Implementação de um Sistema Especialista Fuzzy para Calcular o Ganho de um Funcionário por Produção e Tempo na Fabricação de Peças Eletrônicas

Lucas Cordeiro de Souza, Mauricio Demonti Amorim

1. Introdução

Neste relatório será apresentado o detalhamento de implementação de um Sistema Especialista Fuzzy utilizando a ferramenta FuzzyClips, com a finalidade de calcular o ganho (em reais) diário de um funcionário fabricante de peças eletrônicas, considerando como variáveis auxiliares o tempo estimado e a quantidade produzida.

As variáveis linguísticas de entrada e saída são mostradas na Tabela 1:

	TEMPO		
PRODUÇÃO	**POUCO**	**MÉDIO**	**MUITO**
MUITO-BAIXA	Baixo	Baixo	Baixo
BAIXA	Médio	Baixo	Baixo
MÉDIA	Alto	Médio	Baixo
ALTA	Alto	Alto	Médio
MUITO-ALTA	Alto	Alto	Alto

Tabela 1. Variáveis linguísticas para calcular o ganho diário de monetização.

Na primeira coluna da tabela estão descritos os valores possíveis de produção de peças, enquanto na primeira linha tempos as possíveis variáveis de tempo (Pouco, médio e muito). Já as demais colunas estão descritos os resultados dos ganhos por produção em seu determinado tempo.

2. Implementação e Testes

Para cada variável linguística foi definido um template. No caso do template Tempo, utilizou-se duas funções pré-definidas (z) no início e (s) para final e uma função trapezoide de três pontos:

```
(deftemplate tempo
0 8 tempo_producao_horas
  ((pouco (z 0 4))
    (medio (2 0)(4 1)(6 0))
    (muito (s 5 8))
  ))
```

A figura 1 ilustra um gráfico com os valores numéricos possíveis para o tempo de trabalho considerando as variáveis linguísticas definidas na Tabela 1 e no template citado.

