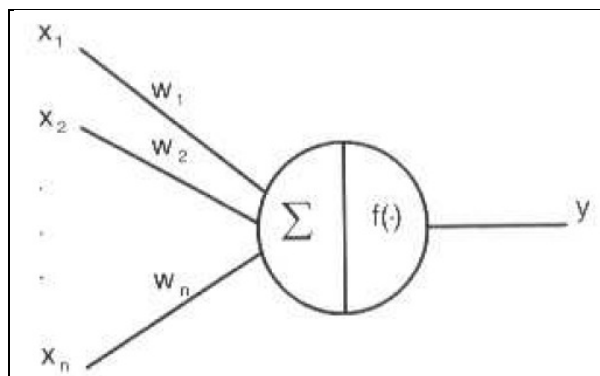


Figura 2.2: Representação de um neurônio artificial, McCulloch e Pitts (BRAGA, 2007).

As entradas do neurônio artificial dadas por x_1, x_2, \dots, x_n podem ser comparadas com os dendritos de um neurônio biológico, onde as entradas são responsáveis por captar estímulos externos. A saída y seria o axônio e, como no neurônio biológico, será ativada dependendo do valor do somatório das entradas multiplicadas por seus respectivos pesos e da função de ativação escolhida nesta estrutura.

Já o corpo é responsável por, de certa forma, interpretar a informação repassada pelas entradas e decidir pela ativação ou não da saída deste neurônio. O corpo realiza o somatório dos resultados da multiplicação do valor da entrada multiplicada por seu respectivo peso. Este valor é avaliado pela função de ativação que define a ativação da saída do neurônio.

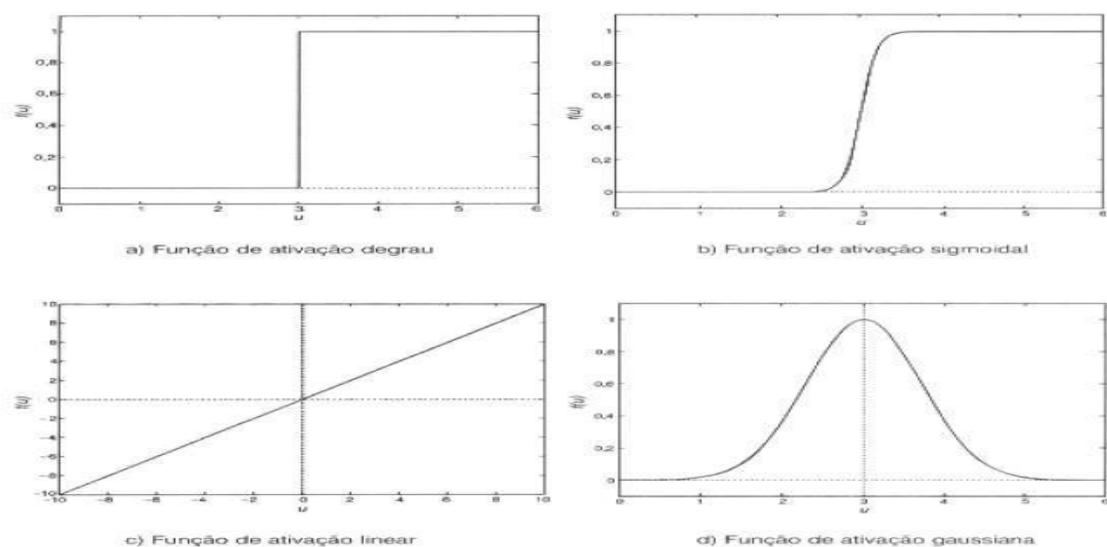


Figura 2.3: Exemplos de funções de ativação (BRAGA, 2007).

2.1.2 Redes de camada única

O trabalho proposto por McCulloch e Pitts mostra uma unidade de processamento que seria o neurônio artificial muito simples e apenas com o trabalho de Rosenblatt, é que se tornou possível a utilização de uma rede de neurônios artificiais utilizando um algoritmo de aprendizado que é realmente confiável, o algoritmo perceptron.

Alguns anos mais tarde, Rosenblatt demonstrou o teorema de convergência do perceptron, que mostra que um neurônio MCP treinado com o algoritmo de aprendizado do perceptron sempre converge caso o problema seja linearmente separável. (BRAGA 2007, p.28.).

O perceptron é a forma mais simples de uma RNA e sua topologia inicial possui três camadas: Retina, associação e resposta.

A camada da retina é responsável por captar os estímulos do meio externo, a camada de associação, que possui pesos definidos antes do treinamento e a camada de resposta.

Sendo assim o perceptron simples só pode resolver problemas de classificação e o problema deverá ser linearmente separável.

Outra estrutura importante é o adaline, com desenvolvimento em época próxima do perceptron, o modelo adaline possui funcionamento muito próximo do funcionamento do perceptron, já que também executa soma ponderada das entradas em sua unidade de processamento. É caracterizado por ter uma função de ativação linear e um termo de polarização que estipula um grau de liberdade, desta forma pode-se dizer que o adaline é um aproximador linear de funções.

Sendo assim, temos o perceptron como uma ferramenta para solução de problemas de classificação, desde que o problema seja linearmente separável. Já o adaline é uma ferramenta de aproximação linear de funções.

2.1.3 Redes de múltiplas camadas

As RNAs de múltiplas camadas foram criadas devido à incapacidade das RNAs de camada única resolverem problemas não-lineares, ampliando a possibilidade de utilização destes modelos para a solução de problemas mais gerais.

Quando se pensa em problemas do cotidiano, na maioria das vezes encontramos problemas com características não-lineares, sendo necessária a utilização de RNAs de múltiplas camadas como uma das possíveis técnicas de solução.

Quando se tem uma RNA onde os neurônios possuem funções de ativação não lineares e estes estão dispostos em camadas tem-se uma RNA de múltiplas camadas. Se os neurônios da camada intermediária possuírem uma função de ativação sigmoideal, esta RNA recebe o nome de Multilayer Perceptron – MLP.

Segundo Cybenko *apud* Braga (2007) “uma rede MLP com uma camada intermediária pode aproximar qualquer função contínua, enquanto a utilização de duas camadas intermediárias permite a aproximação de qualquer função”.

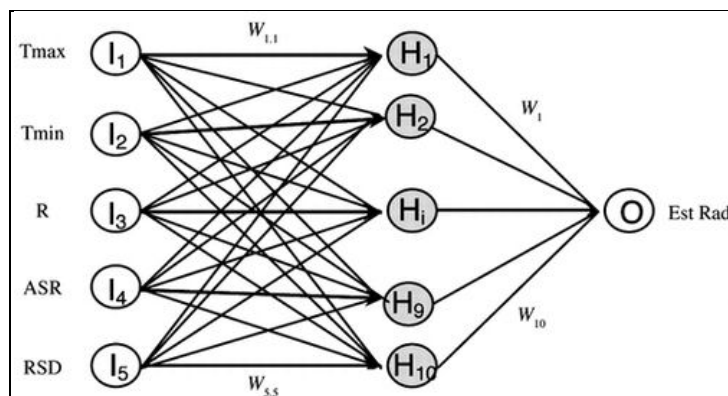


Figura 2.4: Exemplo de uma MLP (BOCCO, 2010).

2.2 Otimização

As regras de demanda e consumo do mundo atual exigem cada vez mais de qualquer processo produtivo, seja ele ligado ao setor alimentício, minerário, siderúrgico dentre outros. É cada vez maior a necessidade de se produzir mais, com maior qualidade e também com o menor custo.

Então, como produzir sendo cada vez mais competitivo?

Quando se tem um problema de complexidade baixa, segundo Soares, "muitas vezes, o fator experiência é suficiente na resolução destes problemas. No entanto, solucionar problemas mais complicados pode não ser tarefa fácil. Neste contexto, surge a otimização." (1997, p.4).

Embora a solução de alguns problemas possa ser encontrada através da análise de algumas observações, não se deve confundir este tipo de solução com uma solução encontrada através da aplicação de técnicas de otimização, uma vez que, dependendo do problema considerado, a simples adição de algumas restrições pode tornar a solução por análise de resultados advindos de tentativa e erro inviável. Desta forma, para que se tenha um ótimo resultado, utilizando o menor tempo na solução do problema, deve-se utilizar, segundo Silva (2011) "métodos computacionais de otimização que realizam uma busca racionalizada da solução ótima".

O conceito de otimização está bem identificado como um mecanismo de análise de decisões complexas, envolvendo seleção de valores para variáveis, com o simples objetivo de quantificar performance e medir a qualidade das decisões. A intenção é encontrar a melhor solução, respeitando, se necessário,

as restrições de viabilidade imposta aos parâmetros do problema. (SOARES, 1997, p.4)

Sendo assim, o uso da otimização como ferramenta na obtenção de resultados muito próximos do ponto ótimo de um determinado processo se torna essencial, já que é a forma mais concreta de encontrar a melhor solução dentro do espaço de soluções factíveis.

2.2.1 Algoritmo Genético

A criação do algoritmo genético - GA, assim como no caso das RNAs, foi baseada na observação de fenômenos da natureza. Enquanto que as RNAs representam uma tentativa de emular o funcionamento de cérebros biológicos, o GA é baseado em um modelo simplificado do processo de evolução biológica.

Segundo Pacheco, os GAs se inspiram em princípios naturais desenvolvidos originalmente por Charles Darwin. A teoria Darwiniana sugere que indivíduos menos adaptados a seu meio seriam mais susceptíveis a uma morte prematura, resultando em uma menor probabilidade de reprodução em relação a outros organismos mais bem adaptados. Desta forma, uma dada população de organismos tenderia, ao longo das gerações, a ser representada cada vez mais por indivíduos que herdaram características de organismos com maior capacidade de sobrevivência – maior *fitness* – levando a altos níveis de especialização e adaptação ao meio.

