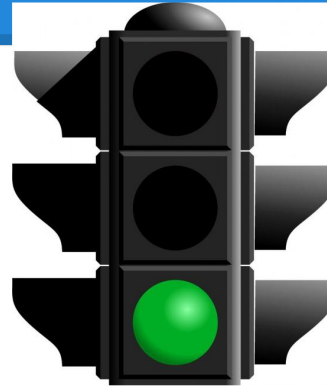


Controlador fuzzy de tráfego com algoritmos genéticos

Nilton Vasques
Victor Costa

Universidade Federal da Bahia
Lab. Inteligência Artificial
Profa. Tatiane Nogueira



Roteiro

- Introdução
- Algoritmos Genéticos utilizados
 - MOGUL
 - LT.RS
- Resultados
- Análises
- Conclusão
- Bibliografia

Introdução

Segundo CORDON et al, para desenvolver um sistema inteligente fuzzy, é necessário o cumprimento de duas tarefas:

1. Selecionar os operadores fuzzy envolvidos no sistema de inferência
 - a. Definir o modo de execução do processo de inferência fuzzy
2. Gerar uma adequada base de regras fuzzy sobre o problema a ser resolvido

A precisão do sistema fuzzy dependerá diretamente desses componentes.

Introdução

- Os Algoritmos Genéticos podem ser empregados em diversas fases de um sistema baseado em regras fuzzy (SBRF).
- Neste trabalho, os AG foram utilizados para geração de regras (MOGUL) e para refinamento (Lateral Tuning + Rule Selection).

Os algoritmos genéticos utilizados

No software R, existem 6 algoritmos genéticos disponíveis para geração de regras.

Tarefas	Regressão	Classificação
Algoritmos	The Thrift's Method	The Ishibuchi's Method based on genetic cooperative-competitive learning
	The genetic fuzzy systems for fuzzy rule learning based on the MOGUL methodology	The Ishibuchi's Method based on hybridization of genetic cooperative-competitive learning
	The genetic for lateral tuning and rule selection of linguistic fuzzy system	The structural learning algorithm on vague environment

Algoritmos de geração de regras

- Método de MOGUL
 - Foi utilizado o modelo: Takagi-Sugeno-Kang.
 - O consequente das regras é uma função das variáveis de entrada.

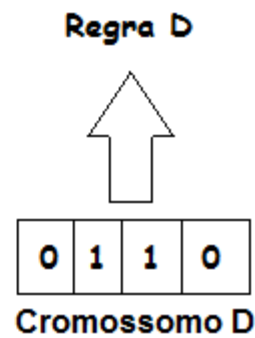
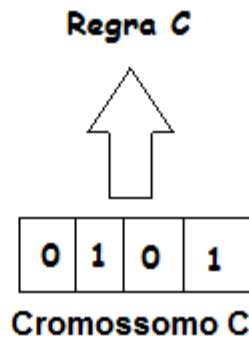
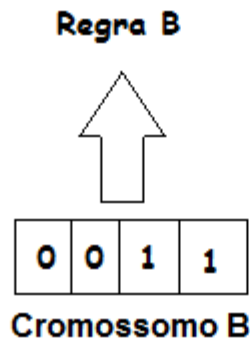
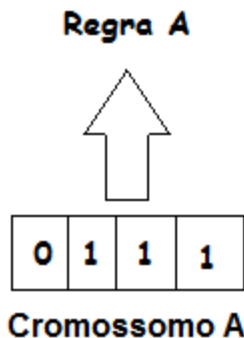
Regra: SE x é A E y é B THEN $z = f(x, y)$

$$z = px + qy + r$$

Algoritmos de geração de regras

- Método de MOGUL

- Baseado na abordagem de Michigan.
 - Cada cromossomo representa uma regra.
 - A função de aptidão do cromossomo é o grau de compatibilidade dessa regra.



Algoritmos de geração de regras

- Fases do método de MOGUL

1. Os cromossomos são gerados pelo método denominado “*Iterative Rule Learning Approach*”. (IRL)
2. Depois que se obtém a população inicial, o algoritmo genético é disparado utilizando os operadores genéticos de mutação, cruzamento e seleção especificados nos parâmetros.
3. No pós-processamento, o resultado é refinado removendo regras redundantes e desnecessárias. Para isso, é utilizado a abordagem de Pittsburgh.

Algoritmos de geração de regras

- O IRL é composto por iterações que funcionam da seguinte maneira:
 1. É selecionado o cromossomo que obteve o maior valor de fitness para compor a população.
 2. A regra associada a este cromossomo é “penalizada”, removendo os exemplos do treinamento que unificaram com a mesma.

Cromossomo	Fitness
A	15%
B	10%
C	5%



1. Este cromossomo é inserido na população final
2. A regra associada a este cromossomo é penalizada, removendo do conjunto de treinamento as amostras que casam com a regra
 - a. Repete o processo n vezes. O valor de n é um parametro do algoritmo.

Algoritmos de geração de regras

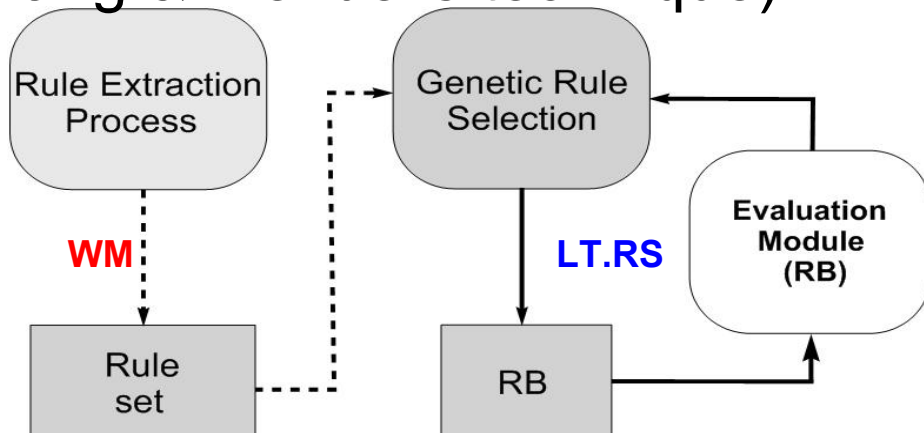
- LT.RS -> Algoritmo genético para deslocamento lateral e seleção de regras de um sistema fuzzy linguístico
 - É um algoritmo para pós-processamento.
 - Será executado em cima da base de regras, que já deve estar criada.

