X Congresso Brasileiro de Inteligência Computacional (CBIC'2011), 8 a 11 de Novembro de 2011, Fortaleza, Ceará

DETERMINAÇÃO DA FUNÇÃO DE TRANSFERÊNCIA DE UM MOTOR CICLO OTTO UTILIZANDO PROGRAMAÇÃO GENÉTICA

Eduardo Siegel Guerra
Universidade Federal Tecnológica do Paraná e Robert Bosh do Brasil S/A
eduardo.guerra@hotmail.de

Heitor Silvério Lopes*e Walter Godoy Jr.
Universidade Federal Tecnológica do Paraná
hslopes@utfpr.edu.br, godoy@utfpr.edu.br

Resumo – Em um motor de ciclo Otto, a determinação do ponto de operação ideal não é uma tarefa trivial. Este ponto deve-se levar em consideração a potência desejada para a menor quantidade de poluente possível emitida pelo motor. As medições realizadas em dinamômetro desta máquina tem como saída uma grande massa de dados que inviabiliza sua análise sem uma ferramenta robusta de computação. O objetivo deste trabalho é investigar o uso de programação genética na determinação da função de transferência de um motor a ciclo Otto, de modo a obter o ponto ótimo de sua operação. O papel da programação genética é encontrar uma função que utilize dados tais como quantidade de combustível, ponto de injeção, rotação do motor, carga, temperatura do ar, e forneça como saída a taxa de emissão NOx e a quantidade de fumaça. A PG conseguiu gerar uma função de transferência adequada para a saída fumaça e observou-se que a emissão de NOx é uma função da outra saída. Os resultados foram satisfatórios e demonstram que a PG é útil para este problema real de regressão simbólica

Palavras-chave – Otimização, Programação Genética, Motor de Ciclo Otto, Função de Transferência, Dinamômetro.

1. INTRODUÇÃO

No desenvolvimento de motores a combustão, uma das etapas mais custosas para a equipe de desenvolvimento é a calibração do ponto ótimo de trabalho do motor. Este ponto ótimo de trabalho deve levar em consideração vários fatores como: as limitações físicas do motor, potência desejada, quantidade de poluentes emitidos, além de outros parâmetros de interesse do consumidor final. O motor que possuir uma relação favorável de potência versus emissões de poluentes, com certeza suprirá os requisitos do cliente final e terá sua calibração aprovada.

Na busca deste ponto ótimo, várias medições são realizadas em bancadas chamadas de dinamômetros. Nele, vários parâmetros de entrada são alterados proporcionando saídas distintas. Algumas entradas importantes são: quantidade de combustível, ponto de injeção, rotação, carga, temperatura de entrada do ar, etc. Na saída, geralmente são analisados o torque, potência e emissão de poluentes. A massa de dados gerada nestes testes é enorme, visto que há muitas variáveis de entradas, e estas, quando combinadas, podem gerar resultados extremamente diferentes. Tal volume de dados se torna uma dificuldade para o técnico calibrador entender qual o ponto ótimo de operação.

Com o intuito de facilitar a análise do calibrador, é proposto neste trabalho o levantamento de uma ou mais funções de transferência de um motor ciclo Otto, para determinados conjuntos de dados de entrada.

Os dados de entrada utilizados serão de um motor à combustão de ciclo Otto (gasolina/etanol), retirados de um dinamômetro na empresa Robert Bosch S/A situada na Alemanha. Em principio objetiva-se descobrir uma função de transferência para a emissão de NOx (veja a seguir), que é um dos parâmetros mais importantes do motor, embora a quantidade de fumaça também seja um parâmetro interessante, mas secundário. Para tanto, será utilizada mais de uma variável de entrada, por exemplo: rotação, carga, injeção de combustível, ignição, pressão de combustível e taxa de EGR. Assim, quando um calibrador necessitar a análise de um ponto que não consta na medição, pode-se aplicar os parâmetros na função de transferência obtida para estimar o valor de emissão de NOx. Também é muito interessante a obtenção de mais de uma função de transferência, mas para cada uma das variáveis de entrada, separadamente. Assim o calibrador pode analisar pontualmente o quanto cada variável de entrada do sistema interfere na saída, isto é, a emissão de NOx.