Os algoritmos genéticos são mecanismos de busca estocástica, direcionado pelas melhores soluções parciais baseadas nos processos de genética e seleção natural. Nessa analogia as possíveis soluções são indivíduos, e a função de otimização simula um ambiente de sobrevivência. Uma função de seleção escolhe quem vive para interagir com os operadores genéticos, resultando em novas soluções parciais. (Soares, 1997 p.19)

Seguindo esta teoria, são criados os GAs que procuram a melhor solução para o problema baseados na criação de populações com indivíduos cada vez mais evoluídos, ou seja, populações que estão cada vez mais próximas da solução do problema.

Tabela 2.1: comparação entre genética artificial e genética natural

Analogia entre sistemas naturais e artificiais.	
Genética Natural	Genética Artificial
gene	caractere
alelo	valor do caractere
cromossomo	cadeia de caracteres
locus	posição do gene na cadeia de caracteres
genótipo	estrutura, indivíduo
fenótipo	conjunto de parâmetros, ponto solução, estrutura decodificada
epistasia	não linearidade

Fonte: SOARES, 1997. p. 10.

Na natureza, os pais repassam suas características aos filhos através dos genes. O conjunto de genes dos filhos é formado por uma combinação dos genes dos pais, desta forma os filhos herdaram as características dos pais e os filhos que conseguirem as características mais importantes para que sejam fortes dentro do seu grupo, terão maiores condições de sobrevivência, desta forma os indivíduos mais fracos vão aos poucos sendo retirados da população.

Já no GA os “genes” (também ocasionalmente chamados de cromossomos) são valores, que podem ser combinações binárias, vetores etc., que compõem a solução do problema dada por um determinado indivíduo. Sendo assim, no momento da criação de um indivíduo filho ele terá como característica valores que são herdados das características de cada um dos pais. Alguns filhos herdarão características importantes para a solução do problema e continuarão na população enquanto outros herdarão características que não são relevantes para a solução do problema e por isto serão retirados da população.

Em geral o processo do GA tem a seguinte estruturação:

Entrada:

Tamanho da população (P);

Espaço de busca (E);

Funções objetivo e de restrição ($f()$, $g()$, $h()$);

Critério de paradas (Q);

Saída:

Estimativa da solução ótima (S);

Início:

$\mathbf{P}^{(0)} = \text{Inicializar a população}(\mathbf{P}, \mathbf{E})$

$t = 0$

Enquanto Q não for atendido faça:

$\emptyset^{(t)} = \text{Avaliar}(\mathbf{P}^{(t)}, \mathbf{f}(\cdot), \mathbf{g}(\cdot), \mathbf{h}(\cdot))$

$\mathbf{S}^{(t)} = \text{Selecionar}(\mathbf{P}^{(t)}, \emptyset^{(t)})$

$\mathbf{V}^{(t)} = \text{Modificar}(\mathbf{S}^{(t)})$

$\mathbf{P}^{(t+1)} = \text{Atualizar População}(\mathbf{P}^{(t)}, \mathbf{V}^{(t)})$

$t = t + 1$

Fim

Fim

Essas etapas serão detalhadas abaixo.

2.2.1.1 Inicializar a população.

A primeira população de um GA pode ser dada ou criada aleatoriamente dentro de uma dada região do espaço de busca. Já nas próximas etapas a criação das novas gerações é realizada através de operadores de variação.

No caso de GAs de codificação binária, as variáveis são expressas como cadeias de bits de comprimento l , dado pela quantidade de caracteres, ou cromossomos, deste indivíduo.

A escolha do tamanho da população, isto é, o número de pontos avaliados a cada iteração; do comprimento do cromossomo; e das probabilidades de cruzamento e de mutação são muito importantes para se obter um bom resultado de um GA. Erros na escolha destes parâmetros podem diminuir o seu desempenho.

O comprimento do cromossomo está ligado com à precisão do GA, quanto maior o cromossomo maior será a precisão, mas também maior será o espaço de possibilidades no qual o GA precisará buscar a solução.

Para a definição do tamanho da população, deve-se levar em consideração o tempo para a entrega da solução, pois o mesmo está ligado à amostragem do espaço de busca. Segundo Soares(1997) uma população pequena pode conduzir o algoritmo a um mínimo local e uma população muito grande contém representatividade de amostras, mas levará a um aumento do custo computacional.

A probabilidade de cruzamento é responsável pelas trocas de informações entre os diversos pontos da população. Uma probabilidade de cruzamento baixa pode levar a uma utilização insatisfatória das informações contidas na população, enquanto que valores muito altos podem levar o algoritmo a uma convergência prematura.

A mutação é responsável por inserir novas informações na população. Se a probabilidade da mutação for pequena também será pequena a quantidade de mutações na população, assim diminuindo a diversificação dos cromossomos. Por outro lado um

alto valor para a probabilidade de mutação pode levar o GA a um comportamento de busca aleatória.

2.2.1.2 Avaliar a população

Nesta etapa todos os indivíduos da população (cujo tamanho é usualmente mantido constante ao longo das iterações) serão avaliados em relação à função objetivo do problema. Estes indivíduos receberão uma nota de acordo com a resposta dada pela função objetivo, e é a partir desta nota que os indivíduos serão selecionados para continuar nas próximas etapas do GA.

2.2.1.3 Selecionar a população

Na etapa de seleção os melhores indivíduos, ou seja, os indivíduos que promovem uma resposta da função objetivo mais próximo da solução do problema serão escolhidos para continuarem nas etapas seguintes do GA.

Segundo Soares (1997) “durante a seleção, os indivíduos com desempenho acima da média poderão ter mais cópias enquanto que aqueles com baixo desempenho podem desaparecer completamente”.

Para selecionar os melhores indivíduos existem diversas estratégias, sendo as mais utilizadas a seleção por roleta e a por torneio.

O método da roleta, também conhecido como método proporcional, é um método estocástico que seleciona os indivíduos a partir de seu desempenho, dando a eles uma probabilidade de seleção para cruzamento baseado na resposta da função objetivo quando avaliada com este indivíduo.

Supondo f como sendo o valor da função de avaliação para um dado cromossomo e s como sendo a soma do valor da função de avaliação de todos os cromossomos, cada cromossomo receberá um setor na roleta de tamanho igual ao ângulo dado por $2\pi f / s$, ou seja, um setor com tamanho proporcional ao grau de adaptabilidade do cromossomo (Neto, p. 38).

A partir daí é montada a roleta, que terá a maior parte ocupada por indivíduos com maior aptidão e a menor parte da roleta é dada aos indivíduos com menor aptidão. Após montada a roleta ela é ‘girada’ quantas vezes forem necessárias para a seleção de pares de indivíduos para reprodução.

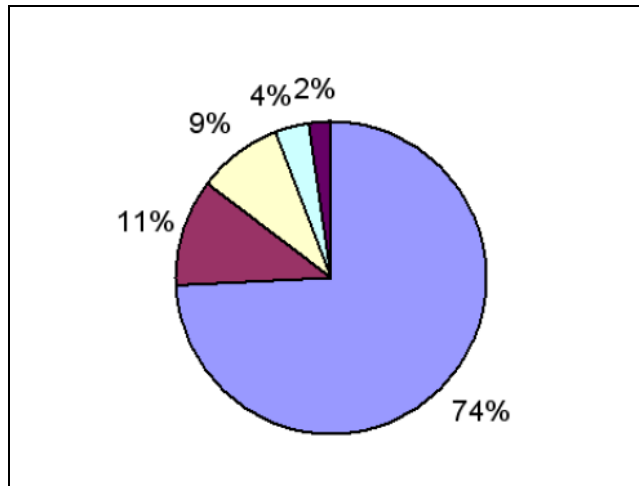


Figura 2.5: Exemplo de uma roleta (Mognon, 2004).

Este método possui a vantagem de atribuir a todos os indivíduos uma probabilidade não-nula de seleção. Entretanto, o mesmo apresenta a desvantagem de ser susceptível a perdas prematuras de diversidade geradas por 'super indivíduos', i.e., pontos que possuam desempenho muito superior aos demais em uma dada geração. Segundo Mognon “uma alta pressão seletiva tende a diminuir a diversidade rapidamente, levando a população a convergir em poucas gerações, o que pode resultar em convergência prematura do GA”.

O método do torneio é baseado em competições dos indivíduos, levando em consideração a aptidão de cada um. Neste método um grupo de N indivíduos é escolhido aleatoriamente e também aleatoriamente são escolhidos dentro deste grupo os indivíduos que competirão. O indivíduo com melhor resposta para a função objetivo será o indivíduo escolhido.

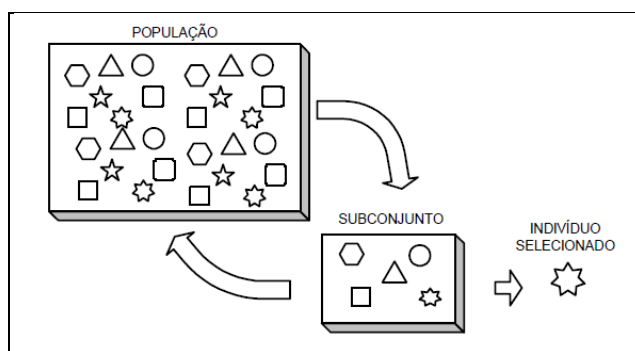


Figura 2.6: Exemplo de um torneio (Mognon, 2004).

2.2.1.4 Modificar a população

Depois de selecionados os indivíduos da população que assumirão o papel de 'pais', os mesmos são utilizados para a geração de novos indivíduos através dos operadores de variação conhecidos como cruzamento e mutação.