Algoritmos de geração de regras

- LT.RS

- O conjunto inicial de regras foi gerado pelo algoritmo de geração de regras: **WM**.

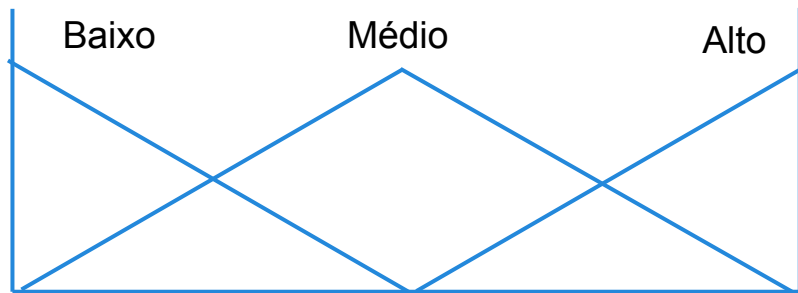
(Wang & Mendel's technique)



Algoritmos de geração de regras

- LT.RS

- Utiliza o modelo Mamdani:
 - Ambos antecedentes e consequentes de uma regra são variáveis linguísticas.



Algoritmos de geração de regras

- LT.RS
 - Processos distintos a serem abordados para compor o cromossomo desse algoritmo genético:
 1. “*Lateral tuning*”
 2. “*Rules selection*”

Algoritmos de geração de regras

- O “*lateral tuning*”
 - É uma variação no formato das funções de pertinencia para induzir a uma melhor cooperação entre as regras finais do sistema.
 - Tem como objetivo encontrar os melhores arranjos das funções de pertinencia para o domínio em questão.

Algoritmos de geração de regras

- O “*lateral tuning*”
 - Para realizar esse processo, as regras tem a sua representação modificada.
 - 2-Tuplas nos lugares dos Conjuntos Fuzzys.
 - O primeiro valor da 2-tupla será o conjunto fuzzy daquela variável linguística.
 - O segundo valor da 2-tupla refere-se a translação simbólica do conjunto fuzzy para a esquerda ou direita. dependendo se o seu valor é negativo ou positivo. $\alpha \in [-0.5, 0.5]$

Algoritmos de geração de regras

- O “*lateral tuning*”: Exemplo prático

- Regras no formato clássico:

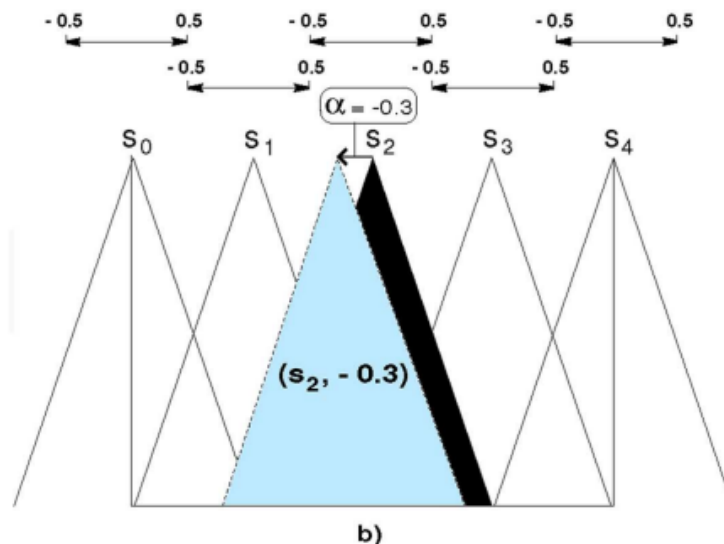
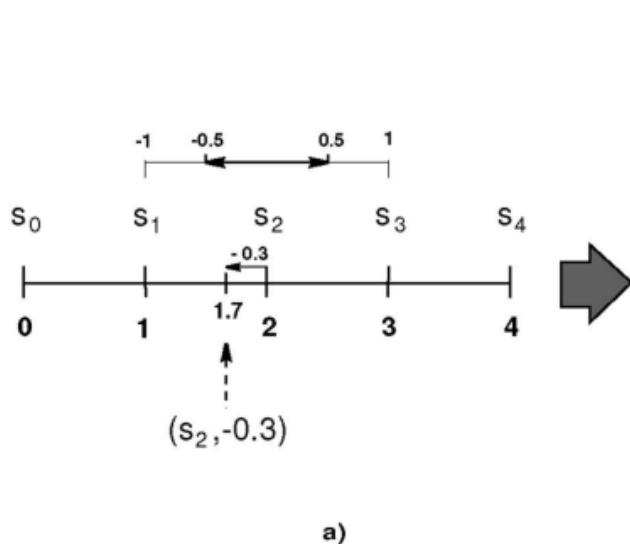
SE Idade é **Média** ENTÃO Peso é **Alto**

- Regras na representação de 2-Tuplas para realizar o “*lateral tuning*”

SE Idade é (**Média**, **-0.3**) ENTÃO Peso é (**Alto**, **0.1**)