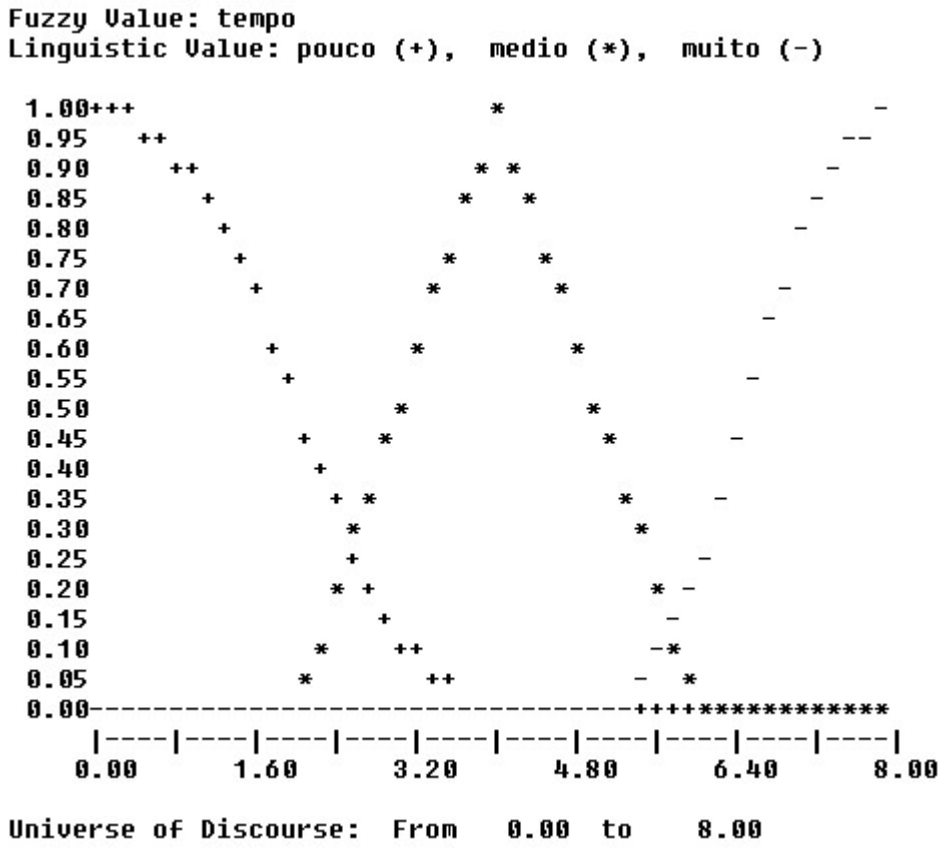

Figura 1. Plotagem dos Valores numéricos possíveis para o tempo de trabalho

A Figura 2 ilustra agora os valores numéricos possíveis para a quantidade de produção de peças, sendo utilizadas duas funções pré-definidas (z e s), e três trapezoides:

```
(deftemplate producao
0 500 qtd_produzida
  (
    (muito-baixa (z 0 110))
    (baixa (90 0)(110 1)(190 1)(210 0))
    (media (190 0)(210 1)(290 1)(310 0))
    (alta (290 0)(310 1)(390 1)(410 0))
    (muito-alta (s 390 500))
  )
)
```

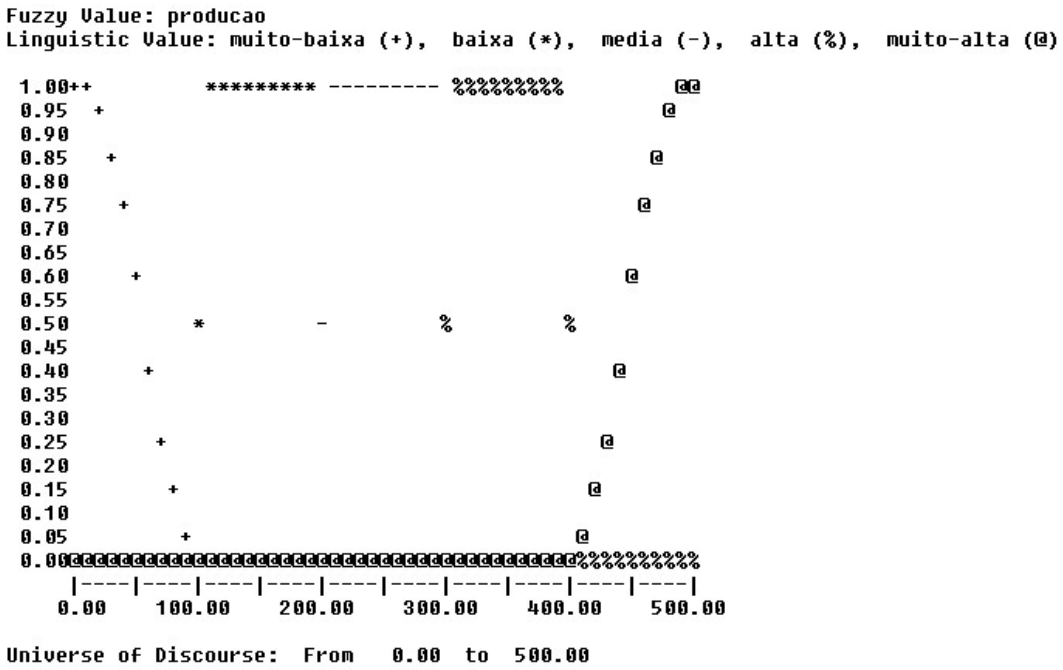

Figura 2. Plotagem dos Valores numéricos possíveis para a quantidade de produção

A Figura 3 ilustra os possíveis ganhos diários relacionados a produção e o tempo de trabalho, sendo composta por duas funções pré-definidas (z e s) e um trapezoide:

```
(deftemplate ganho
0 300 ganho_producao/tempo
  (
    (baixo (z 0 110))
    (medio (90 0)(110 1)(190 1)(210 0))
    (alto (s 190 300))
  )
)
```