2. MOTOR DE CICLO OTTO

O ciclo "Otto" foi idealizado originalmente pelo engenheiro francês Alphonse Beau de Rochas em 1862, que teve seu princípio aprimorado e batizado pelo engenheiro alemão Nikolaus Otto, 14 anos mais tarde, em 1876 [1]. O ciclo Otto é um ciclo onde um determinado gás executa repetidamente transformações termodinâmicas, resultando em trabalho, com aplicações em: motores, turbinas, aquecimento ou refrigeração.

*Autor para contato

O motor em questão funciona baseado no ciclo Otto, no qual o combustível reage com o oxigênio do ar no interior do cilindro produzindo grande quantidade de gases, cuja pressão é capaz de mover um êmbolo. Antes de entrar no cilindro, o combustível é misturado com o ar num dispositivo que pode ser um carburador ou um servo-injetor. Atualmente, muitos motores são equipados com um sistema de injeção direta, no qual o ar e o combustível são injetados separadamente, ocorrendo a mistura no interior do cilindro.

O ciclo Otto divide-se em fases distintas, que são denominadas tempos: quando a mistura chega ao cilindro, o êmbolo desce (1º tempo: admissão); em seguida fecha-se a válvula de admissão e o êmbolo sobe e comprime a mistura (2º tempo: compressão). Neste momento, um dispositivo produz uma faísca elétrica que provoca a ignição, e a pressão dos gases aumenta ainda mais, deslocando o êmbolo para baixo (3º tempo: combustão ou explosão). Quando o êmbolo chega ao ponto inferior, abre-se a válvula de escapamento; o êmbolo sobe e os gases resultantes da queima são expulsos do interior do cilindro (4º tempo: expansão), iniciando-se em seguida um novo ciclo. Observa-se, portanto, que o ciclo Otto completa-se em quatro tempos conforme mostra a Figura 1.

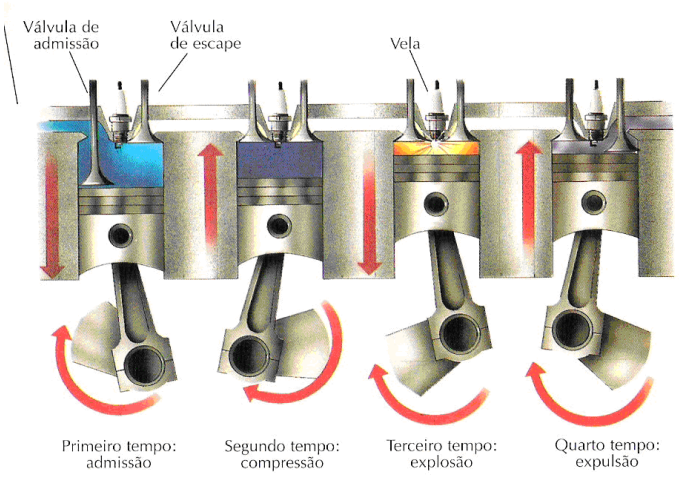

Figura 1: Funcionamento de um motor ciclo Otto.

Alterando-se variáveis de entrada, tais como, o tempo de injeção, a pressão de combustível, a quantidade de gases de escapes a serem recirculados na admissão, o ângulo de injeção e a temperatura do motor, é possível obter comportamentos distintos e valores muito diferentes para as variáveis de saída. Além do torque e potência resultantes da queima de combustível, as emissões de gases poluentes é o objeto de análise deste trabalho como variável de saída. A concentração de NOx (óxido nitroso e derivados) e particulados (fumaça), serão os parâmetros de análise da eficiência enérgica de um motor e sua operação ótima. Portanto, a determinação de uma função de transferência que leve em consideração as entradas e saídas citadas auxiliam e muito a determinação do ponto ótimo de operação pelo fabricante do motor [2].

3. COMPUTAÇÃO EVOLUCIONÁRIA E PROGRAMAÇÃO GENÉTICA

A ideia de simular inteligência e aprendizado em computadores utilizando princípios evolutivos pode ser observada em um conjunto de técnicas propostas. Além da Programação Genética – PG [3], pode-se citar ainda a Programação Evolucionária, Vida Artificial, Estratégias Evolucionárias e Algoritmos Genéticos. Estas técnicas estão incluídas na grande área de pesquisa denominada de Computação Evolucionária que, por sua vez, é incluída na Inteligência Computacional [4].