Algoritmos de geração de regras

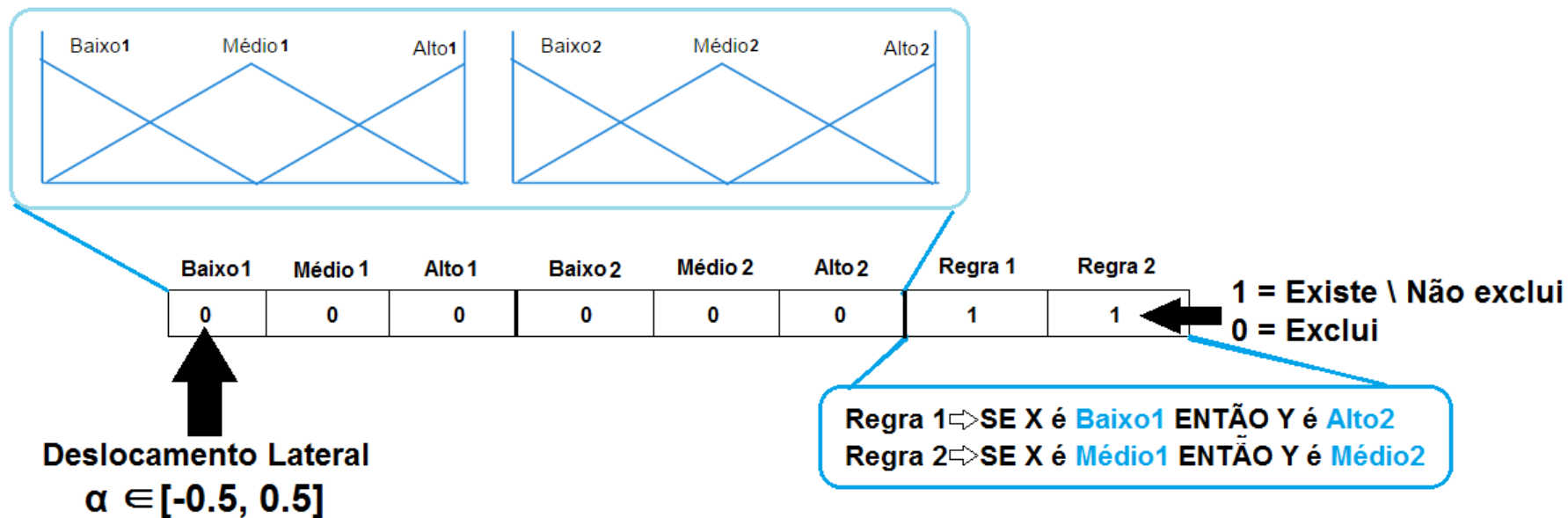
- O “*lateral tuning*”: Resultado



Algoritmos de geração de regras

- O “rules selection”
 - Nada mais é do que um processo para eliminar regras irrelevantes, redundantes, erradas e conflitantes, que atrapalham o desempenho do SBRF ao coexistirem com as outras regras.

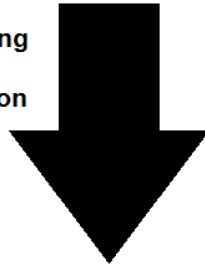
Cromossomo inicial



Exemplo dos outros cromossomos

Mutação no Cromossomo Inicial

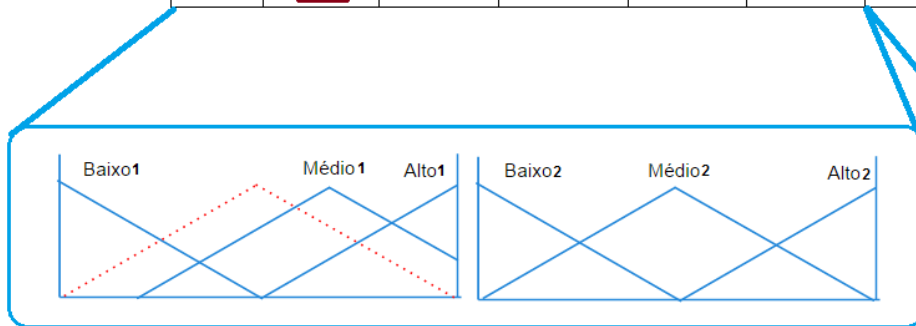
Lateral Tuning
+
Rule Selection



Lateral Tuning

Rule
Selection

Baixo1	Médio 1	Alto1	Baixo2	Médio 2	Alto2	Regra 1	Regra 2
0	<u>0,2</u>	0	0	0	0	1	<u>0</u>



Regra 1 \Rightarrow SE X é Baixo1 ENTÃO Y é Alto2

Algoritmos de geração de regras

- “*Global tuning*” X “*Local tuning*”
 - No “**Global tuning**”, cada conjunto fuzzy de uma variável linguística só pode ter um deslocamento lateral. Se determinarmos que “*Idade é (Média, 0.3)*”
todas as regras que tiverem “*Idade é Média*” serão substituídas por “*Idade é (Média, 0.3)*”.

Algoritmos de geração de regras

- “*Global tuning*” X “*Local tuning*”
 - No “**Local tuning**”, o deslocamento lateral dos conjuntos fuzzy nas regras são independentes. Ou seja, em uma regra pode-se determinar que

“Idade é (*Média*, 0.3)”

E em outra regra, pode-se determinar que

“Idade é (*Média*, 0.1)”