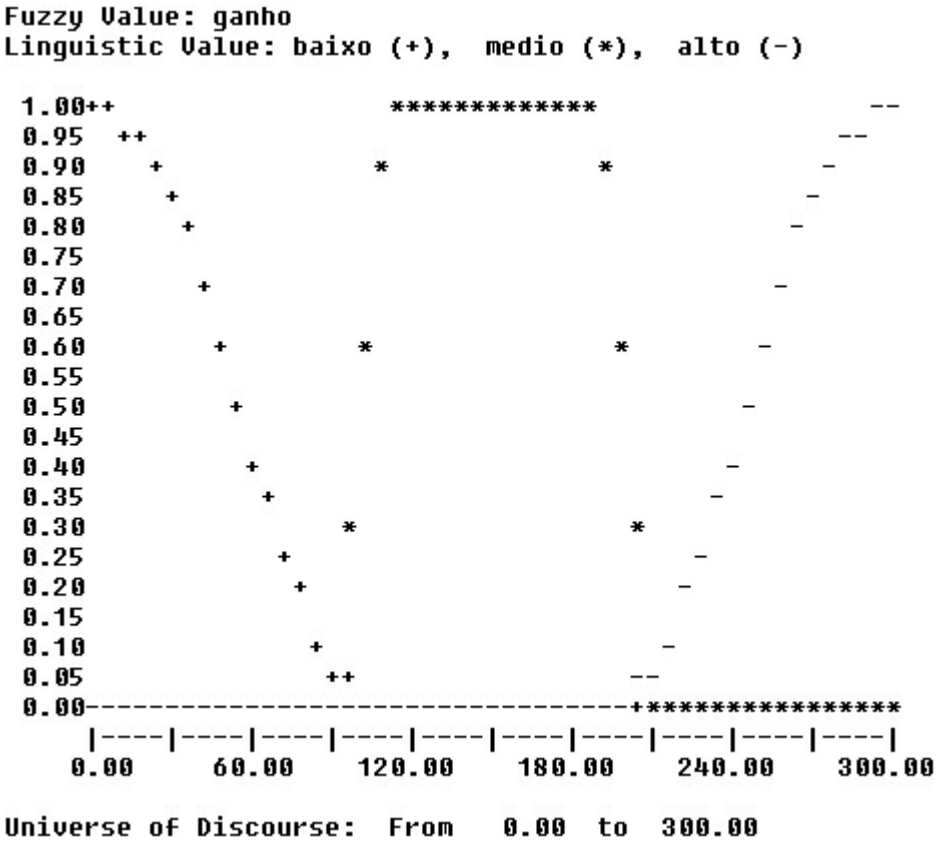

Figura 3 Plotagem dos Valores numéricos possíveis para o ganho diário.

A definição do resultado de Ganho é gerada por três regras distintas (defrule baixo, defrule medio e defrule alto), uma para cada variável linguística. Foi utilizada a declaração da salience para que as regras fossem executadas antes da regra de defuzzificação.

```
(defrule baixo
  (declare (salience 10))
  (or (and (producao muito-baixa) (tempo pouco))
      (and (producao muito-baixa) (tempo medio))
      (and (producao muito-baixa) (tempo muito))
      (and (producao baixa) (tempo medio))
      (and (producao baixa) (tempo muito))
      (and (producao media) (tempo muito))
)
=>
  (assert (ganho baixo))
)
```

```
(defrule medio
  (declare (salience 10))
  (or (and (producao baixa) (tempo pouco))
      (and (producao media) (tempo medio))
      (and (producao alta) (tempo muito))
)
=>
  (assert (ganho medio))
)

(defrule alto
  (declare (salience 10))
  (or (and (producao media) (tempo pouco))
      (and (producao alta) (tempo pouco))
      (and (producao alta) (tempo medio))
      (and (producao muito-alta) (tempo pouco))
      (and (producao muito-alta) (tempo medio))
      (and (producao muito-alta) (tempo muito))
)
=>
  (assert (ganho alto))
)
```