Dentro da Computação Evolucionária, o Algoritmo Genético (AG), inicialmente propostos por John Holland [5] é o paradigma mais conhecido e mais aplicado com sucesso em uma grande quantidade de problemas, principalmente na área de otimização. Essencialmente a técnica engloba uma população indivíduos codificada como cromossomos, tipicamente usando uma representação binária, cada indivíduo da população representa uma solução para o problema em questão. Como estas representações são mais fenotípicas, facilitariam sua utilização em determinados ambientes, onde essa transformação "fenótipo – genótipo" é muito complexa.

Para medir a adaptação destes indivíduos ao ambiente (problema a ser resolvido) é definida uma função de fitness, que é aplicada a cada solução candidata. Um critério de seleção vai fazer com que, depois de muitas gerações, o conjunto inicial de indivíduos gere indivíduos mais aptos. A maioria dos métodos de seleção são projetados para escolher preferencialmente indivíduos com valores maiores de aptidão, embora não exclusivamente, a fim de manter a diversidade da população.

Um método de seleção muito utilizado é o método da roleta, onde indivíduos de uma geração são escolhidos para fazer parte da próxima geração, através de um sorteio de roleta. Neste método, cada indivíduo da população é representado na roleta proporcionalmente ao seu valor de aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Um conjunto de operações é necessário para que, dado uma população, seja possível gerar populações sucessivas que, espera-se, melhorem sua aptidão com o tempo.

Estes operadores são: cruzamento e mutação. Eles são utilizados para assegurar que a nova geração possua, de alguma forma, características de seus pais, ou seja, a população se diversifica e mantém características de adaptação adquiridas pelas gerações anteriores.

Para prevenir que os melhores indivíduos não desapareçam da população pela manipulação dos operadores genéticos, eles podem ser automaticamente colocados na próxima geração, através da reprodução elitista, que consiste em manter os melhores indivíduos de uma geração para outra.

A Programação Genética (PG) tem sido considerada por diversos autores como uma extensão ou variação dos algoritmos genéticos devido à semelhança entre as duas abordagens. A diferença fundamental entre ambas está na forma de representação das soluções. Enquanto os Algoritmos Genéticos utilizam uma representação abstrata usualmente sob a forma de um *string* de bits, a Programação Genética apresenta como solução estruturas complexas sob a forma de "programas", compostos de elementos de uma linguagem pré-definida. O objetivo central, portanto, na PG é descobrir uma sequência de instruções que possa resolver o problema proposto.

Muitos problemas semelhantes em aprendizado de máquina, inteligência artificial e processamento simbólico podem ser vistos como uma busca ou descoberta de um programa de computador que produza determinadas saídas a partir de um conjunto específico de entradas. Este tipo de problema está presente tanto na engenharia quanto na computação e a PG pode prover soluções interessantes [6–9]

A partir da hipótese de que um programa correto possa ser induzido a partir de um conjunto de casos de testes, pode-se considerar esta atividade como uma tarefa de indução, que por sua vez está sujeita a todos os méritos e problemas do raciocínio indutivo. Um dos principais problemas é que uma hipótese indutiva está ligada diretamente à capacidade que casos de teste tem de representar o problema inteiro. A maioria dos problemas possui um conjunto grande de hipóteses possíveis. A PG é um passo além em relação a AG no que se refere à representação das possíveis soluções, pois manipula estruturas de complexidade arbitrária denominadas de programas (mais usualmente representados como árvores). As operações são realizadas diretamente com os programas hierarquicamente estruturados, que podem variar em forma e tamanho. A tarefa pode ser resumida na busca pelo programa mais adaptado, no espaço de todos os possíveis programas.

A Figura 2 ilustra os passos empregados na PG. Na simbologia empregada, M é o tamanho da população, i o número de indivíduos a população e Pr e Pc significam as probabilidades de aplicação dos operadores de reprodução e cruzamento, respectivamente.

4. MODELAGEM DO PROBLEMA

O problema em questão busca uma minimização do erro entre valores medidos em testes reais e valores teóricos a serem obtidos por uma função de transferência. Os valores medidos provêm de uma massa de dados obtida por medições realizadas em dinamômetro, e a função de transferência será obtida pela PG. Portanto, busca-se encontrar uma função matemática que utilize elementos definidos pelo usuário, e seja capaz de representar o comportamento mecânico do sistema de combustão interna ciclo Otto, com o mínimo de erro possível.