Segundo Neto, após a escolha dos pares o operador de cruzamento é responsável pela criação de novos indivíduos. Mognon sugere que “o cruzamento ocorre com alta probabilidade, com valores entre 0.7 e 1.0”. Há varias maneiras de se realizar o cruzamento entre os indivíduos como o ponto de corte, que pode ser com um, dois ou mais pontos e também entre vários indivíduos.

O ponto de corte é escolhido aleatoriamente entre o primeiro e o último gene. As figuras 2.7 e 2.8 ilustram a criação de novos indivíduos a partir do cruzamento com um e dois pontos de corte, respectivamente, enquanto que a figura 2.9 ilustra o procedimento para o cruzamento conhecido como *uniforme*, onde múltiplos pontos são aleatoriamente selecionados.

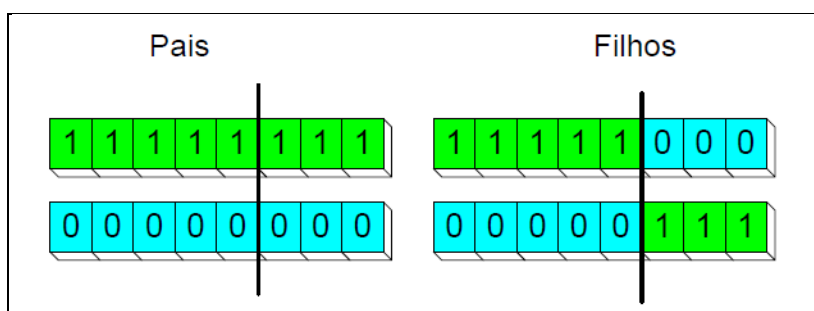


Figura 2.7: Exemplo de cruzamento por ponto duplo (Mognon, 2004).

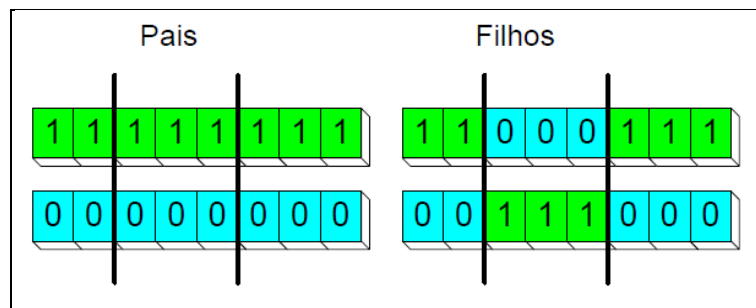


Figura 2.8: Exemplo de cruzamento por ponto duplo (Mognon, 2004).

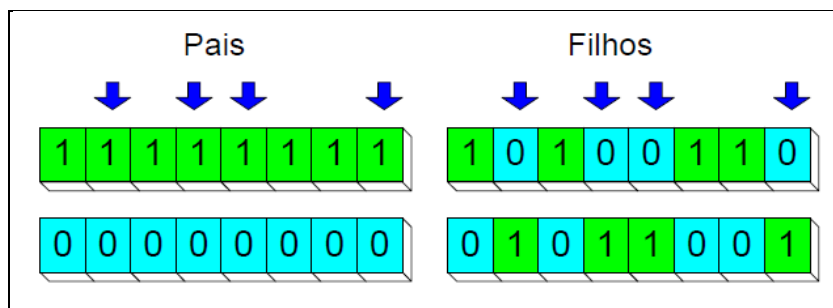


Figura 2.9: Exemplo de cruzamento aleatório (Mognon, 2004).

A partir dessas operações, o operador de cruzamento é capaz de gerar filhos que carreguem informações herdadas dos indivíduos-pais. Uma desvantagem deste operador é não conseguir inserir novas informações na população, ou seja, uma busca baseada apenas em cruzamento estaria limitada a valores existentes na população inicial.

Em sistemas evolutivos naturais, segundo Soares (1997), “para contornar o problema, a natureza insere material genético diferente através do processo conhecido como mutação”. Nos GAs, a mutação é um operador que insere probabilisticamente uma perturbação em alguns pontos dos indivíduos, e.g., uma inversão de bit no caso de codificação binária.

Ao contrário do operador de cruzamento que deve ter sua probabilidade de acontecimento alta, a probabilidade de acontecer uma mutação é usualmente mantida baixa nos GAs, para evitar a destruição de padrões que possam ser interessantes na busca pela solução ótima.

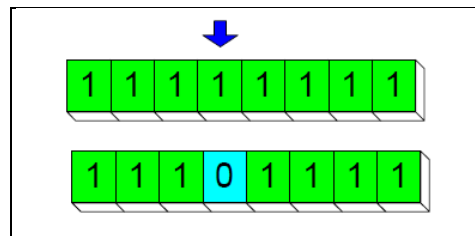


Figura 2.11: Exemplo de mutação (Mognon, 2004).

2.2.1.5 Critérios de convergência

Após selecionar os melhores indivíduos, seguindo as métricas escolhidas, realizar os cruzamentos seguindo a probabilidade P_{cross} e as mutações seguindo P_{mut} deve-se verificar o critério de parada do algoritmo.

A escolha de um critério de parada de forma incorreta pode fazer com que o algoritmo continue sua busca mesmo já tendo encontrado o melhor valor. A figura abaixo mostra um exemplo onde o algoritmo não consegue mais melhorar o valor da função objetivo e mesmo assim ele não interrompe a busca.

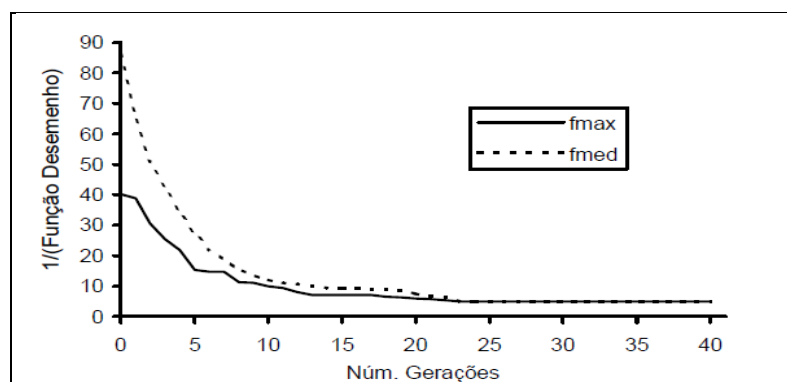


Figura 2.12: Exemplo de um critério de parada escolhido de forma errada (Soares, 1997).

Os critérios de parada mais usados levam em consideração várias métricas do GA como número de gerações, convergência do melhor indivíduo, convergência da população, dentre outros.

O critério de parada por número de gerações é muito simples, sempre que o algoritmo atingir um dado número de gerações o mesmo é interrompido e o melhor indivíduo desta população é retornado como a solução encontrada para o problema.

Já no critério de parada por estabilização do melhor indivíduo, verifica-se a diferença no valor do melhor desempenho entre gerações consecutivas:

$$F_{\max(p)} - F_{\max(p-1)} < \varepsilon$$

Se valores inferiores a ε se repetirem por um determinado número de vezes o algoritmo é interrompido e o valor do melhor indivíduo da população atual é retornado.

Já o critério de parada por convergência da população considera a diversidade da população, interrompendo o algoritmo quando a totalidade da população se torna muito similar em termos de desempenho:

$$1 - \frac{F_{\text{medio}}}{F_{\text{max}}} = F_{\text{conv}}$$

Segundo Soares “pode-se considerar que a população está homogênea quando $F_{\text{conv}} \leq 0.01$ ”.

3 O PROBLEMA DE MISTURA DE CARVÕES

3.1 Mistura de carvões

O carvão betuminoso ou carvão metalúrgico é uma rocha formada através da ação do tempo sobre a matéria orgânica. A ação de microorganismos, temperatura, pressão e o tempo são, então, capazes de transformar a matéria orgânica em um tipo de rocha que pode ser utilizada como combustível. Essa rocha é chamada de carvão betuminoso.

Os tipos de carvões metalúrgicos são normalmente classificados de acordo com a norma ASTM-D388, que divide os carvões em 4 tipos: linhitico, sub-betuminoso, betuminoso e antracítico. Já no Brasil, os tipos de carvões são classificados levando em consideração apenas a matéria volátil contida nos carvões:

- Baixo volátil;
- Médio volátil;
- Alto volátil.



Figura 3.1: Processo Geológico de formação do carvão mineral (trufa, sub-betuminoso, betuminoso e antracito) (Casagrande, p.7).

A partir da classificação de alto volátil, médio volátil e baixo volátil inicia-se o processo de criação da mistura de carvões metalúrgicos que será utilizada no processo da coqueria. Outro parâmetro que é levado em consideração na escolha dos carvões é a capacidade destes em gerar um coque de qualidade, parâmetro este que é requerido pelo alto forno e relacionado à operação da coqueria.

Para o processo de coqueificação é necessário que o carvão possua propriedades coqueificantes, isto é, quando aquecido em ausência de ar ele deve amolecer, inchar, aglomerar e, finalmente, solidificar na forma de um sólido poroso e rico em carbono chamado coque. Os carvões que formam o coque são chamados de carvões coqueificantes. (Osório, p. 10)

Os maiores países produtores de carvão estão concentrados no hemisfério norte, a maior parte dos carvões produzidos no mundo são carvões destinados à produção de energia elétrica, com apenas uma parte pequena representando carvões coqueificáveis. Devido a esta dificuldade de encontrar bons carvões metalúrgicos, o preço deste tipo de carvão é muito alto.

Um caminho para contornar este problema é utilizar uma mistura de carvões metalúrgicos alto, médio e baixo voláteis de forma que se obtenha um coque com a especificação desejada e com o menor custo possível. Segundo Osório, a siderurgia nacional utiliza na mistura de carvões algo em torno de sete a dez carvões diferentes com intuito de diminuir o custo da mistura.