Foi utilizado o procedimento de defuzzificação para se obtermos um resultado. Para a defuzzificação, é criada uma variável global e uma regra que também faz a plotagem do valor numérico encontrado. A regra defuzifica foi declarada com salience 0, forçando-a a ser executada após as demais regras.

```
(defglobal
?*g_resultado* = 0
)
(defrule defuzifica
  (declare (salience 0))
  ?v_tmp <- (ganho ?)
=>
  (bind ?*g_resultado* (moment-defuzzify ?v_tmp))
```

```
  (plot-fuzzy-value t "*" nil nil ?v_tmp)
  (retract ?v_tmp)
  (printout t "Ganho: ")
  (printout t ?*g_resultado* crlf)
)
```

Para testar as regras apresentadas, geramos os deffacts, obtendo valores numéricos como resultado para as regras e fatos apresentados, devido a defuzificação citada acima.

Deffacts - 1

```
(deffacts producao/tempo
  (producao muito-alta)
  (tempo pouco))
```

A figura 4 ilustra o resultado obtido a partir dos fatos citados acima.

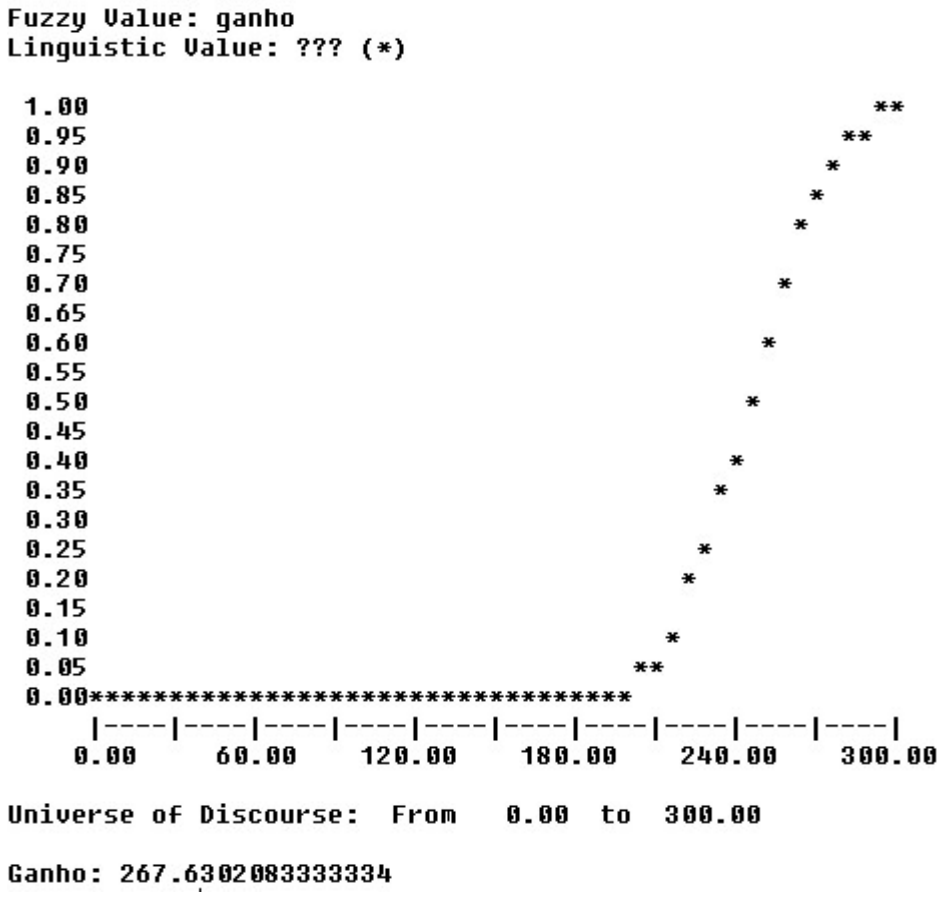

Figura 4. Plotagem do Deffacts – 1. Ganho de uma produção muito alta em pouco tempo.

