

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**EVOLSYS: UM AMBIENTE DE CONFIGURAÇÃO E
ANÁLISE DE ALGORITMOS EVOLUTIVOS PARA
SINTONIA DA BASE DE REGRAS *FUZZY* DO
SISTEMA DE CONTROLE DE UM FMS**

MAYKON ROCHA SANTANA

ORIENTADOR: PROF. DR. ORIDES MORANDIN JUNIOR

São Carlos - SP

Dezembro/2015

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**EVOLSYS: UM AMBIENTE DE CONFIGURAÇÃO E
ANÁLISE DE ALGORITMOS EVOLUTIVOS PARA
SINTONIA DA BASE DE REGRAS *FUZZY* DO
SISTEMA DE CONTROLE DE UM FMS**

MAYKON ROCHA SANTANA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Orides Morandin Junior

São Carlos - SP

Dezembro/2015

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

S231e Santana , Maykon Rocha
Evolsys: um ambiente de configuração e análise de algoritmos evolutivos para sintonia da base de regras fuzzy do sistema de controle de um FMS / Maykon Rocha Santana . -- São Carlos : UFSCar, 2016. 212 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2015.

1. Sistema de Manufatura Flexível . 2. Sistemas Fuzzy . 3. Algoritmos Genéticos. 4. Algoritmos Evolutivos. 5. Base de Regras Fuzzy. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Maykon Rocha Santana, realizada em 14/12/2015:

Prof. Dr. Orides Morandin Junior
UFSCar

Profa. Dra. Veronica Oliveira de Carvalho
UNESP

Prof. Dr. Danilo Sipoli Sanches
UTFPR

A todos aqueles que, de alguma forma, me apoiaram nessa conquista.

AGRADECIMENTO

À Deus que foi o guia e o meu principal orientador no desenvolver desse projeto.

À minha família por ter me guiado por esse caminho, em especial, a minha irmã Lorena e aos meus pais, Jário e Hirêne que com garra e esforço me propiciaram toda a educação necessária para que eu chegasse até aqui.

À minha namorada Marcela que muito me apoiou e torceu por mim todos esses anos nunca me deixando desanimar ante qualquer obstáculo.

Ao meu orientador, Prof. Dr. Orides Morandin Junior, pela dedicação, paciência e por todos os ensinamentos nesses anos.

À Marinete e ao Carlos, pais do Melo, a Dorotí, mãe da Marcela, ao Zé Luís, tio da Marcela e ao Gilberto e a Edna, pais da Jeh, por me aturar nos meus dias de folga.

Ao pessoal do laboratório, Rafael Sanchez, Vinícius Caridá, Rogers Cristo e Monique Viana que me propiciaram discussões enriquecedoras nesses anos.

Ao Seu Ademir, que me conduziu todos os dias para o DC – UFSCar. Ele, com sua simpatia habitual, fez os meus dias de trabalho começarem sempre alegres.

E não posso esquecer do pessoal que tentou me atrapalhar bastante: Jéssica Poellnitz (Jeh), Gustavo Sales Barbosa (Filô, Filord, Filong, Filurso, Filoturno, Filord do Amor, Filô Pão, FilôCaré, ...), Carol Raimundo (Carol), Diogo Melo (Melo), Richard Valefuogo (Rich), Marcela Valefuogo, Caio Evaristo (Caião), Lucas Zago (Bixão), Alan Henrique (Baianão), Tamyris Marconi, Danilo Vargas (Cureza), Wellington Puerta (Punho), André Oliveira, Roger, Nayara (Nay), Anderson (Mr. Anderson), Denise (Dêda) e tantos outros. Na verdade, não ajudaram em nada nesse trabalho, mas o Rich disse que seria injusto não mencioná-los já que fizeram meus dias mais felizes durante esses anos.

Finalmente, aos professores Dr. Danilo Sanches e Dra. Verônica Carvalho que aceitaram participar dessa banca.

Meu muito obrigado a todos.

"Filô, eu vou ser o seu braço direito".

Diogo Melo

RESUMO

Nos últimos anos, empresas tem usado técnicas de Inteligência Artificial (AI) para auxiliar o processo de tomada de decisão em sistemas de manufatura. O uso dessas técnicas possibilita o aumento do desempenho dos Sistemas Flexíveis de Manufatura (FMS), uma vez que a automatização do processo com o uso de recursos computacionais permite uma análise mais profunda das condições do sistema o que, por vezes, resulta em uma melhor tomada de decisão. Neste sentido, a Lógica *Fuzzy* vem sendo usada para realizar essa tarefa, pois ela tem a característica de lidar facilmente com informações imprecisas, codificando o conhecimento do especialista nas chamadas Regras *Fuzzy*. Entretanto, à medida que a complexidade do sistema aumenta, a tarefa de gerar uma Base de Regras *Fuzzy* (FRB) adequada ao sistema proposto se torna cada vez mais difícil. Para auxiliar esse processo de geração da FRB, várias técnicas podem ser usadas e dentre elas destaca-se a técnica de busca denominada Algoritmo Evolutivo (EA). O EA pode ser usado, por exemplo, para a sintonia da Base de Regras *Fuzzy* do Sistema de Controle de um FMS por intermédio da redução de valores de variáveis de otimização como *Makespan* ou *Tardiness*. No caso da variável denominada *Makespan*, a sintonia ocorre quando o EA gera uma FRB que reduz os valores do *makespan* do FMS em questão. Entretanto, a construção do EA que efetivamente gera uma FRB sintonizada para um FMS não é trivial, pois é necessário que haja, nesse processo, a construção de vários tipos de EA com métodos de seleção diferentes, taxas de cruzamento e mutação diferentes dentre outras configurações, até que se encontre o EA adequado à uma dada situação. Sendo assim, no presente trabalho, o objetivo é a construção de um ambiente de configuração e análise de desempenho de EAs para sintonia da FRB do Sistema de Controle de um FMS, ou seja, pretende-se investigar qual o cenário de parâmetros ideal do EA usado na sintonia da FRB do referido sistema de controle. No presente trabalho, o EA usado foi uma extensão do Algoritmo Genético (GA). Para implementação da proposta, um Sistema Evolutivo para configuração e análise dessa variante do GA foi criado. Nesse sistema, intitulado “*EvolSys - Evolutionary System*”, parâmetros dos sistema como Número de Variáveis de Entrada da FRB, Número de Variáveis de Saída da FRB, Tamanho da População, Taxa de Mutação e Taxa de Cruzamento do EA, dentre outros são configurados e, por consequência, uma FRB é gerada. Com isso, há a possibilidade da análise do EA para a escolha de uma FRB que venha propiciar a redução do *makespan* em FMSs. Portanto, é possível concluir, a partir desse trabalho, que o uso de EAs em colaboração com os sistemas *Fuzzy* pode vir a se tornar uma importante ferramenta para sintonia da Base de Regras do sistema responsável pelo sequenciamento das operações de um FMS e, nesse sentido, o ambiente criado cumpre a etapa de configuração e análise do desempenho de EAs.

Palavras-chave: Sistema de Manufatura Flexível, Sistemas *Fuzzy*, Algoritmos Genéticos, Algoritmos Evolutivos, Base de Regras *Fuzzy*, Sistema de Controle *Fuzzy*, FMS, GA, EA

ABSTRACT

In recent years, companies have used Artificial Intelligence (AI) techniques to facilitate the decision-making process in manufacturing systems. The use of these techniques allows increased performance of Flexible Manufacturing System (FMS). The automation of the process using computational resources allows a deeper analysis of the system conditions, which sometimes result in a better decision taking. In this sense, the Fuzzy Logic has been engaged to carry out this task, because it has the characteristic of dealing easily with inaccurate information and encoding knowledge specialist in Fuzzy rules. However, as soon as the system complexity increases, the task of generating a Fuzzy Rule Base (FRB) appropriate to the proposed system becomes increasingly difficult. To assist this process of generation of the FRB, several techniques can be used and among them stand out the search technique called Evolutionary Algorithm (EA). The EA is used, for example, for tuning the FRB of the FMS through the reduction of the optimization variables values as Makespan or Tardiness. In the case of variable called Makespan, the tuning occurs when the EA generates an FRB that reduces the makespan values of a FMS. However, the construction of the EA that effectively generates a tuning FRB is not trivial. It is required to be in the process, the construction of various EA with different selection methods and different mutation rates among other settings until an appropriate EA for a given situation appears. Therefore, in this study we aim to build an environment configuration and performance analysis of EAs in order to define the tuning FRB of the Fuzzy Control System of an FMS, i.e., it is intended to investigate how the EA ideal parameter scenario used for tuning the FRB of the said control system. In this study, the used EA was an extension of Genetic Algorithm (GA). For implementing the proposal, an evolutionary system for configuration and analysis of this variant of the GA was created. In this system, entitled "EvolSys - Evolutionary System" parameters of the system as Number of Input Variables of FRB, Number of Output Variables of FRB, Population Size, Mutation Rate and the EA Crossover Rate, among others are configured and then, one FRB is generated. Using this, there is an EA analysis of the possibility for choosing a FRB that will provide the reduction of makespan in FMS. Consequently, through this study, we may conclude that the use of EAs in collaboration with Fuzzy system may become an important tool for turning the system responsibility to the sequences of an FMS operation. Accordingly, the environment created meets the configuration step and analysis of EAs.

Keywords: Flexible Manufacturing System, Fuzzy Systems, Genetic Algorithms, Evolutionary Algorithms, Fuzzy Rule Bases, Fuzzy Control Systems, FMS, GA, EA.

LISTA DE FIGURAS

Figura 2-1 - Função de Pertinência Triangular <i>Fuzzy</i>	35
Figura 2-2 - Função de Pertinência Trapezoidal <i>Fuzzy</i>	36
Figura 2-3 - Gráfico <i>Fuzzy</i> das funções de pertinência A e B.....	36
Figura 2-4 - Gráfico da União dos Conjuntos A e B.....	37
Figura 2-5 - Gráfico da Intersecção dos Conjuntos A e B.....	37
Figura 2-6 - Gráfico do Complemento do Conjunto A ('A)	37
Figura 2-7 - Gráfico do Complemento do Conjunto B ('B).....	38
Figura 2-8 - Representação do Sistema Inferência <i>Fuzzy</i> Mamdani	41
Figura 2-9 - Estrutura de um GA (REZENDE, 2005).....	47
Figura 2-10 - Codificação Binária para uma população com 8 indivíduos - Adaptado de Coello, Lamont e Veldhuizen (2007)	48
Figura 2-11 - Codificação Real para uma população com 4 indivíduos - Adaptado de Coello, Lamont e Veldhuizen (2007)	49
Figura 2-12 - Método da Roleta para os indivíduos da Tabela 2-3	51
Figura 2-13 - Algoritmo Método da Roleta – Adaptado de Mitchell (1998)	52
Figura 2-14 - Exemplo de seleção de indivíduos pelo Método do Torneio (LINDEN, 2012).....	53
Figura 2-15 - Algoritmo para selecionar os indivíduos para o torneio por meio de uma probabilidade previamente definida – Adaptado de Mitchell (1998).....	54
Figura 2-16 - Exemplo de pontos de corte - Adaptado de (LINDEN, 2012)	56
Figura 2-17 - Operador de Cruzamento com 1 ponto de corte	56
Figura 2-18 - Operador de Cruzamento com 2 pontos de corte	57
Figura 2-19 - Exemplo da operação de Mutação.....	58
Figura 2-20 - Operador de Cruzamento e Mutação - Adaptado de Linden (2012)	59
Figura 3-1 - Cronologia dos projetos voltados para o controle de sistemas de manufatura desenvolvidos pelo Tear	72
Figura 3-2 - Rede de Petri com destaque nos pontos onde podem ocorrer conflitos. ..	73
Figura 3-3 - Rede de Petri com os “Places de Controle”, em vermelho, adicionados. 74	
Figura 3-4 - Perspectiva do modelo com visão para o Sistema de Sequenciamento de Operações	74
Figura 4-1 - Diagrama de Blocos do FMS (modelo proposto por Reis et al. (2013))..	80

Figura 4-2 - Diagrama de Blocos do FMS Simulado	81
Figura 4-3 - Sistema de Inferência <i>Fuzzy</i> Mamdani - Destaque para a Base de Regras	83
Figura 4-4 - Detalhando a Base de Regras <i>Fuzzy</i> - Em destaque o ponto de atuação do projeto (saída das regras).....	83
Figura 4-5 - Exemplo de modelagem dos indivíduos do EA para o exemplo da Figura 4-4.....	84
Figura 4-6 - Etapas do Sistema Evolutivo	85
Figura 4-7 - Configuração do cromossomo do Exemplo 1 para 3 variáveis de entrada (Variável 1 => 5 Valores Possíveis; Variável 2 => 3 Valores Possíveis; Variável 3 => 2 Valores Possíveis) e 2 variáveis de saída (Variável 1 => 3 Valores Possíveis; Variável 2 => 5 Valores Possíveis).....	87
Figura 4-8 - FRB Ideal (Cromossomo do Indivíduo que leva ao valor de <i>fitness</i> máximo)	89
Figura 4-9 - Cálculo do <i>Fitness</i> de 3 indivíduos para uma dada FRB Ideal.	90
Figura 4-10 - Seleção dos possíveis progenitores para realização dos cruzamentos ...	92
Figura 4-11 - Cortes no cromossomo dos progenitores para geração de um indivíduo da nova população.....	95
Figura 4-12 - Descrição do processo de mutação quando um indivíduo selecionado sofre mutação em 2 genes.....	96
Figura 5-1 - Fases do UP adaptadas para o presente projeto.....	99
Figura 5-2 - Diagrama de Caso de Uso do Sistema de Controle de FMS	100
Figura 5-3 - Diagrama de Caso de Uso do Sistema Evolutivo.....	100
Figura 5-4 - Diagrama de Caso de Uso do Algoritmo Evolutivo.....	102
Figura 5-5 – Diagrama de Caso de Uso do Cálculo do <i>Fitness</i>	104
Figura 5-6 - Diagrama de Classes do Sistema Evolutivo	105
Figura 5-7 - Diagrama de Sequência do Algoritmo Evolutivo.....	106
Figura 5-8 - Protótipo de tela principal sobreposta sobre a tela que leva à janela de configuração de parâmetros e execução do Sistema Evolutivo.....	107
Figura 5-9 - Protótipo da tela de configuração de parâmetros	108
Figura 5-10 - Diagrama Entidade-Relacionamento do <i>EvolSys</i>	110
Figura 5-11 – Tela do Menu Principal do <i>EvolSys</i>	111
Figura 5-12 - Tela de Configuração Parâmetros do <i>EvolSys</i>	112
Figura 5-13 - Gráfico dos Resultados da Execução do EA	114

Figura 6-1 - Parte da tela referente à execução do teste com entradas corretas para avaliação do método GerarPopulacaoInicial	121
Figura 6-2 - Parte da tela referente à execução do teste com entradas incorretas para avaliação do método GerarPopulacaoInicial	122
Figura 6-3 - Parte da tela referente à entradas utilizadas para avaliação do método GerarPopulacaoInicial	122
Figura 6-4 - Configurações inseridas no banco de dados para a tabela Parâmetros...	126
Figura 6-5 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 9 com 9 valores possíveis para cada gene.....	128
Figura 6-6 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 27 com 9 valores possíveis para cada gene.	128
Figura 6-7 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 81 com 9 valores possíveis para cada gene.	128
Figura 6-8 - Gráfico do Melhor Indivíduo, Média da População, Pior Indivíduo, Variância da População e Desvio Padrão da População.....	130
Figura 8-1 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-1 - Execução Padrão (Código 01)	144
Figura 8-2 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-2 - Regras <i>Fuzzy</i> (Código 02)	145
Figura 8-3 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-2 - Regras <i>Fuzzy</i> (Código 03)	146
Figura 8-4 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-2 - Regras <i>Fuzzy</i> (Código 04)	146
Figura 8-5 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras <i>Fuzzy</i> (Código 05)	148
Figura 8-6 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras <i>Fuzzy</i> (Código 06)	149
Figura 8-7 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras <i>Fuzzy</i> (Código 07)	149
Figura 8-8 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras <i>Fuzzy</i> (Código 08)	151
Figura 8-9 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras <i>Fuzzy</i> (Código 09)	152

Figura 8-10 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras <i>Fuzzy</i> (Código 10)	152
Figura 8-11 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras <i>Fuzzy</i> (Código 11)	154
Figura 8-12 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras <i>Fuzzy</i> (Código 12)	155
Figura 8-13 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras <i>Fuzzy</i> (Código 13)	155
Figura 8-14 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 14).....	157
Figura 8-15 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 15).....	158
Figura 8-16 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 21).....	158
Figura 8-17 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-7 - Cotas (Código 22).....	160
Figura 8-18 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-8 - Cotas (Código 23).....	162
Figura 8-19 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-9 - Cotas (Código 24).....	164
Figura 8-20 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 25).....	165
Figura 8-21 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 26).....	166
Figura 8-22 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 27).....	166
Figura 8-23 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 28).....	168
Figura 8-24 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 29).....	169
Figura 8-25 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 30).....	169
Figura 8-26 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 31).....	171

Figura 8-27 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 32).....	172
Figura 8-28 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 33).....	172
Figura 8-29 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 34).....	174
Figura 8-30 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 35).....	175
Figura 8-31 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 36).....	175
Figura 8-32 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 38).....	177
Figura 8-33 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 39).....	178
Figura 8-34 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 40).....	178
Figura 8-35 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 41).....	180
Figura 8-36 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 42).....	181
Figura 8-37 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 43).....	181
Figura 8-38 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-16 - Grupos de Progenitores (Código 44).....	183
Figura 8-39 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-16 - Grupos de Progenitores (Código 45).....	184
Figura 8-40 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 46).....	185
Figura 8-41 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 47).....	186
Figura 8-42 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 48).....	186
Figura 8-43 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 49).....	188

Figura 8-44 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 50).....	189
Figura 8-45 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 51).....	189
Figura 8-46 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 52).....	190
Figura 8-47 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 53).....	191
Figura 8-48 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 54).....	192
Figura 8-49 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 55).....	192
Figura 8-50 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 56).....	193
Figura 8-51 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 57).....	193
Figura 8-52 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-20 - Seleção de Progenitores (Código 58).....	195
Figura 8-53 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-20 - Seleção de Progenitores (Código 59).....	196
Figura 8-54 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 60).....	197
Figura 8-55 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 61).....	198
Figura 8-56 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 62).....	198
Figura 8-57 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-22 - Cruzamentos (Código 63).....	200
Figura 8-58 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-22 - Cruzamentos (Código 64).....	201
Figura 8-59 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-23 - Cruzamentos (Código 65).....	202
Figura 8-60 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-23 - Cruzamentos (Código 66).....	203

Figura 8-61 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 67).....	204
Figura 8-62 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 68).....	205
Figura 8-63 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 69).....	205
Figura 8-64 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 70).....	206
Figura 8-65 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 71).....	207
Figura 8-66 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 72).....	208
Figura 8-67 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 73).....	208
Figura 8-68 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 74).....	209
Figura 8-69 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 75).....	209
Figura 8-70 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 76).....	210
Figura 8-71 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 77).....	211
Figura 8-72 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 78).....	212
Figura 8-73 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 79).....	212

LISTA DE TABELAS

Tabela 2-1 - Propriedades Algébricas dos Conjuntos <i>Fuzzy</i>	38
Tabela 2-2 - Sistema Natural e GAs –Adaptado de Pacheco (1999) e Linden (2012).	45
Tabela 2-3 - Exemplo de seleção de indivíduos pelo Método da Roleta	50
Tabela 4-1 - Configuração das variáveis de entrada para o Exemplo 1.	86
Tabela 4-2 - Configuração das variáveis de saída para o Exemplo 1.....	87
Tabela 4-3 - Relação entre as variáveis de saída e seus respectivos valores linguísticos nas saídas das regras <i>Fuzzy</i>	88
Tabela 5-1 - Parâmetros a serem configurados quando da execução do <i>EvolSys</i>	113
Tabela 6-1 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para os testes de unidades	119
Tabela 6-2 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Carga	119
Tabela 6-3 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Estresse	119
Tabela 6-4 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Banco de Dados	120
Tabela 6-5 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para os Testes de Validação	120
Tabela 6-6 - Execuções do teste de carga.....	123
Tabela 6-7 - Execuções do teste de carga.....	125
Tabela 6-8 - Configurações com espaços de busca variados - Cromossomos de tamanho 9 (Código 02), 27 (Código 03) e 81 (Código 4) para uma gramática de tamanho 9.....	127
Tabela 8-1 - Execução Padrão	144
Tabela 8-2 - Regras <i>Fuzzy</i> - Quantidade de Variáveis de Entrada	145
Tabela 8-3 - Regras <i>Fuzzy</i> - Quantidade de Valores Possíveis para Cada Variável de Entrada.....	148
Tabela 8-4 - Regras <i>Fuzzy</i> - Quantidade de Variáveis de Saída.....	151
Tabela 8-5 - Regras <i>Fuzzy</i> - Quantidade de Valores Possíveis para Cada Variável de Saída	154
Tabela 8-6 - População - Quantidade de Indivíduos	157
Tabela 8-7 - Cotas - 100% de Elitismo	160

Tabela 8-8 - Cotas - 100% de Cruzamento.....	162
Tabela 8-9 - Cotas - 100% de Mutação	164
Tabela 8-10 - Cotas - Elitismo e Cruzamento	165
Tabela 8-11 - Cotas - Elitismo e Mutação.....	168
Tabela 8-12 - Cotas - Cruzamento e Elitismo	171
Tabela 8-13 - Cotas - Cruzamento e Mutação.....	174
Tabela 8-14 - Cotas - Mutação e Elitismo.....	177
Tabela 8-15 - Cotas - Mutação e Cruzamento.....	180
Tabela 8-16 - Grupos de Progenitores - 2 Grupos.....	183
Tabela 8-17 - Grupos de Progenitores - 3 Grupos.....	185
Tabela 8-18 - Grupos de Progenitores - 4 Grupos.....	188
Tabela 8-19 - Tabela referente aos Grupos de Progenitores - 5 Grupos	191
Tabela 8-20 - Seleção de Progenitores - Métodos de Seleção.....	195
Tabela 8-21 - Seleção de Progenitores - Taxa de Seleção.....	197
Tabela 8-22 - Cruzamentos - Quantidade de Progenitores por Cruzamentos	200
Tabela 8-23 - Cruzamentos - Quantidade de Pontos de Corte	202
Tabela 8-24 - Cruzamentos - Quantidade de Progenitores e Quantidade de Pontos de Corte por Cruzamentos.....	204
Tabela 8-25 - Mutação - Taxa de Mutação.....	207
Tabela 8-26 - Execução - Número Máximo de Gerações	211

LISTA DE ABREVIATURAS E SIGLAS

AI	Inteligência Artificial (do inglês: <i>Artificial Intelligence</i>)
AGV	Veículo Alto Guiado (do inglês: <i>Automated Guided Vehicle</i>)
AMS	Sistema de Manufatura Automatizado (do inglês: <i>Automated Manufacturing System</i>)
ANN	Rede Neural Artificial (do inglês: <i>Artificial Neural Network</i>)
PLC	Controlador Lógico Programável (do inglês: <i>Programmable logic controller</i>)
CPN	Rede de Petri Colorida (do inglês: <i>Coloured Petri Net</i>)
DB	Base de Dados (do inglês: <i>Data Base</i>)
EA	Algoritmo Evolutivo (do inglês: <i>Evolutionary Algorithm</i>)
EDD	Menor Data de Entrega (do inglês: <i>Earliest Due Date</i>)
EDP	Problema de Despacho Econômico (do inglês: <i>Economic Dispatch Problem</i>)
FCFS	Primeiro a Chegar, Primeiro a ser Atendido (do inglês: <i>First Come First Served</i>)
FIS	Sistema de Inferência <i>Fuzzy</i> (do inglês: <i>Fuzzy Inference System</i>)
FMS	Sistema Flexível de Manufatura (do inglês: <i>Flexible Manufacturing System</i>)
FRB	Base de Regras <i>Fuzzy</i> (do inglês: <i>Fuzzy Rule Base</i>)
FRBS	Sistema <i>Fuzzy</i> Baseado em Regras (do inglês: <i>Fuzzy Rule-Based Systems</i>)
FS	Sistema <i>Fuzzy</i> (do inglês: <i>Fuzzy System</i>)
GA	Algoritmo Genético (do inglês: <i>Genetic Algorithm</i>)
IM	Mecanismo de Inferência (do inglês: <i>Inference Mechanism</i>)
KB	Base de Conhecimento (do inglês: <i>Knowledge Based</i>)
KBGA	Algoritmo Genético com Base de Conhecimento (do inglês: <i>Knowledge Based Genetic Algorithm</i>)
OSS	Sistema de Sequenciamento de Operações (do inglês: <i>Operation Sequence System</i>)
PN	Rede de Petri (do inglês: <i>Petri Net</i>)
RB	Base de Regra (do inglês: <i>Rule Base</i>)
SPT	Menor tempo de processamento (do inglês: <i>Shortest Processing Time</i>)
SQL	Linguagem de Consulta Estruturada (do inglês: <i>Structured Query Language</i>)
UML	Linguagem de Modelagem Unificada (do inglês: <i>Unified Modeling Language</i>)
UP	Processo Unificado (do inglês: <i>Unified Process</i>)

VPN

Redes de Petri Virtuais (do inglês: *Virtual Petri Net*)

SUMÁRIO

1	INTRODUÇÃO	24
1.1	CONTEXTUALIZAÇÃO.....	24
1.2	MOTIVAÇÃO.....	26
1.3	JUSTIFICATIVA.....	27
1.4	OBJETIVO.....	27
1.5	DELIMITAÇÕES DO TRABALHO.....	28
1.6	ORGANIZAÇÃO DO TRABALHO.....	28
2	FUNDAMENTAÇÃO TEÓRICA	30
2.1	CONSIDERAÇÕES INICIAIS.....	30
2.2	CONCEITOS FUNDAMENTAIS.....	31
2.3	LÓGICA FUZZY.....	32
2.3.1	<i>Histórico da Lógica Fuzzy</i>	32
2.3.2	<i>Conjuntos Fuzzy</i>	34
2.3.3	<i>Funções de Pertinência</i>	34
2.3.4	<i>Operações em Conjuntos Fuzzy</i>	36
2.3.5	<i>Variáveis Linguísticas</i>	38
2.3.6	<i>Proposições e Regras em Conjuntos Fuzzy</i>	39
2.3.7	<i>Sistema Fuzzy</i>	40
2.4	ALGORITMOS GENÉTICOS.....	42
2.4.1	<i>Histórico dos Algoritmos Genéticos</i>	43
2.4.2	<i>Terminologia</i>	44
2.4.3	<i>Características Gerais dos Algoritmos Genéticos</i>	45
2.4.4	<i>Codificação</i>	47
2.4.5	<i>Operador Evolutivo de Seleção</i>	50
2.4.6	<i>Operador Evolutivo de Elitismo</i>	54
2.4.7	<i>Operador Evolutivo de Cruzamento</i>	55
2.4.8	<i>Operador Evolutivo de Mutação</i>	57
2.4.9	<i>Parâmetros Evolutivos</i>	59
2.5	CONSIDERAÇÕES FINAIS.....	61
3	ARTIGOS RELACIONADOS	62

3.1	CONSIDERAÇÕES INICIAIS.....	62
3.2	LÓGICA <i>FUZZY</i> E ALGORITMOS GENÉTICOS NA RESOLUÇÃO DOS MAIS DIVERSOS PROBLEMAS.....	62
3.3	ALGORITMOS GENÉTICOS PARA GERAÇÃO DA BASE DE REGRAS DE UM SISTEMA <i>FUZZY</i>	66
3.4	RESOLUÇÃO DE PROBLEMAS DE CONTROLE EM SISTEMAS DE MANUFATURA	68
3.5	LÓGICA <i>FUZZY</i> E ALGORITMOS GENÉTICOS PARA CONTROLE DE FMS	69
3.6	PESQUISAS NO LABORATÓRIO TEAR – DEPARTAMENTO DE COMPUTAÇÃO – UFSCAR	71
3.7	CONSIDERAÇÕES FINAIS	77
4	PROPOSTA DO TRABALHO	79
4.1	CONSIDERAÇÕES INICIAIS.....	79
4.2	PROPOSTA DO TRABALHO.....	79
4.2.1	<i>Visão Global do Projeto de Controle de Sistema de Manufatura Flexível</i>	<i>80</i>
4.2.2	<i>Proposta de Sistema Evolutivo</i>	<i>82</i>
4.2.3	<i>Definição dos Genótipos dos Indivíduos</i>	<i>85</i>
4.2.4	<i>Cálculo do Fitness</i>	<i>88</i>
4.2.5	<i>Operadores Evolutivos de Elitismo, Seleção, Cruzamento e Mutação</i>	<i>91</i>
4.3	CONSIDERAÇÕES FINAIS	97
5	IMPLEMENTAÇÃO DA PROPOSTA	98
5.1	CONSIDERAÇÕES INICIAIS.....	98
5.2	PROJETO DO SISTEMA EVOLUTIVO.....	98
5.3	FERRAMENTAS	108
5.4	BANCO DE DADOS DO PROJETO	109
5.5	SISTEMA EVOLUTIVO - <i>EVOLSYS</i>	111
5.6	CONSIDERAÇÕES FINAIS	114
6	TESTES.....	116
6.1	CONSIDERAÇÕES INICIAIS.....	116
6.2	PLANEJAMENTO.....	116
6.2.1	<i>Recursos</i>	<i>117</i>
6.2.2	<i>Escopo</i>	<i>117</i>
6.2.2.1	<i>A Fase de Teste de Unidade.....</i>	<i>117</i>
6.2.2.2	<i>A Fase de Teste de Sistema</i>	<i>118</i>
6.2.3	<i>Estratégia de Testes</i>	<i>118</i>
6.2.3.1	<i>Estratégia para o Teste de Unidade.....</i>	<i>118</i>
6.2.3.2	<i>Estratégia para o Teste de Sistema.....</i>	<i>119</i>
6.3	TESTES REALIZADOS	120
6.3.1	<i>Execução dos Testes de Unidade.....</i>	<i>120</i>
6.3.2	<i>Execução dos Testes de Sistema.....</i>	<i>122</i>

6.3.2.1	Execução dos Testes de Carga	122
6.3.2.2	Execução dos Testes de Estresse	124
6.3.2.3	Execução dos Testes de Banco de Dados.....	126
6.3.2.4	Execução dos Testes de Validação.....	126
6.4	CONSIDERAÇÕES FINAIS	130
7	CONCLUSÕES E TRABALHOS FUTUROS	132
7.1	CONCLUSÕES	132
7.2	TRABALHOS FUTUROS.....	133
	REFERÊNCIAS.....	135
	APÊNDICE A.....	143
8	TESTES DE VALIDAÇÃO	143
8.1	CONSIDERAÇÕES INICIAIS.....	143
8.2	EXECUÇÃO PADRÃO	144
8.3	REGRAS <i>FUZZY</i> - QUANTIDADE DE VARIÁVEIS DE ENTRADA.....	145
8.4	REGRAS <i>FUZZY</i> - QUANTIDADE DE VALORES POSSÍVEIS PARA CADA VARIÁVEL DE ENTRADA	148
8.5	REGRAS <i>FUZZY</i> - QUANTIDADE DE VARIÁVEIS DE SAÍDA	151
8.6	REGRAS <i>FUZZY</i> - QUANTIDADE DE VALORES POSSÍVEIS PARA CADA VARIÁVEL DE SAÍDA.....	154
8.7	POPULAÇÃO - QUANTIDADE DE INDIVÍDUOS.....	157
8.8	COTAS - 100% DE ELITISMO	160
8.9	COTAS – 100% DE CRUZAMENTO.....	162
8.10	COTAS - 100% DE MUTAÇÃO.....	164
8.11	COTAS – ELITISMO E CRUZAMENTO	165
8.12	COTAS – ELITISMO E MUTAÇÃO.....	168
8.13	COTAS - CRUZAMENTO E ELITISMO	171
8.14	COTAS - CRUZAMENTO E MUTAÇÃO	174
8.15	COTAS - MUTAÇÃO E ELITISMO.....	177
8.16	COTAS - MUTAÇÃO E CRUZAMENTO	180
8.17	GRUPOS DE PROGENITORES - 2 GRUPOS.....	183
8.18	GRUPOS DE PROGENITORES - 3 GRUPOS.....	185
8.19	GRUPOS DE PROGENITORES - 4 GRUPOS.....	188
8.20	GRUPOS DE PROGENITORES - 5 GRUPOS.....	191
8.21	SELEÇÃO - MÉTODOS DE SELEÇÃO	195
8.22	SELEÇÃO - TAXA DE SELEÇÃO	197
8.23	CRUZAMENTOS - QUANTIDADE DE PROGENITORES POR CRUZAMENTOS.....	200
8.24	CRUZAMENTOS - QUANTIDADE DE PONTOS DE CORTE.....	202

8.25	CRUZAMENTOS - QUANTIDADE DE PROGENITORES E DE PONTOS DE CORTE POR CRUZAMENTO.....	204
8.26	MUTAÇÃO - TAXA DE MUTAÇÃO.....	207
8.27	EXECUÇÃO - NÚMERO MÁXIMO DE GERAÇÕES.....	211

Capítulo 1

INTRODUÇÃO

1.1 Contextualização

Os Sistemas Flexíveis de Manufatura têm por características envolver situações relacionadas a eventos discretos, compartilhamento de recursos, concorrência, flexibilidade de roteamento, etc. (LI; WU; RONG, 2014). Essas características introduziram, por consequência, novas dificuldades ao processo fabril e, devido a isso, surgiu a necessidade de implantar meios de evitar ou tratar tais ocorrências.