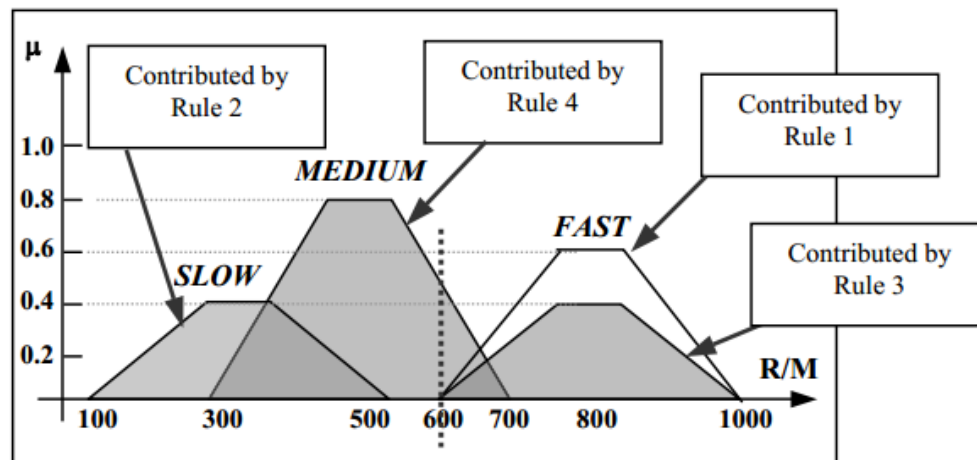
Exemplo dos outros cromossomos

- A função de aptidão (fitness)
 - A função de aptidão do cromossomo será inversamente proporcional ao Erro Quadrático Médio (MSE) da base de regras representada por ele.
- Cruzamento
 - Cruzamento de dois pontos na Seção das Regras e cruzamento na Seção dos Deslocamentos Laterais.
- Mutação
 - É aplicado na Seção das Regras antes do cruzamento.

Defuzzificação

COG

- “The most prevalent and physically appealing of all the defuzzification methods [Sugeno, 1985; Lee, 1990]”
- Alta Complexidade!
É um dos métodos que mais requerem recursos computacionais.



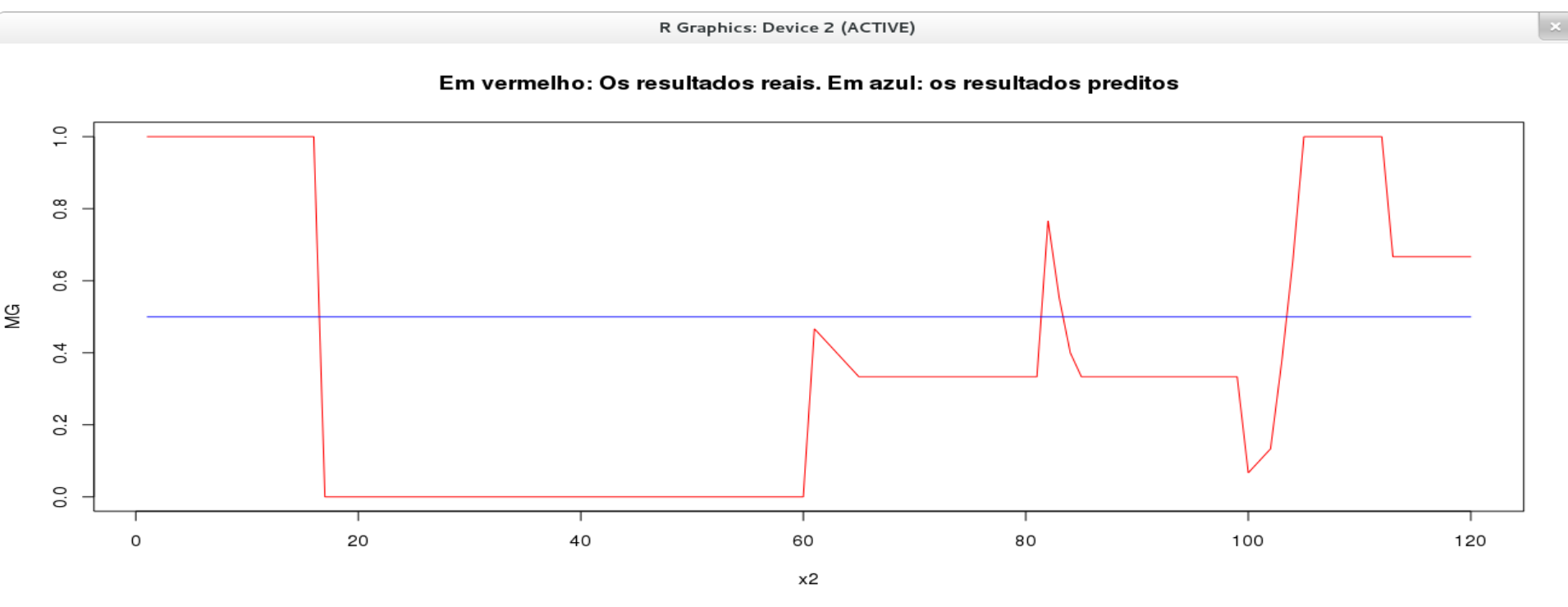
Resultados - MSE (Erro Quadrado Médio)

MOGUL+COG	0.15674886 (~15%)
WM+COG	0.05071358 (~5%)
LT+COG (200 gerações)	0.0462144 (~5%)
DENFIS+COG	0.03334778 (~3%)