3.2 Introdução de RNAs como restrições no problema

Para se obter uma operação estável de um alto forno, é necessário um controle muito rigoroso sobre a matéria prima utilizada na produção do ferro-gusa. As propriedades físico-químicas e metalúrgicas das matérias primas são muito importantes, pois tem uma influência muito grande na qualidade final do ferro-gusa e na estabilidade operacional do alto-forno.

O coque é um insumo de extrema importância no processo do alto forno, pois ele é principal combustível utilizado. Juntamente ao coque podem-se utilizar outros combustíveis como gás natural, carvão pulverizado dentre outros.

Sendo assim o custo do coque também é preponderante em relação ao custo do ferro-gusa, pois o custo de compra dos carvões metalúrgicos e o custo de transformação dos carvões metalúrgicos em coque é muito alto.

Conforme Casagrande apud Coelho (2003):

As principais matérias-primas para a produção do ferro-gusa em fornos de grande porte são: carga metálica (sinter, pelotas e minérios granulados), combustíveis (coque metalúrgico e finos de coque para PCI) e fundentes. A participação do coque no custo total de gusa é de aproximadamente 40,5%, sendo considerada a matéria-prima mais cara da carga do alto-forno.

Uma propriedade do coque que é muito importante no controle metalúrgico do alto forno é o CSR. Segundo Casagrande a reserva térmica do alto forno vem do coque, que fornece energia térmica através de sua combustão. Além disso, o coque deve apresentar também uma resistência mecânica capaz de suportar o peso da carga e permitir a passagem dos gases.

Desta forma, faz-se extremamente necessário prover ferramentas que possam prever o valor do CSR do coque. O parâmetro CSR é muito importante, mas pela dificuldade no processo de apuração do seu valor, ele é, de certa forma, um valor muito escasso para a avaliação dos metalurgistas, uma vez que no processo laboratorial em que se

obtem o valor é muito demorado. Normalmente a quantidade de leituras do CSR é da ordem de um resultado por dia.

Para se obter este valor, o coque é coletado após sair da bateria e é levado ao laboratório. No laboratório o procedimento que apura do valor do CSR leva aproximadamente 20 horas. Normalmente quando este valor chega até o metalurgista é o valor do último coque desenfornado, ou seja, é o valor da mistura de carvões enfornada cerca de 20 horas antes. Sendo assim, toda mistura de carvões utilizada na coqueria neste período tende a ter o mesmo valor para o CSR ou pelo menos um valor muito próximo do valor apurado.

Se neste momento o metalurgista recebe um valor de CSR que está fora do acordado com o controle metalúrgico do Alto Forno, não há muito que fazer com todos os fornos das baterias de coque que já estão com a mistura de carvões em processo de coqueificação, apenas mudanças na forma de operar a bateria como aumentar ou diminuir o tempo em que a mistura de carvões fica no interior do forno, aumentar a temperatura média da bateria ou de um forno específico dentre outras.

Devido a estes problemas tem-se a necessidade de se ter uma forma de predição confiável do CSR no momento da criação da mistura de carvão. Para a solução desse problema foi criada uma RNA que tem a capacidade de prever um CSR médio e que será utilizada em conjunto com o otimizador. A RNA em questão utiliza entradas médias de 14 dias. Foram utilizadas leituras do laboratório compreendidas em um período de aproximadamente 10 anos de operação, de 2003 a 2013.

Ao definir quais parâmetros fariam parte da entrada da RNA foi inicialmente realizado um estudo sobre a correlação entre o CSR medido e alguns parâmetros da mistura. Os resultados encontrados são detalhados na tabela 3.1.

Tabela 3.1: Correlação entre parâmetros de qualidade da mistura de carvões e o coque

Propriedade 1	27,10%
Propriedade 2	-7,61%
Propriedade 3	6,18%
Propriedade 4	-13,13%
Propriedade 5	34,97%
Propriedade 6	19,99%
Propriedade 7	4,73%
Propriedade 8	-13,31%
Propriedade 9	-8,14%
Propriedade 10	16,90%
Propriedade 11	18,93%
Propriedade 12	-9,65%
Propriedade 13	-43,45%
Propriedade 14	15,80%
Propriedade 15	-10,99%

As correlações acima foram escolhidas para comporem o critério de decisão no momento de seleção das entradas do modelo. Para escolher estas entradas foram estabelecidos 2 critérios:

1. A correlação entre os valores do CSR e da variável: as variáveis com o valor da correlação abaixo de 10% com a saída não seriam utilizadas.
2. Conhecimento dos metalurgistas: foram utilizadas variáveis que, de acordo com a experiência dos metalurgistas, impactariam diretamente no resultado do CSR, mesmo que apresentassem baixos valores de correlação. Dentro deste critério se encaixaram as seguintes variáveis: Tempo Bruto de Coqueificação, Vitrinita, Logaritmo de Máxima Fluidez, Matéria Volátil e Taxa de Retorno.

Algumas variáveis utilizadas como entradas do modelo são dadas em função de outros parâmetros da mistura de carvões. As variáveis que são compostas por valores de outras variáveis são: basicidade, reativos, granulometria, álcalis dentre outras. Alguns exemplos são descritos a seguir.

$$\text{Basicidade} = \frac{\%Fe_2O_3 + \%CaO + \%Na_2O + \%K_2O + \%MgO}{\%SiO_2 + \%Al_2O_3}$$

$$\text{Reativos} = \frac{(\%Vitrinita + \%Cutinita + \%Esporinita + \%Resinita + (\%Semifusinita/3))}{5}$$

Um ponto importante no momento de levantar o modelo por RNA é que esta seja uma rede generalista. Para aumentar a generalidade do modelo optou-se por utilizar o método de validação cruzada.

No método de validação cruzada o conjunto de dados de treinamento é dividido em K subgrupos. Destes K subgrupos é escolhido 1 que será utilizado como conjunto de dados de validação e os demais K-1 subgrupos são utilizados no treinamento. Desta forma o processo de validação cruzada é repetido K vezes de forma que todos os K subgrupos de dados sejam utilizados como validação.

A RNA escolhida para prever o CSR possui 15 neurônios na camada intermediária e um neurônio na camada de saída. Para os neurônios da camada intermediária foi utilizada como função de ativação a função sigmoideal.

$$F(x) = \frac{1}{1 + e^{-ax}}$$

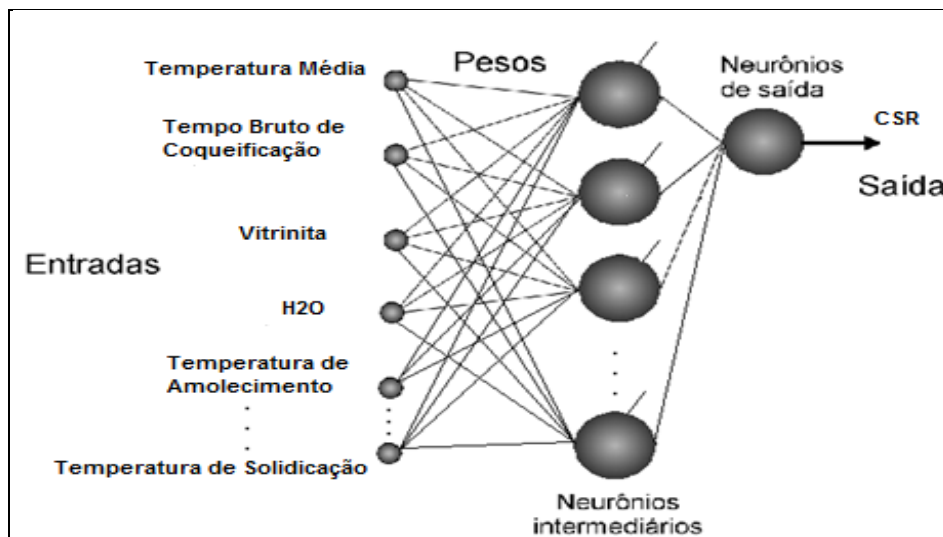


Figura 3.2: RNA para predição do CSR.

3.3 O otimizador de carvão

3.3.1 O problema da mistura de carvões e a RNA do CSR

De posse da RNA de previsão do CSR, o desafio deste trabalho então passa a ser resolver o problema da mistura de carvões, porém com uma restrição especial que é a restrição do valor do CSR.

Normalmente o problema da mistura de carvões é tratado como o problema de programação matemática linear, onde todas as restrições impostas na a solução do problema são restrições lineares. Sendo assim, para solucionar a previsão do CSR era utilizado um modelo linear que não atingia as expectativas dos metalurgistas da coqueria.

No problema abordado tem-se a adição de uma nova restrição, ou uma restrição antiga que era tratada como uma restrição linear e que agora será tratada como uma restrição não-linear, a restrição do valor do CSR.

O algoritmo escolhido para solução do problema de mistura de carvões foi o Algoritmo Genético.

Dentre todas as restrições colocadas pelos metalurgistas, a única restrição que não era linear é a restrição do valor do CSR. Para a maioria das restrições lineares a formulação da restrição foi a seguinte:

$$P_{jMist} = \sum_{i=1}^N P_{jC_i} * \%C_i$$

Onde:

P_{jMist} = valor da propriedade j da mistura.

P_{jC_i} = valor da propriedade j da do carvão i .

$\%C_i$ = porcentagem do carvão i na mistura.

N = Número de carvões.

Já nas restrições do valor de DI, CRI, Cz e S do coque foram utilizadas equações lineares levantadas pelos metalurgistas da coqueria.

Outra restrição utilizada foi a que trata da quantidade máxima da soma de todos os carvões, ou seja, se somar a porcentagem de todos os carvões este valor não pode ultrapassar o valor de 1 ou 100%. Esta restrição foi modelada da seguinte maneira:

$$1 = \sum_{i=1}^N \%C_i$$

Para a função objetivo do problema foi considerado o valor do custo da mistura de carvões. Que é dado a partir do somatório da participação do custo de todos os carvões. A equação do custo é a seguinte:

$$\text{Custo da Mistura} = \sum_{i=1}^N \text{Custo}_{C_i} * \%C_i$$

E finalizando, a restrição especial que é a restrição do valor do CSR.