Para que o algoritmo da PG possa encontrar uma função que consiga representar o sistema em análise, deve-se utilizadas funções matemáticas arbitrariamente escolhidas pelo usuário. Por um lado, se forem utilizadas muitas funções, o algoritmo pode ter dificuldade em convergir para uma solução útil, devido à complexidade de combinar as funções de forma adequada. Por outro lado, se as funções utilizadas não poderem suprir pelo menos em partes as características do sistema sendo modelado, então a PG não conseguirá chegar a um resultado satisfatório. Assim, é necessário estabelecer um compromisso entre quantidade e expressividade das funções matemáticas utilizadas na PG [10]. O conjunto de funções utilizados foi definido como: $\mathrm{F} = \{+, -, \times, \%, \sin, \cos, \cosh, \sinh, \tanh, \exp, \log, \sqrt{}, sinc, sig\}$, onde $\%$ é a divisão protegida, $sinc(x) = \sin(x)/x$ e $sig(x) = 1/(1 - e^{-x})$.

Como elementos terminais, sobre os quais operam as funções, foram utilizados seis parâmetros de entrada do sistema, além de uma constante efêmera aleatório no intervalo $[-1..1]$ (denominada c). Assim, o conjunto de terminais é definido como $\mathrm{T} = \{rot, load, inj, ign, pcomb, EGR, c\}$, onde: rot é a rotação do motor (em rpm), $load$ é a carga, inj é a quantidade de combustível injetado, ign é o ângulo de ignição, $pcomb$ é a pressão de combustão e EGR é a taxa de recirculação de gases de escape. Cada elemento do conjunto T, exceto c, é representado como x_i, onde $i = 1..6$.

Para se avaliar as possíveis soluções geradas pela PG, foi utilizada uma função de avaliação ou função de *fitness*, dada pela Equação 1:

$$fitness = \sum_{j=1}^{N_{cf}} |S(j) - R(j)| \tag{1}$$

onde N_{cf} é o número de casos de *fitness*, isto é, o número de *t*-plas de entrada/saída que serão testados; $S(j)$ é o valor numérico obtido pela aplicação da função de transferência obtida pela programação genética ao *j*-ésimo conjunto de dados de entrada; e $R(j)$ é o *j*-ésimo valor de saída.

Para avaliar a quantidade de "acertos" das expressões evoluídas pela PG ao longo das gerações, é calculado para cada ponto de análise a tolerância em relação ao ponto de referência. Caso seja menor ou igual a 20%, é computado um acerto.