Nesse sentido, algumas questões relacionadas a *deadlocks* e conflitos vêm sendo estudadas quando surgem situações onde há a necessidade de tomada de decisão, competição ou compartilhamento de recursos como pode ser visto nos trabalhos de Reis et al. (2013) e Yufeng, Zhiwu e Al-Ahmari (2013). Entretanto, para resolver essas questões, a modelagem e o controle dos FMSs tem de ser feitos de forma adequada a fim de propiciar um maior desempenho do sistema.

Especificamente no que diz respeito à questão de conflitos, quando o mesmo ocorre em um sistema de manufatura, uma decisão tem de ser tomada a fim de resolvê-lo. Essa decisão está ligada ao processo de controle do sistema de manufatura e tem importante papel na gerência e na qualidade do sistema uma vez que o desempenho do FMS pode ser afetado de acordo com a regra de programação da produção escolhida (TUNCEL, 2007).

Técnicas de Inteligência Artificial podem ser usadas para auxiliar o processo de tomada de decisão e, dessa forma, corroborar com o aumento do desempenho do FMS. (CHAN; CHAN; LAU, 2002).

Dentre essas técnicas para tomada de decisão e que, conseqüentemente, propiciam o controle e a resolução de conflitos em um FMS, destaca-se a abordagem que usa a Lógica *Fuzzy* (ANGSANA; PASSINO, 1994; KEUNG; IP; YUEN, 2003; QIN; BAI; RALESCU, 2011; REIS et al., 2013). Essa abordagem lida facilmente com informação incerta e incompleta, e dessa forma, o conhecimento dos especialistas pode ser codificado nas chamadas regras *Fuzzy*. Devido a isso, nos últimos anos, o uso da Lógica *Fuzzy* em programação de FMS aumentou (NANVALA, 2011).

Para tanto, na construção de um sistema *Fuzzy*, a tarefa mais importante é encontrar um conjunto de regras *Fuzzy* adequado a um problema específico (CORDÓN; HERRERA, 2001; CASTRO; CAMARGO, 2004).

Entretanto, na medida em que a complexidade do sistema aumenta, a tarefa de gerar uma Base de Regras *Fuzzy* correspondente a um determinado problema torna-se cada vez mais difícil quando se usa apenas o conhecimento do especialista para isso. Castro e Camargo (2004) afirmam que o especialista, em muitas situações não conhece a teoria *Fuzzy* e o engenheiro, por vezes, não é familiar ao problema que se quer tratar o que dificulta a criação da FRB.

Portanto, devido à complexidade da FRB uma grande quantidade de técnicas automáticas foi proposta para gerar essa base automaticamente. Algoritmo Genético, que é uma classe de Algoritmo Evolutivo, pode ser citado como exemplo de técnica usada nesse processo (NAUCK; KRUSE, 1997; CORDÓN et al., 2001; LIAO; CELMINS; HAMMELL, 2003).

No caso do presente trabalho tem-se a ideia de que, uma extensão do GA desenvolvido por Holland¹, pode ser usada para geração de uma FRB que leva à redução, por exemplo, da variável de otimização denominada *Makespan*², ou seja, à sintonia da FRB de um dado FMS. Neste caso, a sintonia ocorre quando o EA gera uma FRB que reduz os valores de *makespan* de um dado FMS. Entretanto, a construção do EA que de fato gera uma FRB sintonizada para um FMS não é uma tarefa simples, pois, é necessário que haja, nesse processo, a construção de vários tipos de EA com métodos de seleção diferentes, taxas de cruzamento e mutação

¹ Para o presente trabalho, o Algoritmo Evolutivo usado é uma extensão do Algoritmo Genético concebido por John Henry Holland na década de 1960. Neste caso, é acrescentado o Operador de Elitismo e ocorrem adaptações em relação às etapas de seleção, cruzamento e mutação.

² Diferença de tempo entre o início e o fim de uma sequência de trabalhos ou tarefas. Tempo total de conclusão das ordens (LUSTOSA et al., 2008).

diferentes, número de indivíduos e pontos de corte do cromossomo diferentes dentre outras configurações, até que se encontre o EA adequado à situação em questão.

Portanto, o foco do presente trabalho está relacionado à definição dos parâmetros de um Algoritmo Evolutivo para possibilitar a configuração da Base de Regras *Fuzzy* que conduz à resolução dos conflitos em FMS.

1.2 Motivação

Os FMS estão inseridos no contexto industrial como sistemas capazes de reagir a mudanças esperadas ou inesperadas quando se trata da fabricação de algum produto. Nesse sentido, essa flexibilidade traz consigo a necessidade de um controle de manufatura capaz de atender adequadamente às diversas situações que surjam e assim propiciar uma tomada de decisão apropriada de acordo com critérios que venham a ser definidos.

Nesse sentido, o Laboratório de Estratégias de Automação (Tear) do Departamento de Computação da UFSCar desenvolve projetos que visam realizar o controle e, conseqüentemente, as tomadas de decisões em FMSs por intermédio do uso de técnicas de Inteligência Artificial como a Lógica *Fuzzy* e os Algoritmos Genéticos e suas variações.

No projeto de Reis et al. (2013) por exemplo, a Lógica *Fuzzy* é usada para realizar o controle de um FMS por meio da construção da chamada Base de Regras *Fuzzy*. Nesse sentido, trabalhos futuros do laboratório Tear trazem a perspectiva de empregar uma variedade de GA, denominada Algoritmo Evolutivo, na seleção das regras que deverão ser usadas pelo Sistema de Controle *Fuzzy*, a fim de realizar a sintonia da FRB do sistema e, conseqüentemente, aumentar o desempenho de um dado FMS.

No entanto, existe uma dificuldade relacionada à definição do EA que conduz à sintonia da FRB de um FMS, pois, devido a quantidade de parâmetros que podem ser configurados, uma grande quantidade de EAs pode ser construída³, sendo que cada um deles leva a diferentes desempenhos do FMS.

³ Neste caso, considera-se que cada configuração dos parâmetros do EA gera um EA diferente.

Desse modo, nesse trabalho espera-se que, com a definição dos parâmetros do EA, seja possível definir qual o EA que gera a FRB que conduz à resolução dos conflitos em FMSs.

1.3 Justificativa

Em um FMS, os sistemas *Fuzzy* podem ser utilizados para auxiliar o processo de tomada de decisão. No entanto, esses sistemas possuem variáveis que, de acordo como são definidas, podem levar a tomadas de decisão melhores ou piores. Nesse sentido, estudos vêm sendo realizados, a fim de melhorar esses sistemas de controle baseado em Lógica *Fuzzy* usando para tanto, técnicas de Inteligência Artificial como os Algoritmos Genéticos.

Nesse contexto, Yuan e Zhuang (1996), Herrera e Lozano (1999), Cordón e Herrera (2001), Fazzolari et al. (2013) e Abdekhak et al. (2014) desenvolvem projetos relacionados ao uso dos Algoritmos Evolutivos, como os GAs, para possibilitar alternativas que podem levar a uma melhor condição do sistema *Fuzzy*.

Além disso, os trabalhos de Morandin Jr. e Kato (2003), Morandin Jr. et al. (2007), Kato, Morandin Jr. e Sgavioli (2010), Reis (2012) e Reis et al. (2013), desenvolvidos no laboratório Tear, direcionam e formam a base para o desenvolvimento de projetos que usam técnicas, como os EAs, para a geração de FRB que conduzem à resolução de conflitos em FMSs.

1.4 Objetivo

Como objetivo geral pretende-se:

- Construção de um ambiente que permita configurar e analisar o desempenho de Algoritmos Evolutivos⁴ para geração da FRB que conduz à resolução de conflitos em FMS.

⁴ Ressalta-se que para o presente trabalho, o Algoritmo Evolutivo usado é uma extensão do Algoritmo Genético concebido por John Henry Holland na década de 1960. Neste caso, é acrescentado o Operador de Elitismo e ocorrem adaptações em relação às etapas de seleção, cruzamento e mutação.

Como objetivos específicos tem-se:

- Configuração de experimentos variados para testes⁵ de todos os parâmetros configuráveis no sistema.
- Análise do impacto de cada um dos parâmetros configuráveis do EA como o Operador de Cruzamento, Operador de Mutação, Número de Indivíduos, Método de Seleção dentre outros parâmetros.

1.5 Delimitações do Trabalho

Especificamente nesse projeto, a ideia é que o ambiente desenvolvido permita configurar os parâmetros de um EA gerando indivíduos cujos cromossomos representam, cada um, uma FRB. Neste caso, o EA atua na(s) variável(is) de saída das regras *Fuzzy* gerando combinações de bases de regras que serão avaliadas de acordo com o problema em questão.

É importante ressaltar que, neste projeto, o EA não atuará na:

- Definição da quantidade de regras *Fuzzy* a serem usadas (redução ou não no número de regras).
- Definição da quantidade de variáveis usadas nas regras *Fuzzy* (redução ou não no número de variáveis usadas).

1.6 Organização do Trabalho

No Capítulo 2 será apresentada a fundamentação teórica referente à Lógica *Fuzzy* e ao tipo de Algoritmo Evolutivo usado nesse projeto, que é uma extensão do Algoritmo Genético concebido por Holland na década de 1960. Esses tópicos são a base para a composição deste trabalho. Histórico, terminologias, características gerais dos sistemas que podem ser formados com essas técnicas e o Algoritmo Evolutivo desenvolvido para o presente trabalho serão temas abordados.

⁵ Detalhes sobre a fase de testes estão presentes no Capítulo 6 - Testes (página 116).

No Capítulo 3 serão apresentados os artigos relacionados à Lógica *Fuzzy*, aos Algoritmos Genéticos e aos Sistemas de Manufatura nas mais diversas situações. Artigos relacionados a área de controle de sistema de manufatura, em especial, os sistemas de controle que utilizam a Lógica *Fuzzy* na tomada de decisão também serão abordados. Além disso, também serão apresentados trabalhos relacionados ao uso dos GAs como sendo uma técnica para automatizar o processo de geração da base de regras em sistemas *Fuzzy*.

O Capítulo 4 será dedicado à apresentação da proposta deste trabalho onde será descrita a principal contribuição da pesquisa e o passo a passo de como se deu o desenvolvimento do projeto para a construção do ambiente de configuração e análise de EA para sintonia da FRB do Sistema de Controle *Fuzzy* de um FMS.

O Capítulo 5 será dedicado à apresentação da implementação da proposta. As funcionalidades referentes ao software desenvolvido serão apresentadas.

O Capítulo 6 será dedicado à apresentação dos testes realizados no sistema desenvolvido. Os Testes Unitários e de Sistema serão apresentados e os resultados obtidos serão apresentados e analisados. Além disso, serão apresentados os testes de Validação para que seja possível verificar se os requisitos estabelecidos para o sistema resultam no impacto esperado para cada um dos parâmetros que podem ser configurados na construção do Algoritmo Evolutivo.

No Capítulo 7 serão apresentadas as conclusões destacando a abrangência dos resultados alcançados, as limitações do trabalho e as contribuições que o mesmo trás para a área de sistema de controle de manufatura e para os projetos do laboratório Tear do Departamento de Computação da UFSCar.

FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações Iniciais

O capítulo é iniciado com uma introdução aos conceitos fundamentais referentes à resolução de conflitos em FMS. Problemas de *deadlocks*, conflitos e tomada de decisão são apresentados. Então é destacada a questão de tratamento de conflitos em FMS e, por conseguinte, destaca-se a possibilidade do uso de Sistemas *Fuzzy* e dos Algoritmos Genéticos a fim de prover a resolução desses conflitos em sistemas de manufatura flexíveis.

A seguir, duas seções serão apresentadas. A primeira se refere à teoria *Fuzzy* e a segunda se refere aos Algoritmos Genéticos.

Vale ressaltar que, como a estrutura do EA desenvolvido apresenta algumas diferenças em relação aos GAs clássico (acrécimo do Operador de Elitismo e adaptações em relação às etapas de seleção, cruzamento e mutação), achou-se por bem não se referir ao Algoritmo Evolutivo desenvolvido como “Algoritmo Genético”. Entretanto, como a técnica que forma a base para o projeto de construção do ambiente proposto é o próprio GA, serão abordadas as questões referentes a essa classe de Algoritmo Evolutivo.

A compreensão desses tópicos é fundamental para o entendimento e a execução do projeto proposto.

Por fim, seguem as considerações finais abordando uma síntese da fundamentação teórica evidenciando a importância do conhecimento tanto da teoria *Fuzzy* como dos GAs para a execução e entendimento do projeto realizado, destacando qual será o foco do mesmo.

2.2 Conceitos Fundamentais

O uso dos FMSs introduziu por consequência novas dificuldades ao processo fabril como *deadlocks* e situações de conflitos.

Deadlock é uma situação que pode ocorrer em um sistema complexo que inclui sistemas de produção automatizados, sistemas de transporte, sistemas operacionais de computador, sistemas de software concorrentes, redes de computador e sistemas de banco de dados distribuídos (FANTI; ZHOU, 2004). Um sistema pode entrar em uma situação de *deadlock* quando sistemas de processos concorrentes exigem exclusividade no uso sequencial de recursos. Essa situação ocorre quando todos os processos em um conjunto estão à espera de um evento que somente pode ser causado por um outro processo do mesmo conjunto fazendo com que esses processos entrem em uma espera circular (FANTI; MAIONE; TURCHIANO, 2000).

Outro fator que deve ser observado quando do uso de FMS é em relação às situações de conflito. Essas situações, quando ocorrem, necessitam de tratamento especial, o que, em muitas vezes, pode resultar em intervenção no sistema (GOMES, 2005). Em FMS, os conflitos ocorrem quando o sistema tem duas ou mais possibilidades e é preciso escolher uma delas. Neste caso o sistema de controle recebe a informação do conflito do sistema de gerenciamento da produção e decide, baseado em algum critério, qual das possibilidades será a escolhida e assim, os processos prosseguem sem problemas.

Dessa forma, essas duas situações que são inerentes ao uso dos FMSs, demonstram a necessidade de serem desenvolvidas políticas de controle eficazes a fim de garantir que uma tomada de decisão apropriada seja implementada garantindo que *deadlocks* sejam evitados e conflitos sejam resolvidos do modo mais eficiente possível (REIS et al., 2013).

Nesse sentido, a Lógica *Fuzzy* e os Algoritmos Genéticos podem atuar de forma colaborativa para auxiliar a tomada de decisão em FMSs e corroborar para que os conflitos sejam resolvidos e *deadlocks* sejam evitados.

Em particular, o foco desse trabalho está na confecção de um ambiente de configuração e análise de desempenho de Algoritmos Evolutivos, em particular, uma extensão do Algoritmo Genético, para sintonia da FRB do Sistema de Controle de um FMS.

A seguir, serão apresentadas as fundamentações teóricas referentes à Lógica *Fuzzy* e aos Algoritmos Genéticos, teorias essas que são imprescindíveis para entendimento da proposta do presente trabalho.

2.3 Lógica *Fuzzy*

No dia a dia, quando se usa da linguagem natural para transmitir alguma informação, naturalmente são usados alguns dados imprecisos. Podem-se citar como exemplo expressões naturais como “Hoje está quente!” ou “Ele é de estatura média”.

Na resolução de problemas matemáticos e computacionais, por vezes, essa situação também ocorre e é necessário que se utilizem recursos que sejam capazes de lidar com tais indefinições de conhecimento. Todavia, a Lógica Clássica não é adequada nesta situação uma vez que a mesma não lida com ambiguidades e se baseia apenas em conceitos relacionados a falso e verdadeiro (MUKAIDONO, 2001).

Já a Lógica *Fuzzy*, que é uma extensão da Lógica Clássica, suporta o tratamento de valores intermediários que se encontrem entre o falso e o verdadeiro. Dessa forma, segundo Nguyen e Walker (2006), a Lógica *Fuzzy* pode ser usadas em situações onde haja indefinições do problema já que a mesma lida com informação incerta, incompleta, imprecisa ou vaga, além de tratar problemas complexos em que a modelagem matemática é difícil de ser implementada.

O sucesso desse método tem sido demonstrado numa variedade de campos como o controle de sistemas complexos, quando modelos matemáticos são difíceis de especificar e em sistemas especialistas quando regras expressam conhecimento e fatos são naturalmente linguísticos (NGUYEN; WALKER, 2006).

2.3.1 Histórico da Lógica *Fuzzy*

Em 1965, o matemático e cientista da computação nascido no Azerbaijão, Lotfi Zadeh escreveu um artigo intitulado *Fuzzy Sets*, onde descrevia os Conjuntos *Fuzzy* como sendo uma classe de objetos com um contínuo de graus (ZADEH, 1965). Esse artigo foi a gênese da teoria dos conjuntos *Fuzzy*. Essa publicação marcou o início de uma nova direção que, ao longo das

últimas décadas, tem levado a uma vasta literatura e a um número crescente de aplicações que vão desde produtos de consumo a sistemas de apoio à decisão⁶.

Vale ressaltar que os estudos de Zadeh foram baseados na teoria anteriormente desenvolvida por Lukasiewicz, que propôs uma variação da Lógica Booleana onde existissem mais valores além do 0 e do 1. Sendo assim, Lukasiewicz definiu que 1 representaria verdadeiro, 0 representaria falso e $\frac{1}{2}$ para é possível (REZNIK, 1997). Dessa forma, Zadeh desenvolveu a Lógica *Fuzzy* combinando conceitos da Lógica Clássica com a teoria de Lukasiewicz definindo assim graus de pertinência (COSENZA; VILLELA; MORE, 2006).

Por meio desses conhecimentos, realizou-se, em 1974, uma aplicação da Lógica *Fuzzy* pelo professor Ebrahim Mamdani do departamento de Engenharia Elétrica e Eletrônica do *Queen Mary College* – atualmente *Queen Mary University of London*⁷. Ele utilizou a Lógica *Fuzzy* em um projeto e publicou os resultados no artigo “*Application of Fuzzy algorithms for control of simple dynamic plant*” (MAMDANI, 1974).

Neste artigo Mamdani descreve um sistema no qual um algoritmo *Fuzzy* é usado para controlar um motor a vapor construído em laboratório. O algoritmo é implementado como um intérprete para um conjunto de regras expressas em declarações condicionais *Fuzzy*.

Destaca-se que, Mamdani, por intermédio de seus trabalhos, ajudou a difundir a Lógica *Fuzzy* abrindo caminho para que novas aplicações usando tal método fossem desenvolvidas (ARAI; WATANABE, 2012).

Com esses e outros estudos e aplicações da Lógica *Fuzzy*, na década de 1980, o Japão iniciou a aplicação dessa técnica em sistemas industriais. Em 1987, a empresa *Omron* desenvolveu os primeiros controladores *Fuzzy* comerciais e, posteriormente, em 1993, a *Fuji Electric* aplicou a Lógica *Fuzzy* para controlar estações de tratamento de água (GARRIDO, 2012).

Atualmente a Lógica *Fuzzy* é aplicada nas mais diversas áreas como a militar, as redes de computadores, *soft computing*, roteamento de veículos, sistemas industriais entre outras aplicações (SINGH et al., 2013).

⁶ Disponível em: < http://www.ieeeghn.org/wiki/index.php/Lotfi_A._Zadeh > Acessado em 10/07/2014

⁷ Disponível em < <http://www.qmul.ac.uk/about/history/index.html> > Acessado em 08/07/2014

2.3.2 Conjuntos *Fuzzy*

Na Lógica Proposicional ou Clássica um subconjunto A de um conjunto U é determinado pela sua função característica X_A que é definida por:

$$X_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad (2.1).$$

Essa função especifica se um elemento pertence ou não ao subconjunto A (NGUYEN; WALKER, 2006). Ela assume o valor 1 (um) se o elemento pertence ao subconjunto A ou assume o valor 0 (zero) se não pertence. Neste caso, percebe-se que apenas esses dois valores podem ser assumidos pela função característica.

Na Lógica *Fuzzy*, essa noção é expandida no sentido de permitir o tratamento de valores que estejam no intervalo $[0,1]$. Desse modo, Zadeh propôs uma caracterização mais ampla, na medida em que sugere que alguns elementos são mais membros de um conjunto do que outros. O fator de pertinência pode então assumir qualquer valor entre 0 e 1, sendo que o valor 0 indica uma completa exclusão e um valor 1 representa completa pertinência do elemento ao conjunto (GOMIDE; GUDWIN; TANSCHKEIT, 1995).

2.3.3 Funções de Pertinência

Na Lógica *Fuzzy*, para um elemento de um conjunto em particular não é determinado se o mesmo pertence ou não a um certo conjunto. É dito apenas que este elemento pertence a tal conjunto com um determinado grau de pertinência o qual está compreendido num valor entre 0(zero) e 1(um) inclusive (REZNIK, 1997).

Nesse sentido, tem-se as chamadas funções de pertinência que são usadas justamente para determinar o grau de pertinência de cada elemento em relação a um conjunto específico.

Os gráficos gerados por intermédio dessas funções podem ser triangulares, trapezoidais dentre outros.

A função triangular é representada da forma:

$$X_A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b \\ \frac{x-c}{b-c}, & \text{se } b < x \leq c \\ 0, & \text{se } x > c \end{cases} \quad (2.2),$$

para algum $a < b < c$.

Pode-se representar a função triangular por meio de gráficos, como pode ser visto na Figura 2-1.

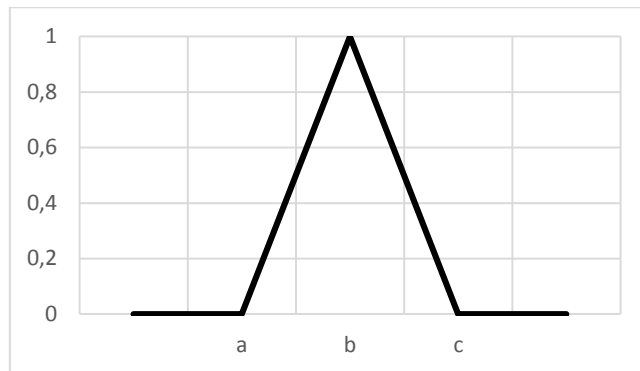


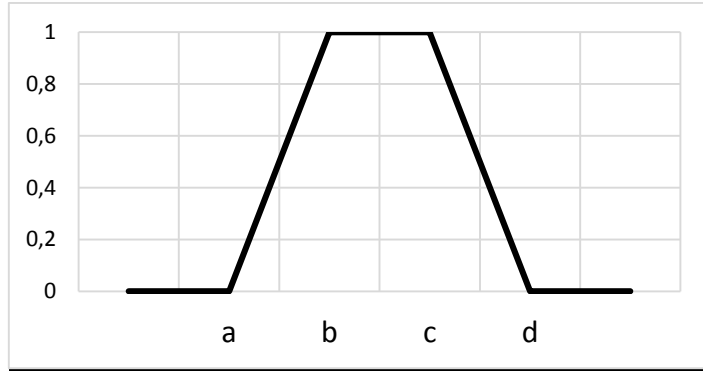
Figura 2-1 - Função de Pertinência Triangular Fuzzy

Já a função trapezoidal é representada como:

$$X_A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b \\ 1, & \text{se } b < x \leq c \\ \frac{x-c}{b-c}, & \text{se } c < x \leq d \\ 0, & \text{se } x > d \end{cases} \quad (2.3),$$

para algum $a < b < c < d$.

Pode-se também representar a função trapezoidal por meio de gráficos, como pode ser visto na Figura 2-2.

Figura 2-2 - Função de Pertinência Trapezoidal *Fuzzy*

2.3.4 Operações em Conjuntos *Fuzzy*

Os conjuntos *Fuzzy* possuem algumas operações que são extensões já existentes nos conjuntos clássicos como, por exemplo, as operações de União, Interseção e Complemento.

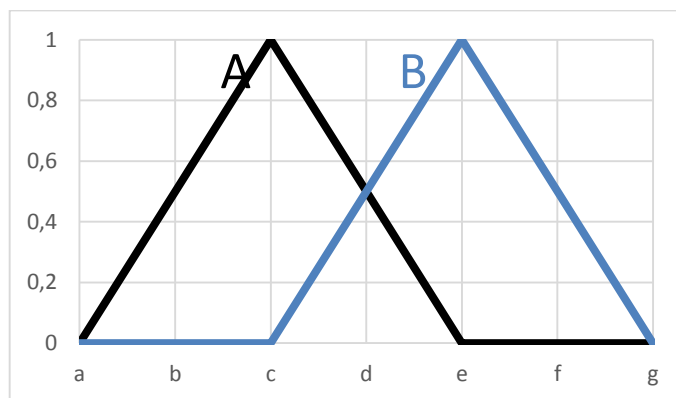
Nguyen e Walker (2006) afirmam que essas operações podem ser representadas por meio de suas funções de pertinência como segue.

$$(A \vee B)(x) = \max\{A(x), B(x)\} = A(x) \vee B(x) \quad \text{União}$$

$$(A \wedge B)(x) = \min\{A(x), B(x)\} = A(x) \wedge B(x) \quad \text{Interseção}$$

$$A'(x) = 1 - A(x) \quad \text{Complemento}$$

Para exemplificar, tem-se os conjuntos A e B, cujas funções de pertinência *Fuzzy* são representadas por intermédio do gráfico da Figura 2-3.

Figura 2-3 - Gráfico *Fuzzy* das funções de pertinência A e B

Sendo A e B dois conjuntos *Fuzzy* em U representados na Figura 3, a união desses conjuntos pode ser vista na Figura 2-4.

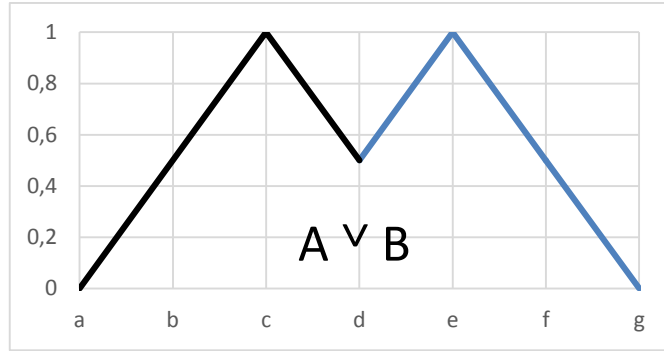


Figura 2-4 - Gráfico da União dos Conjuntos A e B

Considerando os mesmos conjuntos da Figura 2-3, a intersecção pode ser representada da seguinte forma (Figura 2-5):

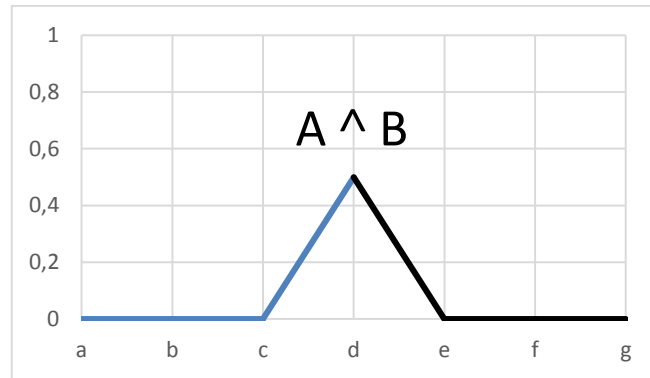


Figura 2-5 - Gráfico da Intersecção dos Conjuntos A e B

Ainda para os mesmos conjuntos da Figura 2-3, os complementos dos respectivos conjuntos podem ser representados das seguintes formas (Figura 2-6 e Figura 2-7):

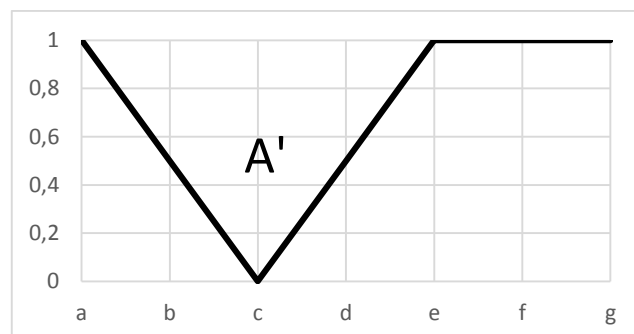


Figura 2-6 - Gráfico do Complemento do Conjunto A (A')

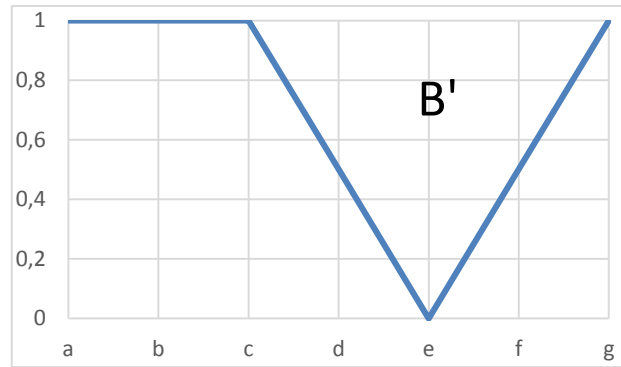


Figura 2-7 - Gráfico do Complemento do Conjunto B ('B)

Pode-se ainda utilizar as definições de união, intersecção e complemento para verificar propriedades algébricas dos conjuntos *Fuzzy* como pode ser visto no artigo de Gomide, Gudwin e Tanscheit (1995). Essas propriedades são apresentadas na Tabela 2-1.

Tabela 2-1 - Propriedades Algébricas dos Conjuntos *Fuzzy*

Involução	$(A')' = A$
Idempotência	$A \cap A = A$
	$A \cup A = A$
Comutatividade	$A \cap B = B \cap A$
	$A \cup B = B \cup A$
Associatividade	$(A \cap B) \cap C = A \cap (B \cap C)$
	$(A \cup B) \cup C = A \cup (B \cup C)$
Distributividade	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Lei Transitiva	Se $A \subset B$ e $B \subset C$ então $A \subset C$
Conjunto Vazio	$A \cap \emptyset = \emptyset$
	$A \cup \emptyset = A$
Conjunto Universo	$A \cap U = A$
	$A \cup U = U$

2.3.5 Variáveis Linguísticas

Zadeh (1973) afirma que a capacidade de sintetizar a informação desempenha um papel essencial na caracterização de fenômenos complexos e que a capacidade do ser humano de resumir as informações encontra a sua manifestação mais acentuada no uso das línguas naturais.