3.3.2 A montagem do problema

Problemas de otimização devem ser montados de uma forma características para que sejam resolvidos. No presente trabalho foi utilizado o Matlab® na solução do problema e sendo assim o problema deve ser montado de uma forma específica, conforme exemplo abaixo:

Minimize $F(x)$ sujeito à:

$$A_{eq} * x = b_{eq} \text{ (equações lineares)}$$

$$A * x \leq b \text{ (inequações lineares)}$$

$$C_{eq}(x) = 0 \text{ (equações não-lineares)}$$

$$C(x) \leq 0 \text{ (inequações não-lineares)}$$

As equações devem estar num formato de matriz onde cada linha representa uma restrição a ser respeitada pela solução. A equação de cada linha é representada da seguinte forma:

$$\sum_{i=1}^N \%P_{jC_i} - 1 = 0$$

$$\left\{ \begin{array}{ccc} x * P_{1c_1} & y * P_{1c_2} & z * P_{1c_m} \\ x * P_{2c_1} & y * P_{2c_2} & z * P_{2c_m} \\ x * P_{3c_1} & y * P_{3c_2} & \ddots \quad z * P_{3c_m} \\ \vdots & \vdots & \vdots \\ x * P_{nc_1} & y * P_{nc_2} & z * P_{nc_m} \end{array} \right\} + \left\{ \begin{array}{ccc} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \ddots \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & -1 \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{array} \right\}$$

Figura 3.3: Exemplo da montagem de matrizes para solução do problema.

Sendo assim, para utilizarmos os dados na solução do problema deve-se montar as matrizes da seguinte forma: na matriz A teremos nas linhas o parâmetro – vitrinita, Al₂O₄, CaO, etc. – de cada carvão utilizado na mistura e nas colunas têm-se os carvões.

$$\left\{ \begin{array}{cccc|cccccc} 268 & 256 & 192 & 240 & 11 & -1 & 0 & 0 & 0 & 0 \\ 412 & 446 & 453 & 442 & 0 & 0 & -1 & 0 & 0 & 0 \\ 500 & 510 & 512 & 522 & \ddots & 444 & 0 & 0 & -1 & 0 \\ 4.249 & 1.672 & 0.954 & 1.663 & \ddots & 0 & 0 & 0 & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \end{array} \right\}$$

Figura 3.4: Exemplo da montagem da matriz Aeq.

Esta matriz será chamada de Aeq.

Continuando a montagem do problema deve-se montar a matriz beq. Esta matriz representa simplesmente o outro lado da equação, ou seja, será uma matriz com uma coluna de zeros contendo o mesmo numero de linhas que a quantidade de restrições.

Montadas as matrizes que representam as equações, deve-se montar as matrizes que vão definir os limites de atuação do algoritmo que solucionará o problema. Estas matrizes contêm os valores máximos e mínimos de cada restrição, valores estes que podem ser de $+\infty$ até $-\infty$. Estas matrizes serão chamadas de LB (Lower Bound) e UB (Upper Bound) e serão matrizes de mesmo tamanho da matriz beq, ou seja, será uma matriz de apenas uma coluna com os valores e contendo o mesmo numero de linhas que a quantidade de restrições.

Desta forma, o algoritmo de solução do problema será capaz de encontrar os valores dos parâmetros para cada restrição e verificar se este valor está compreendido no espaço restringido pelos valores de LB e UB da restrição em questão.

$$LB = \begin{Bmatrix} -\infty \\ 0 \\ 10 \\ 0 \\ \vdots \\ 1200 \end{Bmatrix} \quad UB = \begin{Bmatrix} 0 \\ +\infty \\ 1000 \\ -10 \\ \vdots \\ 0 \end{Bmatrix}$$

Figura 3.6: Exemplo da montagem das matrizes LB e UB .

3.4 Resultados

Nesta parte do trabalho serão inicialmente mostrados os resultados referentes à RNA de previsão do CSR, seguidos pelos resultados do otimizador de misturas.

3.4.1 Resultados da RNA

Para o treinamento da RNA foram utilizados os seguintes comandos no Matlab®.

```
%REDE NEURAL
%Cria a Rede
net1 = newff(Matriz_Completa_de_entrada',Matriz_Completa_de_saida', FimEntrada,
{'tansig' 'purelin'});

net1.trainParam.epochs=200;
net1.trainParam.max_fail=100;
net1.performFcn = 'mse';

%Treina a rede
net1=train(net1,U1',Y1');

%Simulação do treinamento
yTrain = sim(net1,U1');
yTrain = yTrain';

%Simulação da validação
yVal = sim(net1, UV);
yVal = yVal';

%Simulação do teste
yTest = sim(net1,UT);
yTest = yTest';
```

Os metalurgistas da coqueria impuseram algumas métricas para validar a RNA durante os testes no processo. Essas métricas estão descritas na tabela abaixo:

Tabela 3.2: Métricas para validação do modelo

RMSE de treinamento	2,66
Erro Máximo de treinamento	5,00 ou -5,00

Vários modelos foram encontrados, porem o modelo que foi aprovado apresentou o seguinte comportamento:

Tabela 3.3: Métricas para validação do modelo

RMSE de treinamento	1,85
Erro Máximo de treinamento	3,86 e -3,23

O comportamento deste modelo é ilustrado nos gráficos a seguir:

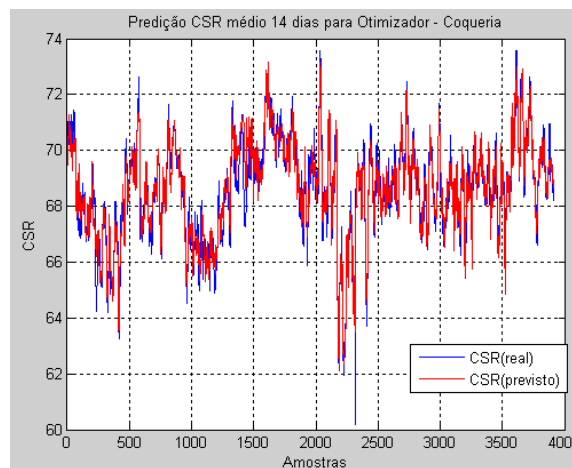


Figura 3.7: Comparação entre CSR real e CSR previsto pela rede neural para massa de dados de treinamento e validação.

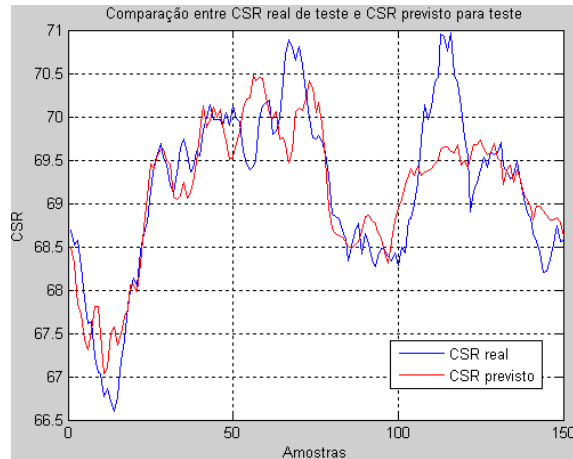


Figura 3.8: Comparação entre CSR real e CSR previsto pela rede neural para massa de dados de teste.

3.4.2 Resultados do otimizador

Após a criação do modelo para previsão do CSR, é necessário criar um otimizador que solucione o problema. Para isso foi escolhido o GA.

Primeiramente decidiu-se criar três otimizadores, um linear e dois não lineares, desta forma comparando o resultado de cada algoritmo. Para facilitar a solução foi criado um arquivo executável binário a partir de um arquivo .m do Matlab® através do Matlab Compiler. Foi criada então uma aplicação em C# como interface entre o metalurgista e o GA, na qual o metalurgista escolhe os carvões que farão parte da otimização e ainda especifica os parâmetros operacionais da coqueria. Esta aplicação cria arquivos .txt contendo as informações necessárias para que o GA resolva o problema.

Na criação do arquivo .m foram utilizados os seguintes comandos:

- linprog;
- fmincon;
- GA;

O linprog é um algoritmo que resolve apenas problemas lineares, este algoritmo foi utilizado na comparação entre os resultados com os modelos lineares e os modelos não-lineares. Já os algoritmos do fmincon e GA são algoritmos que tratam restrições não-lineares, porém de forma diferente.

Como descrito acima foi criada uma aplicação em C# que faz a interface entre o usuário e o algoritmo de solução do problema e é nesta aplicação que são montadas as matrizes Aeq, beq, LB, UB etc. e esta parte será explicada mais a frente neste capítulo.