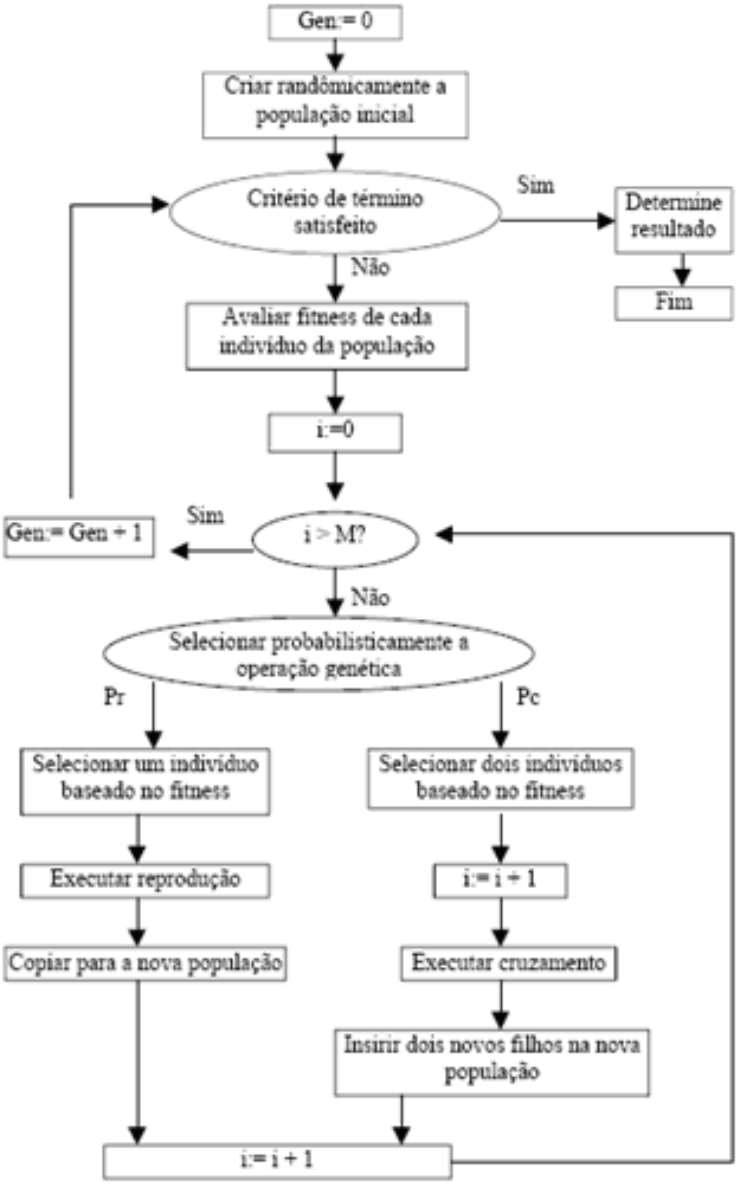

Figura 2: Resumo do método de PG.

5. EXPERIMENTOS

Para a implementação desse algoritmo, foi utilizado o software de distribuição gratuita *lilgp* [1], codificado em linguagem ANSI-C e rodando em sistema DOS baseado no sistema operacional Windows XP. Foram implementadas as alterações para que o software seja capaz de realizar a regressão simbólica proposta e utilizar todas as funções de entrada definidas na seção anterior. Os dados de entrada provenientes das medições em dinamômetro foram carregados no algoritmo, através de rotina que lê os dados de um arquivo no formato texto.

Foram utilizados para o teste do algoritmo 200 casos de *fitness*, especialmente selecionados de toda a massa de dados, sendo estes realmente representativos do problema. Em média, foram utilizados 15 valores diferentes para cada valor de rotação do motor. Como mencionado anteriormente, a função de fitness do algoritmo leva em consideração para a análise da qualidade da árvore gerada (programa) o número de acertos dos pontos dados no algoritmo. A métrica usada é de acerto do valor dos pontos, com no máximo 20% de erro. Um valor ligeiramente grande, mas que não influencia na obtenção da função de transferência.

Inicialmente foram utilizados parâmetros propostos na modelagem do problema, com o número de indivíduos da população de 5000, o número de gerações de 100, taxa de *crossover* e reprodução de 0,9 e 0,5 respectivamente, a profundidade máxima da árvore de 50.

Primeiramente foram analisadas as funções e terminais citadas na modelagem. Foi adotada então, uma estratégia de alterar as funções e terminais do algoritmo, combinando todos os elementos possíveis. Foram realizadas aproximadamente 100 simulações com a troca destas funções. Nestes experimentos observado que as variáveis de entrada rot, $load$ e inj são as mais utilizadas

[1] Disponível em: http://garage.cse.msu.edu/software/lil-gp/

nos programas, bem commo as funções sig e $sinc$. Este fato levou à seleção de apenas as funções que mais apareciam nas melhores soluções encontradas pela PG e, portanto, apresentam maior correlação com a natureza física da solução procurada. Nos experimentos, foi testado como variável de saída não o NOx, como mencionado anteriormente.

Para se explorar mais os dados de entrada do sistema, foram alteradas as variáveis de entrada. Em vez de se utilizar apenas as 3 variáveis principais (rot, $load$ e inj) e buscar apenas o valor de emissão de NOx na saída, foram realizados mais testes com a combinação das outras variáveis de entrada disponíveis (ign, $pcomb$, EGR) com as duas variáveis de saída (fumaça ou NOX). Nestes experimentos notou-se que, para alguns casos, o resultado é muito ruim, em virtude da não linearidade dos dados. Por exemplo, a variável de entrada rot se tem comportamento linear em vários pontos (aproximadamente 15), no entanto, quando se leva em consideração a variável EGR, os resultados são muito ruins, pois estes valores oscilam muito e sem uma sequência lógica.