Assim, para ele, cada palavra x numa linguagem natural L pode ser vista como uma descrição resumida de um subconjunto *Fuzzy* $M(x)$ de um universo U , com $M(x)$ que representa o significado de x . Neste sentido, Zadeh afirma que a linguagem como um todo pode ser considerada como um sistema de atribuição de rótulos atômicos (por exemplo *alto*, *médio*, *baixo*) e compostos (por exemplo, *muito alto*, *pouco alto*, *mais ou menos alto*), ou seja, palavras, frases e sentenças para os subconjuntos *Fuzzy* de U .

Segundo Gomide, Gudwin e Tanscheit (1995) a principal função das variáveis linguísticas é fornecer uma maneira sistemática para uma caracterização aproximada de fenômenos complexos ou mal definidos. Em essência, o uso do tipo de descrição linguística empregada por seres humanos, e não de variáveis quantificadas, permite o tratamento de sistemas que são muito complexos para serem analisados por intermédio de mecanismos matemáticos convencionais.

2.3.6 Proposições e Regras em Conjuntos *Fuzzy*

Gomide, Gudwin e Tanscheit (1995) caracterizam a Proposição *Fuzzy* como sendo da forma $(\Pi \text{ é } A)$, onde Π é o nome de uma variável linguística e A é um subconjunto *Fuzzy* definido no universo de discurso U de Π .

Essa proposição *Fuzzy* pode ser combinada por meio de operadores, gerando novas proposições.

Considerando X e Y como variáveis linguísticas e A , B , C e D subconjuntos *Fuzzy* definidos no universo de discurso U de X e Y , pode-se utilizar os operadores lógicos E e OU para gerar novas proposições compostas do tipo:

$$[(X \text{ é } A) \quad E \quad (Y \text{ é } B)] \quad OU \quad [(X \text{ é } C) \quad E \quad (Y \text{ é } D)].$$

As Regras *Fuzzy* são criadas a partir do uso do operador de implicação SE (*antecedente*) $ENTÃO$ (*consequente*), também conhecido como condicional *Fuzzy*, em combinação com proposições *Fuzzy* simples ou compostas. As Regras *Fuzzy* portanto, podem ser escritas no formato:

$$SE \quad (X \text{ é } A) \quad ENTÃO \quad (Y \text{ é } B).$$

Vale ressaltar que o antecedente e o consequente de uma Regra *Fuzzy* podem assumir formatos de proposições compostas, gerando regras do tipo:

SE (X1 é A1) **E** (X2 é A2) **ENTÃO** (Y1 é B1) **OU** (Y2 é B2)

onde X1, X2, Y1 e Y2 são variáveis linguísticas e A1, A2, B1 e B2 são subconjuntos *Fuzzy* definidos no universo de discurso U de X1, X2, Y1 e Y2.

2.3.7 Sistema *Fuzzy*

Sistemas *Fuzzy* dizem respeito a um sistema para representar e processar informação linguística, com mecanismos para lidar com a incerteza e imprecisão (CASTRO; CAMARGO, 2004).

Para o presente trabalho, serão abordados os chamados Sistemas *Fuzzy* Baseados em Regras (FRBS).

Cintra e Camargo (2007) caracterizam os FRBS – também conhecidos como Sistemas de Inferência *Fuzzy* (FIS) – como sistemas compostos por dois componentes principais chamados de Base de Conhecimento (KB) e de Mecanismo de Inferência (IM).

A KB é formada pela Base de Dados (DB) e pela Base de Regras (RB). A DB contém as definições dos conjuntos *Fuzzy* relacionados aos termos linguísticos usados na regra *Fuzzy* e a RB armazena o conjunto de Regras *Fuzzy* que modelam um determinado problema.

Ainda em Cintra e Camargo (2007) o IM é caracterizado como o responsável pela aplicação de um processo de raciocínio que usa inferências para derivar a saída ou conclusão do sistema, baseado nas regras e fatos conhecidos.

Rabêlo et al. (2011) caracterizam os Sistemas de Inferência *Fuzzy* como sistemas baseados em regras de produção que fazem uso das variáveis linguísticas para executar processos de tomada de decisão.

Mamdani (1977) e o Takagi e Sugeno (1985) são os tipos mais comuns de FIS sendo usados em diferentes aplicações (SHLEEG; ELLABIB, 2013).

Sheleeg e Ellabib (2013) diferenciam os FIS Mamdani e Takagi-Sugeno pelo caminho da saída dos Conjuntos *Crisp* (também chamados de convencionais ou clássicos) que é gerada

a partir das entradas *Fuzzy*. O método Mamdani usa a técnica de conversão Escalar-*Fuzzy* de uma saída *Fuzzy*, enquanto o método Takagi-Sugeno usa uma média ponderada para calcular a saída *Crisp*. Assim, Mamdani possui funções de pertinência de saída enquanto Takagi-Sugeno não tem funções de pertinência de saída.

Dessa forma, o FIS do tipo Mamdani tem de converter o resultado do processo de inferência (que é um conjunto *Fuzzy*) em uma saída numérica exata. Já no FIS do tipo Takagi-Sugeno a saída já é um número exato não necessitando de qualquer tipo de conversão ao final do processo.

O método Mamdani, em particular, é amplamente aceito para a captura de conhecimento especializado e permite descrever a experiência de forma mais intuitiva, ou seja, mais próximo do pensamento humano (SHLEEG; ELLABIB, 2013).

O método Mamdani, o qual será empregado neste trabalho, pode ser visto na Figura 2-8.

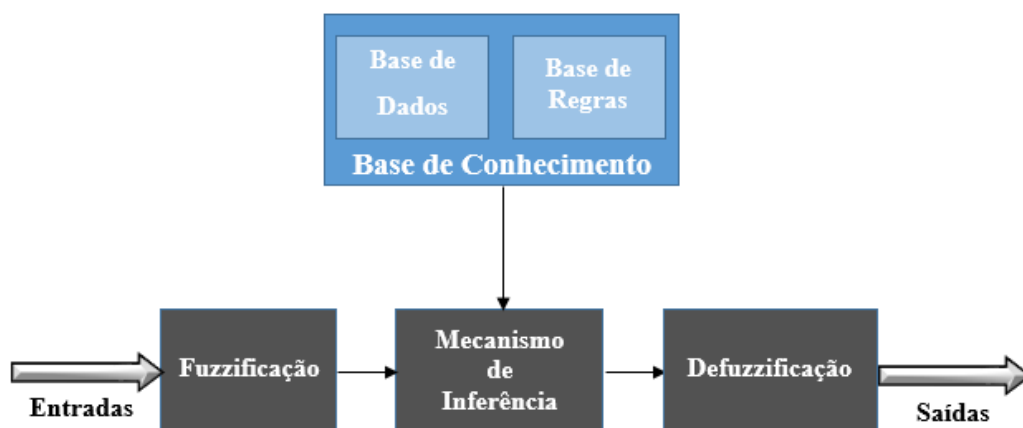


Figura 2-8 - Representação do Sistema Inferência *Fuzzy* Mamdani

Como pode ser visto, o FIS Mamdani possui várias etapas, as quais são responsáveis por realizar as tarefas necessárias para gerar as informações que serão usadas no processo de tomada de decisão.

Na etapa chamada de Fuzzificação, as entradas determinísticas obtidas por intermédio de medições ou observação são transformadas em variáveis *Fuzzy*. Nesta etapa as regras *Fuzzy* são ativadas.

A seguir, o processo de Inferência é executado e é nessa etapa que ocorrem as operações com os conjuntos *Fuzzy*. Esse processo usa a Base de Conhecimento que é constituída da Base de Dados e da Base de Regras.

A Base de Dados é responsável por definir quais serão as funções de pertinência do conjunto *Fuzzy* e a Base de Regras possui as regras, dispostas como variáveis linguísticas, que são criadas de acordo com o problema em questão.

Vale ressaltar que, por vezes, essas regras são construídas a partir do conhecimento do especialista no problema específico. Entretanto, não é possível garantir que a Base de Regras gerada apenas com o conhecimento do especialista irá produzir os resultados esperados. Neste caso, processos automáticos para geração da RB são usados a fim de produzir resultados melhores do que os gerados apenas com o conhecimento do especialista.

Por fim, ocorre o processo de conversão Escalar-*Fuzzy* que usa os conjuntos *Fuzzy* obtidos no processo de inferência, interpreta-os e gera saídas não *Fuzzy* (precisas) para que as aplicações práticas possam utilizar as informações geradas no processo.

A Lógica *Fuzzy* é, portanto, uma importante ferramenta de tomada de decisão, uma vez que ela lida com a imprecisão de informação que é uma característica intrínseca ao presente projeto.

2.4 Algoritmos Genéticos

Por vezes, em situações de aplicações práticas para a indústria e o comércio, os desenvolvedores de sistema se deparam com problemas que são considerados difíceis de serem tratados, ou seja, requerem recursos significativos para serem resolvidos. Na tentativa de resolução dessas situações, diversas técnicas são usadas e dentre elas se encontra os Algoritmos Genéticos⁸.

⁸ Como dito anteriormente, no presente trabalho, a técnica que forma a base para o projeto de construção do ambiente de configuração e análise de Algoritmos Evolutivos é o Algoritmo Genético. Nesse sentido, o presente tópico abordará as questões referentes a essa classe de Algoritmo Evolutivo.

Os Algoritmos Genéticos, são procedimentos de busca inerentemente paralelos, probabilísticos, projetados para trabalhar em grandes espaços e que utilizam um conjunto de amostras distribuídas a partir do espaço (de uma população de cadeias) para gerar um novo conjunto de amostras (GOLDBERG; HOLLAND, 1988).

Os GAs foram criados, portanto, com o intuito de imitar, de forma simples e eficiente, determinados processos observados na evolução natural das espécies e se mostram como técnicas capazes de auxiliarem na resolução dos mais diversos problemas.

2.4.1 Histórico dos Algoritmos Genéticos

Em 1859, o naturalista britânico Charles Robert Darwin publicou o livro “Sobre a Origem das Espécies por Meio da Seleção Natural, ou a Preservação de Raças Favorecidas na Luta pela Vida” (do original, em inglês, “*On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life*” (DARWIN, 1859)). Neste livro, Darwin apresenta a teoria da Seleção Natural que explica a adaptação e especialização das espécies por meio de características favoráveis (em relação a determinado ambiente) que se tornam comuns e de características desfavoráveis que se tornam incomuns⁹.

Já em 1865, o monge e botânico austríaco Gregor Johann Mendel publicou o trabalho “Ensaio com Plantas Híbridas” (do original, em alemão, “*Versuche über Pflanzen-hybriden*” (MENDEL, 1866)). Esta publicação introduziu as leis da hereditariedade que deram origem ao que atualmente é chamado de Genética¹⁰.

A junção dessas duas pesquisas deu origem a Síntese Evolutiva Moderna (DEPEW; WEBER, 2011). Essa síntese foi a inspiração para a criação da técnica de Inteligência Artificial chamada de Algoritmos Genéticos (GOLDBERG, 1989).

Os GAs foram inventados na década de 1960 pelo físico e matemático estadunidense John Henry Holland e foram desenvolvidos e difundidos por ele e seus colegas e alunos da Universidade de Michigan nas décadas de 1960 e 1970. Seu objetivo principal era desenvolver maneiras de utilizar os mecanismos de adaptação natural em sistemas de computador (MITCHELL, 1998).

⁹ Disponível em: <<http://www.biography.com/people/charles-darwin-9266433>> Acessado em 10/07/2014

¹⁰ Disponível em: <<http://www.biography.com/people/gregor-mendel-39282#synopsis>> Acessado em 10/07/2014

A partir de então, grandes avanços foram feitos na área, principalmente a partir da publicação do livro intitulado “Adaptação em Sistemas Naturais e Artificiais: Uma Análise Introdutória com Aplicações à Biologia, Controle e Inteligência Artificial” (do original, em inglês “*Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*” (HOLLAND, 1975)).

Na década de 1980 surgiram com mais intensidade as ideias de combinar GA com outras técnicas de inteligência artificial gerando um intenso campo de pesquisa (KOEHN, 1994). Foi nessa década que o cientista da computação estadunidense David Edward Goldberg, publicou seu livro intitulado “Algoritmos genéticos em busca, otimização e aprendizado de máquina” (do original, em inglês, “*Genetic algorithms in search, optimization, and machine learning*” (GOLDBERG, 1989)), livro este, que auxiliou nas pesquisas em GAs nas décadas posteriores.

Na década de 1990 os algoritmos genéticos foram usados em vários problemas de engenharia, e demonstraram ter vantagens numéricas em relação aos métodos clássicos de otimização, especificamente nos casos de problemas matemáticos complexos (DIAS; CORRÊA, 2006).

Nos últimos anos, os GAs têm sido amplamente usados no mundo para modelagem e resolução de problemas. Eles podem ser aplicados a muitos problemas de tecnologia, problemas científicos, problemas de engenharia, entretenimento, negócios entre outros. Também há uma grande quantidade de aplicações que utilizam tal técnica como a robótica, design de engenharia, design automotivo etc. (TABASSUM; MATHEW, 2014). Há aplicações que envolvem ainda áreas como sistemas de manufatura (SOUIER; SARI; HASSAM, 2012), eletroquímica (HAN et al., 2014), os sistemas especialistas (ARDESTANI; MOAZEN; JIN, 2014) e sistemas elétricos (TEIMOURZADEH; ZARE, 2014).

2.4.2 Terminologia

Os GAs, por serem uma técnica de busca estocástica¹¹ baseada na Síntese Evolutiva Moderna, utilizam várias analogias relacionada à biologia dos seres vivos. Termos como genes,

¹¹ GAs são técnicas estocásticas e não determinísticas, ou seja, um GA com a mesma população inicial e o mesmo conjunto de parâmetros pode gerar soluções diferentes a cada vez que é executado (LINDEN, 2012). Entretanto, os GAs não são buscas aleatórias não-direcionadas, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos (REZENDE, 2005).

cromossomos, população, mutação, descendentes, cruzamento são empregados na resolução dos mais variados problemas.

Na Tabela 2-2 é apresentada a relação entre o sistema natural de representação de características e os Algoritmos Genéticos.

Tabela 2-2 - Sistema Natural e GAs –Adaptado de Pacheco (1999) e Linden (2012)

Natureza	Algoritmos Genéticos
Cromossomo	Palavra binária, vetor, <i>string</i> .
Gene	Característica do problema.
Alelo	Valor da característica.
Loco	Posição no cromossomo.
Genótipo	Representa a informação contida no cromossomo – Estrutura.
Fenótipo	Produto da interação do conteúdo genético com o ambiente – Conjunto de Parâmetros.
Indivíduo	Representado pelo cromossomo. Solução para um dado problema.
População	Conjunto de Indivíduos.
Geração	Indivíduos de uma população, por meio de recombinação e mutação, geram descendentes.

Outro termo usado em GA é a chamada Função de Aptidão ou Função *Fitness* que diz respeito à qualidade de um indivíduo como solução de um problema qualquer. Ela recebe um indivíduo como entrada e retorna um número (avaliação) como saída (RUSSELL; NORVIG, 2009; LINDEN, 2012).

Existem ainda os operadores que trabalham sobre cromossomos, chamados de operadores de reprodução, que são a recombinação (também chamada de cruzamento ou *crossover*) e a mutação. Esses operadores fazem analogia aos principais mecanismos de evolução natural, ou seja, a recombinação e a mutação gênica. A recombinação é aplicada, em geral, a um par de cromossomos. Os indivíduos selecionados para o processo de recombinação são chamados pais. A mutação é aplicada a um simples cromossomo, modificando-o aleatoriamente (GABRIEL; DELBEM, 2008).

Outros termos serão apresentados no decorrer dos próximos tópicos.

2.4.3 Características Gerais dos Algoritmos Genéticos

Segundo Rezende (2005) o GA difere de outros métodos tradicionais de otimização por quatro aspectos principais:

1. Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
2. Trabalham com uma população e não com um único ponto;
3. Utilizam informações de custo ou recompensa e não com derivadas ou outro conhecimento auxiliar;
4. Utilizam regras de transição probabilística (estocástica) e não determinística.

Os GAs são, portanto, programas que necessitam somente de informações locais ao ponto avaliado não necessitando de qualquer outra informação adicional (LINDEN, 2012). Eles tentam imitar as características do processo evolutivo com a ideia básica de criar uma população de indivíduos (possíveis soluções para um dado problema) que evolui com o uso de princípios como seleção natural e hereditariedade (FORREST, 1993).

Em Linden (2012), é dito que os GAs são programas simples e que realizam processos iterativos, nos quais cada iteração é chamada de geração ou ciclo. Durante cada geração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Por meio da seleção, são determinados quais indivíduos conseguirão se reproduzir, gerando um número determinado de descendentes para a próxima geração, com a probabilidade de passar à frente suas características determinadas pela sua Função *Fitness*.

Um Algoritmo Genético possui a forma apresentada na Figura 2-9.

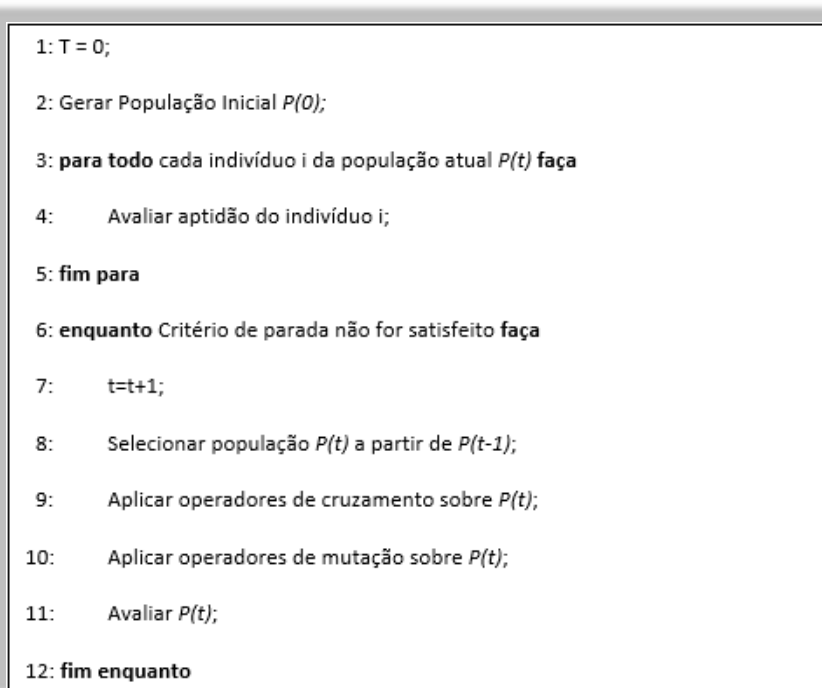


Figura 2-9 - Estrutura de um GA (REZENDE, 2005)

Primeiramente, inicia-se o contador do tempo e a seguir é iniciada a população com soluções candidatas (indivíduos), por vezes, aleatória. Em seguida é feita uma avaliação de cada indivíduo da população e posteriormente, inicia-se o processo de seleção da população por intermédio da população anteriormente gerada levando em consideração a aptidão ou o *fitness* de cada indivíduo. Aplicam-se, então, operadores de cruzamento – com o intuito de gerar novos descendentes – e mutação – a fim de propiciar diversidade à população gerada. Uma nova avaliação é feita e, a seguir, o processo é repetido até que surjam indivíduos que representem soluções desejadas ou próximas ao esperado para o problema em questão.

2.4.4 Codificação

Em um GA, a forma com que as soluções candidatas são codificadas tem importância fundamental no sucesso da técnica (MITCHELL, 1998). A codificação consiste em uma forma de traduzir as informações do problema para o computador.

Nesse sentido, várias são as formas de realizar essa codificação e dentre elas destacam-se a Codificação Binária, que é a mais comum, e a Codificação Real, que é aplicada em muitos problemas (GAFFNEY; GREEN; PEARCE, 2010; KUMAR, 2013).

A Codificação Binária foi usada inicialmente por Holland (1975) e se caracteriza pelo fato do genótipo do indivíduo ser definido por uma sequência de bits (0s ou 1s). Neste caso, cada bit representa uma característica do problema e o conjunto dos bits representa uma possível solução para o mesmo.

A Codificação Binária pode ser vista na Figura 2-10.

		Loco (Posição)								
		1	2	3	4	5	6	7	8	
População	Indivíduo 1 →	0	0	1	1	1	0	0	0	→ Cromossomo
	Indivíduo 2 →	1	1	0	0	0	0	0	1	→ Cromossomo
	Indivíduo 3 →	1	1	1	1	1	0	1	0	.
	Indivíduo 4 →	1	0	1	1	0	0	1	1	.
	Indivíduo 5 →	1	1	1	0	1	1	0	0	.
	Indivíduo 6 →	0	1	0	0	1	1	0	1	.
	Indivíduo 7 →	1	1	0	1	1	1	1	0	.
	Indivíduo 8 →	0	1	1	0	0	1	1	1	→ Cromossomo

Alelo (Valor) = 1 Alelo (Valor) = 0

Figura 2-10 - Codificação Binária para uma população com 8 indivíduos - Adaptado de Coello, Lamont e Veldhuizen (2007)

A Codificação Binária é normalmente usada em situações onde a presença de uma característica é denotada pelo valor 1 (um) e a ausência dessa característica é denotada pelo valor 0 (zero). Ela tem a vantagem de possibilitar uma grande quantidade de cromossomos mesmo com uma pequena quantidade de alelos, entretanto, Herrera, Lozano e Verdegay (1998) afirmam que a representação binária tem dificuldade de lidar com múltiplas dimensões de variáveis contínuas, especialmente quando uma grande precisão é requerida. Isso acontece porque uma quantidade grande de bits será necessária para realizar a representação e, portanto, o cromossomo ficará muito grande dificultando as operações com o GA.

Outro ponto importante diz respeito ao chamado Abismo de Hamming (LINDEN, 2012). Isso acontece em situações onde a mudança de um valor unitário leva a necessidade de mudar vários, ou até mesmo todos os bits de um número binário. Para mudar, por exemplo, o número 7(0111) para o número 8(1000) precisa-se alterar todos os bits de uma única vez. Já para mudar do número 8(1000) para o 9(1001), apenas um bit é alterado. Isso resulta no fato de que mudanças com o mesmo efeito requerem alterações completamente diferentes em relação a cada indivíduo.

Para resolver esse problema pode-se utilizar a representação em Código Gray (DORAN, 2007) onde se converte a representação binária clássica em uma codificação binária em que o Abismo de Hamming não ocorre.

Contudo, Linden (2012) aponta que pode ainda ocorrer um problema relacionado à existência de bits mais significativos na codificação binária. Para um número binário 1000, por exemplo, caso seja trocado o primeiro bit a mudança será de 8 para 0 (diferença de 8), enquanto que se a troca for no último bit a mudança será de 8 para 9 (diferença de 1). Conclui-se, portanto, que a depender do bit, a mudança terá um impacto grande podendo levar a efeitos nocivos na qualidade da avaliação do cromossomo.

Vale salientar que, em algumas situações, essas mudanças bruscas nos cromossomos podem ser benéficas uma vez que o indivíduo criado poderá realizar busca em espaços ainda não explorados.

Pode-se, no entanto, utilizar outro tipo de codificação a fim de evitar os problemas relacionados à Codificação Binária. Nesse sentido, pesquisas tem adotado a chamada Codificação Real. Em Catarina e Bach (2003) é dito que essa codificação é uma das formas de codificação mais usada e consiste em representar, em um gene ou cromossomo, uma variável numérica contínua por intermédio de seu próprio valor real.

A Codificação Real pode ser vista na Figura 2-11.

		Loco (Posição)			
		1	2	3	
População	Indivíduo 1 →	4,3852	0,5837	8,3853	→ Cromossomo
	Indivíduo 2 →	6,3964	5,5495	1,0937	⋮
	Indivíduo 3 →	1,0937	8,3853	9,3856	⋮
	Indivíduo 4 →	6,3964	1,0645	0,5837	→ Cromossomo

Figura 2-11 - Codificação Real para uma população com 4 indivíduos - Adaptado de Coello, Lamont e Veldhuizen (2007)

Percebe-se na Figura 2-11 que o máximo de precisão pode ser usada nesse tipo de codificação e o cromossomo tem um tamanho pequeno pelo fato de que, nesse tipo de representação, cada característica do problema é representada por um gene. Apesar dessas vantagens, a Codificação Real é mais difícil de ser implementada em relação a Codificação

Binária pois, por vezes, é necessário o desenvolvimento de métodos de cruzamento e mutação específicos para o problema.

Várias questões devem ser levadas em consideração na escolha da codificação a ser usada, além do fato de que nem sempre é possível identificar qual o método de codificação mais eficiente em relação a determinado problema. Dessa forma, é necessária a realização de uma análise profunda nas características do problema a fim de escolher a codificação que seja mais adequada e natural possível.

2.4.5 Operador Evolutivo de Seleção

Em um GA, é necessário que sejam escolhidos os indivíduos que, a partir de suas características, darão origem a uma nova geração da população. O operador responsável por esse processo é chamado de Operador de Seleção e tem um papel importante para a evolução do GA até a solução desejada.

A seleção é realizada usando-se o valor do *fitness* (retornado pela Função de Aptidão ou Função de *Fitness*) de cada indivíduo. Nesse processo, a escolha dos membros da população para o processo de reprodução privilegia os indivíduos de *fitness* mais alto sem, entretanto, desconsiderar os de *fitness* mais baixo.

O fato de se utilizar (em menor escala) os indivíduos de menor *fitness* no processo de reprodução é destacado em Linden (2012). Ele afirma que, se apenas os melhores indivíduos se reproduzirem, a população tenderá a ser composta de indivíduos cada vez mais semelhantes e faltará diversidade à população, o que pode impactar, de forma negativa, no processo evolutivo do GA. O uso, portanto, de indivíduos de menor *fitness*, pode inserir características importantes à população, as quais podem não existir nos indivíduos de maior aptidão.

Tendo isso em vista, Goldberg (1989) desenvolveu um método chamado de Método da Roleta onde cada indivíduo possui uma fatia de uma roleta virtual proporcional ao seu valor de aptidão. A cada giro da roleta, algum indivíduo é selecionado de uma forma que os mais aptos (maior *fitness* → maior fatia da roleta) tenham maior probabilidade de serem selecionados.

Um exemplo desse processo pode ser visto na Tabela 2-3.

Tabela 2-3 - Exemplo de seleção de indivíduos pelo Método da Roleta

Indivíduos	Cromossomo	<i>Fitness</i>	Fatia da Roleta (°)
------------	------------	----------------	---------------------

Ind1	0001	5	18
Ind2	0010	75	270
Ind3	0011	15	54
Ind4	0100	5	18
TOTAL		100	360

No exemplo da Tabela 2-3, tem-se 4 indivíduos e os seus respectivos valores de *fitness* que nesse caso totalizam o valor 100. A seguir, a proporção de cada um na roleta é calculada multiplicando o valor total da circunferência da roleta (360°) pelo valor de *fitness* de cada indivíduo. Em seguida, esse valor é dividido pelo valor total do *fitness* (nesse caso 100) e assim a fatia da roleta correspondente a cada um é calculada.

Na Figura 2-12 é apresentada a roleta para cada indivíduo da Tabela 2-3.

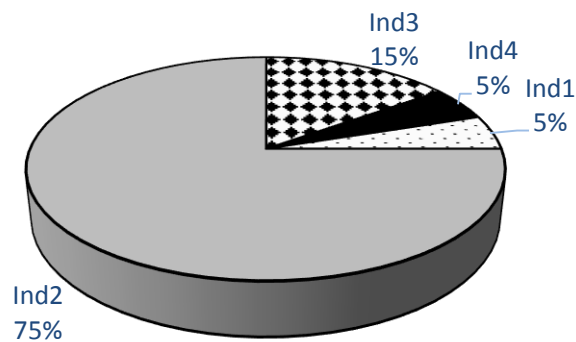


Figura 2-12 - Método da Roleta para os indivíduos da Tabela 2-3

O algoritmo que implementa a ideia do Método da Roleta pode ser visto na Figura 2-13.

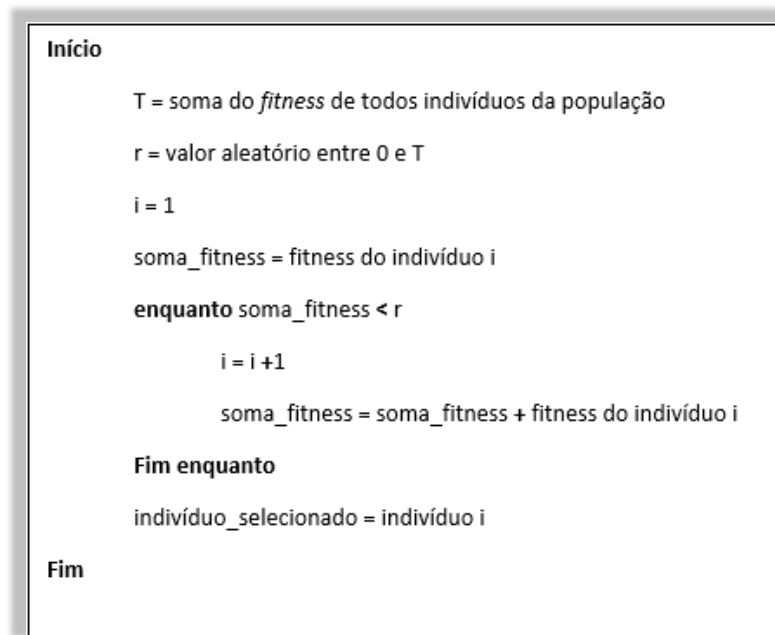


Figura 2-13 - Algoritmo Método da Roleta – Adaptado de Mitchell (1998)

No algoritmo da Figura 2-13, a variável T guarda a soma do valor de *fitness* de todos os indivíduos. Uma variável r guarda um número aleatório entre 0 e T. Percorre-se então os indivíduos da população, somando-se os valores esperados, enquanto a soma for menor que r. O indivíduo cujo valor esperado coloca a soma acima deste limite será o selecionado.

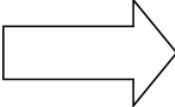
O Método da Roleta apesar de ser um método muito comum e simples de ser implementado, tem a desvantagem de, para populações pequenas, gerar número de filhos atribuídos a um indivíduo muito longe do valor esperado¹². Mitchell (1998) chama atenção para a alta variância do método. Ele demonstra que no início do processo a variação de aptidão na população é alta e um pequeno número de indivíduos são muito mais aptos do que outros. Com isso, em uma população pequena, os melhores indivíduos se multiplicam rapidamente fazendo com que a diversidade da população diminua o que pode causar uma convergência prematura do algoritmo para uma solução não desejada, ou seja, para um máximo local e não para o máximo global. Neste caso, outros métodos de seleção podem ser usados como, por exemplo, o Método do Torneio.

¹² Percebe-se na Figura 2-12 que o Indivíduo 2 (Ind2) tem fatia igual a 75% da roleta (ou 270°). Isso significa que a cada 4 sorteios a tendência é que o Ind2 seja escolhido 3 vezes, contudo, Linden (2012) destaca que a teoria da probabilidade demonstra que essa afirmação apenas será verdadeira se o número de sorteios tender ao infinito, o que para o GA significa um valor relativamente grande de indivíduos na população.