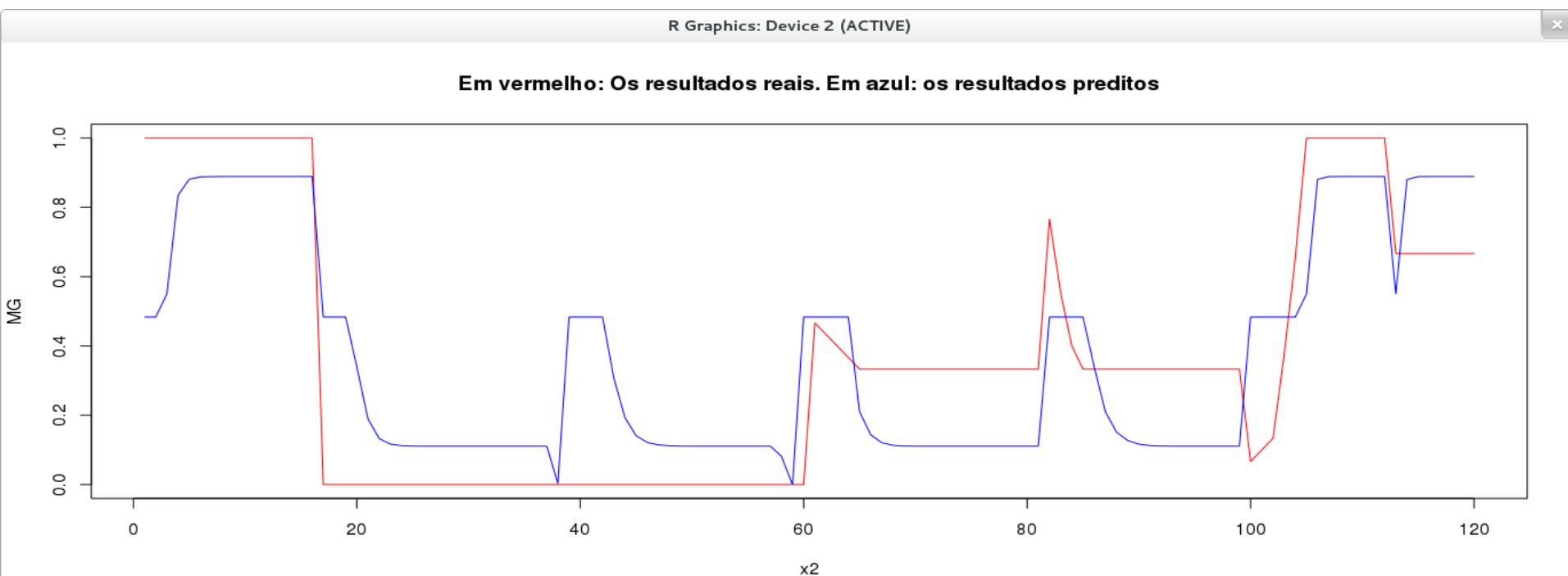
Resultados - RMSE (Raiz do Erro Quadrado Médio)

MOGUL+COG	0.39731033
WM+COG	0.22519676
LT+COG (200 gerações)	0.2149753
DENFIS+COG	0.18261375