O arquivo .m do Matlab® recebeu o nome de ResolveProblema e a partir daqui será referenciado por RP. Para iniciar o RP foram introduzidos os seguintes comandos:

```

clear all;
close all;

global net_com_restricoes Temperatura TempoBruto H2O TaxaRetorno Granulometria

global NoCarv NoProp Otimiz_com_restricoes W Phi Beta Gama W_sem_restricoes
Phi_sem_restricoes Beta_sem_restricoes Gama_sem_restricoes

global mInputXMin mInputXMax mInputYMin mInputYMax mOutputXMin mOutputXMax
mOutputYMin mOutputYMax

global mInputXMin_sem_restricoes mInputXMax_sem_restricoes mInputYMin_sem_restricoes
mInputYMax_sem_restricoes mOutputXMin_sem_restricoes mOutputXMax_sem_restricoes
mOutputYMin_sem_restricoes mOutputYMax_sem_restricoes

clc;

```

A partir disso necessita-se carregar as matrizes do problema, e para isto foram introduzidos no RP os seguintes comandos, que permitem ler as informações do problema:

```

%Lê as informações

fobj = load('Fobject.txt');

Aeq = load('Aeq.txt');
beq = load('beq.txt');

ub = load('UB.txt');
lb = load('LB.txt');

CSR_Aeq = load('CSR_Aeq.txt');
S_Aeq = load('S_Aeq.txt');
DI_Aeq = load('DI_Aeq.txt');
CRI_Aeq = load('CRI_Aeq.txt');
Cz_Aeq = load('Cz_Aeq.txt');
Basicidade_Aeq = load('Basicidade_Aeq.txt');

param = load('param.txt');

```

Para solucionar o problema a partir do linprog foram utilizados os seguintes comandos:

```

%Resolve o problema linear

Aeq_lin = [Aeq, zeros(NoProp,6)];

Aeq_lin = [Aeq_lin; [ones(1,NoCarv) CSR_Aeq -1 0 0 0 0 0]; [zeros(1,NoCarv) S_Aeq 0
-1 0 0 0 0]; [zeros(1,NoCarv) CRI_Aeq 0 0 -1 0 0 0]; [zeros(1,NoCarv) Cz_Aeq 0 0 0 -1 0
0]; [zeros(1,NoCarv) DI_Aeq 0 0 0 0 -1 0]; [zeros(1,NoCarv) Basicidade_Aeq 0 0 0 0 0 -
1]];

beq_lin = [beq;-CSR_beq;-S_beq;-CRI_beq;-Cz_beq;-DI_beq; -Basicidade_beq ];

```

Após a montagem das matrizes, é o momento de parametrizar o algoritmo e usá-lo na solução do problema. Nesta etapa foram utilizados vários parâmetros diferentes, e como métrica mais importante neste trabalho é a agilidade na solução do problema, então foi acordado com os metalurgistas que um tempo razoável seria de no máximo 15 minutos. Sendo assim para o linprog foram utilizados os seguintes parâmetros:

```

optLinprog = optimset('linprog');
optLinprog.TolFun = 1e-5;

fobj = [fobj;zeros(6,1)];

```

```
[x_sol_lin, FVAL, EXITFLAG]=linprog(fobj,A,b,Aeq_lin,beq_lin,lb,ub,[],optLinprog)

x_sol_lin_bit = [EXITFLAG; x_sol_lin];
save x_sol_lin.txt x_sol_lin_bit -ASCII;
```

Para utilizar o GA ou o Fmincon foram carregadas e montadas as matrizes com os dados do problema, para que o mesmo fosse resolvido com a adição da restrição especial do CSR. Para carregar os dados das matrizes foram utilizados os seguintes comandos:

```
Aeq_ga = [Aeq, zeros(NoProp,6)];

beq_ga = [beq];

%Cria a função objetivo
f = @(x) fobj'*x';
```

A principal métrica para os algoritmos continua sendo o tempo de solução do problema em questão. Na busca do atendimento desta meta o Fmincon foi parametrizado da seguinte forma:

```
% solução utilizando o Fmincon.

opt = optimset();
opt.PlotFcns = @optimplotfval;
opt.Algorithm = 'sqp';
opt.TolCon = 1e-5;
opt.TolFun = 1e-5;
opt.TolX = 1e-5;
opt.MaxFunEvals = 9000;

opt.TolFun = 1e-5;
```

Com as matrizes montadas os comandos para o Fmincon foram os seguintes:

```
[x_sol_fmincon,FVAL_FMINCON,EXITFLAG_fmincon]=
fmincon(f,x_sol_lin',A_ga,b,Aeq_ga,beq_ga,lb,ub,@nonlcon3,opt)

x_sol_fmincon_bit = [EXITFLAG_fmincon; x_sol_fmincon'];

%Salva o resultado em um arquivo
save x_sol_fmincon.txt x_sol_fmincon_bit -ASCII;
```

Com as matrizes do problema não-linear carregadas é necessário apenas parametrizar o GA para utilizá-lo na solução do problema. E para isto foram utilizados os seguintes comandos:

```
options = gaoptimset();

%Cria a população inicial baseado na solução do problema pelo Fmincon
options.InitialPopulation = x_sol_fmincon;

options.PopulationSize = 100;
options.Generations = 300;%5000;
options.StallGenLimit = 50;%2500;
options.StallTimeLimit = 1000;
options.TolCon = 1e-4;
options.TolFun = 1e-4;
options.SelectionFcn = @selectiontournament;
options.CrossoverFcn = @crossoverarithmetic;
options.HybridFcn = @fmincon;
options.EliteCount = 50;
%options.PlotFcns = @gaplotmaxconstr;
options.PlotFcns = @gaplotbestf;
```

```

%Resolve o problema

[x_sol_ga,FVAL_GA,EXITFLAG_GA]=
ga(f,size(Aeq_ga,2),A_ga,b,Aeq_ga,beq_ga,lb,ub,@nonlcon3,options)

x_sol_ga_bit = [EXITFLAG_GA; x_sol_ga'];

%Salva o resultado em um arquivo
save x_sol_ga.txt x_sol_ga_bit -ASCII;

```

Tanto na chamada do GA quanto na chamada do Fmincon há um termo especial chamado de @nonlcon3 e este termo é a chamada para a restrição especial do CSR. O nonlcon3 nada mais é do que uma função que representa as restrições não lineares. Para mostrar a implementação da RNA de predição do CSR, foi implementada toda a RNA no nonlcon3, conforme abaixo:

```

function [c, ceq] = nonlcon3(var)

A = 1;
B = 0;

for j=1:size(W_sem_restricoes,1) %neurônios
    X1_neuronio(j) = 0;
    X_neuronio(j) = 0;
    Y1_neuronio(j) = 0;
end

Y2_neuronio = 0;
Y_neuronio = 0;
CSR_saida = 0;

for j=1:size(W_sem_restricoes,1) %neurônios
for i=1:size(W_sem_restricoes,2) %entradas

    X1_neuronio(j) = X1_neuronio(j) + var_net(i)*W_sem_restricoes(j,i);
end
end

for j=1:size(W_sem_restricoes,1)
    X_neuronio(j) = X1_neuronio(j) + Phi_sem_restricoes(j);
    Y1_neuronio(j) = 2/(1+exp(-2* X_neuronio(j)))-1;
end

for j=1:size(W_sem_restricoes,1)
    Y2_neuronio = Y2_neuronio + Y1_neuronio(j)*Beta_sem_restricoes(j);
end
Y_neuronio = Y2_neuronio + Gama_sem_restricoes;

CSR_saida = (Y_neuronio - mOutputYMin_sem_restricoes)*(mOutputXMax_sem_restricoes
-
mOutputXMin_sem_restricoes)/(mOutputYMax_sem_restricoes-
mOutputYMin_sem_restricoes)+mOutputXMin_sem_restricoes;

```

Após a RNA calcular o valor do CSR é necessário montar as equações do Ceq.

```

ceq_CSR = var(NoCarv+NoProp+1) - (A*CSR_saida+B);

```

Para verificar a eficiência do trabalho proposto foram realizados vários testes. Em uma primeira etapa foram testadas várias parametrizações do Linprog, do Fmincon e do GA na mesma mistura de carvões e desta forma verificando eficiência dos algoritmos. Inicialmente verificou-se que o GA nunca atingiu a métrica de tempo para solucionar o problema, já os demais sempre alcançaram este objetivo. O GA apresentou respostas

próximas dos valores alcançados pelo Fmincon porém sempre com um tempo acima de 30 minutos na solucionar o problema, desta forma inviabilizando o uso deste algoritmo desta forma.

Para continuar com a opção de utilizar o GA decidiu-se utilizar um comando no matlab® que cria a população inicial com base em um indivíduo específico. Este indivíduo deverá orientar o algoritmo a encontrar um valor de resposta para o problema dentro de um tempo razoável.

Primeiramente testou-se a solução dada pelo Linprog como um indivíduo da população inicial do GA, conforme trecho de código abaixo:

```
options = gaoptimset();  
options.InitialPopulation = x_sol_lin';
```

Verificou-se então que o GA continuou sendo muito lento. O GA conseguiu alcançar valores próximos do encontrado pelo Fmincon porém para que ele chegasse ao valor encontrado pelo Fmincon ou ultrapassasse este valor o GA continua com valor de tempo para encontrar a solução do problema muito acima da métrica colocada.

A última tentativa então foi inserir como indivíduo base na população inicial do GA a resposta dada pelo Fmincon conforme abaixo:

```
options = gaoptimset();  
options.InitialPopulation = x_sol_fmincon;
```

A partir deste ponto temos o GA melhorando o valor encontrado pelo Fmincon e na maioria das vezes alcançando o resultado para o problema com o tempo dentro da métrica prevista mas ainda com um tempo alto.