Neste método um subconjunto da população é sorteado e os melhores indivíduos são selecionados para reprodução. No caso, o tamanho do subconjunto pode variar entre 2 e o número da população – 1.

Um exemplo desse método pode ser visto na Figura 2-14.

Indivíduo	<i>fitness</i>
X1	200
X2	100
X3	9500
X4	100
X5	100
X6	10000
X7	1
X8	40



Torneios		
X1	X7	X8
X2	X3	X5
X6	X4	X4
X2	X7	X1
X5	X5	X5
X3	X4	X2
X4	X2	X6
X4	X6	X5

Figura 2-14 - Exemplo de seleção de indivíduos pelo Método do Torneio (LINDEN, 2012)

No exemplo da Figura 2-14 o tamanho do subconjunto da população selecionado é igual a 3. Os indivíduos sorteados competem no torneio sendo o vencedor aquele que tiver o maior valor de *fitness*. Os vencedores dos torneios estão destacados em cinza.

É importante ressaltar que, neste método, a única vantagem dos indivíduos mais aptos é que, se selecionados, eles vencerão o torneio.

Um problema nesse método é o fato de que a chance de os indivíduos menos aptos passarem seus genes é muito baixa, uma vez que eles apenas vencerão os torneios em que os sorteios ocorrem entre eles próprios ou indivíduos de mesmo *fitness*. Nessa situação, quanto maior for o subconjunto, menor será a chance de o vencedor do torneio ser o de *fitness* mais baixo. Devido a esse fato, os torneios ocorrem entre 2 indivíduos, a fim de aumentar as chances dos indivíduos menos aptos passarem suas características para a próxima geração.

Outra forma de aumentar as chances dos indivíduos menos aptos é fazer a seleção nos torneios por meio de uma probabilidade previamente definida. Essa forma pode ser vista no algoritmo da Figura 2-15.

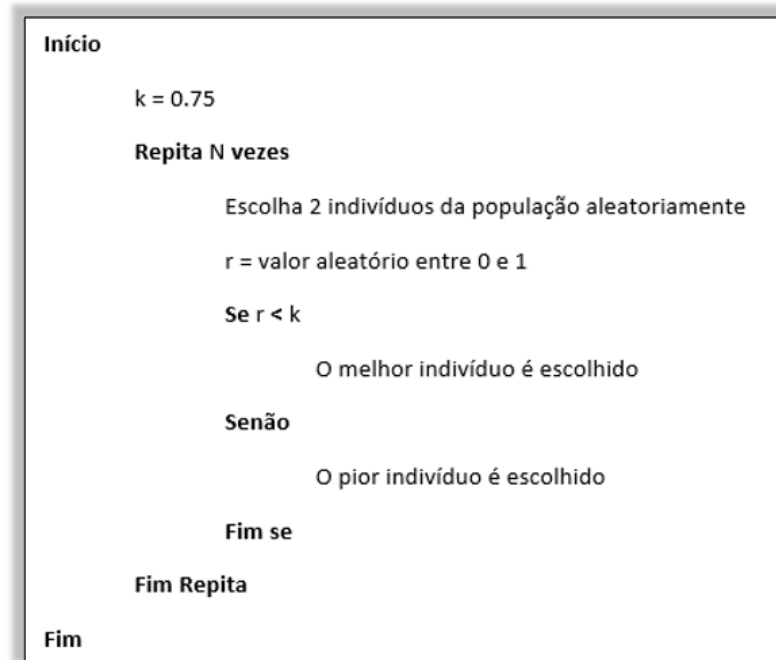


Figura 2-15 - Algoritmo para selecionar os indivíduos para o torneio por meio de uma probabilidade previamente definida – Adaptado de Mitchell (1998)

O algoritmo da Figura 2-15 inicia atribuindo o valor 0,75 (75%) a variável k que é usada para determinar, baseado em um valor aleatoriamente escolhido, qual o indivíduo será selecionado. Esse método atribui uma probabilidade adequada para o indivíduo menos apto fazendo com que haja diversidade na população. A vantagem é que, dessa forma, evita-se a convergência prematura do GA para um máximo local.

Assim sendo, a escolha do método dependerá do problema proposto no que diz respeito ao tamanho da população, as restrições do problema, o tempo disponível para busca da solução, dentre outros fatores. A ideia é, portanto, escolher o método de seleção que mais se adeque ao problema proposto.

2.4.6 Operador Evolutivo de Elitismo

O processo de Elitismo foi introduzido pelo cientista da computação estadunidense Kenneth De Jong¹³ em seu artigo “Uma Análise do Comportamento de uma Classe de Sistemas

¹³ Disponível em: < <http://mitpress.mit.edu/authors/kenneth-de-jong> > Acessado em 01/09/2014

Genéticos Adaptativos” (do original, em inglês, “*An Analysis of the Behavior of a Class of Genetic Adaptive Systems*” Jong (1975)).

Esse processo é um adendo ao método de seleção que faz com que os melhores cromossomos de uma geração sejam mantidos na seguinte. Sem o uso desse processo, os melhores indivíduos de uma geração podem ser perdidos devido a cruzamentos, mutações ou procedimentos de seleção para reprodução. Muitos pesquisadores descobriram que o elitismo melhora significativamente o desempenho em relação ao GA (MITCHELL, 1998).

Linden (2012) destaca que a manutenção do melhor indivíduo da geração t na população da geração $t+1$ garante que o melhor indivíduo da geração $t+1$ é pelo menos igual ao melhor indivíduo da geração t (no pior caso, em que nenhum indivíduo melhor é gerado). Esse processo, portanto, garante que o gráfico de avaliação do melhor indivíduo no decorrer das gerações é uma função monotonamente crescente.

Jong (2006) afirma que, quanto mais elitista for o algoritmo, mais rápido ele converge, todavia, o risco de não encontrar a melhor solução aumenta (convergência para um ótimo local). Ele afirma que aumentando o grau de elitismo, aumenta-se o que ele chama de “*greediness*” ou ganância do algoritmo.

Ainda em Jong (2006) é destacado que se reduzindo o grau de elitismo, diminui-se a taxa de convergência do algoritmo e aumenta-se a probabilidade de encontrar a melhor solução. No entanto, se diminuir o grau de elitismo, a resposta retornada pode ser pior que os algoritmos mais elitistas.

Percebe-se, portanto, que os parâmetros definidos para um GA impactam diretamente no desempenho do processo sendo, desse modo, específicos para cada problema e determinados, muitas vezes, por meio de experimentação o que demonstra a viabilidade da proposta apresentada nesse projeto de construção de um ambiente para configuração e análise de EAs para sintonia da FRB do Sistema de Controle de FMS.

2.4.7 Operador Evolutivo de Cruzamento

O Operador de Cruzamento (também chamado de Operador de *Crossover* ou Recombinação) tem a função de misturar características de indivíduos genitores a fim de gerar novos descendentes. Nesse processo, um ou mais pontos de corte são escolhidos e então as

características são permutadas gerando, conseqüentemente, novos indivíduos (soluções candidatas).

Um ponto de corte constitui uma posição entre dois genes de um cromossomo. Cada indivíduo de n genes contém $n-1$ pontos de corte, e este ponto de corte é o ponto de separação entre cada um dos genes que compõem o material genético de cada pai (LINDEN, 2012).

A Figura 2-16 mostra os possíveis pontos de corte em um cromossomo com 5 genes.

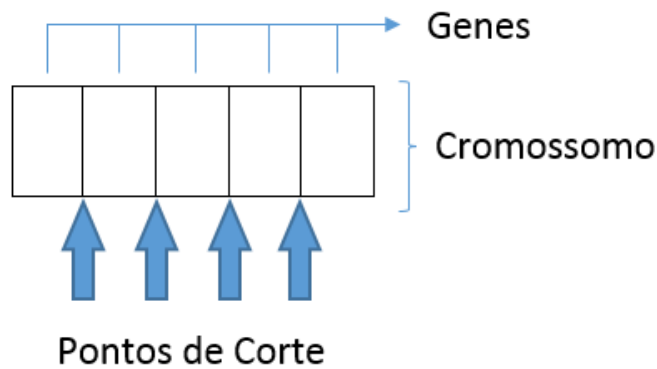


Figura 2-16 - Exemplo de pontos de corte - Adaptado de (LINDEN, 2012)

Definido(s) o(s) ponto(s) de corte, ocorre então, a permutação entre os genes dos pais, gerando novos filhos, o que pode ser visto na Figura 2-17.

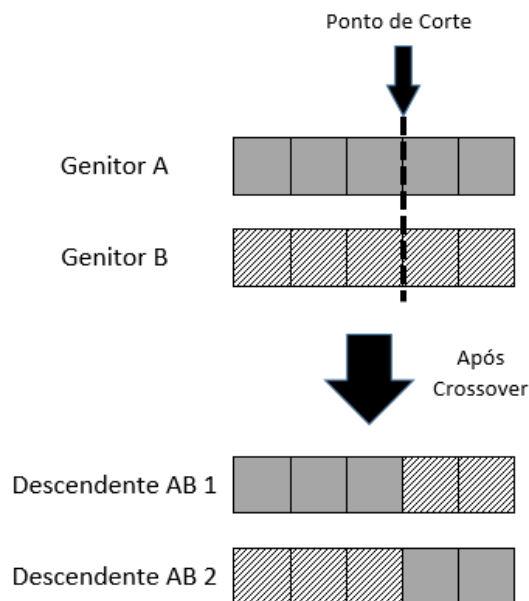


Figura 2-17 - Operador de Cruzamento com 1 ponto de corte

Podem ocorrer situações onde mais de um ponto de corte é definido. Isso pode ser visto na Figura 2-18.

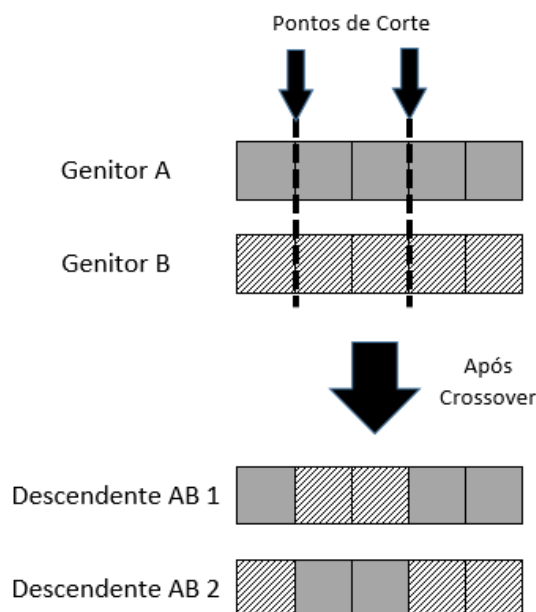


Figura 2-18 - Operador de Cruzamento com 2 pontos de corte

Na situação onde há 2 ou mais pontos de cortes, a operação é dita *multi-point crossover*.

Jong (2006) afirma que, em seus estudos, Holland já destacava a importância da recombinação como fonte de variação na prole produzida (HOLLAND, 1967). Ele afirma que o cruzamento, juntamente com a mutação ocasional, fornece o potencial para uma exploração muito mais agressiva do espaço de busca. As ideias de Holland introduziram as bases para a realização do cruzamento de 1 ponto e, posteriormente, a ideia foi expandida para o cruzamento de 2 a $n-1$ pontos (considerando n a quantidade de genes) (JONG, 1975).

2.4.8 Operador Evolutivo de Mutação

Após a operação de recombinação ser executada, é a vez do Operador de Mutação. Esse operador tem o objetivo de modificar alguma(s) característica(s) de um indivíduo, de forma aleatória.

Nesse procedimento, é implementado um sorteio onde cada gene tem uma probabilidade de ser alterado ou não. Caso o sorteio indique que haverá mudança, ocorre uma alteração do

alelo do gene correspondente por um valor presente no alfabeto genético da população em questão.

Um exemplo de mutação pode ser visto na Figura 2-19.

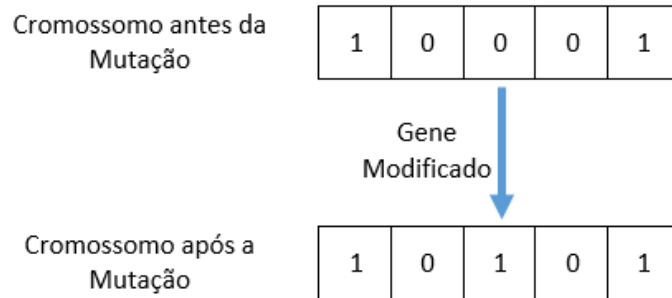


Figura 2-19 - Exemplo da operação de Mutação

Mitchell (1998) e Jong (2006) destacam que o uso do Operador de Mutação é necessário para que seja inserida, e se mantenha, a diversidade em uma dada população, fazendo com que o algoritmo desenvolvido tenha a possibilidade de explorar espaços diferentes de soluções.

Na Figura 2-20 é mostrado o processo de cruzamento e mutação até a geração de novos indivíduos.

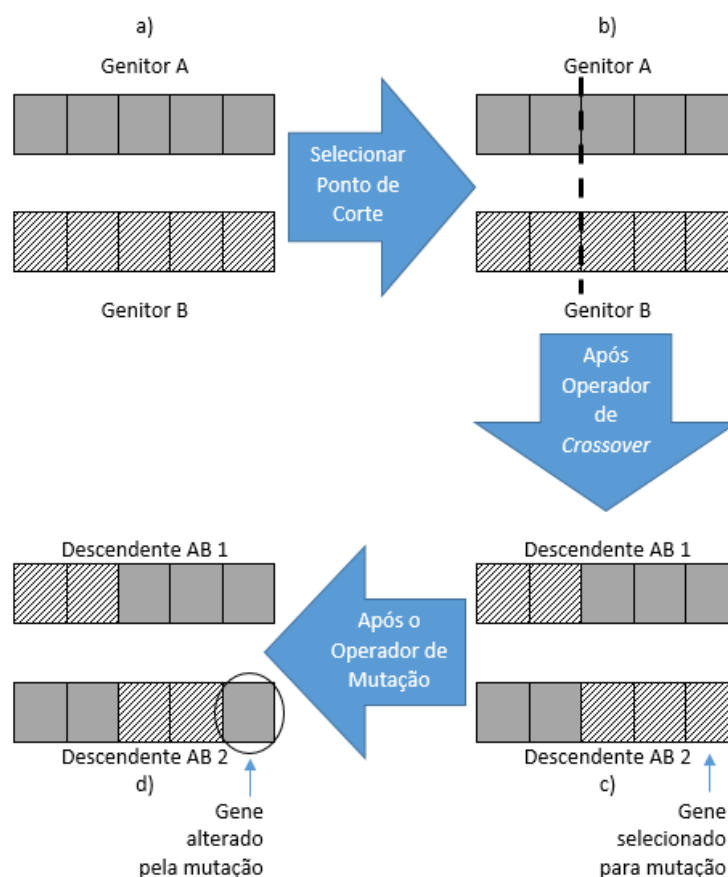


Figura 2-20 - Operador de Cruzamento e Mutação - Adaptado de Linden (2012)

2.4.9 Parâmetros Evolutivos

Os parâmetros definidos para um GA podem afetar diretamente o seu desempenho, provocando, por vezes, segundo Jong (2006), uma convergência prematura para um ótimo local. Linden (2012) destaca que os valores dos parâmetros são particulares de cada problema e, para defini-los, podem ser realizados testes com valores variáveis, para que se obtenha a definição do que seria o ideal para a situação que se deseja tratar.

Os principais parâmetros a serem definidos para um GA são (GOLDBERG; HOLLAND, 1988; MITCHELL, 1998; MEDEIROS, 2005; JONG, 2006; BARBOZA, 2010):

- **Tamanho da População:** Esse parâmetro afeta o desempenho do GA no sentido em que, se for definido um valor muito pequeno, o GA não terá indivíduos suficientes para explorar todo espaço de busca, podendo ser incapaz de encontrar boas soluções. Em contrapartida, caso esse número seja grande demais, pode ocorrer uma queda de desempenho, no que diz respeito ao fato do GA, nessa

situação, demorar demais para encontrar uma boa solução. Jong apud Mitchell (1998) sugere que experimentos indicam um valor entre 50 e 100 indivíduos como adequado, todavia, a própria Mitchell alerta para o fato de não se ter estudos que comprovem a eficiência desses números para problemas fora do escopo dos tratados por Jong.

- **Taxa de Cruzamentos:** Parâmetro que define a porcentagem de indivíduos que deverão ser recombinados a fim de gerar novos indivíduos. Esse valor afeta o desempenho do GA no sentido que se for pequeno o GA pode demorar muito para encontrar uma solução uma vez que os melhores indivíduos passarão seus genes com uma frequência muito baixa e, desse modo, o GA demorará para convergir para o ótimo desejado. Em compensação, se esse valor for muito alto, a maior parte da população é substituída e, assim, podem ocorrer perdas de genes que poderiam ser importantes no processo de busca da solução desejada. Em Jong apud Mitchell (1998) é sugerida uma taxa de 60%.
- **Taxa de Mutação:** Esse parâmetro, se mal ajustado, pode fazer com que o GA se comporte de uma maneira não esperada, fazendo com que não sejam procuradas soluções em novas regiões do espaço de busca ou, simplesmente, a busca se torne aleatória (*Random Walk*). Jong sugere um valor da taxa de mutação (probabilidade que um gene tem de ser alterado) na ordem de 0,1% e Linden sugere valores na faixa de 0,5%.
- **Número de Gerações:** Esse parâmetro diz respeito a quantidade de iterações que serão executados para busca da solução. Um valor pequeno pode fazer com que o GA não seja eficiente na medida em que pode incorrer em soluções encontradas em máximos locais. Um valor grande faz com que o desempenho do GA aumente, pois evita conversões prematuras para máximos locais, entretanto, o tempo de processamento também aumenta, o que pode ser um problema em situações que demandam respostas rápidas. O valor dependerá do critério de parada. Goldberg apud Barboza (2010) considera o desvio padrão dos valores de aptidão dos indivíduos da população como critério para definir o número de gerações. Compara-se o desempenho da geração atual com a anterior e, se o desvio padrão for igual ou menor ao estabelecido, o processo é encerrado.

Jong (2006) destaca que não há resultados conclusivos sobre o que é melhor, e a maioria das pessoas usa o que tem funcionado bem nos casos semelhantes anteriormente relatados na

literatura. Já Hinterding, Michalewicz e Eiben (1997) chamam atenção para a possibilidade de uso de parâmetros variáveis para a avaliação da qualidade dos mesmos durante a execução do GA.

2.5 Considerações Finais

Neste capítulo foram abordados os conceitos fundamentais que envolvem a Lógica *Fuzzy* e os Algoritmos Genéticos. Questões como histórico, terminologias e características gerais dos sistemas que podem ser formados com as técnicas foram tratadas. Abordou-se também a questão do Algoritmo Evolutivo destacando que este, trata-se de um adendo ao GA proposto por Holland tendo, adaptações realizadas tanto na forma como as seleções, os cruzamentos e a mutações são realizados como no acréscimo do Operador de Elitismo proposto por Jong (1975).

Destaca-se ainda o fato dos parâmetros a serem definidos, quando da execução de um GA (ou mesmo de um EA), variarem de acordo com cada problema. Esse fato corrobora com a ideia da construção de um ambiente que permita a configuração e a análise de desempenho de EAs a fim de definir qual a configuração que gerará a FRB que leva à sintonia da FRB do Sistema de controle de um FMS.

ARTIGOS RELACIONADOS

3.1 Considerações Iniciais

Neste capítulo serão apresentados os artigos relacionados ao presente projeto abordando trabalhos que lidam com EAs, em especial os GAs, Lógica *Fuzzy* e Sistemas de Manufatura.

Trabalhos referentes à resolução de problemas de controle de sistema de manufatura, Algoritmos Genéticos para geração da base de regras de um Sistema *Fuzzy*, uso dos sistemas *Fuzzy* e Genéticos para resolução dos mais diversos problemas e trabalhos que utilizam *Fuzzy* e GA para prover o controle de um sistema de manufatura são apresentados e discutidos. Além disso, haverá um tópico relacionado às pesquisas no laboratório Tear da UFSCar e suas relações com o projeto aqui proposto.

Por fim, seguem as considerações finais abordando uma síntese dos temas abordados e apresentando o foco do presente trabalho.

3.2 Lógica *Fuzzy* e Algoritmos Genéticos na Resolução dos Mais Diversos Problemas

A Lógica *Fuzzy* e os Algoritmos Genéticos são técnicas da AI que podem ser usadas para resolução de vários problemas, principalmente, em casos onde outras técnicas não apresentam resultados satisfatórios ou, até mesmo, não são capazes de encontrar as soluções.

A Lógica *Fuzzy*, em particular, é extremamente útil em pesquisas e desenvolvimento de projetos, incluindo áreas como engenharia (elétrica, mecânica, civil, química, aeroespacial, agrícola, biomédica, informática, meio ambiente, geológica, industrial e mecatrônica), matemática, desenvolvimento de software, ciências naturais (biologia, química, ciências da terra, e física), pesquisas médicas, ciências sociais (economia, administração, ciência política e psicologia), políticas públicas, negócios e juristas (SINGH et al., 2013).

Um exemplo de uso da Lógica *Fuzzy* pode ser visto no trabalho de Royes e Bastos (2001). Esse trabalho se insere na área de ciências políticas e tem o objetivo de demonstrar a viabilidade da construção de um sistema especialista que pode auxiliar no processo de tomada de decisão política. O sistema final corresponde a verificação da chance de reeleição de um presidente atual, governador ou prefeito. Nesse trabalho, a qualidade das previsões em relação às chances de reeleição dos candidatos, indica a viabilidade do uso de Lógica *Fuzzy* em ciência política como um bom meio para tomadas de decisões. O sistema criado foi definido por uma amostragem de opinião e também por meio de reuniões sucessivas com um especialista em política. Além disso, uma experimentação foi feita com diferentes candidatos avaliados na pesquisa da revista e os resultados obtidos pelo sistema *Fuzzy* foram muito semelhantes aos resultados oficiais da eleição.

Outros exemplos de aplicação da Lógica *Fuzzy* são os trabalhos de Coelho et al. (2012) e de Lingala et al. (2014) que se inserem na área médica, mais precisamente no auxílio a diagnósticos.

Em Coelho et al. (2012) é apresentado um método de clusterização, chamado *Fuzzy c-means*, usado na segmentação de imagens dermatoscópicas de lesão de pele. Esse método usa a Lógica *Fuzzy* para lidar com as incertezas referentes às regiões de uma dada imagem. O método consiste em calcular o grau de pertinência de um dado pixel em relação às regiões que ele pode pertencer de acordo com determinada vizinhança. Neste trabalho, este método foi aplicado em três imagens de lesão de pele sendo, dois melanomas¹⁴ e um nevo¹⁵, obtendo-se três classes de *clusters* para cada imagem. Estes *clusters* foram usados para calcular dois valores

¹⁴ O melanoma cutâneo é um tipo de câncer de pele que tem origem nos melanócitos (células produtoras de melanina, substância que determina a cor da pele) e tem predominância em adultos brancos. Disponível em: < http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/pele_melanoma/definicao > Acessado em 10/07/2014

¹⁵ Nevos são pequenas manchas marrons regulares na pele, salientes ou não. São popularmente conhecidos por pintas e verrugas. A maioria das pintas surge em decorrência da exposição solar, e possui um formato regular. Disponível em: < <http://www.sbd.org.br/doencas/nevos-displasicos/> > Acessado em 10/07/2014

de limiar. Estes limiares foram aplicados no algoritmo de binarização, para a segmentação da imagem. Com o intuito de verificar a eficiência deste método nestes tipos de imagem, as imagens segmentadas por meio do método *Fuzzy c-means* foram comparadas com as mesmas imagens segmentadas por meio do algoritmo de Otsu¹⁶. Como resultado, obteve-se uma segmentação melhor com o uso do método *Fuzzy c-means* em relação ao algoritmo de Otsu, melhorando os processos de diagnósticos por imagem.

Em Lingala et al. (2014) a Lógica *Fuzzy* é empregada como ferramenta de auxílio em diagnósticos médicos. Nesse trabalho, ela foi usada para auxiliar a análise de cores em imagens dermatológicas a fim de detectar casos de melanoma. Foram analisados três tons de azul (azul lavanda, azul claro e azul escuro) para determinar os pontos de lesão benigna usando um conjunto de treinamento para determinar os parâmetros da função membro. Nesse processo, a detecção das áreas azuis ocorreu em duas etapas: o conjunto de treinamento e em seguida o conjunto de teste. No conjunto de treinamento, houve a análise de 22 imagens de melanoma com marcação manual das áreas em azul para determinação dos parâmetros e posterior implementação dos conjuntos *Fuzzy*. Para o conjunto de testes houve a análise de um conjunto de 866 imagens para extração das características e classificação das imagens. Como resultado, de acordo com os testes, obteve-se uma precisão acima de 82,7% na detecção de melanoma. Com o uso de um reconhecedor de padrões a precisão obtida foi de 79,9-80,1% para cores individuais e 81,4% para vários tons. O artigo conclui que, com o uso dos conjuntos *Fuzzy*, obteve-se melhores resultados do que os resultados obtidos com Conjuntos *Crisp* (convencionais).

Em relação aos Algoritmos Genéticos, estudos demonstram que eles têm sido usados para encontrar as melhores soluções para problemas complexos em vários domínios, como biologia, engenharia, ciência da computação e ciência social, dentre outros (KUMAR et al., 2010).

Como exemplo tem-se o trabalho de Ouiddir, Rhali e Abdelhakem-Koridak (2005) que usa GAs para resolver o chamado Problema de Despacho Econômico (EDP)¹⁷. Neste estudo, utilizou-se o GA para resolver o EDP em uma rede elétrica disposta no oeste da Argélia. Dois

¹⁶ Algoritmo de Otsu consiste em encontrar um limiar global automático e, a partir deste, binarizar a imagem (COELHO et al., 2012).

¹⁷ Esse problema diz respeito ao escalonamento das saídas das unidades de geração conveniadas para encontrar a demanda de carga consumidora a um custo mínimo de operação, satisfazendo as restrições inerentes às unidades geradoras usadas e restrições de igualdade e desigualdade impostas pelo problema (CHOWDHURY; RAHRNAN, 1990; COELHO; MARIANI, 2006).

casos foram analisados, sendo que no primeiro as perdas de transmissão da linha foram calculadas e mantidas constantes e no segundo caso as perdas nas linhas de transmissão foram consideradas como uma função linear. Para o primeiro caso, o uso de GA traria um lucro equivalente a 2,35% por ano à geradora analisada quando da comparação com os métodos tradicionais. No segundo caso o GA proporcionaria um lucro de 2,82%, sendo que, boa parte desse resultado, é devido ao custo de produção usando GA que é mais baixo em relação aos métodos tradicionais. Portanto, os resultados obtidos nos casos estudados indicam que o GA pode ser usado para determinar o controle ótimo para a geração de energia, com custo baixo e resultados melhores que os métodos clássicos.

Outro exemplo é o trabalho de Srivastava e Kim (2009) que apresenta um método para otimizar os processos de testes de software, identificando o caminho mais crítico de um programa. Isso foi feito por intermédio de variáveis, que, quando ajustadas adequadamente, propiciam a otimização e a seleção dos caminhos considerando o quão crítico são os mesmos. Nesse projeto, desenvolve-se uma abordagem mais seletiva para testes, concentrando-se nas partes que são mais críticas para que esses caminhos fossem testados. Segundo os autores, ao identificar os caminhos mais críticos, a eficiência de testes pode ser aumentada. Por fim, é concluído que, com o uso dos GAs, obtém-se uma forma mais eficaz de realizar os testes o que ajuda a refinar esforço e estimativa de custo desta fase.

Existem casos ainda que as técnicas *Fuzzy* e GA são usadas em conjunto para a obtenção de melhores resultados no tratamento de problemas.

O artigo de Herrera, Lozano e Verdegay (1994) por exemplo, mostra como GA pode ser aplicado para resolver problemas de otimização *Fuzzy*. É mostrado que o uso de GA pode ser de grande valia na resolução de problemas associados a Lógica *Fuzzy* pois, devido ao seu potencial em ambientes de incerteza, ele é capaz de atuar como uma ferramenta flexível para problemas de otimização e busca.

Outro exemplo é o trabalho de Tosatti et al. (2008), onde é apresentada a proposta de uma abordagem alternativa baseada em sistemas Genético-*Fuzzy* aplicado ao problema de determinação do estágio do câncer de próstata. Nesse trabalho é feita uma comparação do método usado com a abordagem clássica de *Pittsburgh*¹⁸. O sistema proposto usa os GAs como

¹⁸ A abordagem Pittsburgh é uma abordagem de GA baseada em população onde cada indivíduo codifica um conjunto de regras de conhecimento. Por ter sido inicialmente proposta na Universidade de Pittsburgh, essa abordagem recebeu o nome da universidade (BERNARDINI, 2006).

um componente dinâmico para busca em um espaço de soluções e o sistema *Fuzzy* como um componente estático representando a relação e graus de influência dos fatores envolvidos. Como resultado, verificou-se que apesar da abordagem *Pittsburgh* utilizar menos recursos computacionais, a abordagem Genética-*Fuzzy* apresentou resultados mais significativos, pois os indivíduos gerados possuíam valores de *fitness* maior que os da abordagem clássica.

Outro trabalho que usa a Lógica *Fuzzy* em conjunto com os Algoritmos Genéticos é o de Jamshidi e Pilevar (2013), onde é proposto um método para automatizar a segmentação de imagens de ressonância magnética e estimar o número correto desses segmentos. Nesse trabalho, o método Genético-*Fuzzy* foi testado em imagens de ressonância magnética e obtiveram-se melhores resultados do que os métodos tradicionais. No caso do presente projeto, o número correto de segmentos foi obtido de forma totalmente automática e o método convergiu rapidamente para a solução desejada.

Percebe-se, portanto, que a Lógica *Fuzzy* e os Algoritmos Genéticos podem ser usados para a resolução de diversos problemas, nas mais variadas áreas e que a junção dessas técnicas, na resolução de um dado problema, deve ser amplamente considerada, haja vista as potencialidades demonstradas nos trabalhos acima citados.

3.3 Algoritmos Genéticos para Geração da Base de Regras de um Sistema *Fuzzy*

Em especial, a Lógica *Fuzzy* vem sendo usada para realizar a tomada de decisão em FMS e, por consequência, mostram-se importantes estudos que busquem formas de aperfeiçoar os sistemas que utilizam tal abordagem.

Um dos pontos que pode ser aperfeiçoados em um Sistema *Fuzzy* é a Base de Regras e os Algoritmos Genéticos podem ser usados nessa tarefa.

Neste sentido, essas técnicas de IA podem ser usadas em conjunto para otimização de um sistema e, neste tópico, alguns trabalhos relacionados ao uso de GA para geração automática da FRB serão apresentados.

Como exemplo, do emprego das 2 técnicas em conjunto tem-se o trabalho de Castro e Camargo, 2004 onde é usado um Algoritmo Genético Autoadaptável para geração da Base de Regras *Fuzzy*. As técnicas são usadas de forma a prover uma otimização do processo de geração da Base de Regras *Fuzzy* com um GA que muda dinamicamente seus parâmetros proporcionando uma diversidade maior na população e evitando convergência prematura. Neste trabalho, o fato de se utilizar um GA Autoadaptável para geração da FRB proporcionou melhores resultados quando comparados com o uso de GA com parâmetros fixos. Neste caso, os resultados revelaram que a alteração dinâmica dos valores dos parâmetros melhorou ainda mais o desempenho do algoritmo evitando convergências prematuras para ótimos locais.