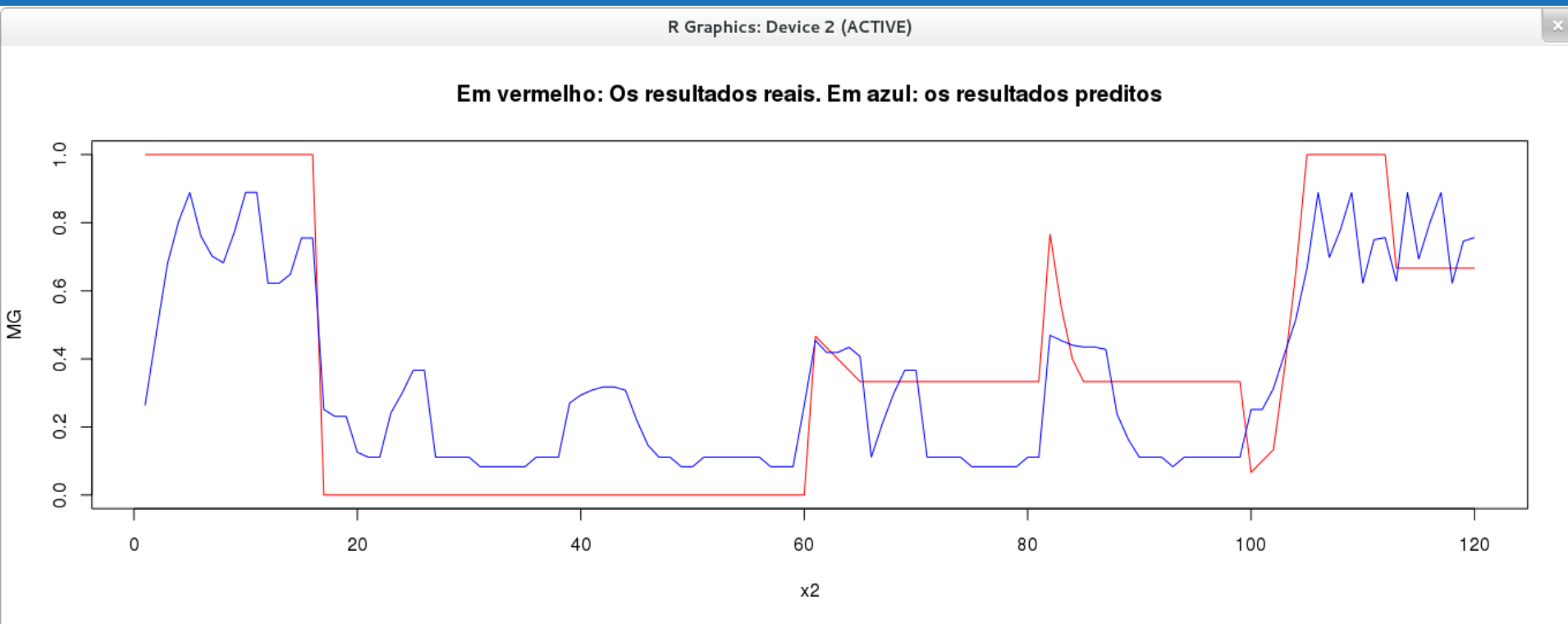
Resultados - Gráfico - MOGUL+COG



Resultados - Gráfico - WM+COG



Resultados - Gráfico - LT+COG



Resultados - Gráfico - DENFIS+COG

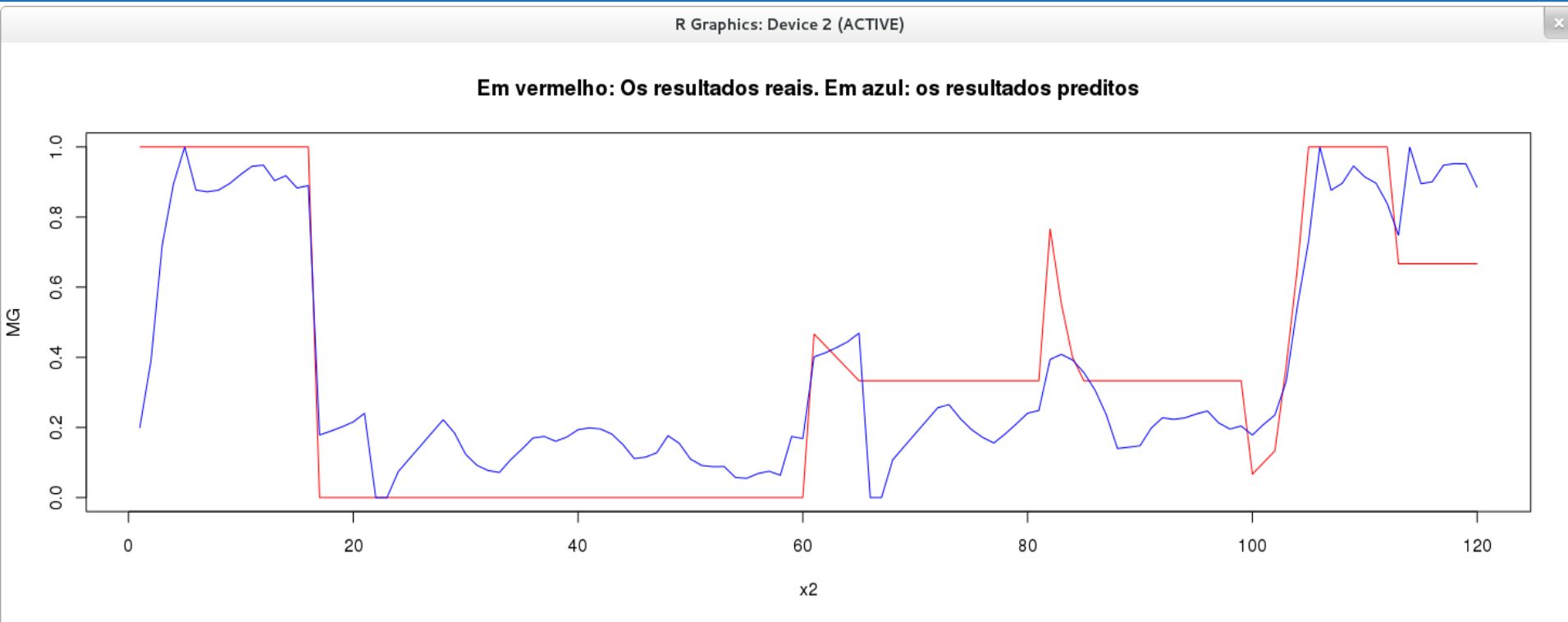


Gráfico - LT+COG (25 gerações)



Gráfico - LT+COG (50 gerações)

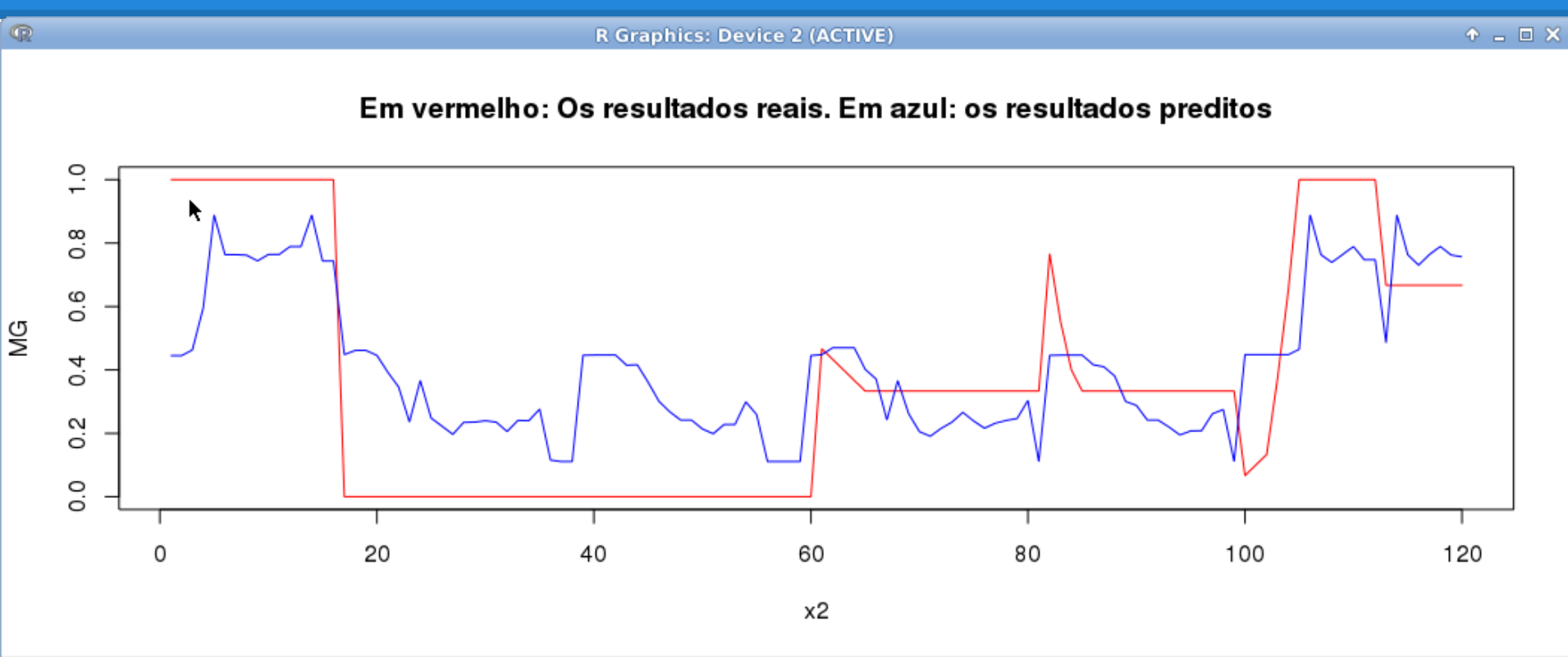


Gráfico - LT+COG (75 gerações)