Para iniciar os testes do GA com a restrição especial do CSR e comparamos os resultados obtidos entre o GA e o Fmincon. No primeiro teste a população do GA foi parametrizada com 300 indivíduos, conforme abaixo:

```
options.PopulationSize = 300;
```

Na tabela abaixo se pode verificar o tempo de execução em segundos dos algoritmos:

Tabela 3.3: Tempo decorrido para solução do problema – população do GA = 300

tElapsedGa	tElapsedFmc	tElapsedLin
1547.2	0.68123	0.1582
728.36	0.90837	0.16429
1152.7	0.68705	0.15683

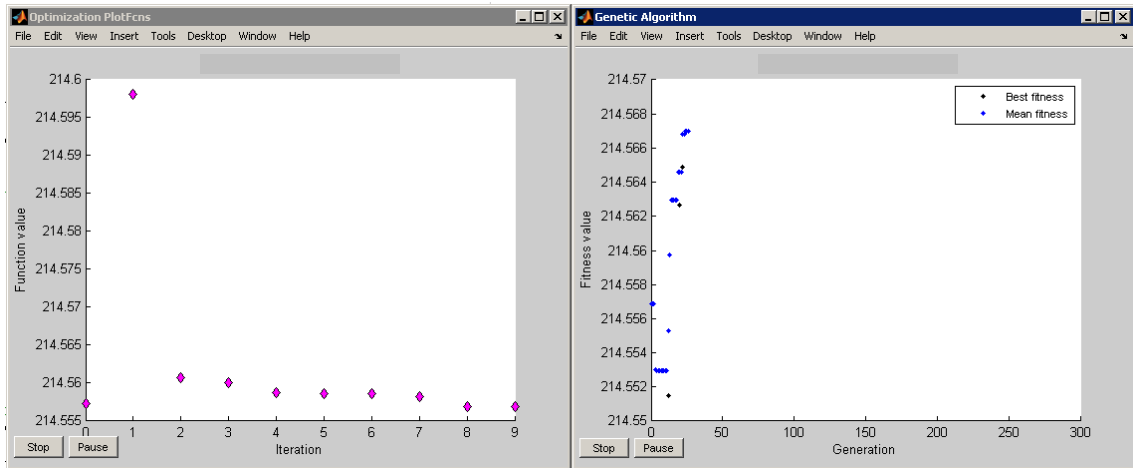


Figura 3.9: Comportamento do GA e do fmincon – população GA = 300.

A partir destes resultados decidiu-se não continuar com os testes e reduzir o valor da população, conforme abaixo:

```
options.PopulationSize = 200;
```

Tabela 3.4: Tempo decorrido para solução do problema – população do GA = 200

tElapsedGa	tElapsedFmc	tElapsedLin
233.51	0.68837	0.16196
355.54	0.72031	0.15891
237.32	0.68506	0.15803
377.61	0.72707	0.15752
396.32	0.71578	0.15821
232.43	0.15733	0.7982
406.52	0.15641	0.72813
267.09	0.1575	0.6956

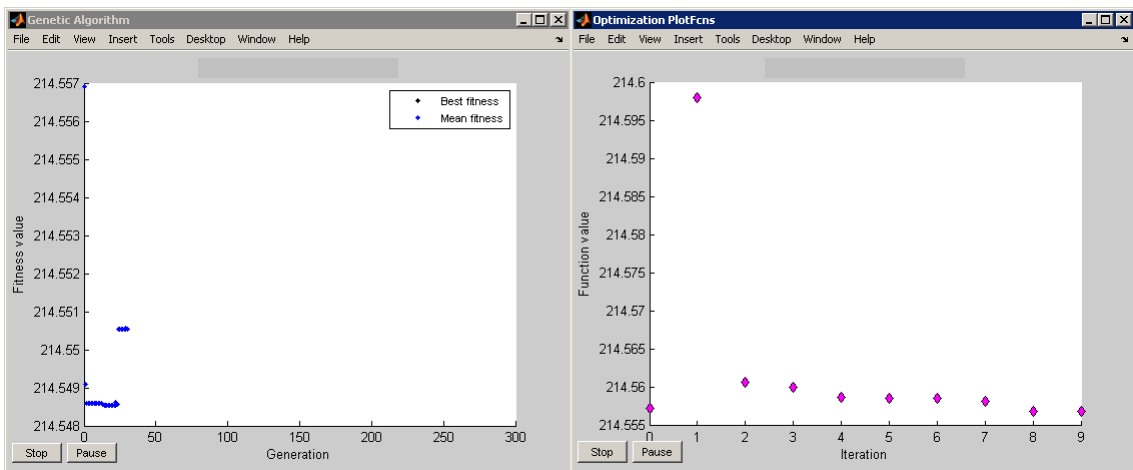


Figura 3.10: Comportamento do GA e do Fmincon – população GA = 200.

Nesta etapa verificou-se uma melhora significativa no tempo necessário para que o GA encontrasse o resultado do problema, então o valor da população do GA foi reduzida mais uma vez, conforme abaixo:

```
options.PopulationSize = 150;
```

Tabela 3.5: Tempo decorrido para solução do problema – população do GA = 150

tElapsedGa	tElapsedLin	tElapsedFmc
87.143	0.15783	0.70426
34.732	0.15885	0.66998
82.349	0.15653	0.67991
38.808	0.15655	0.73262
90.289	0.15682	0.66892

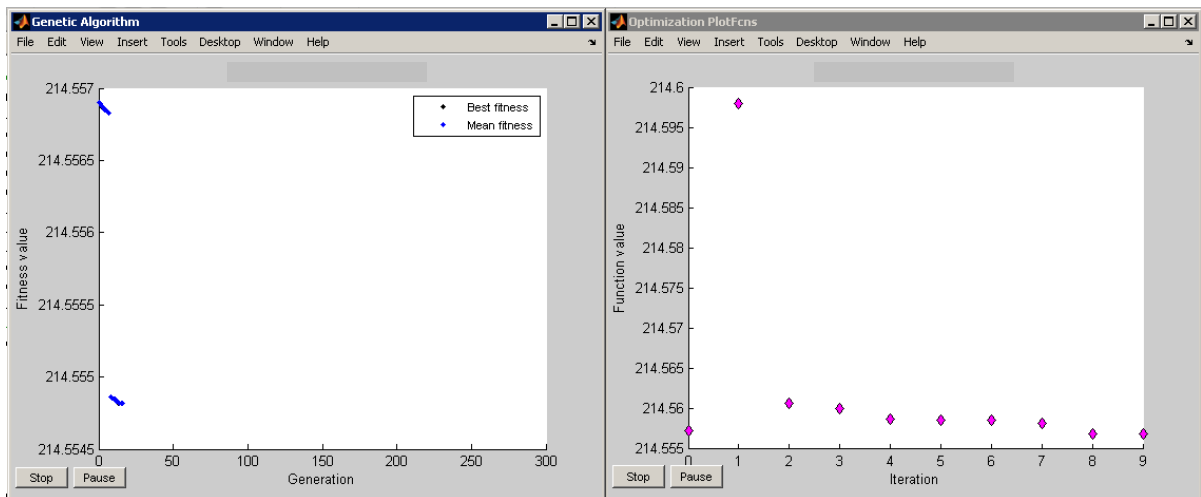


Figura 3.11: Comportamento do GA e do Fmincon – população GA = 150.

Mais uma vez verificou-se a melhora no tempo de execução do GA e decidiu-se mais uma vez reduzir o tamanho da população do GA, conforme abaixo:

```
options.PopulationSize = 100;
```

Tabela 3.6: Tempo decorrido para solução do problema – população do GA = 100

tElapsedGa	tElapsedLin	tElapsedFmc
61.489	0.15623	0.79004
16.684	0.15657	0.72469
23.082	0.15642	0.78248
70.958	0.15649	0.79222
19.234	0.15645	1.3919

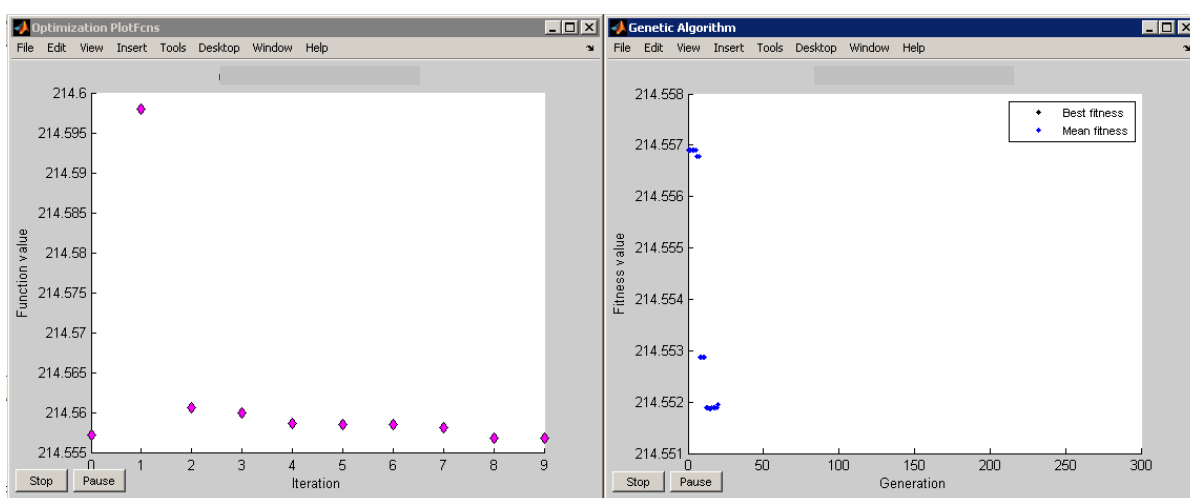


Figura 3.12: Comportamento do GA e do Fmincon – população GA = 100.