Outro trabalho é o de Cintra et al., 2007 que compara dois métodos para geração automática da FRB a partir de dados numéricos. Um dos métodos aborda a criação da FRB usando GA em associação com uma heurística de pré-seleção de regras candidatas e o outro método aborda o uso de classificadores Bayesianos também para a geração da FRB. Neste trabalho, a geração automática da FRB com GA tem seu espaço de busca definido por uma combinação de um certo número de regras de todas as regras possíveis, considerando as variáveis do problema e os conjuntos *Fuzzy* definidos. Já o método que usa o classificador Bayesiano reduz o número e a complexidade do conjunto de regras, entretanto, não garante um conjunto de regras mínimo, fazendo com que sejam necessários, em algumas situações, passos extras para obtenção do conjunto mínimo de regras. Neste caso, a técnica com classificadores Bayesianos se mostrou mais relevante para aplicações que possui um conjunto de atributos irrelevantes. Os resultados demonstraram ainda que o uso da pré-seleção das regras no GA provocou um número menor de interações necessárias para conversão para um resultado ótimo, reduzindo assim, o tempo de processamento.

Outro artigo é o de Abdelhak et al., 2014 que propõe uma metodologia para otimizar a configuração de um sistema de energia híbrido (Energia Eólica e Solar). Para isso é usada uma abordagem para geração da Base de Regras *Fuzzy* de forma automática por intermédio de um Algoritmo Genético Adaptativo *Fuzzy*. Neste trabalho, as taxas de cruzamento e mutação são modificadas dinamicamente garantindo a diversidade da população e evitando convergência prematura. Esta metodologia permitiu a obtenção do número ideal de painéis fotovoltaicos, turbinas eólicas e unidade de armazenamentos do sistema de energia híbrido demonstrando que o uso de um GA adaptativo apresentou resultados melhores que o uso do GA padrão.

3.4 Resolução de Problemas de Controle em Sistemas de Manufatura

Como técnica de controle para sistemas de manufatura destacam-se as Redes de Petri(PN)¹⁹, a Lógica *Fuzzy* e os Algoritmos Genéticos.

As Redes de Petri, em particular, tornaram-se uma ferramenta para a modelagem, simulação, programação e controle fabril e muitas pesquisas tem usado dessa técnica para resolver problemas relacionados a sistemas de manufatura (LI; WU; RONG, 2014).

Tem-se como exemplo o trabalho de Morandin Jr. e Kato (2003), onde foi proposto um método para modelagem e controle de FMS baseado em Redes de Petri Virtuais. O método proposto apresentou uma estratégia modular considerando os recursos compartilhados e o planejamento de processo alternativo. Os autores propuseram uma estratégia que visava a facilitação do processo de modelagem de sistemas de manufatura complexos. Essa estratégia propiciou a diminuição da dificuldade de compreensão do modelo em sistemas com grande quantidade de elementos e com isso propiciou um melhor controle do FMS.

Outro exemplo é o trabalho de Kato, Morandin Jr. e Sgavioli (2010) que tem como objetivo utilizar Redes de Petri Colorida (CPN)²⁰ para modelar um sistema de produção e resolver os conflitos que ocorrem nos processos durante a competição por recursos. Regras *Fuzzy* são construídas para dar prioridades para o processo de conflito e dessa forma realizar o controle do sistema de manufatura. Nesse trabalho, informações em tempo real sobre o estado do sistema são usadas, o que possibilita a resolução de problemas relacionados à programação da produção e, conseqüentemente, um melhor controle do sistema.

Outro exemplo é o trabalho de Prakash, Chan e Deshmukh (2011) que desenvolveu um Algoritmo Genético com Base de Conhecimento (KBGA)²¹ para resolução de problemas relacionados a controle de sistema de manufatura, mais precisamente o problema de programação da produção. Neste artigo o KBGA foi testado em 10 problemas de tamanho moderado para mostrar a eficácia da técnica sobre o GA. No algoritmo proposto, os operadores

¹⁹ Uma Rede de Petri é definida como um grafo bipartido dirigido contendo lugares, transições e arcos direcionados que ligam lugares a transições e transições aos lugares (LI; WU; RONG, 2014).

²⁰ Redes de Petri são uma ferramenta de modelagem gráfica e matemática aplicável a muitos sistemas desenvolvida para descrição e estudos de sistemas de processamento de informação, que se caracterizam como sendo concorrentes, assíncronos, distribuídos, paralelos, não-determinísticos e/ou estocásticos(MURATA, 1989).

²¹ O KBGA é uma técnica de procura estocástica que une a capacidade inerente do GA e a força do conhecimento para melhorar o desempenho de um sistema (PRAKASH; CHAN; DESHMUKH, 2011).

genéticos são usados com a base de conhecimento a fim de reduzir o tempo de processamento e alcançar a solução esperada de uma forma mais eficiente. Os resultados mostraram que o uso do KBGA forneceu bons resultados, demonstrando, desse modo, a eficácia do uso de GAs em conjunto com uma Base de Conhecimento para resolução de problemas relacionados a programação da produção.

Percebe-se, deste modo, que várias técnicas podem ser usadas, inclusive em conjunto, para a resolução de problemas relacionados a controle de FMS. Nesse contexto, modelagens adequadas e tomadas de decisões apropriadas podem evitar e tratar ocorrências relacionados a *deadlocks* e conflitos em sistemas de manufatura.

3.5 Lógica *Fuzzy* e Algoritmos Genéticos para Controle de FMS

Em AI algumas técnicas podem ser combinadas para a resolução de problemas, gerando assim os chamados Sistemas Inteligentes Colaborativos e os Sistemas Inteligentes Híbridos.

Sistemas Inteligentes Colaborativos são caracterizados pelo fato dos agentes do sistema trabalharem em conjunto para um objetivo comum com a necessidade crítica de interação um com o outro (IVAN; CIUREA, 2009).

Os Sistema Inteligentes Híbridos caracterizam-se pela integração de 2 ou mais técnicas de IA para a construção de um sistema com mais vantagens e menos deficiências quando da comparação com os sistemas agindo isoladamente (AHMED; ADEEL, 2007; DESOUKY; SCHWARTZ, 2009; HASOON; JASIM, 2013).

Como exemplo tem-se os Algoritmos Genéticos e a Lógica *Fuzzy* que podem atuar em conjunto, de maneira colaborativa ou híbrida, no controle de um sistema de manufatura, tornando-o mais eficiente.

Como exemplo de sistemas que empregam GA e Lógica *Fuzzy* para controle de manufatura tem-se o trabalho de Khoo, Ang e Zhang (2000). Neste artigo é descrito o desenvolvimento de uma abordagem híbrida que integra a teoria dos grafos, os conjuntos *Fuzzy* e os Algoritmos Genéticos para diagnóstico de falhas em um sistema de manufatura. Primeiramente uma técnica foi proposta e desenvolvida para a identificação de possíveis caminhos de propagação de falhas. Como o diagnóstico de falhas envolve a interpretação de

incertezas, a Lógica *Fuzzy* é usada para tratar essas situações. Diferentemente dos sistemas convencionais, onde o diagnóstico assume que todos os componentes do sistema ou os nós do modelo são mensuráveis, o sistema proposto usa um motor de busca baseado em GA capaz de lidar com nós que não podem ser, ou não são, mensurados. Portanto, essa abordagem permite que os nós do sistema não mensurados possam ser diagnosticados e ainda permite aos engenheiros uma tomada de decisão mais precisa durante a resolução de problemas físicos.

Outro trabalho é o de Chi e Yan (2004) onde é proposto um sistema para realizar o agrupamento de peças e máquinas em um sistema de manufatura celular por meio da Lógica *Fuzzy* e dos Algoritmos Genéticos. Os objetivos por trás dessa abordagem incluem a minimização do número de movimentos entre células, a maximização da densidade de produção dentro das células e a minimização da variação de carga de produção dentro das células. Este trabalho desenvolve um modelo de análise de agrupamento Genético-*Fuzzy* que usa os dados de entrada da matriz de máquinas e peças, a oscilação da demanda de produção e a viabilidade técnica de uma máquina específica como variáveis do sistema para produzir um modelo que resolva os problemas relacionados a células de manufatura.

Ainda tratando de problemas relacionados a células de manufatura tem-se o trabalho de Boulif e Atif (2008) onde é abordada a formação dessas células considerando o comportamento dinâmico do sistema de produção. Para isso, duas estratégias são consideradas. A primeira estratégia, chamada de passiva, mantém a mesma composição das máquinas durante o horizonte de planejamento global, enquanto a segunda, chamada de ativa, permite a execução de uma composição diferente a cada período. Para o problema de formação de células passivas é provado que o mesmo equivale a um problema de formação de células em um ambiente estático. Para o problema de formação de células ativas, é proposto um método baseado em uma heurística de busca de caminho mínimo e em GA. A Lógica *Fuzzy* e os GA são necessários na tomada de decisão de qual a estratégia mais adequada a cada ambiente de manufatura. Os resultados demonstraram que melhores resultados são obtidos quando as técnicas *Fuzzy* e GA atuam em conjunto no controle do sistema de manufatura.

O trabalho de Park et al. (2014) apresenta um método para determinar o desempenho de sistema de manufatura. Este trabalho combina GA e um método de análise de dados a fim de selecionar o melhor cenário para alocação de operador. O método proposto gera, por meio do GA, um conjunto de cenários de alocação de operador e posteriormente esses cenários são avaliados. Como as variáveis do sistema podem ser imprecisas, a Lógica *Fuzzy* é usada para

auxiliar classificação dos dados e na escolha do melhor cenário. Os resultados obtidos a partir de um estudo de caso prático são apresentados e ilustram a eficácia da metodologia com GA e *Fuzzy* atuando em conjunto.

3.6 Pesquisas no Laboratório Tear – Departamento de Computação – UFSCar

O Laboratório de Estratégias de Automação (Tear) do Departamento de Computação da UFSCar desenvolve projetos em que se busca realizar o controle e, conseqüentemente, tomadas de decisões em sistemas de manufatura, com o uso das Redes de Petri e suas variantes e técnicas de Inteligência Artificial como a Lógica *Fuzzy* e os Algoritmos Evolutivos.

Nesse contexto, é apresentado na Figura 3-1 a cronologia dos projetos voltados para o aperfeiçoamento do controle de sistemas de manufatura desenvolvidos aos logo dos anos no laboratório Tear.

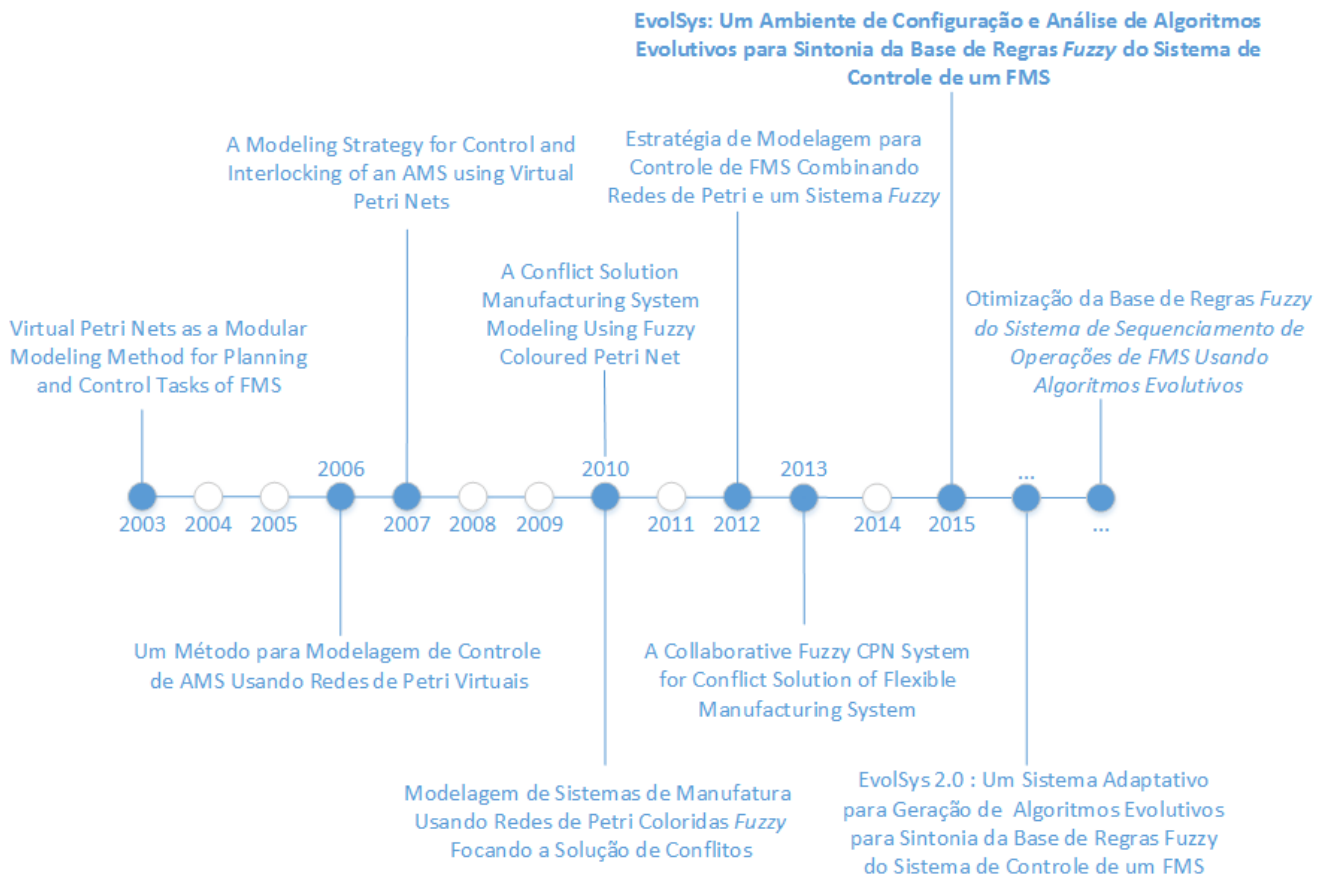


Figura 3-1 - Cronologia dos projetos voltados para o controle de sistemas de manufatura desenvolvidos pelo Tear

Em 2003 foi publicado o artigo “*Virtual Petri Nets as a Modular Modeling Method for Planning and Control Tasks of FMS*” (MORANDIN JR.; KATO, 2003). Nele, foi introduzido um método de modelagem modular baseado no compartilhamento de recursos e no planejamento de processos em FMSs, usando as chamadas Redes de Petri Virtuais (VPN)²². Neste artigo, a ideia era definir uma abordagem para a tarefa de modelagem de um Sistema de Manufatura Automatizado (AMS) propiciando uma alta coesão e um baixo acoplamento do sistema, provocando assim, uma diminuição da dificuldade de compreensão e análise do modelo, além de melhorar a manutenibilidade do mesmo.

Já em 2006 foi apresentado o projeto intitulado “Um Método para Modelagem de Controle de AMS Usando Redes de Petri Virtuais” (ARAÚJO, 2006). Neste projeto, foi proposta uma estratégia de modelagem modular para o planejamento do controle e do

²² Segundo Morandin Jr e Kato (2003), Rede de Petri Virtual ou VPN é um extensão da Rede de Petri Lugar-Transição com a adição de elementos considerados virtuais. Esses elementos são a base para o tratamento das questões relacionadas à modularidade dos AMSs.

intertravamento de um AMS que considere as características inerentes ao seu controle usando as Redes de Petri Virtuais. São propostas no trabalho oito etapas de elaboração do modelo que ponderam os elementos em um nível de especificação mais genérico e que, gradualmente, vão sendo mais detalhadas.

Com esse trabalho, foi possível definir uma arquitetura de sistema que resultou em um desacoplamento entre o intertravamento e o sistema responsável pela tomada de decisão em sistemas de manufatura. Essa nova estratégia de modelagem torna possível evitar eventuais *deadlocks* e tratar, de uma forma mais eficiente, os possíveis conflitos que possam surgir. A ideia foi portanto, separar o sistema de intertravamento do sistema de sequenciamento das operações. Foram usadas regras simples para comprovar a capacidade de usar um modulo separado do intertravamento para realização das tomadas de decisão em sistemas de manufatura.

Em 2007, foi então publicado o artigo intitulado “*A Modeling Strategy for Control and Interlocking of an AMS Using Virtual Petri Nets*” (MORANDIN JR et al., 2007), referente ao projeto de Araújo (2006).

Para exemplificar a estratégia de modelagem proposta em Araújo (2006) e em Morandin Jr. et al (2007) tem-se a Figura 3-2, onde é apresentada uma simplificação de uma Rede de Petri com destaque para os blocos em azul, que são os pontos em que podem ocorrer conflitos.

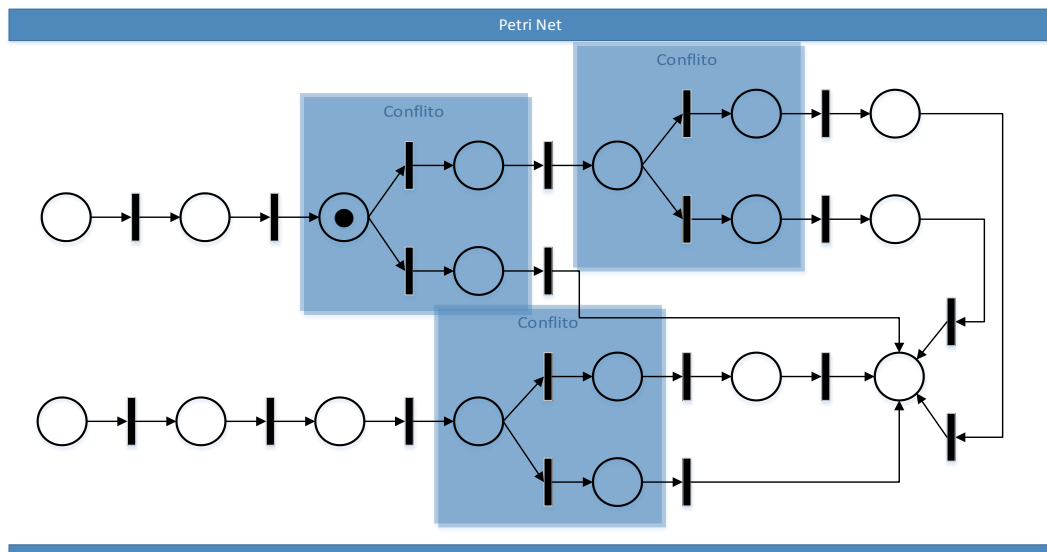


Figura 3-2 - Rede de Petri com destaque nos pontos onde podem ocorrer conflitos.

Na Figura 3-3 é apresentada a remodelagem da Rede de Petri simplificada da Figura 3-2, onde são adicionados os chamados “Places de Controle” propostos em Araújo (2006), em vermelho.

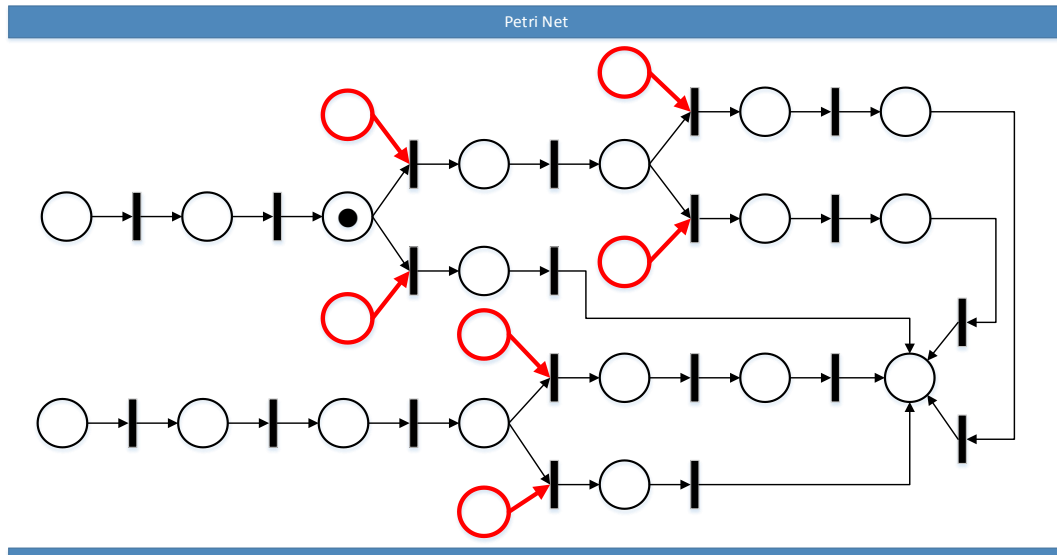


Figura 3-3 - Rede de Petri com os “Places de Controle”, em vermelho, adicionados.

Os “Places de Controle” atuam de modo a permitir a tomada de decisão no sistema e, dessa forma, resolver os conflitos existentes. Todavia, quando esses novos places são adicionados, a rede é dita morta (*dead network*) uma vez que o sistema não consegue prosseguir, pois não há marca em nenhum dos places anteriores em vermelho.

Nesse sentido, a Figura 3-4, apresenta uma visão em perspectiva do modelo onde é apresentado o Sistema de Sequenciamento de Operações (OSS).

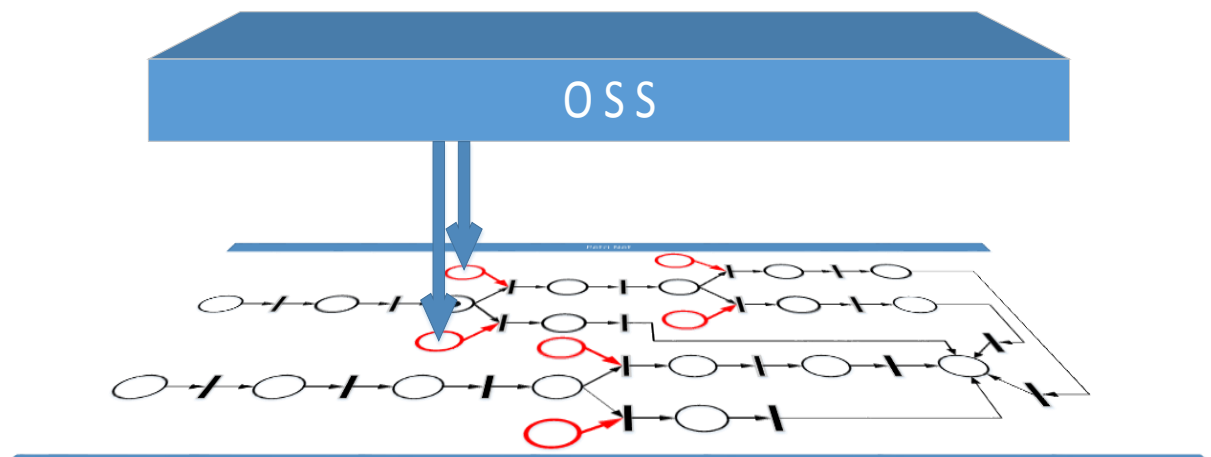


Figura 3-4 - Perspectiva do modelo com visão para o Sistema de Sequenciamento de Operações

O OSS, que é um componente do Sistema de Controle, é o responsável pela tomada de decisão do sistema, na medida em que decide, usando alguma lógica, em qual “Place de Controle” será colocada a marca que indica uma decisão para os conflitos. Na Figura 3-4 tem-se as setas em azul que indicam que o OSS pondera entre colocar a marca em um dos places em vermelho e, desse modo, decide os respectivos conflitos.

Os trabalhos de Araújo (2006) e Morandin et al. (2007) avançaram até esse ponto, contudo, havia a necessidade de averiguar o desempenho dos sistemas quando da utilização de Redes de Petri Colorida *Fuzzy* para solução dos conflitos em sistemas de manufatura.

Surge então o trabalho intitulado “Modelagem de Sistemas de Manufatura Usando Redes de Petri Coloridas *Fuzzy* Focando a Solução de Conflitos” (SGAVIOLI, 2010) e o artigo intitulado “A *Conflict Solution Manufacturing System Modeling Using Fuzzy Coloured Petri Net*” (KATO; MORANDIN JR.; SGAVIOLI, 2010). Nesses trabalhos buscou-se investigar a capacidade de um sistema híbrido CPN – *Fuzzy* em resolver os conflitos de um sistema de manufatura usando, contudo, regras mais complexas do que as empregadas em Araújo (2006) e Morandin et al. (2007). Desse modo, apesar do intertravamento ter sido modelado na CPN, os cálculos para o sequenciamento das operações, embora resolvidos com o uso da Lógica *Fuzzy*, ainda ocorriam internamente à própria CPN. Foi então observado que sistemas CPN – *Fuzzy*, apesar de tratarem bem os conflitos referentes ao sequenciamento das operações, não resolvem os problemas relacionados à complexidade da modelagem, manutenibilidade do sistema e ao alto acoplamento do modelo.

Em 2012, foi então apresentado o projeto “Estratégia de Modelagem para Controle de FMS Combinando Redes de Petri e um Sistema *Fuzzy*” (REIS, 2012). Neste projeto, foi elaborado um sistema, seguindo a estratégia proposta por Araujo (2006) de desacoplamento do sistema em relação ao intertravamento e ao sequenciamento de operações, acrescentando, todavia, um sistema baseado em Regras *Fuzzy* para apoio às tomadas de decisões.

Nesse projeto, uma parte do sistema é destinada a questão do intertravamento, o qual é pautado no modelo em Redes de Petri Coloridas e, de maneira colaborativa, a outra parte do sistema é destinada ao sequenciamento das operações que é pautado na Lógica *Fuzzy*. Desse modo, o encaminhamento da produção, as rotas dos AGVs (*Automated Guided Vehicle*) e as máquinas e *buffers* são modelados em CPN e um sistema baseado em regras *Fuzzy* para a tomada de

decisão é construído usando a ferramenta *Matlab*²³. Além disso, foi desenvolvido um sistema que tinha o objetivo de comunicar a modelagem em CPN com o Sistema de Controle *Fuzzy*.

Em suma, a ideia é que o intertravamento seja realizado por meio das CPNs sendo executado em um Controlador Lógico Programável (PLC) e o sequenciamento das operações seja realizado separadamente, por intermédio de um Sistema *Fuzzy*, este último sendo executado em um computador. Isso torna possível, realizar cálculos mais complexos para decidir as questões relacionadas aos FMSs, mais especificamente aos conflitos, levando em consideração critérios de otimização como *makespan*, *tardiness*, dentre outros. Dessa forma, foi possível controlar o sistema de manufatura garantindo a eficiência do modelo quando ocorrem eventos inesperados, como a quebra de máquinas ou mudanças na gestão da produção, pois são usadas informações em tempo real com base no estado do sistema, para, assim, resolver conflitos em ambientes de manufatura. Com essa proposta, aumenta-se a representatividade e reduz-se o acoplamento do modelo proporcionando baixos esforços na adaptação a diferentes roteiros de fabricação.

Em 2013, foi então publicado o artigo intitulado “*A Collaborative Fuzzy CPN System for Conflict Solution of Flexible Manufacturing System*” (REIS et al., 2013) referente ao projeto de Reis (2012). Nesse artigo, foi usada a modelagem de uma fábrica simulada baseada em um sistema de manufatura que utilizou 6 máquinas compostas por *Load/Unload* (carregamento e descarregamento) e *buffers* de entrada e saída, 3 AGVs, uma estação de *Load/Unload* e um *AGV Park* (estacionamento para AGVs). Desse modo, um sistema de controle foi construído e testado a partir dessa estratégia de modelagem.

A ideia seguinte é desenvolver projetos relacionados à otimização da Base de Regras *Fuzzy* do sistema de sequenciamento de operações de um FMS usando Algoritmos Evolutivos. Entretanto, foi observado que, há a necessidade de, primeiramente, realizar um projeto que propicie a definição de qual configuração de Algoritmo Evolutivo deve ser usada para realizar essa otimização, ou seja, é necessário definir, em um primeiro momento, qual o conjunto de parâmetros que apresentam os resultados mais satisfatórios para o problema em questão.

No entanto, percebe-se que, a definição do EA que, de fato, gera a FRB sintonizada para o Sistema de Controle *Fuzzy* de um FMS não é trivial, haja vista a quantidade de EAs que

²³ Matlab é um ambiente interativo para cálculo numérico e programação. Pode ser usado para análise de dados, desenvolvimento de algoritmos e criação de modelos e aplicações.

Disponível em: < <http://www.mathworks.com/products/matlab/> > Acessado em 29/08/2014

podem ser criados²⁴, com as mais diversas configurações e que resultam em desempenhos distintos do sistema de manufatura. Essas configurações dizem respeito ao número de indivíduos, métodos de seleção, formas de recombinação, número de gerações, taxas de mutação, etc., todas com influência na convergência do EA.

Foi então concebido o presente projeto que busca propiciar, por intermédio da construção de um ambiente de configuração e análise de desempenho, a definição do Algoritmos Evolutivos que seja capaz de gerar uma Base de Regras *Fuzzy* que conduz à sintonia do Sistema de Controle de um dado FMS.

Projetos futuros estão relacionados à construção de um sistema adaptativo para geração de Algoritmos Evolutivos para sintonia da FRB do Sistema de Controle de um FMS e à proposta de otimização da FRB do sistema de sequenciamento de operações de FMS usando Algoritmos Evolutivos.

3.7 Considerações Finais

Neste capítulo alguns artigos correlatos foram analisados, abordando temas que envolviam a aplicação das técnicas estudados nos mais variados problemas que foram desde auxílio a diagnóstico médico até resolução de conflitos e tomada de decisão em sistemas de manufatura. Além disso, abordou-se as questões relacionadas às pesquisas do laboratório Tear do Departamento de Computação da UFSCar e suas relações com o projeto aqui proposto.

Deste modo, percebe-se que a Lógica *Fuzzy* e os GAs podem atuar em conjunto, colaborativamente ou de forma híbrida na resolução de problemas em diversas áreas, em especial em sistemas de manufatura, obtendo, com isso, resultados significativos. Nesses artigos, foi possível perceber que as técnicas atuam no sentido de propiciar uma melhor tomada de decisão e solução de conflitos o que, no caso de sistemas de manufatura, corrobora com a melhoria dos processos produtivos e conseqüentemente com o controle do sistema fabril.

No presente trabalho, haja vista a dificuldade inerente à geração do EA (várias configurações são possíveis) que gere de fato a FRB sintonizada para um FMS, a ideia é a

²⁴ Levando em consideração que configurações de parâmetros diferentes geram EAs diferentes.

construção de um ambiente que permita a configuração e a análise de Algoritmos Evolutivos ou EAs para possibilitar a definição dessa FRB sintonizada.

PROPOSTA DO TRABALHO

4.1 Considerações Iniciais

Neste capítulo será apresentada a proposta de construção de um ambiente que permita a configuração e a análise de desempenho de Algoritmos Evolutivos para permitir a definição da FRB sintonizada para o Sistema de Controle de um FMS.

Também será apresentada a visão global do projeto de Controle de Sistema de Manufatura Flexível do Laboratório de Estratégias de Automação (Tear) do Departamento de Computação da UFSCar destacando a relação com o presente trabalho.

Além disso, serão apresentadas as propostas relacionadas à definição dos genótipos dos indivíduos, à forma como se dará o cálculo do *fitness* e à implementação dos operadores evolutivos de elitismo, seleção, cruzamento e mutação.

Por fim, serão abordadas as considerações finais a respeito da proposta e o que se espera com a execução da mesma.

4.2 Proposta do Trabalho

Um Sistema de Manufatura Flexível fornece eficiência à produção automatizada mantendo a flexibilidade da produção (NANVALA, 2011). Entretanto, esses benefícios vêm acompanhados de importantes decisões que devem ser tomadas em relação à programação da produção e ao sequenciamento das operações.

Nesse processo, o auxílio de ferramentas computacionais é essencial para que as decisões sejam tomadas da melhor forma possível.

Nesse sentido, técnicas inteligentes são usadas para a tomada de decisão na programação de FMS destacando-se a Lógica *Fuzzy* e os Algoritmos Genéticos (NANVALA, 2011).