Deffacts - 2

```
(deffacts producao/tempo
  (producao muito-baixa)
  (tempo muito)
)
```

A figura 5 ilustra o resultado utilizando as informações contidas no Deffacts - 2.

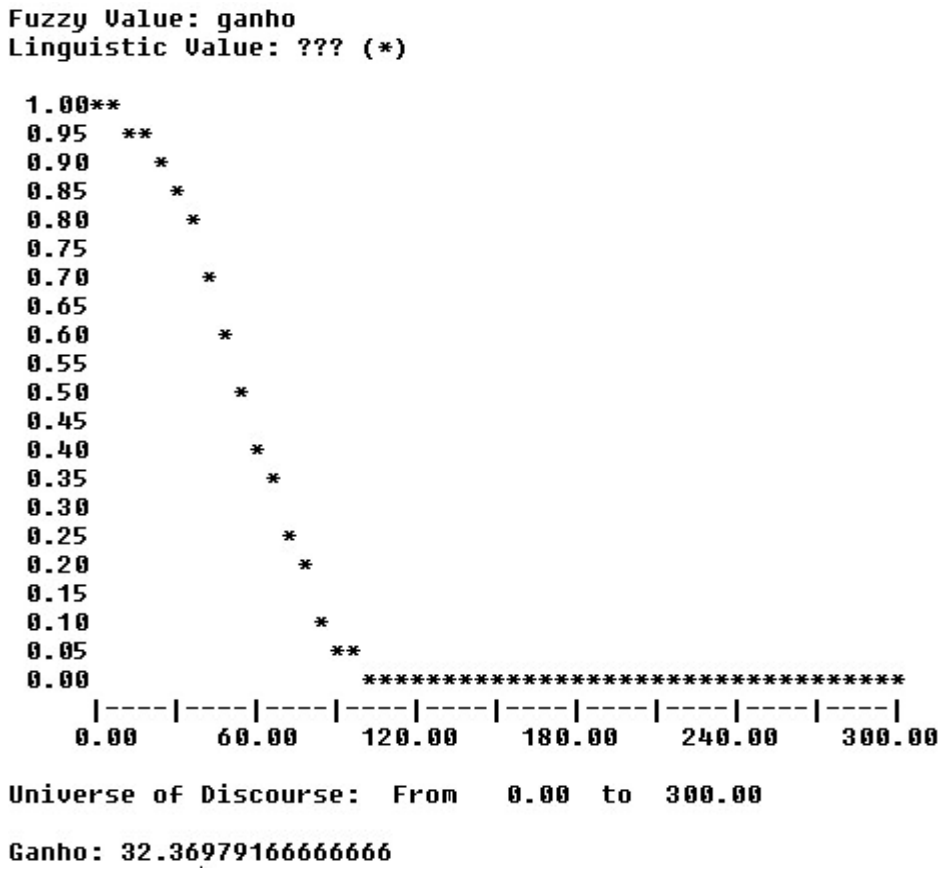

Figura 5. Plotagem do Deffacts – 2. Ganho de uma produção muito baixa em muito tempo.

3. Conclusão

Nos testes apresentados verificamos que implementação de um Sistema Especialista Fuzzy em empresa pode ser muito importante. Com regras e templates bem detalhados podemos ter uma ferramenta capaz de emitir diagnósticos precisos sobre a situação operativa. Como vimos, o sistema trabalha com grandes variedades de entradas, vagas e incertas, as quais podem ser traduzidas para valores reais, onde resulta no valor mais preciso possível.

Um sistema como esse poderia ser muito bem aplicado em qualquer área de produção ou até mesmo adaptado para outros fatores do cotidiano.