De acordo com os resultados obtidos decidiu-se mais uma vez reduzir o valor da população do GA conforme abaixo:

```
options.PopulationSize = 80;
```

Tabela 3.7: Tempo decorrido para solução do problema – população do GA = 80

tElapsedGa	tElapsedLin	tElapsedFmc
15.285	0.16185	0.70396
64.293	0.15724	0.67719
11.164	0.1566	0.68184
10.665	0.15678	0.68579
56.031	0.1578	0.67175

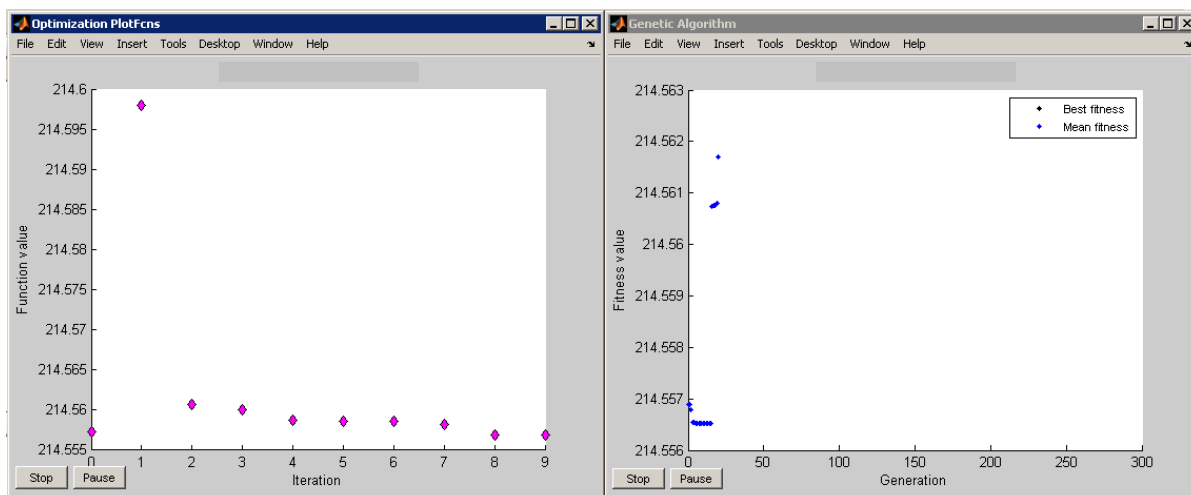


Figura 3.13: Comportamento do GA e do Fmincon – população GA = 80.

As informações acima mostram uma melhora significativa, no tempo gasto pelo GA para entregar uma resposta ao problema proposto, a diminuição da população até o valor de 100 indivíduos. Sendo assim, pode-se verificar na figura 3.14 o comportamento do Algoritmo proposto em relação ao tamanho da população de soluções-candidatas e a melhora no tempo gasto na solução o problema.

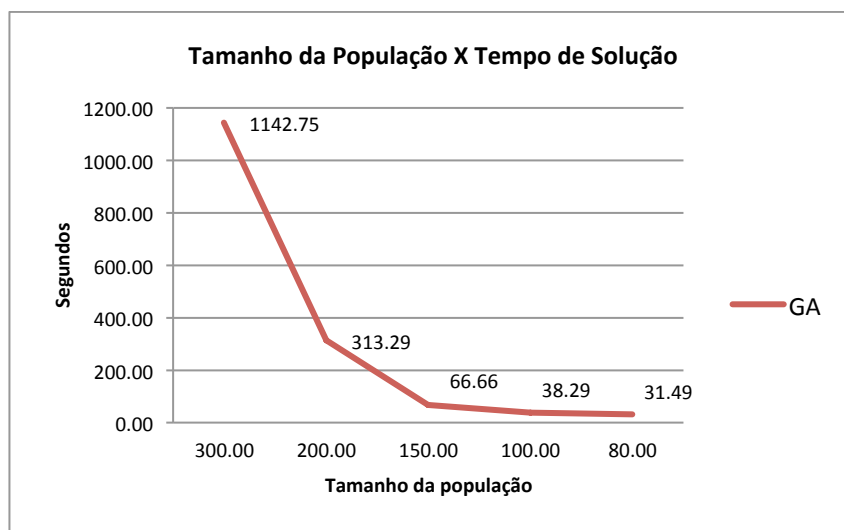


Figura 3.14: Tamanho da população e o tempo para solução.

Os testes mostraram que os valores de função objetivo encontrados entre a otimização realizada pelo Linprog, ou seja, utilizando o modelo linear e o GA foram sempre muito próximos. Pode-se questionar o porquê de não utilizar apenas o modelo linear e desprezar o modelo não-linear, mas ao verificar a diferença entre os valores de RMSE dos modelos apresentados neste trabalho pode-se fazer a seguinte análise baseada no valor de RMSE dos modelos linear e não-linear. De forma a obter um intervalo de confiança de 90% para o acerto de pelo menos do valor mínimo de CSR acordado entre a coqueria e o alto forno, o valor da restrição do CSR que deverá ser

trabalhado na otimização utilizando o modelo linear será sempre maior do que o valor mínimo para o CSR utilizando o modelo não linear, desta forma tem-se uma possibilidade de acerto maior, e com um menor custo.

4 CONCLUSÃO

O trabalho proposto constituiu no desenvolvimento de uma solução para o problema de mistura de carvões com o uso de uma restrição não linear para a predição do CSR. A RNA implementada reduziu o RMSE da predição do CSR. O modelo linear tinha um RMSE de 2,66 e a RNA tem um RMSE de 1,85, desta forma aumentando a confiança na predição do CSR diário, já que quando é especificada uma mistura de carvões que será utilizada na coqueria o RMSE de predição do CSR do coque – produto da coqueria – já tem um controle muito eficiente mesmo antes da mistura ser utilizada na coqueria.

O otimizador utiliza basicamente dois algoritmos, o Fmincon é o algoritmo utilizado para encontrar uma resposta muito próxima do melhor valor. A partir desta resposta o GA busca o melhor valor para os carvões utilizados na mistura. Esta solução foi adotada pela situação do GA gastar muito tempo na entrega de uma solução para o problema, por isso decidiu-se utilizar o resultado obtido pelo Fmincon como um valor próximo do valor da solução ideal do problema e depois utilizar o GA para encontrar o melhor valor.

Com relação ao valor do custo da mistura verificou-se uma variação na maioria das vezes entre o Fmincon e o GA na terceira casa decimal, mas quando verifica-se a quantidade de milhares de toneladas de carvões comprados e o valor dos carvões, mesmo uma variação na terceira casa decimal torna o uso do GA além do Fmincon necessários.

Outra possibilidade de uso para o otimizador é durante a preparação na aquisição de novos carvões para utilização durante o ano todo ou durante um período um pouco menor. Nesta situação o metalurgista pode deixar os valor de máximo e mínimo de utilização de cada carvão e deixar que o GA, mostre a melhor configuração de um pacote de compra de carvão. Neste cenário o metalurgista pode escolher vários carvões alto-voláteis, médio-voláteis e baixo-voláteis e deixar que a ferramenta escolha a melhor divisão de cada carvão para ser adquiridos já com a visão do melhor valor do CSR.

A situação que mostra a real necessidade do uso do otimizador é quando se compara o valor do custo da mistura encontrado pelos métodos tradicionais da área com o valor encontrado pelo otimizador. Neste momento verifica-se um valor discrepante entre o valor da função objetivo dado pelo otimizador e o valor do custo da mistura conforme tabela abaixo:

Tabela 4.1: Variação entre o valor da mistura encontrada pelos metalurgistas e encontrada pelo GA.

M1	6,65%
M2	2,41%
M3	6,58%
M4	3,42%
M5	0,07%
M6	0,38%
M7	2,06%
M8	0,57%
M9	-0,99%

Como pode ser visto na tabela acima, o GA sempre é superior ao valor encontrado pelos metalurgistas, apenas na mistura M9 pode ser verificado que o GA teve desempenho abaixo do metalurgista, mas neste processo é comum que o metalurgista assuma que uma mistura seja válida mesmo que o valor do modelo linear de predição do CSR apresente um valor fora do valor acordado entre o alto forno e a coqueria. Desta forma a ferramenta mostra uma eficiência muito grande em relação ao processo praticado hoje.

Este resultado mostra a eficiência do trabalho proposto e como melhorias futuras pode-se criar novas RNAs para predição dos parâmetros de CRI, DI, Cz e S do coque. Desta forma todos os principais parâmetros de controle do coque estarão bem controlados mesmo antes da mistura estar no processo da coqueria, ou seja, a mistura que estará sendo utilizada na coqueria será criada utilizando modelos muito mais acurados para predição dos parâmetros do coque.

REFERÊNCIAS

SILVA, Guilherme Liziero R. Silva; DESTRO, Elton; MARINHO, Geraldo Magela; ASSIS, Paulo Santos. **Primeiro Seminário de carvão: Problema da otimização de mistura de carvão na produção de coque metalúrgico**. Gramado, 2011.

SILVA, Guilherme Liziero R. Silva; DESTRO, Elton; MARINHO, Geraldo Magela. **Otimização da mistura de carvão na produção de coque metalúrgico**. 2011. 157f. Dissertação (Mestrado em Engenharia de Materiais) – REDEMAT, UFOP, Ouro Preto.

LUENBERGER, David G.; YE, Ynyu. **LINEAR AND NONLINEAR PROGRAMMING**. 3rd. Edition. New York, 2008

AGUIRRE, Luis Antonio. **Introdução á identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais**. 3ª Ed. Belo Horizonte: Editora UFMG, 2007.

SOARES, Gustavo Luís. **Algoritmos Genéticos: Estudos, Novas Técnicas e Aplicações**. 1997. Dissertação, UFMG, Belo Horizonte.

NETO, Silvio Petrolí. **Computação Evolutiva: desvendando os algoritmos genéticos**. Dissertação, FAJ, Jaguariúna.

SILVA, Emílio Carlos Nelli. **Técnicas de Otmização Aplicadas no Projeto de Peças Mecânicas**. USP. São Paulo.

PACHECO, Marco Aurélio C. **Algoritmos Genéticos: Princípios e Aplicações**. 1999. Rio de Janeiro.

Instituto Aço Brasil. Disponível em:

<<http://www.acobrasil.org.br/site/portugues/numeros/estatisticas.asp?par=201212>>.

Acesso em: julho. 2013.

BRAGA, Antônio P. **Redes neurais artificiais: teoria e aplicações**. 2ª. ed. Rio de Janeiro: LTC, 2007.

BOCCO, Mónica.; Willington, Enrique.; Arias. Mónica. Comparison of Regression and Neural Networks Models to Estimate Solar Radiation. **Chilean journal of agricultural research**. Chillán, v.70, n.30 Dez. 2010.