Portanto, a proposta no presente trabalho é realizar a construção de um ambiente que permita a configuração e a análise de desempenho de Algoritmos Evolutivos, em especial, uma extensão do Algoritmo Genético, para sintonia da FRB do Sistema de Controle *Fuzzy* de um dado FMS. A ideia é empregar um sistema baseado em Algoritmos Evolutivos na geração de Bases de Regras de um Sistema *Fuzzy* de um Sistema de Manufatura Flexível e assim tornar possível a verificação do comportamento dos EAs nesse contexto.

Para um melhor entendimento da proposta, será demonstrada, a seguir, a visão global do projeto de controle de sistema de manufatura culminando na delimitação do escopo do projeto aqui proposto.

4.2.1 Visão Global do Projeto de Controle de Sistema de Manufatura Flexível

Dado o modelo do FMS proposto por Reis et al. (2013), que pode ser visto na Figura 4-1, tem-se o diagrama de blocos do FMS com a Fábrica, o Sistema de Sequenciamento de Operações (OSS), o Sistema *Fuzzy* (FS) e a Base de Regras *Fuzzy* (FRB).

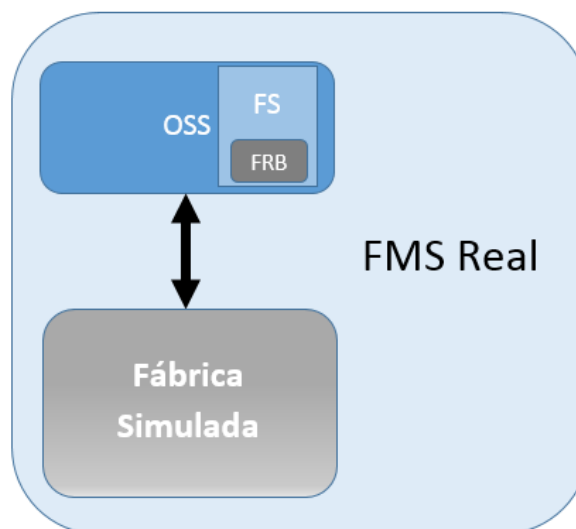


Figura 4-1 - Diagrama de Blocos do FMS (modelo proposto por Reis et al. (2013))

Neste modelo, o sistema é dividido em blocos, sendo o bloco designado Fábrica Real responsável pelo intertravamento e o bloco OSS responsável pelos cálculos mais complexos para o sequenciamento das operações no FMS. Neste projeto, proposto por Reis et al. (2013), a FRB, gerada pelo especialista, possibilita a tomada de decisão no sistema uma vez que contém regras que determinam o sequenciamento das operações do FMS em questão.

A partir desse ponto, a ideia é que, posteriormente, se tenha um sistema simulado em software e que contenha o mesmo conjunto de componentes do, aqui chamado, FMS Real proposto em Reis et al. (2013), com o advento de um Sistema Evolutivo. Esse sistema, chamado de FMS Simulado, pode ser visto na Figura 4-2.

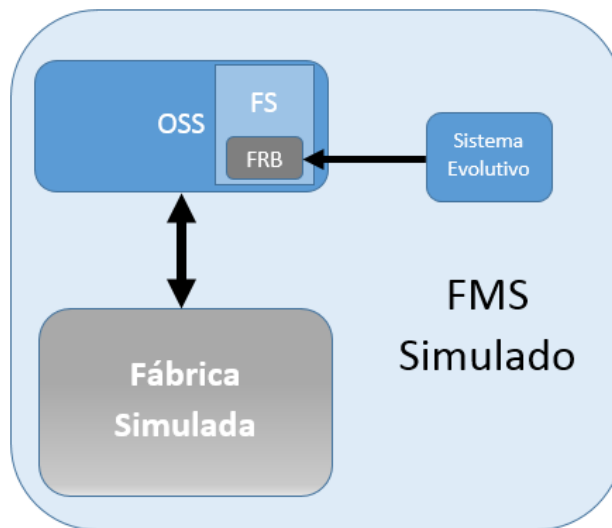


Figura 4-2 - Diagrama de Blocos do FMS Simulado

No sistema apresentado na Figura 4-2, a cada simulação, o Sistema Evolutivo trocará a Base de Regras do FS até encontrar aquela que satisfaça um dado critério de otimização. A ideia é que, enquanto o FMS simulado busca um conjunto de regras sintonizada para o sistema, a Fábrica Real continua executando com o conjunto de regras atual (regras que já estavam sendo usadas anteriormente). Quando a simulação encontra o conjunto de regras que satisfaz um ou mais critérios de otimização definidos (*makespan*, *due date*, *tardiness*, etc.), a FRB do FMS Real é substituída pela FRB gerada pelo Sistema Evolutivo do FMS Simulado.

Todavia, para que a colaboração entre a Lógica *Fuzzy* e o Algoritmo Evolutivo aconteça a contento, é necessário, primeiramente, que seja construído um ambiente que permita a configuração e a análise dos EAs. A concepção e o uso desse ambiente tornam possível, por

meio de testes, definir quais serão os parâmetros que gerarão a Base de Regras *Fuzzy* sintonizada para um dado FMS.

Para a concretização desse projeto, é necessário, portanto, que seja projetado o Sistema Evolutivo. A proposta, conseqüentemente, é a de construir um Sistema Evolutivo para permitir a configuração e a análise de desempenho dos EAs e, desse modo, proporcionar a definição da FRB sintonizada do Sistema de Controle de um FMS. Nesse sentido, foi construído o sistema que será apresentado no Capítulo 5 no tópico 5.5 – Sistema Evolutivo - *EvolSys* (página 111).

4.2.2 Proposta de Sistema Evolutivo

No projeto aqui proposto, foi elaborado um Sistema Evolutivo para possibilitar a configuração e a análise de EAs para possibilitar a definição da FRB sintonizada do Sistema de Controle *Fuzzy* de um FMS.

No Sistema Evolutivo deste trabalho, os parâmetros propostos para serem configurados para a geração da FRB serão:

1. Parâmetros relacionados ao Sistema *Fuzzy*
2. Parâmetros relacionados ao EA
3. Parâmetros relacionados à Execução do EA

Mais detalhes sobre a configuração desses parâmetros podem ser vistos no Capítulo 5 no tópico 5.2 – Projeto do Sistema Evolutivo (página 98) quando da descrição do Caso de Uso Configurar Parâmetros.

Em relação ao sistema *Fuzzy*, a atuação se dá na Base de Regras, como é destacado na Figura 4-3.

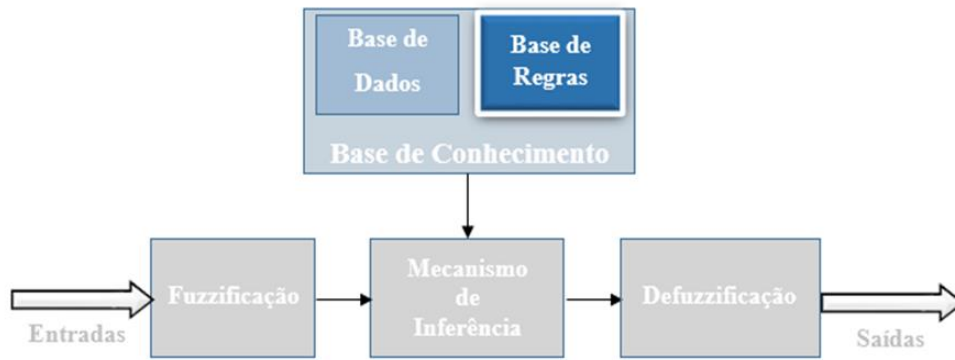


Figura 4-3 - Sistema de Inferência *Fuzzy* Mamdani - Destaque para a Base de Regras

Na Figura 4-3 tem-se o Sistema de Inferência *Fuzzy* Mamdani onde a Base de Regras é destacada, pois será nesse ponto que se pretende atuar no projeto.

Em relação à Base de Regras *Fuzzy*, a Figura 4-4 destaca o ponto de atuação do Sistema Evolutivo.

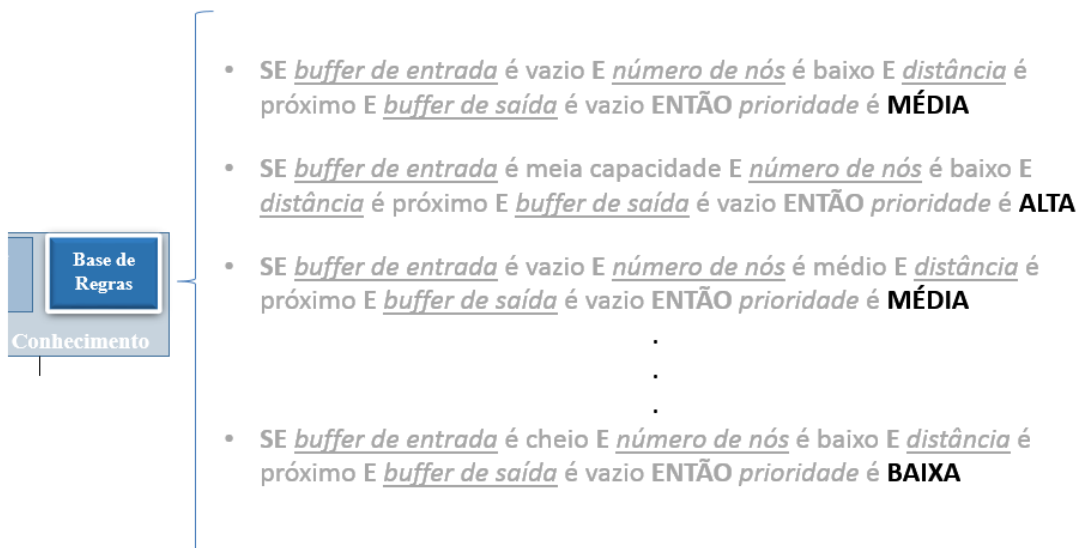


Figura 4-4 - Detalhando a Base de Regras *Fuzzy* - Em destaque o ponto de atuação do projeto (saída das regras)

Na Figura 4-4 tem-se exemplos de regras que possuem as variáveis de entrada²⁵:

- “*Buffer de entrada*” (vazio, meia capacidade ou cheio)
- “*Número de nós*” (baixo, médio ou alto)

²⁵ Vale ressaltar que tanto a quantidade de variáveis de entrada quanto a quantidade de variáveis de saída das regras podem ser alteradas no ambiente proposto a depender da situação que se pretende abordar.

- “*Distância*” (próximo, médio ou longe)
- “*Buffer de saída*” (vazio, meia capacidade ou cheio)

Tem-se também, a variável de saída:

“*Prioridade*” (baixa, média ou alta)

No presente projeto, o Sistema Evolutivo atuará em pontos específicos da FRB. Esses pontos serão as saídas das regras²⁶, ou seja, o sistema atuará na(s) variável(eis) de saída das regras. Na Figura 4-4 a variável usada será a “*Prioridade*”. Neste caso, serão gerados n cromossomos, onde cada gene dos cromossomos representa uma das saídas da FRB.

Na Figura 4-5 tem-se a representação de como são modelados os cromossomos para o exemplo da Figura 4-4.

	0	Baixa
	1	Média
	2	Alta

	R1	R2	R3	R4	. . .	Rn
Cromossomo FRB1	1	2	1	2	. . .	0
Cromossomo FRB2	0	2	2	1	. . .	2
					⋮	
Cromossomo FRBm	0	1	1	2	. . .	1

Figura 4-5 - Exemplo de modelagem dos indivíduos do EA para o exemplo da Figura 4-4

Na Figura 4-5 tem-se a definição de uma gramática com 3 valores (0,1 e 2) que representam as possíveis saídas da FRB da Figura 4-4. Cada indivíduo (cromossomo) representa um FRB do problema e cada um dos genes representa uma saída de uma regra da FRB. Como pode ser visto, pode-se ter n regras (de R1 até Rn) e m indivíduos (FRB1 até FRBm) a depender da configuração que é feita para execução do Sistema Evolutivo²⁷.

²⁶ O EA proposto atua apenas na(s) variável(is) de saída das regras *Fuzzy* gerando combinações de bases de regras que serão avaliadas de acordo com o problema em questão.

²⁷ Importante destacar que, no presente trabalho, todas as regras são tratadas e, por consequência, não há exclusão de regras que podem ser consideradas irrelevantes.

A ideia, portanto, é que, baseados no problema e na forma como se pretende abordá-lo, os parâmetros do Sistema Evolutivo sejam configurados e uma FRB seja gerada a cada novo teste.

Como pode ser visto na Figura 4-6, após o sistema ser parametrizado executa-se o EA e, por fim, uma FRB é gerada.

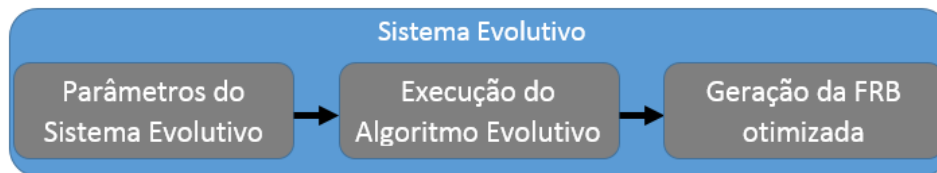


Figura 4-6 - Etapas do Sistema Evolutivo

O intuito, com a realização desse projeto, é permitir a configuração e a avaliação do comportamento de cada EA criado no que diz respeito à criação de um FRB sintonizada para o Sistema de Controle *Fuzzy* de um FMS.

4.2.3 Definição dos Genótipos dos Indivíduos

Para definição dos genótipos, ou seja, das características genéticas dos indivíduos no EA, é necessário definir o tamanho do cromossomo e o tamanho da gramática.

O tamanho do cromossomo diz respeito à quantidade de genes, ou seja, quantidade de regras que cada FRB gerada (indivíduo gerado) irá conter. O seu valor é dado pela multiplicação entre cada um dos valores possíveis para cada variável de entrada. Esses valores são definidos quando da parametrização do Sistema Evolutivo, o que pode ser visto na Figura 5-12 (página 112).

Considere um exemplo (aqui chamado de Exemplo 1) onde o valor do campo Quantidade de Variáveis de Entrada é igual a 3 ou seja, tem-se, para esse exemplo, 3 variáveis de entrada. Considere também que para as variáveis 1, 2 e 3 as quantidades de valores possíveis são 5, 3 e 2 respectivamente. Tem-se então a situação apresentada na Tabela 4-1.

Tabela 4-1 - Configuração das variáveis de entrada para o Exemplo 1.

3 Variáveis de Entrada		
Nome	Quantidade de Valores Possíveis para Cada Variável de Entrada	Termo Linguístico
Variável 1	5 Valores possíveis	Muito Baixo Baixo Médio Alto Muito Alto
Variável 2	3 Valores possíveis	Baixo Médio Alto
Variável 3	2 Valores possíveis	Baixo Alto

Para esse exemplo, o tamanho do cromossomo é igual a multiplicação entre 5, 3 e 2 que são os valores referentes a “Quantidade de Valores Possíveis para Cada Variável de Entrada”. Essa multiplicação resulta no valor 30, ou seja, tem-se para esse problema, um cromossomo de tamanho 30.

Já o tamanho da gramática trata dos possíveis valores que podem ocupar cada gene do cromossomo de um indivíduo, ou seja, diz respeito aos valores de saída de cada regra de cada FRB gerada (indivíduo gerado). O seu valor é dado pela multiplicação entre cada um dos valores possíveis para cada variável de saída. Esses valores são definidos quando da parametrização do Sistema Evolutivo, o que pode ser visto na Figura 5-12 (página 112).

Ainda neste mesmo exemplo, considere que o valor do campo Quantidade de Variáveis de Saída é igual a 2 ou seja, tem-se, para esse exemplo, 2 variáveis de saída. Considere também que para as variáveis 1 e 2 as quantidades de valores possíveis são 3 e 5 respectivamente. Tem-se então a situação apresentada na Tabela 4-2.

Tabela 4-2 - Configuração das variáveis de saída para o Exemplo 1.

2 Variáveis de Saída		
Nome	Quantidade de Valores Possíveis para Cada Variável de Saída	Termo Linguístico
Variável 1	3 Valores possíveis	Baixo Médio Alto
Variável 2	5 Valores possíveis	Muito Baixo Baixo Médio Alto Muito Alto

Portanto, para esse exemplo, o tamanho da gramática é igual a multiplicação entre 3 e 5 que são os valores referentes a “Quantidade de Valores Possíveis para Cada Variável de Saída”. Essa multiplicação resulta no valor 15, ou seja, tem-se para esse problema, uma gramática que varia entre 0 a 14 formas de saída inclusive.

A Figura 4-7 ilustra a representação do cromossomo do Exemplo 1 para valores de saída das regras fictícios.

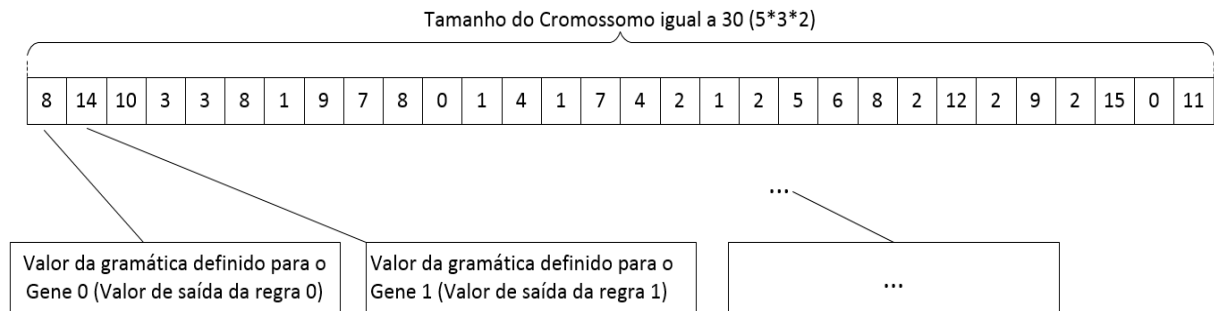


Figura 4-7 - Configuração do cromossomo do Exemplo 1 para 3 variáveis de entrada (Variável 1 => 5 Valores Possíveis; Variável 2 => 3 Valores Possíveis; Variável 3 => 2 Valores Possíveis) e 2 variáveis de saída (Variável 1 => 3 Valores Possíveis; Variável 2 => 5 Valores Possíveis)

A Tabela 4-3 mostra a representação das variáveis linguísticas para cada uma das possíveis saídas para o Exemplo 1.

Tabela 4-3 - Relação entre as variáveis de saída e seus respectivos valores linguísticos nas saídas das regras *Fuzzy*

Variáveis de Saída (Alelos presentes nos Genes dos Indivíduos)	Variável 1 (3 Valores Possíveis) +	Variável 2 (5 Valores Possíveis)
0	Baixo	Muito Baixo
1	Baixo	Baixo
2	Baixo	Médio
3	Baixo	Alto
4	Baixo	Muito Alto
5	Médio	Muito Baixo
6	Médio	Baixo
7	Médio	Médio
8	Médio	Alto
9	Médio	Muito Alto
10	Alto	Muito Baixo
11	Alto	Baixo
12	Alto	Médio
13	Alto	Alto
14	Alto	Muito Alto

Na Tabela 4-3 portanto, tem-se que, caso o valor de um determinado gene seja, por exemplo, igual a 8, tem-se que o valor de saída linguístico para a regra em questão é igual a Médio-Alto e assim por diante para todas as regras presentes nos genes dos indivíduos do EA.

4.2.4 Cálculo do *Fitness*

Para o presente projeto, foi necessário definir um método para realização do cálculo do *fitness* de cada um dos indivíduos das populações do Algoritmo Evolutivo.

Neste caso, como ainda não há um ambiente para avaliação do EA criado, o processo de cálculo do *fitness* foi desenvolvido por intermédio da definição de uma Base de Regras *Fuzzy* Ideal (indivíduo com *fitness* máximo), ou seja, uma base de regras que retorna o melhor valor possível para um dado critério de otimização (*makespan*, *tardiness*, *due date*, dentre outros).

A FRB Ideal foi definida levando em consideração a forma como, em geral, um especialista define uma FRB para um sistema de controle de manufatura. Nesse sentido, o cromossomo do indivíduo ideal, ou seja, da FRB Ideal, é dividido em grupos de acordo com a quantidade de variáveis de saída, sendo que os valores dos alelos de cada gene do cromossomo são distribuídos em cada um dos grupos²⁸.

²⁸ Os significados das terminologias “cromossomo”, “gene”, “alelos”, “loco”, etc., no contexto de algoritmos genéticos, podem ser vistas no tópico 2.4.2 - Terminologia (página 44).

Para demonstrar o processo de criação da FRB Ideal considere um novo exemplo o qual é exibido na Figura 4-8.

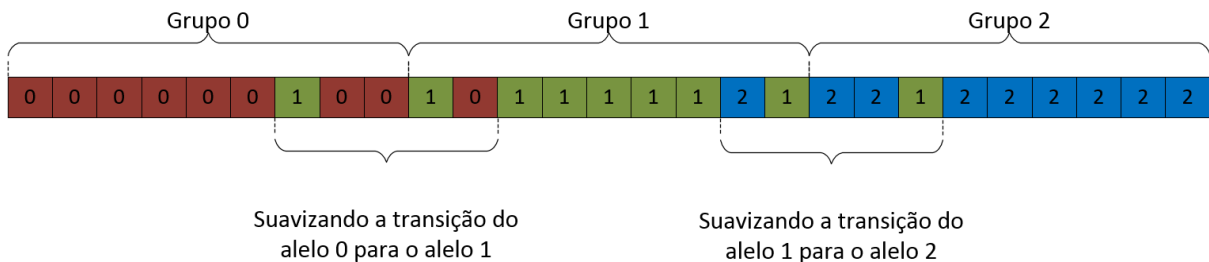


Figura 4-8 - FRB Ideal (Cromossomo do Indivíduo que leva ao valor de *fitness* máximo)

Na Figura 4-8 tem-se um cromossomo de tamanho igual 27. Para chegar a esse valor considerou-se que a quantidade de variáveis de entrada é igual a 3 e a quantidade de valores possíveis para cada variável de entrada também é igual a 3. Neste caso, multiplicando cada uma das 3 variáveis entre si ($3 * 3 * 3$) tem-se um tamanho de cromossomo igual a 27. Considerou-se também, que existe apenas uma variável de saída com 3 valores de saída possíveis, resultando em uma gramática de tamanho 3 (cada gene pode ter um alelo com valor igual a 0, 1 ou 2).

Com esses dados, é possível realizar a construção da FRB Ideal. Nesse processo, divide-se o cromossomo em 3 grupos (3 valores de saída possíveis) e, a partir da inserção dos alelos nos genes, suaviza-se a transição entre os grupos como pode ser visto na Figura 4-8.

Com a FRB Ideal construída, é possível realizar o cálculo do *fitness* dos indivíduos. Esse cálculo é realizado comparando o cromossomo de cada um dos indivíduos com a FRB Ideal para verificar o quão distinto é esse cromossomo em relação à própria FRB Ideal.

Para demonstração do cálculo do *fitness* considere o exemplo a seguir.

Considere, por exemplo, para a construção de uma base de regras, a definição de que se tem 2 Variáveis de Entrada, com 2 valores possíveis para cada uma delas. Neste caso, o tamanho do cromossomo será igual a 2(dois valores possíveis para a Variável 1) vezes 2(dois valores possíveis para a Variável 2) que resulta no tamanho de cromossomo igual a igual a 4. Em relação às saídas, tem-se 3 Variáveis de Saída com 2 valores possíveis para a Variável 1, 2 valores possíveis para a Variável 2 e 3 valores possíveis para a Variável 3. O resultado da multiplicação desses valores ($2 * 2 * 3$) é igual a 12, ou seja, para cada gene é possível que cada alelo correspondente pode assumir algum valor entre 0 e 11 inclusive.

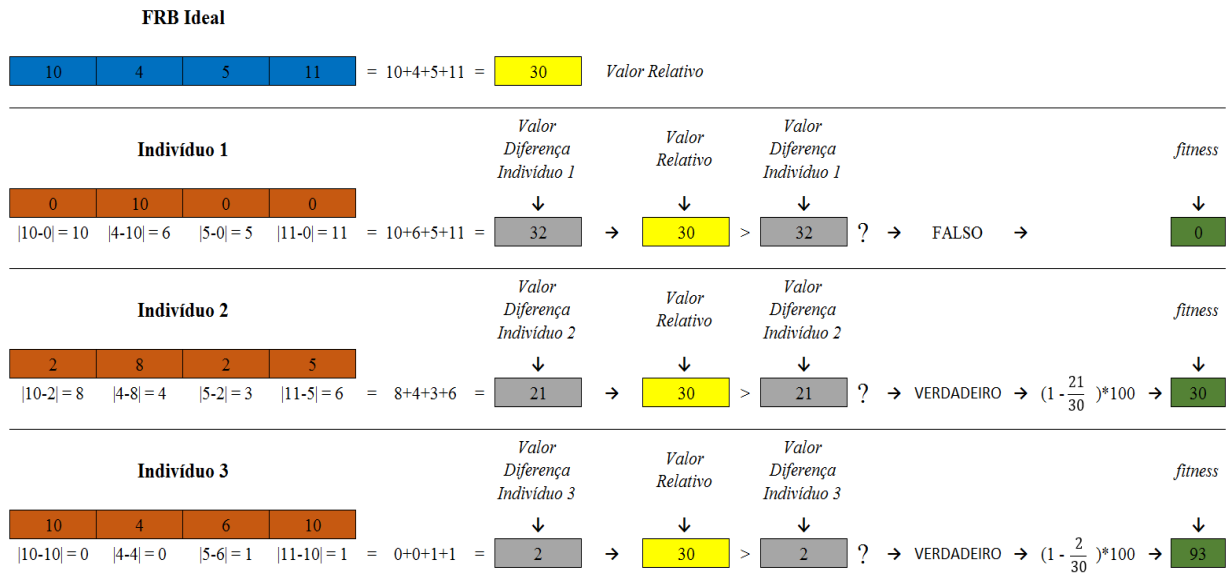


Figura 4-9 - Cálculo do *Fitness* de 3 indivíduos para uma dada FRB Ideal.

Na Figura 4-9 é possível visualizar 3 exemplos de cromossomos que serão utilizados para demonstrar como se comportará o algoritmo criado para a definição do valor do *fitness* de cada um dos respectivos indivíduos. Neste caso, quanto menor o valor obtido a partir da comparação entre o cromossomo do indivíduo e a FRB Ideal, menor será a diferença entre eles, o que, conseqüentemente, resulta em um *fitness* alto. O contrário também é válido, ou seja, quanto maior for o valor obtido na comparação, maior será a diferença entre o cromossomo do indivíduo e a FRB Ideal, o que, conseqüentemente, resulta em um *fitness* baixo. Em outras palavras, quanto maior for a semelhança entre o indivíduo e a FRB Ideal maior será o valor do *fitness* e quanto menor for a semelhança entre o indivíduo e a FRB Ideal menor será o valor do *fitness*.

Esse processo pode ser descrito como um cálculo de similaridade entre vetores, como na chamada **Distância de Manhattan**. Na Figura 4-9 pode perceber como se comporta o algoritmo para realização do cálculo do *fitness* dos indivíduos no EA

Primeiramente, calcula-se o módulo da diferença entre cada um dos genes da FRB Ideal (em azul) em relação aos genes, (no mesmo loco – posição no cromossomo), do cromossomo do indivíduo (em laranja). Para o Indivíduo 1 do exemplo da Figura 4-9, o valor obtido para o primeiro gene foi igual a 10. Esse valor é o resultado da subtração (em módulo) entre o alelo do gene[0] da FRB Ideal e o alelo do gene[0] do Indivíduo 1.

Após o cálculo dos módulos das diferenças em todos os genes, soma-se esses valores resultando num valor (em cinza), aqui chamado de Valor Diferença, que demonstra o quão distinto é o cromossomo do indivíduo em relação à FRB Ideal. Como dito anteriormente, quanto menor for esse valor, menor é a diferença entre o cromossomo do indivíduo e a FRB Ideal e, consequentemente, quanto maior for esse valor, maior é a diferença entre o cromossomo do indivíduo para com a FRB Ideal.

A seguir, é verificado se o valor resultante da soma de todos os alelos da FRB Ideal (em amarelo), aqui chamado de Valor Relativo, é maior que o Valor Diferença dos respectivos indivíduos (em cinza).

Se a resposta for, em valor verdade, igual a FALSO, considera-se que o indivíduo tem uma diferença muito grande em relação a FRB Ideal e, por consequência, é atribuído um *fitness* igual a 0 (zero) para esse indivíduo. Este caso pode ser visto na Figura 4-9 para o Indivíduo 1.

Caso a resposta da comparação resulte em um valor verdade igual a VERDADEIRO é necessário realizar um cálculo para definir, o quão distinto é o cromossomo do indivíduo em relação à FRB Ideal e, por conseguinte, definir qual será o *fitness* do indivíduo em questão. Este caso também pode ser visto na Figura 4-9 para os indivíduos 2 e 3.

O cálculo é realizado subtraindo 1 (valor máximo considerado para o *fitness*) da relação entre o Valor Diferença para o indivíduo em questão (em cinza) e o Valor Relativo da FRB Ideal (em amarelo). Esse valor é multiplicado por 100 para se obter um valor que varia de 0 a 100 inclusive. Para o Indivíduo 2 da Figura 4-9 obteve-se o *fitness* igual a 30 para uma escala que varia de 0 a 100 inclusive o que representa que o Indivíduo 2 se parece com a FRB Ideal em 30%. Para o Indivíduo 3 da Figura 4-9 obteve-se o *fitness* igual a 93 para uma escala que varia de 0 a 100 o que representa que o Indivíduo 3 se parece com a FRB Ideal em 93%.

Com isso, percebe-se que o Indivíduo 1 é bem distinto da FRB Ideal, o Indivíduo 2 possui uma semelhança baixa e o Indivíduo 3 é muito semelhante à FRB Ideal como pode ser constatado na Figura 4-9.

4.2.5 Operadores Evolutivos de Elitismo, Seleção, Cruzamento e Mutação

Para execução do EA, foi necessário definir a forma de aplicação dos operadores evolutivos de Elitismo, Cruzamento e Mutação.

Para a execução do operador de Elitismo, obtém-se a população e, baseado na cota estipulada para o elitismo (porcentagem de indivíduos da geração seguinte gerados a partir de elitismo), salva-se uma quantidade n de indivíduos (depende de configuração) da geração atual, na próxima geração. Esses indivíduos são selecionados de acordo com o *fitness* de cada um, sendo que os que tiverem maior *fitness* passarão para a próxima geração.

Em relação ao Operador de Seleção, este se dá a partir da definição da quantidade de grupos de progenitores, da definição da proporção dos grupos dos possíveis progenitores e da taxa de seleção (quantidade de indivíduos que serão selecionados para serem os possíveis progenitores). A Figura 4-10 demonstra como se dá esse processo.