Portanto, a melhor combinação encontrada foi com as variáveis de entrada tradicionais (rot, $load$ e inj), que são realmente na prática as mais importantes no estudo da combustão de um motor. Esta foi a primeira conclusão importante dos testes, pois constatou-se uma relação da prática com os resultados da simulação do algoritmo. Foram testadas então, 8 diferentes combinações das variáveis de entrada e saída ao total. Para cada uma delas, foram realizadas 50 simulações com alterações dos parâmetros de controle da PG e algumas funções.

Durante toda a análise da combinação das variáveis de entrada (terminais) e a combinação das funções de entrada, notou-se que com o aumento do número de gerações acima de 200, não houve melhoria nenhuma no número de acertos dos pontos. Assim, o número ideal de gerações testado em várias condições foi entre 50 e 120. Abaixo disto o algoritmo não encontrou soluções interessantes e, acima, gastou-se muito tempo de processamento sem ganho significativo.

O parâmetro relacionado à população inicial foi muito importante durante as simulações. Para valores pequenos, em torno de 500, o algoritmo não obteve soluções interessantes e teve um baixo desempenho. No entanto, com o aumento deste numero para 10000 ou até 20000, o algoritmo teve uma evolução considerável, tendo o número de acertos um aumento expressivo, mais de 100 acertos. No entanto, o tempo computacional necessário aumenta drasticamente com o aumento da população. Empiricamente, então, foi ajustado o tamanho da população para 12000 indivíduos.

O valor da taxa de *crossover* utilizado pouco influenciou no desempenho do algoritmo. O valor ideal encontrado foi de 0,9, mas para valores abaixo disto não houve melhora. Para valores perto de 0,1 o algoritmo também não teve bom desempenho.

Outro parâmetro testado foi a taxa de reprodução, que teve uma influência significativa nos resultados dos testes. Para valores de 0,5 o algoritmo conseguiu efetivamente passar dos pais para os filhos as características "boas" durante a evolução dos indivíduos e da população. Valores muito baixos ou excessivamente altos não foram úteis para os testes.

6. RESULTADOS E DISCUSSÃO

Na Tabela 1 estão representados os testes realizados para a combinação das variáveis de entrada elegidas nas simulações, para as duas saídas possíveis, NOx e fumaça.

Tabela 1: Desempenho da PG nas simulações.

	Saída NOx	**Saída Fumaça**
Variáveis	**Acertos**	**Acertos**
$\{rot, load, inj\}$	120	147
$\{rot, pcomb, inj\}$	111	135
$\{rot, EGR, load\}$	89	98
$\{inj, load, EGR\}$	131	123
$\{inj, ign, load\}$	56	54
$\{rot, EGR, ign\}$	67	34
$\{load, ign, EGR\}$	65	67
$\{EGR, pcomb, ign\}$	56	87
$\{rot, ign, pcomb\}$	89	67

Observa-se que o maior número de acertos foi conseguido quando se combinou a saída "fumaça" e as variáveis rotação, carga e quantidade injeção de combustível. Isto se deve ao fato de que tais variáveis serem, de fato, as mais representativas da função de transferência do motor a combustão interna. Embora as outras variáveis sejam importantes, a PG não obteve um bom resultado com seu uso. Pode-se observar este fato, quando se utilizou o ponto de ignição (ign) nas simulações, onde foram obtidos os piores resultados em relação ao núumero de acertos.

De maneira geral, os melhores resultados foram foram para a variável de saída "fumaça". Isto se deve ao fato da maior linearidade deste parâmetro de saída e apresentar poucos valores discrepantes nos dados de entrada. Os experimentos corroboraram a prática onde se conhece que a variável fumaça é mais representativa, já que a medição de quantidade de NOx é um subproduto da fumaça e, ainda, depende de fatores relacionados ao meio de medição (bancada de testes de emissões), que geram uma certa inconstância dos valores lidos.

Na prática, os valores são sempre analisados em conjunto para se ter uma avaliação global mais precisa do motor em relação ao seu potencial de poluição.

Na Figura 3 representada a curva de *fitness* para a melhor combinação de variáveis e, consequentemente, a melhor solução encontrada durante as simulações.