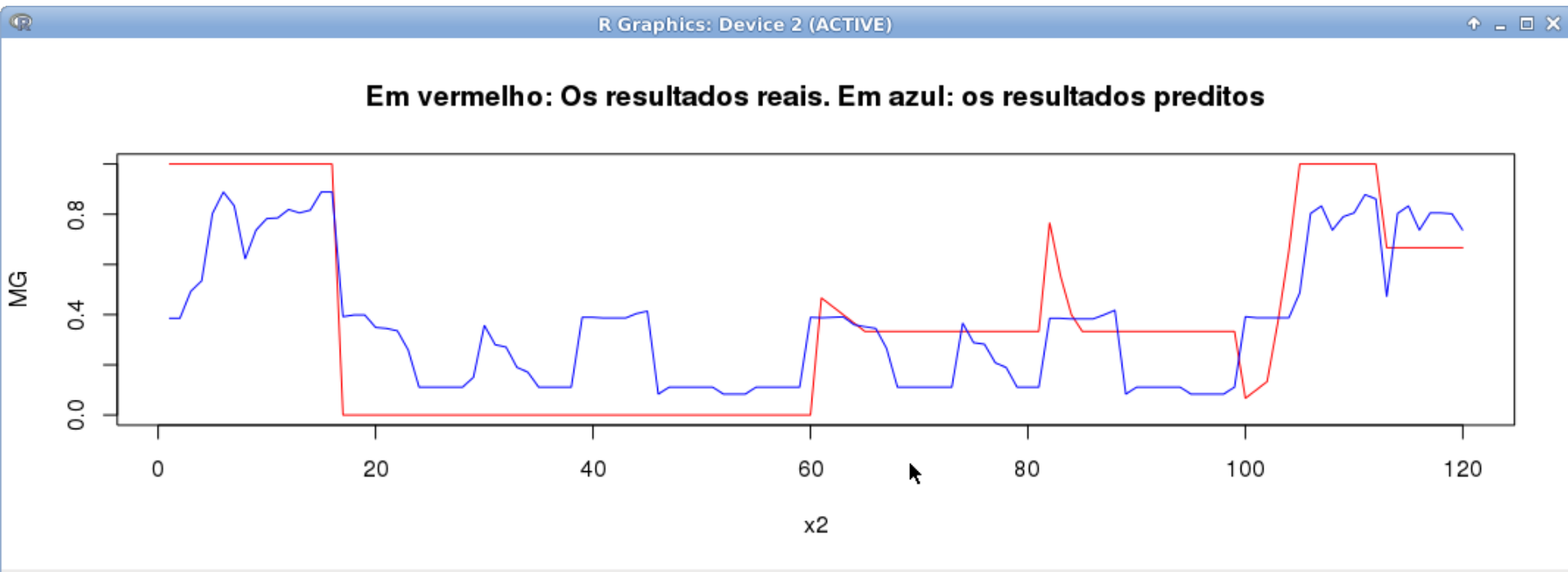


Gráfico - LT+COG (100 gerações)