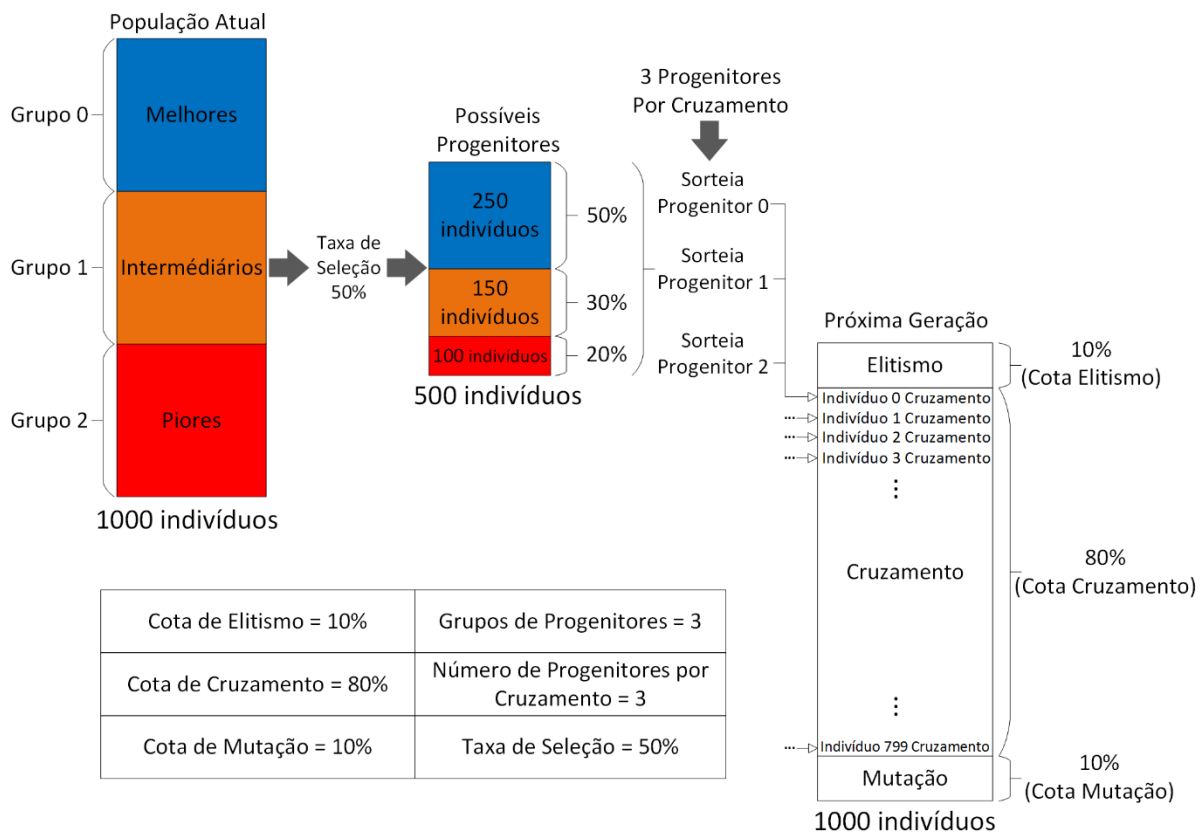


Figura 4-10 - Seleção dos possíveis progenitores para realização dos cruzamentos

Considerando o exemplo da Figura 4-10, tem-se um caso onde foi definido o número de 3 grupos de progenitores, ou seja, a população foi dividida em 3 grupos. Proporcionalmente, os grupos, como são 3, foram divididos em Melhores, Intermediários e Piores. Como a Taxa de Seleção é igual a 50%, serão selecionados, para uma população de 1000 indivíduos, 500 indivíduos para serem os possíveis progenitores. Essa seleção se dá de acordo com a proporção estabelecida para os grupos de possíveis progenitores. Para o exemplo da Figura 4-10 a

proporção dos 3 grupos foi definida em 50% para o grupo 0 (em azul), 30% para o grupo 1 (em laranja) e 20% para o grupo 2 (em vermelho). A formação do grupo dos possíveis progenitores se deu da seguinte forma:

- O grupo 0 é composto de 250 indivíduos (50% dos 500 selecionados para serem os possíveis progenitores) selecionados a partir do grupo dos Melhores Indivíduos (em azul).
- O grupo 1 é composto de 150 indivíduos (30% dos 500 selecionados para serem os possíveis progenitores) selecionados a partir do grupo dos Intermediários (em laranja).
- O grupo 2 é composto de 100 indivíduos (20% dos 500 selecionados para serem os possíveis progenitores) selecionados a partir do grupo dos Piores (em vermelho).

A partir daí, tem-se a fase de cruzamentos, onde são definidos, por intermédio de alguns dos métodos de seleção (roleta, torneio de 2, torneio de 3), quais serão os progenitores responsáveis por gerar os indivíduos da próxima população no que diz respeito ao Operador de Cruzamento.

Para a execução do Operador de Cruzamento os parâmetros necessários são:

- A população onde serão aplicados os cruzamentos
- A cota de cruzamento (porcentagem de indivíduos da geração seguinte gerados a partir de cruzamentos)
- A lista dos possíveis progenitores selecionados a partir da população
- O método de seleção dos possíveis progenitores (torneio de 2, roleta, etc.)
- A quantidade de progenitores por cruzamento
- A quantidade de cortes no cromossomo (em quantas partes os cromossomos dos indivíduos serão divididos para a realização da recombinação - cruzamento).

No exemplo da Figura 4-10 foi definido que seriam usados 3 progenitores por cruzamento. Isso significa que os genes de 3 indivíduos serão recombinados para a geração de um único indivíduo da nova população. Os processos de seleção dos 3 progenitores (para o exemplo dado) e de recombinação para geração de um novo indivíduo se repetem até que toda a cota de cruzamentos seja atingida. Para o exemplo da Figura 4-10, como a cota de cruzamento é igual a 80%, ocorrerão 800 cruzamentos para geração de 800 indivíduos (80% de 1000 indivíduos) para a composição da população em relação ao operador de cruzamento.

Em relação ao cruzamento, ou seja, a recombinação dos genes para geração de um indivíduo, tem-se o exemplo da Figura 4-11.

Neste exemplo, definiu-se como 3 o número de indivíduos responsáveis por serem os progenitores de um dos indivíduos da nova população. Ainda nesse exemplo, foi definido como 2 o número de cortes dos cromossomos dos progenitores.

Após os cortes nos cromossomos dos progenitores, obteve-se 3 grupos de subcromossomos. Esses grupos foram chamados de Grupos de Cortes e cada um deles contém uma parte (nas mesmas posições em relação aos cromossomos dos progenitores) do cromossomo de cada um dos progenitores como pode ser visto na Figura 4-11.

Em seguida, esses grupos são inseridos em uma lista, aqui chamada de Grupo Corte Total, a qual contém, em cada uma das suas posições, um dos Grupos de Corte obtidos após a realização dos cortes.

A seguir, a lista Grupo Corte Total é percorrida e, em cada uma das posições, é sorteado um dos subcromossomos para formar o cromossomo do novo indivíduo.

No caso do exemplo da Figura 4-11 para a primeira posição da lista Grupo Corte Total sorteou-se o valor 1. Neste caso, a primeira parte do novo indivíduo será gerado pelo subcromossomo em laranja do grupo de Corte 0. Em seguida, a lista continua sendo percorrida até que o cromossomo do novo indivíduo seja completamente gerado. Em seguida, um novo grupo de progenitores é sorteado para geração de um outro indivíduo e o processo se repete até que todos os indivíduos que devem ser gerados por cruzamento sejam obtidos.

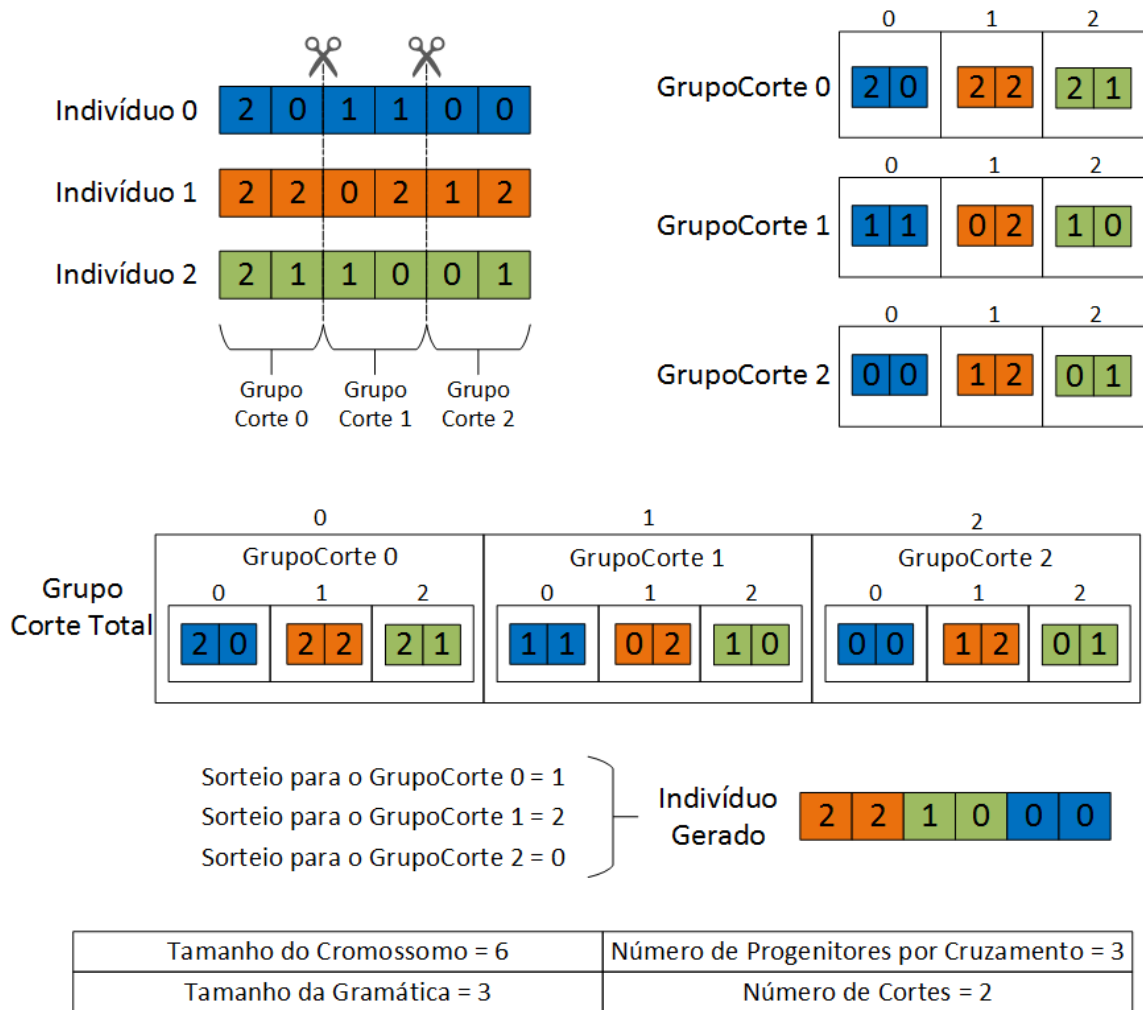


Figura 4-11 - Cortes no cromossomo dos progenitores para geração de um indivíduo da nova população

É importante ressaltar que no cruzamento, o novo indivíduo deve ter em seu cromossomo partes de indivíduos diferentes para que esse processo seja, de fato, uma recombinação genética e não uma aplicação de elitismo ou clonagem.

Para o Operador de Mutação o processo ocorre como na Figura 4-12.

Na Figura 4-12 tem-se o algoritmo usado para execução do operador de mutação do Algoritmo Evolutivo. Baseado na cota de mutação, tem-se a definição de quantos indivíduos deverão sofrer mutação. A partir daí, um indivíduo da população é sorteado e o seu cromossomo é percorrido gene a gene para aplicar, ou não o processo de mutação. A condição de aplicação da mutação depende da Taxa de Mutação estipulada. Por exemplo, se a Taxa de Mutação for de 10%, isso significa que cada gene do indivíduo selecionado tem a chance de 1 em 10 de sofrer mutação. No algoritmo, para realizar a verificação de mutação em um gene é realizado

primeiramente um sorteio de um número de 0 a 100. Caso esse número sorteado seja menor ou igual que a Taxa de Mutação aplica-se a mutação, modificando o alelo do gene em questão por um outro alelo diferente e contido na gramática do EA. Caso esse número seja superior à Taxa de Mutação, o gene não é alterado e o processo se repete para o próximo gene do indivíduo até que todos os genes tenham sido verificados. Como pode ser observado, pode ocorrer a situação onde nenhum gene do indivíduo é alterado. Neste caso, o contador de indivíduos mutacionados não é incrementado e seleciona-se outro individuo da população para sofrer mutação. Esse processo se encerra quando a cota estipulada para indivíduos gerados a partir do operador de mutação seja atingida.

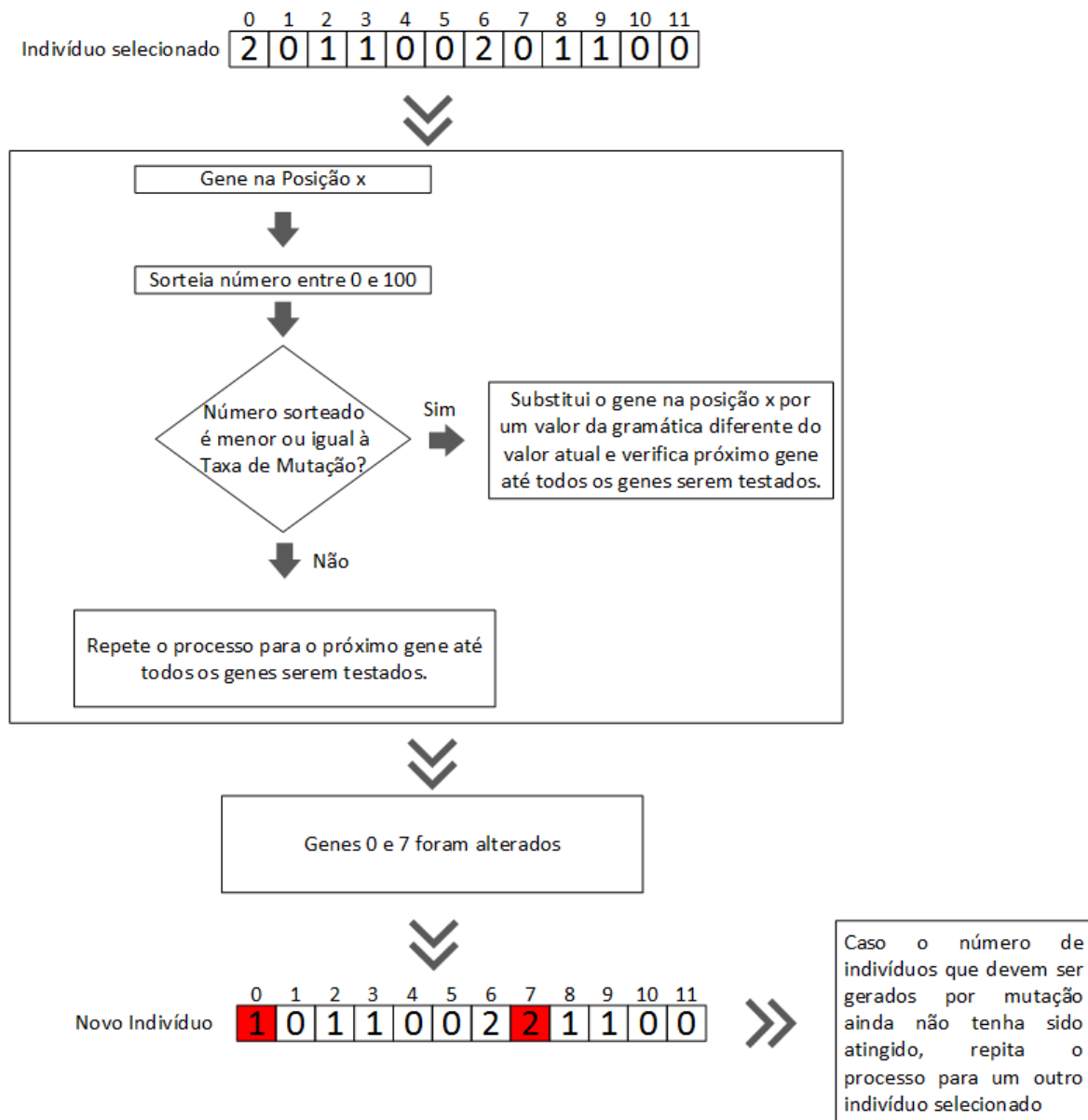


Figura 4-12 - Descrição do processo de mutação quando um indivíduo selecionado sofre mutação em 2 genes

4.3 Considerações Finais

Neste capítulo destacou-se a visão global do projeto de controle de sistema de manufatura flexível com apresentação do diagrama de bloco do FMS Real de acordo com a proposta de Reis et al. (2013) e com a apresentação do projeto de construção de um sistema em software, aqui chamado de FMS Simulado, o qual contém as mesmas características do FMS Real com o advento de um Sistema Evolutivo. Em seguida, foi apresentada a proposta de criação do Sistema Evolutivo e a forma como se dará a definição dos genótipos dos indivíduos do EA criado. Posteriormente, apresentou-se a proposta para o cálculo do *fitness* e a proposta de implementação dos operadores evolutivos de elitismo, seleção cruzamento e mutação.

Em suma, a ideia desse trabalho está na construção de um ambiente que permite, por meio de testes, configurar e analisar o desempenho de Algoritmos Evolutivos para sintonia da FRB do Sistema de Controle de um FMS.

IMPLEMENTAÇÃO DA PROPOSTA

5.1 Considerações Iniciais

Serão expostos nesse capítulo os passos que foram necessários para a implementação do Sistema Evolutivo responsável por possibilitar a configuração e a análise de desempenho de EAs para geração de uma FRB sintonizada para um Sistema de Controle *Fuzzy* de um FMS. A modelagem do sistema com os respectivos diagramas (caso de uso, de sequência e de classes), os protótipos das telas e o modelo do Banco de Dados (BD) criado também serão apresentados.

5.2 Projeto do Sistema Evolutivo

Para a realização dos testes da proposta, um Sistema Evolutivo intitulado “*EvolSys – Evolutionary System*” foi implementado. Para modelagem desse sistema, foi usada uma fração da metodologia de desenvolvimento do chamado Processo Unificado (UP) que é, segundo Hirama (2012), um modelo híbrido para desenvolvimento de software, dirigido a Casos de Uso e proposto no cenário da abordagem Orientada a Objetos.

Neste projeto foram realizadas as fases de Concepção, Elaboração e Construção seguindo as disciplinas de Modelagem, Levantamento dos Requisitos, Análise e Projeto, Implementação e Testes que, nesse modelo, foram executadas, em menor ou maior escala, em todas as fases do sistema. A Figura 5-1 exibe um gráfico de baleia do UP adaptado que foi usado nesse projeto. Esse gráfico indica a intensidade em que cada disciplina foi explorada em cada fase do projeto.

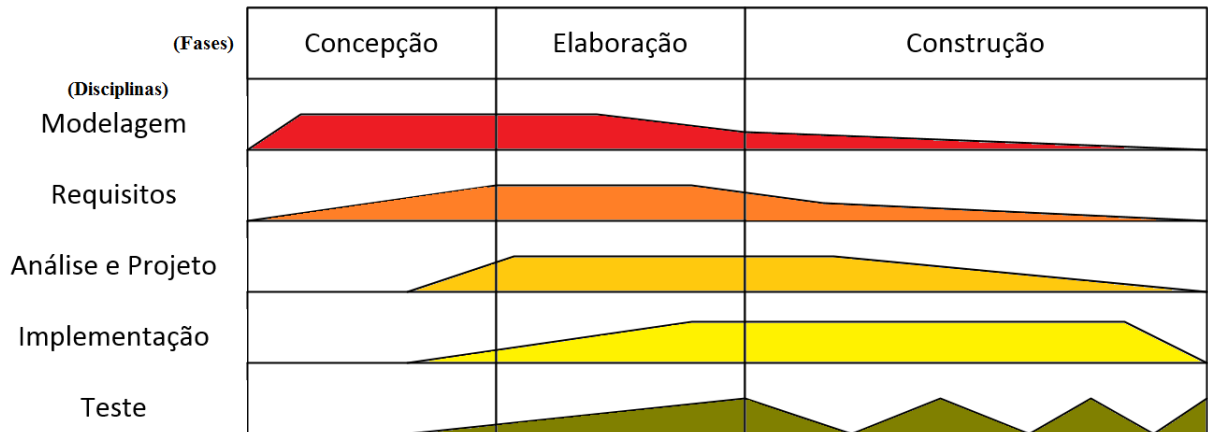


Figura 5-1 - Fases do UP adaptadas para o presente projeto

Na fase de Concepção foi estabelecido o escopo do projeto no sentido do controle de sistema de manufatura. Neste caso, ficou definido que o sistema seria projetado para permitir a configuração e a análise do desempenho de EAs em relação à geração de FRBs sintonizadas para um FMS.

Na fase de Elaboração foi definida a arquitetura do sistema com a especificação dos Casos de Uso do Sistema de Controle, do Sistema Evolutivo e do Algoritmo Evolutivo. Também foram elaborados os diagramas de Classe e de Sequência do sistema proposto.

Na fase de Construção o software foi implementado e testes foram realizados a fim de se obter um sistema de software em funcionamento e com a documentação a ele relacionada disponível.

Os Casos de Uso definidos na fase de Elaboração podem ser vistos na Figura 5-2 (Casos de Uso do Sistema de Controle de FMS), na Figura 5-3 (Casos de Uso do Sistema Evolutivo) e na Figura 5-4 (Casos de Uso do Algoritmo Evolutivo).

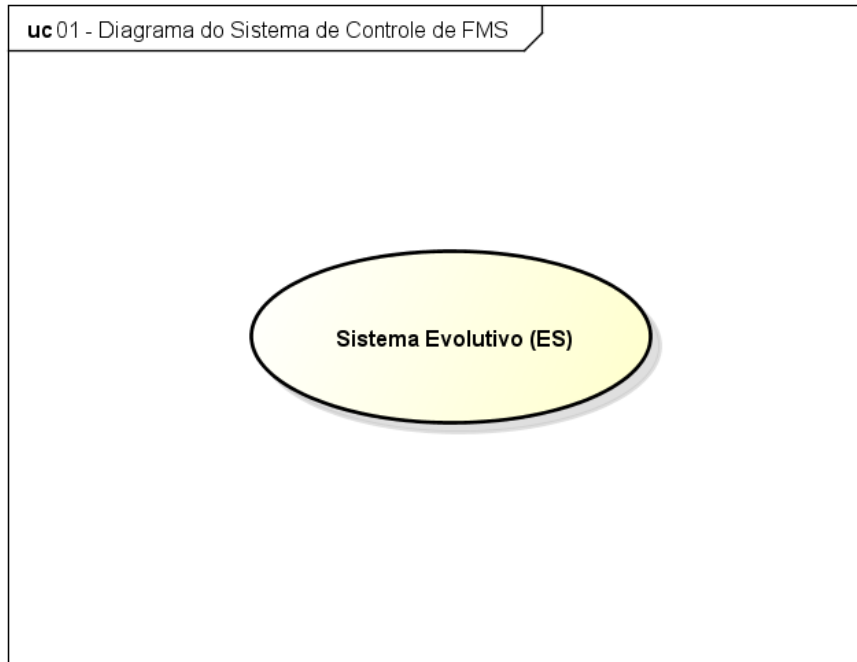


Figura 5-2 - Diagrama de Caso de Uso do Sistema de Controle de FMS

Na Figura 5-2 tem-se o Caso de Uso do Sistema de Controle de FMS que é apresentado aqui para demonstrar que o sistema, como um todo, trata-se de um Sistema Evolutivo.

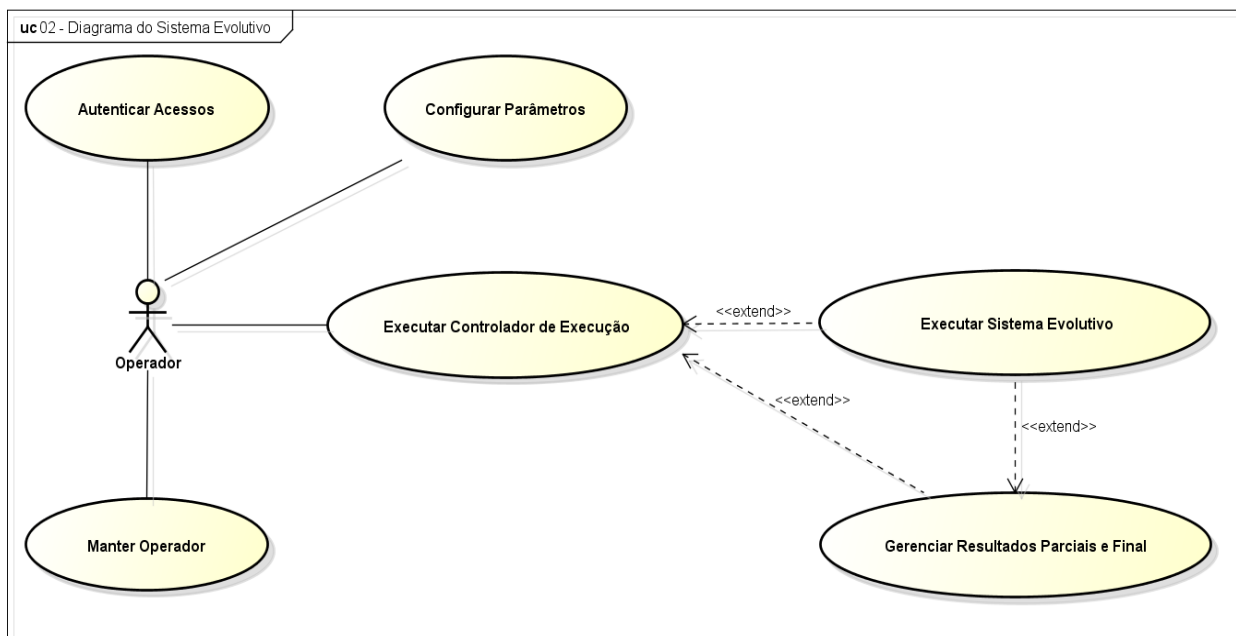


Figura 5-3 - Diagrama de Caso de Uso do Sistema Evolutivo

Como uma especificação do diagrama da Figura 5-2 tem-se o diagrama da Figura 5-3 que trata dos detalhes concernentes ao Sistema Evolutivo.

Neste diagrama pode-se observar os casos de uso relacionados à autenticação dos acessos no sistema (Autenticar Acessos) e à gerência dos operadores do sistema (Manter Operador). Observa-se também o caso de uso de configuração dos parâmetros do sistema (Configurar Parâmetros). Esse caso de uso diz respeito a configuração dos parâmetros relacionados às Regras do Sistema *Fuzzy*, a População do EA e aos processos de Elitismo, Seleção, Cruzamento e Mutação

O caso de uso Configurar Parâmetros trata da solicitação para o usuário da configuração dos parâmetros relacionados ao(à):

1. Sistema Fuzzy

- Quantidade de variáveis de entrada
- Valores possíveis de cada variável de entrada
- Quantidade de variáveis de saída
- Valores possíveis de cada variável de saída

2. Algoritmo Evolutivo

- Quantidade de indivíduos da população
- Cotas de (porcentagem de indivíduos gerados na geração seguinte):
 - Elitismo
 - Cruzamento
 - Mutação
- Grupos (Progenitores divididos por grupos de acordo com o *fitness*)
 - Quantidade de grupos de progenitores
 - Proporção dos grupos de progenitores (Porcentagem de indivíduos em cada grupo)
- Seleção
 - Método de seleção (Roleta, Torneio de 2, Torneio de 3)
 - Taxa de seleção (Quantos indivíduos serão selecionados para cruzamento)
- Cruzamentos
 - Quantidade de progenitores por cruzamento
 - Quantidade de pontos de corte no cromossomo dos indivíduos
- Mutação
 - Taxa de Mutação (Probabilidade de um gene ser alterado)

3. Execução

- Número máximo de gerações
- Tempo máximo de execução

Outros casos de uso são os relacionados à execução do EA (Executar Algoritmo Evolutivo) e a gerência dos resultados obtidos (Gerenciar Resultados Parciais e Final).

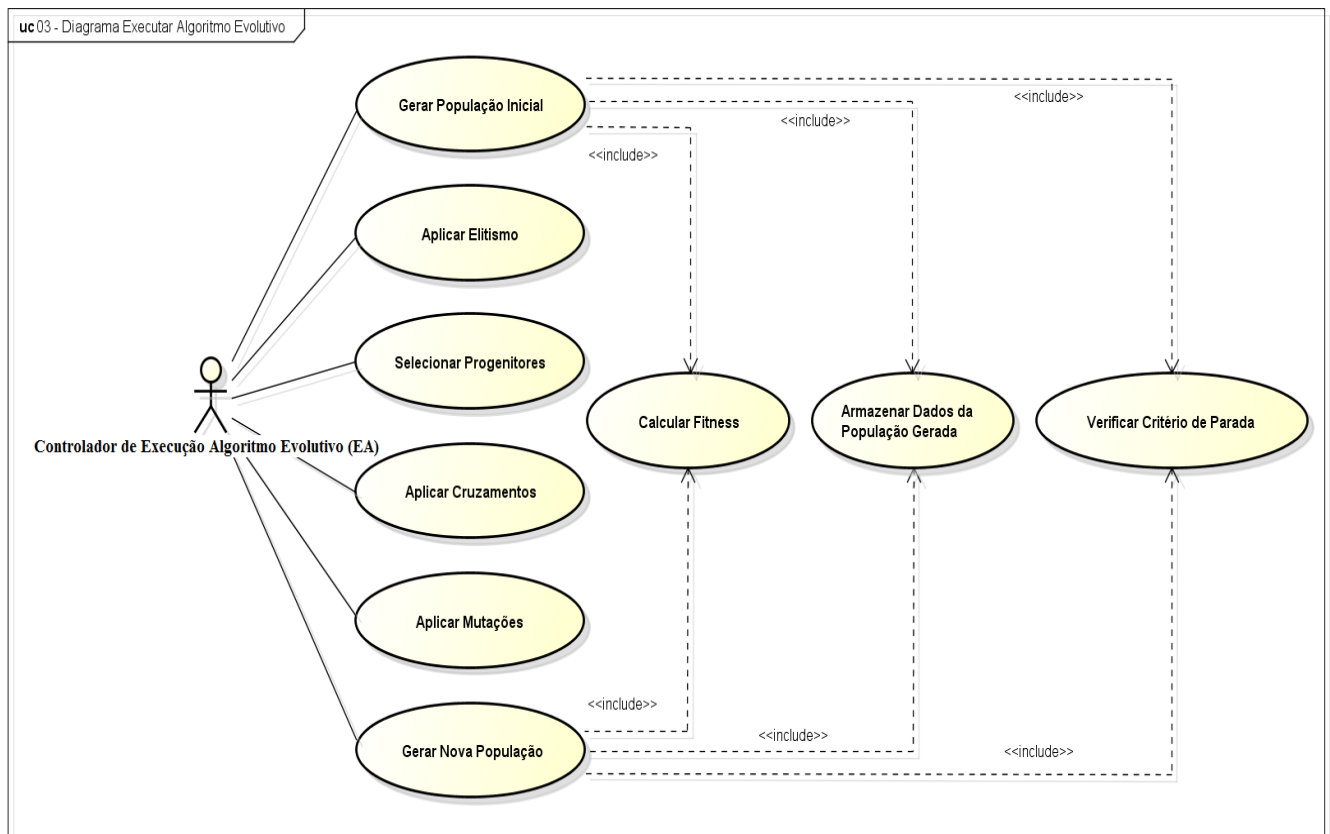


Figura 5-4 - Diagrama de Caso de Uso do Algoritmo Evolutivo

Na Figura 5-4, tem-se o diagrama de caso de uso do Algoritmo Evolutivo que apresenta as ações necessárias para a execução do EA como foi concebido. Os objetivos de cada um dos casos de uso são:

- O caso de uso Gerar População Inicial trata da obtenção da população inicial com os indivíduos sendo escolhidos de forma aleatória.