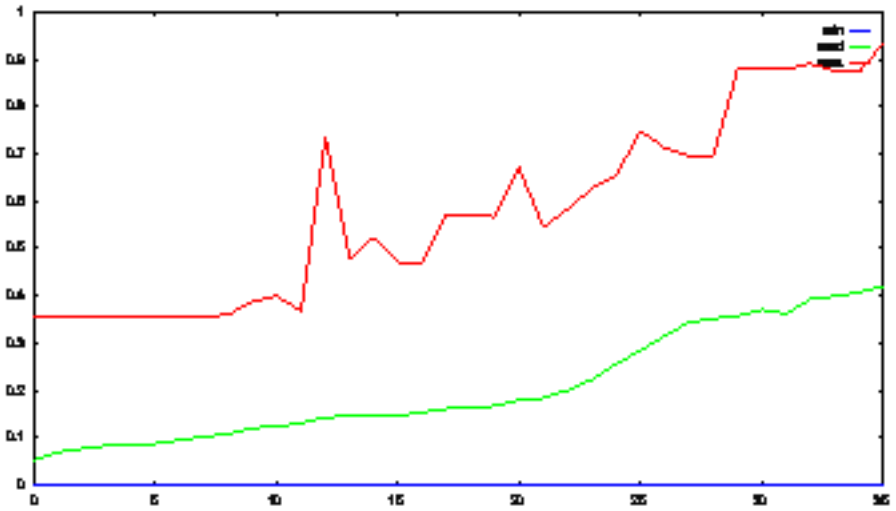

Figura 3: Curva de fitness da melhor solução encontrada.

Nota-se que, para um número de gerações maior do que 25, não há evolução significativa e o *fitness* nessa região chega a 0,8, levando a uma taxa de acerto de 147 dentro dos 200 pontos de análise.

Como a função de transferência do sistema tem três variáveis de entrada, não é possível plotar um gráfico com todos os pontos de entrada (em todas as dimensões) e saída. Assim, para se avaliar de maneira visual como está a distribuição dos pontos encontrados pela função dada pelo algoritmo, foi traçado gráfico da Figura 4. A curva em azul representa os pontos originais (*fitness cases*) para o problema, enquanto que os pontos em vermelho representam os pontos encontrados pela função de transferência sugerida pelo algoritmo. No total, foram acertados 147 pontos. Estes estão dentro da tolerância originalmente definida. A função de transferência encontrada já simplificada está representada a seguir.

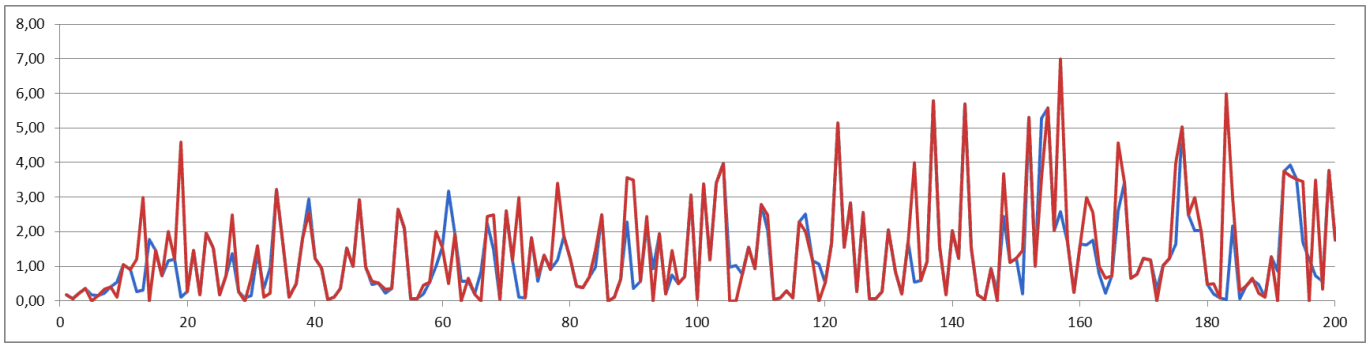

Figura 4: Comparação dos pontos obtidos com pontos referencia.

$$
\begin{aligned}
&(+(rlog(sig(-(-(+-4.61977(-(-(sinc(\sin inj))X)\\
&(+-inj(/(-rot(-(sinh(+load-1.89893))\\
&(\tanh -4.61977)))-3.33429))))-7.70065)\\
&(\sin(--inj(\tanh(rlog(load(\cos(--9.63510(sig(-(\cos(load(\sin -2.19607)))\\
&(\tanh(inj(\tanh rot)))))))))))))))
\end{aligned}
$$

7. CONCLUSÕES E TRABALHOS FUTUROS

Observou-se em geral que a função de transferência encontrada conseguiu gerar pontos muito próximos dos pontos de referência. O algoritmo conseguiu acertar 147 pontos dentro da tolerância exigida, sendo 200 o total de pontos disponíveis fornecidos. Estes dados foram selecionados de um banco de dados de 10000 pontos. Entretanto, esta massa de dados inclui muita informação errada ou inadequada para a regressão simbólica.