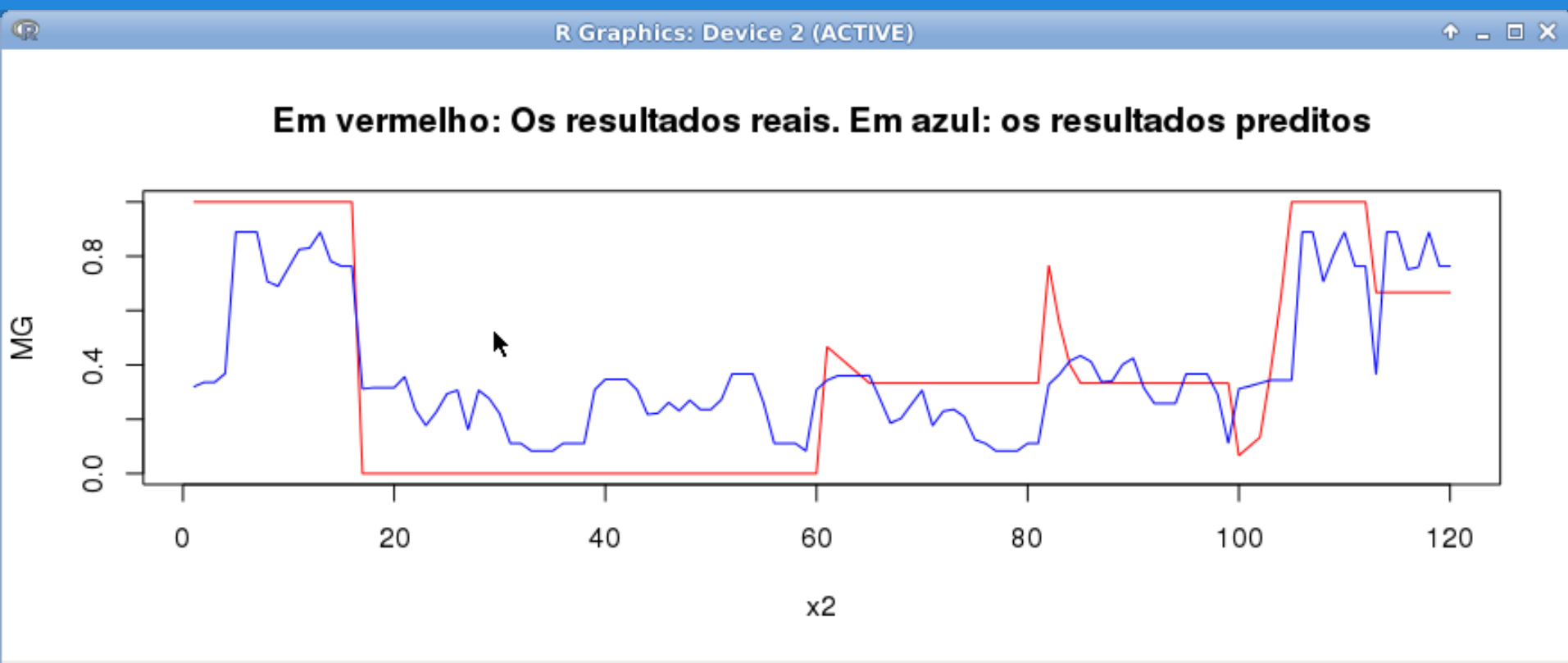
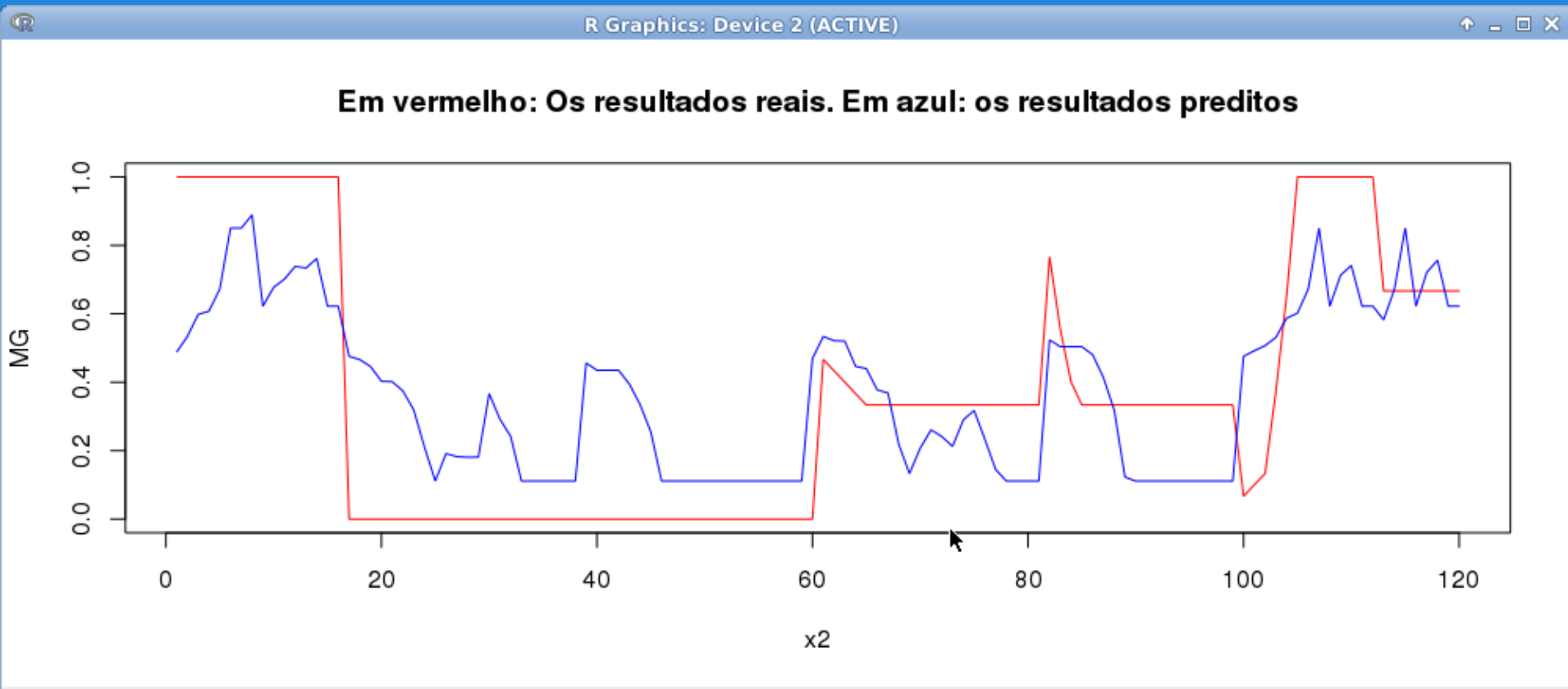


Gráfico - LT+COG (150 gerações)



Justificativa Resultados

- Segundo HERRERA et al: *“os algoritmos genéticos não necessariamente encontram uma solução ótima para qualquer problema, mas podem encontrar boas soluções para problemas que são resistentes a outras técnicas”*.
- Os métodos de geração de regras utilizados no nosso último estudo mostraram resultados satisfatórios em geral. Estes métodos foram:
 - WM
 - HyFIS
 - DENFIS

O resultado dos algoritmos genéticos não foram satisfatórios porque já encontramos ótimos resultados com os métodos antigos. Com destaque ao DENFIS

Trabalhos futuros

- A partir dos resultados, pensou-se em unir as duas estratégias de Michigan e Pittsburg em uma solução.
- Onde utilizaria a abordagem de michigan em base de regras isoladas, com intuito de refiná-las. A exemplo, o WM e o HyFIS.
- E posteriormente a utilização do método de Pittsburg para cruzar essas bases.

Código

<https://github.com/niltonvasques/traffic-control-r>



Bibliografia

- [1]- <http://sci2s.ugr.es/publications/ficheros/ijis-14%2811%29-1123-1153.pdf>
- [2]- F. Herrera, M. Lozano, J.L. Verdegay. Applying Genetic Algorithms in Fuzzy Optimization Problems. University of Granada, 1994
- [3]- citeseerx.ist.psu.edu/viewdoc/download;jsessionid=653202F59EA1C2C77D55F0ECFB5FA106?doi=10.1.1.57.2606&rep=rep1&type=pdf
- [4]- <https://github.com/cran/frbs>
- [5]- https://docs.google.com/presentation/d/http://sci2s.ugr.es/publications/ficheros/ijis-14%2811%29-1123-1153.pdf/edit#slide=id.g43b672417_276
- [6]- <http://sci2s.ugr.es/gfs/>

Dúvidas



Obrigado!