No entanto notou-se a eficácia da programação genética, no que diz respeito ao caráter heurístico da busca pela solução ou programa que mais represente ou resolva o problema de regressão simbólica.

O resultado do trabalho foi muito satisfatório e demostrou a viabilidade de se utilizar programação genética na análise de emissões em motores de combustão com ciclo Otto. A combinação das variáveis mostrou resultados que podem ser efetivamente aplicados na prática e seguem realmente o comportamento do motor à combustão. Com isto, a ferramenta pode ser ainda mais aprimorada e pode, num futuro próximo, ajudar muito as equipes de calibradores de motores a entender as influências das variáveis de entrada e saída em um motor a combustão.

O desenvolvimento dessa pesquisa demostrou o quão poderosa é a computação evolucionária através da implementação de programação genética. Foi visto que mesmo com uma grande massa de dados é possível se levantar uma função de caracterize uma característica física. Com isto, uma proposta para trabalho futuro está a extensão deste, utilizando a programação genética com muitas variáveis de entrada de um sistema, como foi o exemplo do motor a combustão, e tenha como saída mais de uma variável de saída. Assim, através da criação de uma interface gráfica possa se alterar as variáveis de entrada e verificar como as mesmas interferem nas de saída e desempenho do sistema. O objetivo é criar um software interativo, que recebe um banco de dados de medições, consiga correlacionar as mesmas nas procura de uma variável de saída e encontre assim o ponto ótimo de funcionamento de um sistema. De maneira mais especifica e focado no exemplo desse trabalho, o SW a ser desenvolvido teria todas as entradas e saída do motor a combustão (¿ de 20) e geraria varias funções de transferência. Estas funções seriam plotadas em uma interface gráfica e que permitam que sejam modificadas pelo usuário na busca de um ponto ótimo de trabalho de um motor a combustão, por exemplo um ponto que utilize menos combustível, melhor potência de saída e baixa emissões de fumaça e NOx.

REFERÊNCIAS

[1] Infomotor. "Funcionamento do Motor a Combustão Interna Ciclo Otto". Disponível na Internet, 2009. http://www.infomotor.com.br.

[2] R. L. Barbosa, F. M. d. Silva, N. Salvador and C. E. S. Volpato. "Desempenho comparativo de um motor de ciclo diesel utilizando diesel e misturas de biodiesel". *Ciência e Agrotecnologia*, vol. 32, no. 5, pp. 1588–1593, 2008.

[3] J. R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, 1992.

[4] A. P. Engelbrecht. *Computational Intelligence: An Introduction*. J. Wiley & Sons, Chichester, UK, second edition, 2007.

[5] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, USA, 1975.

[6] C. Bojarczuk, H. Lopes, A. Freitas and M. Michalkiewicz. "A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets". *Artificial Intelligence in Medicine*, vol. 30, no. 1, pp. 27–48, 2004.

[7] H. Lopes. "Genetic programming for epileptic pattern recognition in electroencephalographic signals". *Applied Soft Computing*, vol. 7, pp. 343–352, 2007.

[8] E. Rodrigues and H. Lopes. "Genetic Programming for Induction of Context-free Grammars". In *Proceedings of* 7^{th} *International Conference on Intelligent Systems Design and Applications*, Piscataway, USA, 2007. IEEE Computer Society Press.

[9] D. Tsunoda, A. Freitas and H. Lopes. "A Genetic Programming Method for Protein Motif Discovery and Protein Classification". *Soft Computing*, 2010.

[10] J. Rosca. "Generality versus Size in Genetic Programming". In *Proceedings of the First Annual Conference on Genetic Programming*, pp. 381–387, Cambridge, USA, 1996. MIT Press.