- O caso de uso Aplicar Elitismo trata da geração dos melhores indivíduos de uma geração anterior na geração seguinte.
- O caso de uso Selecionar Progenitores trata da seleção dos indivíduos que estarão entre os possíveis geradores dos indivíduos da nova população.
- O caso de uso Aplicar Cruzamento trata da recombinação dos indivíduos selecionados para reprodução gerando novos indivíduos.
- O caso de uso Aplicar Operador de Mutação trata da aplicação da mutação genética para determinados indivíduos da população
- O caso de uso Gerar Nova População trata da geração da nova população após a aplicação dos operadores genéticos.
- O caso de uso Calcular *Fitness* trata da geração de um valor de aptidão correspondente ao desempenho de cada indivíduo, ou seja, a cada base de regras gerada no processo.
- O caso de uso Armazenar Dados da População Gerada trata do armazenamento dos dados da população a cada geração. Esses dados são:
 - Melhor base de regras (melhor indivíduo) encontrada.
 - Pior base de regras (pior indivíduo) encontrada.
 - Média da população
 - Variância da população
 - Desvio padrão da população
 - Média Elitismo
 - Variância Elitismo
 - Desvio padrão Elitismo
 - Média Cruzamento
 - Variância Cruzamento
 - Desvio padrão Cruzamento
 - Média Mutação
 - Variância Mutação
 - Desvio padrão Mutação
 - Número da geração
 - Tempo da iteração
- O caso de uso Verificar Critério de Parada diz respeito a verificação quanto ao fim da execução do EA caso os critérios de parada tenham sido atingidos.

Na Figura 5-5 tem-se o caso de uso Calcular *Fitness* onde um indivíduo é selecionado e o *fitness* para esse indivíduo é calculado. A seguir, é verificado se todos os indivíduos foram

avaliados. Se não, um novo indivíduo é selecionado e avaliado. Se sim, a população avaliada é retornada.

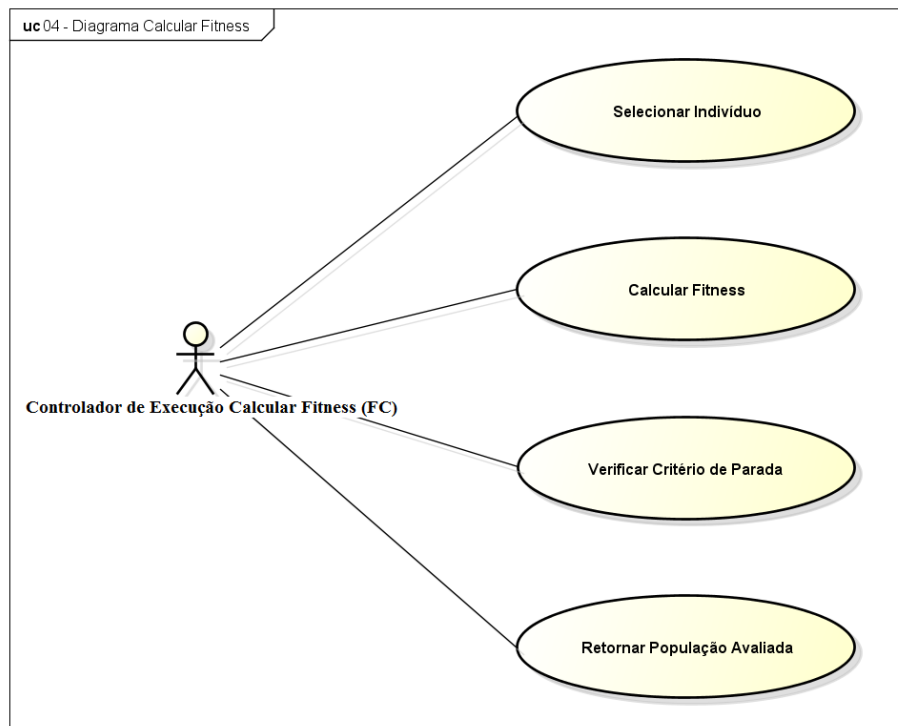


Figura 5-5 – Diagrama de Caso de Uso do Cálculo do *Fitness*

Em relação às disciplinas, na disciplina de Modelagem o sistema foi modelado usando os casos de uso elaborados.

Na disciplina de Requisitos, os agentes do sistema foram identificados e os casos de uso foram desenvolvidos para identificar e modelar os requisitos do sistema.

Na disciplina de Análise e Projeto os diagramas de sequência e de classes foram desenvolvidos.

Na Figura 5-6 pode ser visto o Diagrama de Classes do Sistema Evolutivo. Nela, as classes e os métodos relacionados ao Algoritmo Evolutivo, à configuração dos parâmetros, ao controle de execução, dentre outros são exibidos.

Na Figura 5-7 pode ser visto o Diagrama de Sequência do Algoritmo Evolutivo criado no presente projeto. Nela, podem ser vistas as sequencias de operações desde a solicitação de execução pelo operador do sistema até a geração dos resultados finais passando pela execução do EA e, conseqüentemente, pelos métodos para gerar a população inicial, para cálculo do *fitness*, aplicação do operador de mutação, dentre outros métodos.

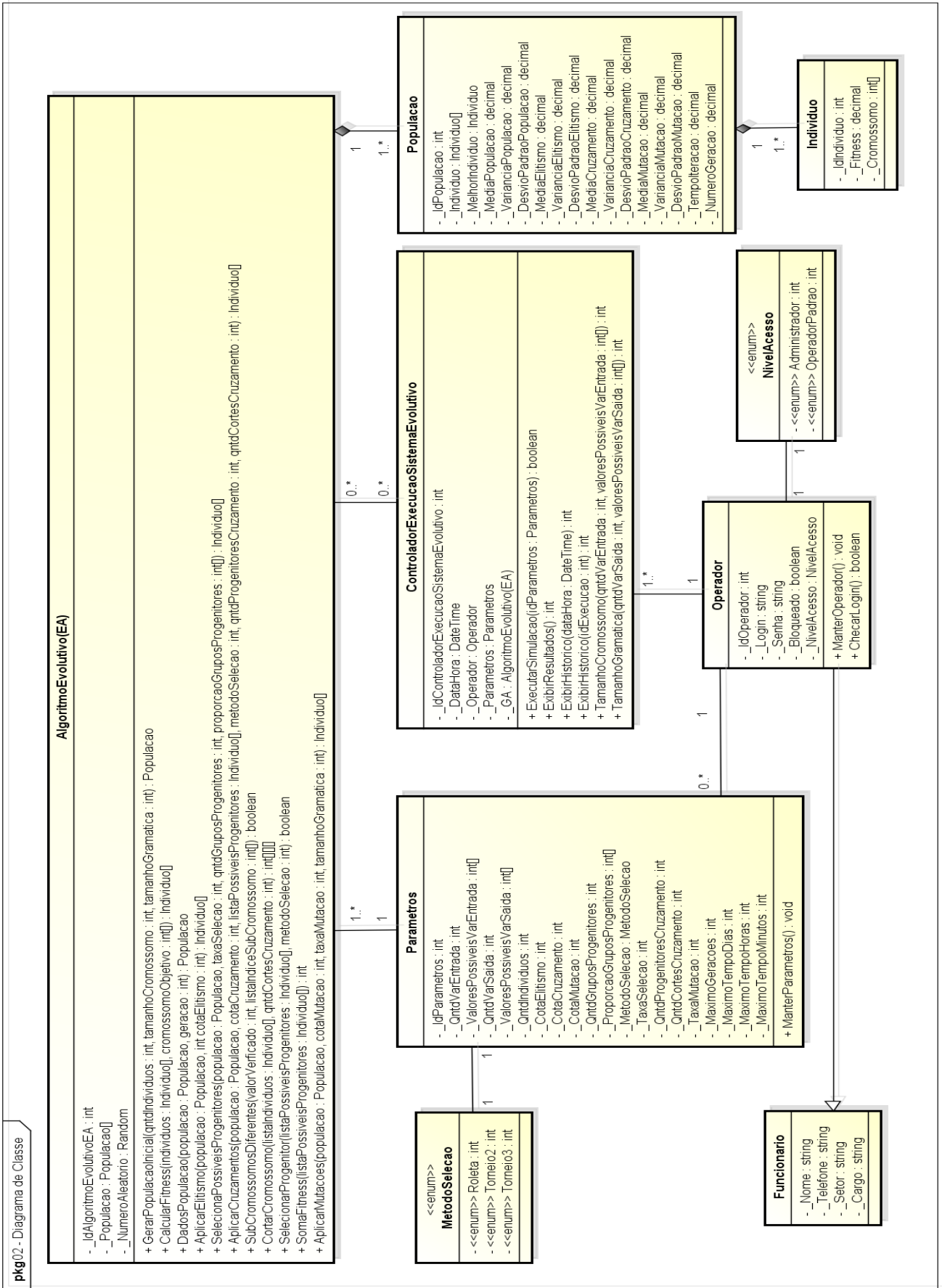


Figura 5-6 - Diagrama de Classes do Sistema Evolutivo

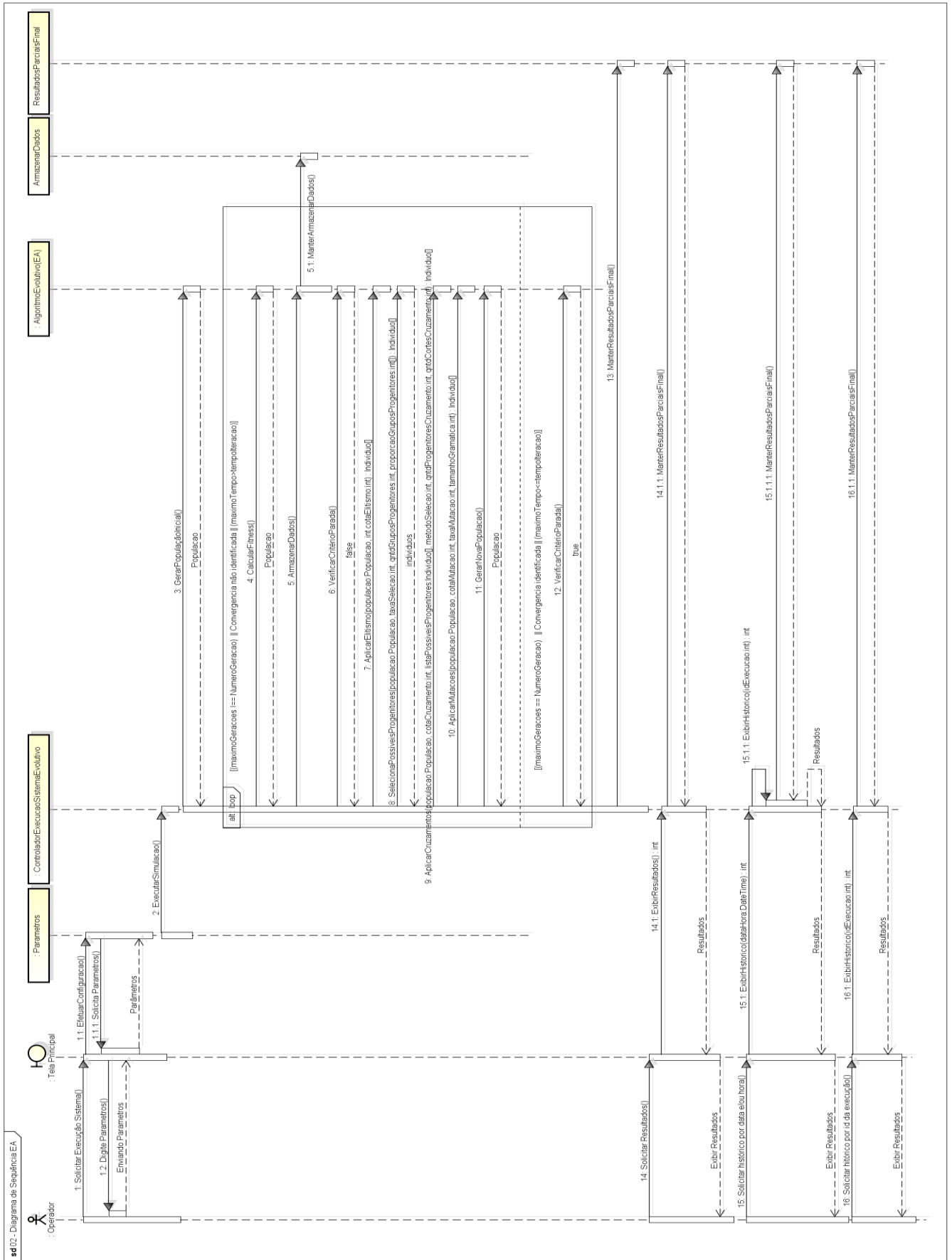


Figura 5-7 - Diagrama de Sequência do Algoritmo Evolutivo

Além disso, foram desenvolvidos protótipos das telas do sistema a fim de auxiliar o desenvolvimento e de propiciar a visualização de como se dará a interação entre o usuário e o sistema ao fim do processo. O protótipos da tela principal sobreposta pela tela que leva à tela de configuração de parâmetros e à execução do Algoritmo Evolutivo pode ser vista na Figura 5-8.

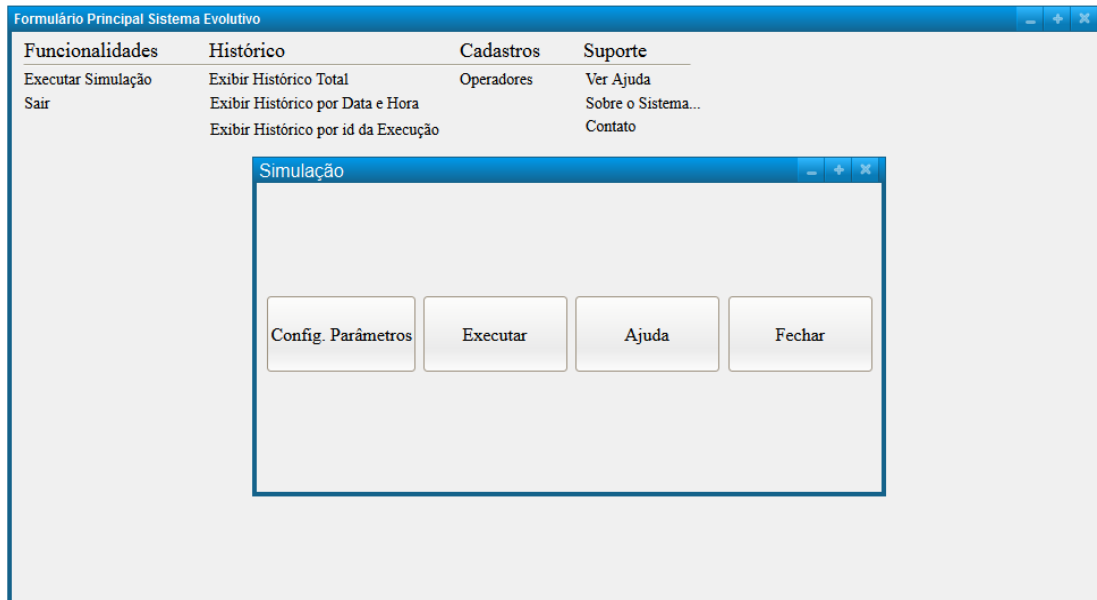


Figura 5-8 - Protótipo de tela principal sobreposta sobre a tela que leva à janela de configuração de parâmetros e execução do Sistema Evolutivo.

Na Figura 5-9 pode ser visto o protótipo da tela de configuração de parâmetros. Nela os parâmetros do EA são configurados e, posteriormente, salvos no Banco de Dados para posterior execução.

O protótipo da tela de configuração de parâmetros apresenta as seguintes seções e campos:

- Regras:** Quantidade de Variáveis de Entrada, Quantidade de Valores Possíveis para Cada Variável de Entrada (Definir), Quantidade de Variáveis de Saida, Quantidade de Valores Possíveis para Cada Variável de Saida (Definir), Default, Validar.
- Seleção:** Método de Seleção (Radio buttons: Roleta, Torneio de 2, Torneio de 3), Taxa de Seleção, Default, Validar.
- População:** Quantidade de Indivíduos, Default, Validar.
- Cruzamentos:** Quantidade de Progenitores por Cruzamento, Quantidade de Pontos de Corte, Default, Validar.
- Cotas:** Cota de Elitismo, Cota de Cruzamento, Cota de Mutação, Default, Validar.
- Mutação:** Cota de Mutação, Taxa de Mutação, Default, Validar.
- Grupos:** Quantidade de Grupos de Progenitores, Proporção dos Grupos de Progenitores (Definir), Default, Validar.
- Execução:** Número Máximo de Gerações, Tempo Máximo de Execução (Dias, Horas, Minutos), Default, Validar.

Botões de ação na base da tela: Limpar Dados, Abrir, Default, Salvar, Ajuda, Fechar.

Figura 5-9 - Protótipo da tela de configuração de parâmetros

Na disciplina de Implementação o sistema foi codificado usado como guia para elaboração do mesmo os casos de uso, os diagramas de sequência e o diagrama de classes elaborados.

Na disciplina de Testes os testes foram realizados e ocorreram em conjunto com a disciplina de implementação. Os testes executados serão discutidos detalhadamente no Capítulo 6 - Testes (página 116).

A seguir, serão apresentadas as ferramentas usadas para o desenvolvimento do projeto e os detalhes de cada um dos passos que foram necessários para a confecção do sistema.

5.3 Ferramentas

Para realização deste projeto foram usadas as seguintes ferramentas:

- Visual Studio
 - Ambiente usado para desenvolver o código em linguagem de alto nível para a confecção do sistema.
 - No presente projeto a linguagem de programação usada foi a C#
 - A versão do ambiente usada foi a:

- *Microsoft Visual Studio Ultimate 2013 Version 12.0.30501.00 Update 2 (64 bits).*
- Astah
 - Software usado para auxiliar a modelagem do sistema em UML (Linguagem de Modelagem Unificada).
 - No presente projeto a versão usada foi a:
 - *Astah Community 6.9.0/b4c6e9 Model Version: 37 (64 bits).*
- Pencil
 - Software usado para elaboração dos protótipos das telas do sistema.
 - No presente projeto a versão usada foi a:
 - *Evolus Pencil Version 2.0.5 (64 bits).*
- DBDesigner
 - Software usado para a modelagem do banco de dados do sistema.
 - No presente projeto a versão usada foi a:
 - *fabFORCE.net DB Designer 4 Version 4.0.5.6 Beta (64 bits).*
- HeidSQL
 - Software usado para realizar a manipulação e criação do banco de dados MySQL (Sistema Gerenciador de Banco de Dados que usa a linguagem SQL (Linguagem de consulta estruturada)).
 - No presente projeto a versão usada foi a:
 - *HeidSQL Versão 8.3.0.4832 (64 bits).*

5.4 Banco de Dados do Projeto

Para confecção do projeto, um Banco de Dados foi criado com o intuito de prover o armazenamento das informações inerentes ao sistema criado. Neste BD 13 tabelas foram criadas para armazenarem os dados referentes ao sistema criado e à execução do Algoritmo Evolutivo.

A Figura 5-10 mostra o Diagrama Entidade-Relacionamento do projeto.

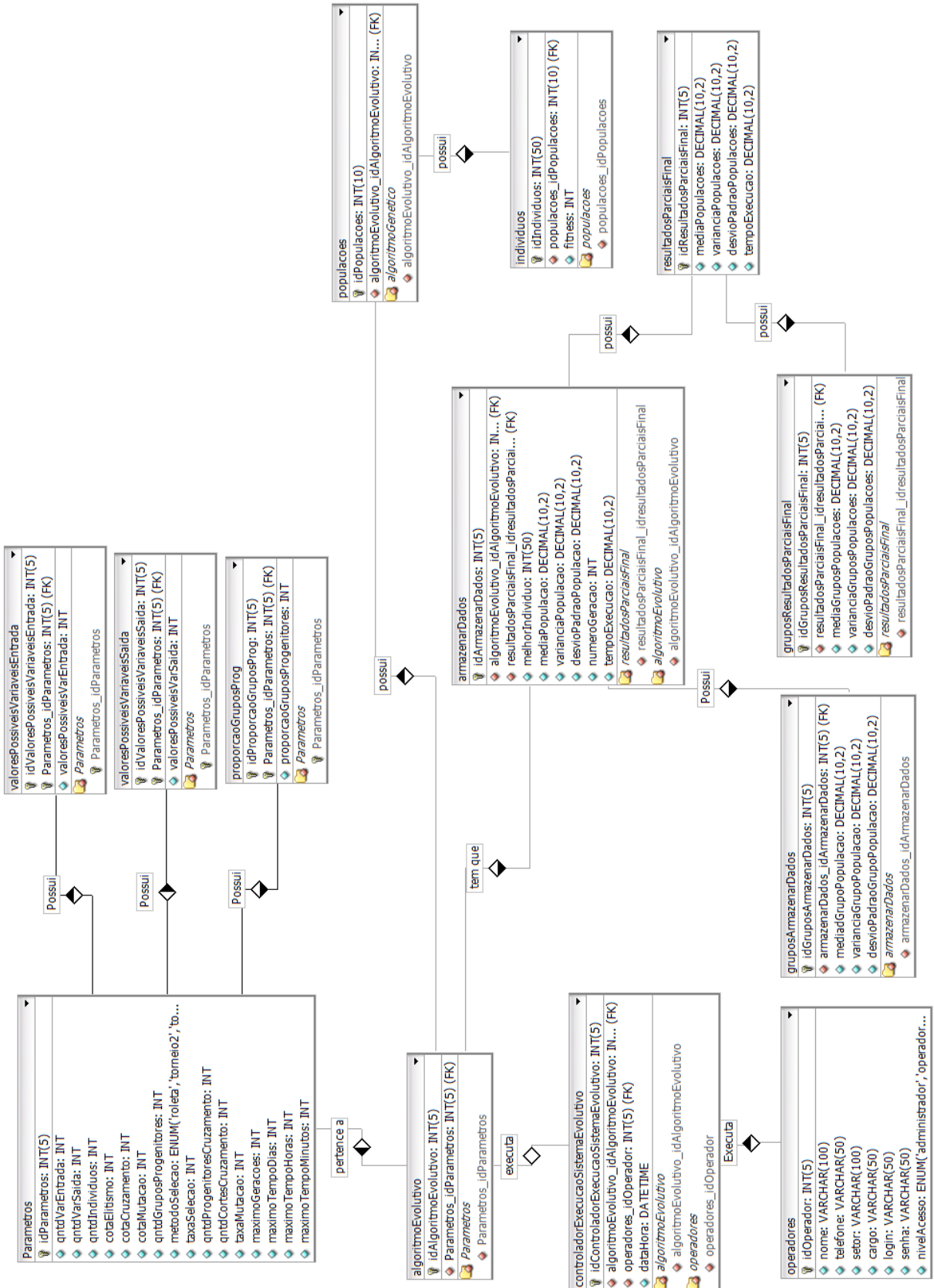


Figura 5-10 - Diagrama Entidade-Relacionamento do EvolSys

Na Figura 5-10 todas as 13 tabelas criadas no banco de dados são exibidas. A partir daí, pode-se criar o BD por intermédio da linguagem SQL e da ferramenta de manipulação de BD chamada HeidiSQL (Capítulo 5 – tópico 5.3 – Ferramentas – página 108).

5.5 Sistema Evolutivo - *EvoISys*

O *EvoISys*, sistema evolutivo criado para análise e configuração de Algoritmos Evolutivos pode ser visto na Figura 5-11.



Figura 5-11 – Tela do Menu Principal do *EvoISys*

Neste sistema, para execução de uma simulação, é necessário, primeiramente, que os parâmetros do sistema sejam configurados. Como dito no tópico 5.2, os parâmetros a serem configurados são os relacionados ao Sistema *Fuzzy*, ao Sistema Evolutivo e à Execução do EA. A Figura 5-12 mostra a tela de configuração para execução de uma simulação.

Figura 5-12 - Tela de Configuração Parâmetros do *EvoISys*

A Tabela 5-1 descreve cada campo da configuração dos parâmetros do Sistema Evolutivo.

Após a realização da configuração executa-se o Sistema Evolutivo para geração da FRB. Neste caso, o *EvoISys* fica em execução até que os critérios de parada sejam atingidos, ou seja, o número máximo de gerações seja alcançado, o tempo limite para execução do EA seja alcançado ou a convergência do EA²⁹ seja identificada. A Figura 5-13 mostra o gráfico (Geração x *Fitness*) resultante da execução do Algoritmo Evolutivo e possibilita a análise da evolução da população durante a execução do mesmo.

Na Figura 5-13 tem-se em destaque a evolução do melhor indivíduo da população em verde, do pior indivíduo em vermelho, da média da população em laranja e da variância da mutação em azul. Nesta tela, é possível escolher quais dados exibir dentre melhor indivíduo (melhor base de regras), pior indivíduo (pior base de regra), média da população, variância da população, desvio padrão da população, média do elitismo, variância do elitismo, desvio padrão do elitismo, média do cruzamento, variância do cruzamento, desvio padrão do cruzamento, média da mutação, variância da mutação e desvio padrão da mutação. Além disso, é possível

²⁹ Mais detalhes sobre verificação da convergência serão vistos no tópico 4.2.4 - Cálculo do *Fitness* (página 88).

definir o intervalo de exibição dos dados para que seja possível visualizar com mais clareza a evolução dos indivíduos durante a execução do EA. No exemplo da Figura 5-13 tem-se um intervalo de 10 ou seja, a distância entre 2 pontos do gráfico equivale a um intervalo de 10 gerações do EA.

Tabela 5-1 - Parâmetros a serem configurados quando da execução do *EvolSys*.

Regras	
Quantidade de Variáveis de Entrada	Define a quantidade de variáveis de entrada presente em cada regra da FRB
Quantidade de Valores Possíveis para Cada Variável de Entrada	Define o valor de cada variável de entrada em cada regra da FRB. A quantidade de valores que deverão ser configurados depende da quantidade de variáveis de entrada que foram selecionadas
Quantidade de Variáveis de Saída	Define a quantidade de variáveis de saída presentes em cada regra da FRB
Quantidade de Valores Possíveis para Cada Variável de Saída	Define o valor de cada variável de saída em cada regra da FRB. A quantidade de valores que deverão ser configurados depende da quantidade de variáveis de saída que foram selecionadas
População	
Quantidade de Indivíduos	Define a quantidade de indivíduos presentes em cada geração do EA.
Cotas	
Cota de Elitismo	Porcentagem de indivíduos da próxima geração que serão gerados a partir de Elitismo
Cota de Cruzamentos	Porcentagem de indivíduos da próxima geração que serão gerados a partir de Cruzamentos
Cota de Mutação	Porcentagem de indivíduos da próxima geração que serão gerados a partir de Mutação
Grupos	
Quantidade de Grupos de Progenitores	Progenitores são divididos por grupos de acordo com o <i>fitness</i> . Campo determina em quantos grupos de progenitores a população será dividida.
Proporção de Grupos de Progenitores	Determina em que proporção se dará a divisão dos grupos de progenitores.
Seleção	
Método de Seleção	Como se dará a seleção dos indivíduos para reprodução. (Roleta, Torneio de 2 ou Torneio de 3)
Taxa de Seleção	Determina quantos indivíduos serão selecionados para serem os possíveis progenitores da próxima geração
Cruzamentos	
Quantidade de Progenitores por Cruzamento	Determina quantos serão os progenitores em cada cruzamento para geração da próxima população
Quantidade de Pontos de Corte	Em quantos pontos o cromossomo será cortado na realização dos cruzamentos
Mutação	
Taxa de Mutação	Probabilidade de um gene ser alterado
Execução	

Além disso, foram apresentadas as ferramentas usadas para a elaboração do projeto com a descrição de cada uma delas, o Diagrama Entidade-Relacionamento do Banco de Dados criado e os diagramas de caso de uso, de classes e de sequência do sistema proposto.

A seguir, será apresentado o capítulo voltado aos testes do sistema desenvolvido e os resultados obtidos a partir desses mesmos testes.

Capítulo 6

TESTES

6.1 Considerações Iniciais

Neste capítulo serão apresentados os testes executados para avaliação do *EvolSys*, ambiente de configuração e análise de EA desenvolvido nesse projeto. Testes de Unidade e de Sistema foram executados e seus resultados foram exibidos e avaliados.

Será demonstrado o planejamento para os testes em relação aos recursos usados, ao escopo definidos para os testes e as estratégias traçadas para cada um dos testes.

Por fim, serão apresentadas as considerações finais a respeito dos testes realizados.

6.2 Planejamento

A finalidade do planejamento nesta etapa do presente projeto, é a de destacar todas as informações necessárias à execução dos testes para o ambiente de análise e configuração de EAs.

Nessa etapa, destacam-se as instruções a serem aplicadas para os testes do *EvolSys* a fim de validar os artefatos produzidos.

A seguir, serão destacados os tópicos que formam o plano de testes do referido projeto.

6.2.1 Recursos

Para confecção e execução de parte dos testes, foi usado o software *Microsoft Visual Studio Ultimate 2013 Version 12.0.30501.00 Update 2 (64 bits)*. A extensão *Unit Test Generator* foi usada para geração e confecção dos testes de unidade.

O *EvolSys* foi testado no Computador com processador *Intel Core i7 – 3537U CPU @ 2.00GHz*, com 8GB de memória RAM e com sistema operacional Microsoft Windows 8 de 64 bits.

6.2.2 Escopo

O *EvolSys* consiste de um sistema para configuração e análise de Algoritmos Evolutivos, e, nesse contexto, configurações de EAs foram elaboradas e testadas. Testou-se também a interface gráfica, e as instruções referentes às operações que devem ser executadas.

Em relação ao processo de testes, o mesmo foi dividido em fases, que se iniciaram desde a fase de implementação, até o final da 1ª versão do protótipo construído, quando todas as partes do projeto já estavam integradas.

Essas fases serão descritas nos tópicos a seguir.

6.2.2.1 A Fase de Teste de Unidade

Segundo Pressman (2010), o Teste de Unidade visa verificar a menor unidade do projeto do software. No caso do presente projeto, as unidades consideradas foram os métodos (ou funções) desenvolvidos para a execução do EA proposto.

Nesses testes, foram inseridos conjuntos de dados com valores de entradas e saídas para os métodos testados. Para esse tipo de teste, os resultados obtidos devem corresponder ao que se espera como resposta. Caso o resultado obtido sejam equivalente ao esperado, o teste é aprovado e caso o resultado não seja o esperado, o teste é reprovado e uma verificação deverá ser executada no método testado. Neste projeto, cada um dos métodos do EA foi verificado.

6.2.2.2 A Fase de Teste de Sistema

De acordo Pressman (2010) o Teste de Sistema corresponde a uma série de diferentes testes cuja finalidade principal é exercitar por completo o sistema baseado em computador.

No presente projeto os teste de Sistema executados foram:

- Teste de Carga – Testar o comportamento do sistema quando o mesmo é submetido a várias condições de carregamento de dados.
- Teste de Estresse – Testar o comportamento do software quando este é submetido a carga máxima de dados.
- Teste de Banco de Dados – Testar a precisão e a integridade dos dados armazenados pelo sistema.
- Teste de Validação – Nesse projeto, o objetivo ao se realizar o teste de validação será verificar se os requisitos estabelecidos para o sistema resultam no impacto esperado na convergência do EA para cada um dos parâmetros configuráveis do sistema.

6.2.3 Estratégia de Testes

A seguir, serão descritas as estratégias utilizadas para a execução de cada um dos testes realizados abordado os objetivos, as técnicas, os critérios de finalização e as respectivas considerações.

6.2.3.1 Estratégia para o Teste de Unidade

A Tabela 6-1 descreve o objetivos, as técnicas, os critérios de finalização e as considerações para os testes de unidade executados.

Tabela 6-1 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para os testes de unidades

Objetivo do Teste	Garantir que as funcionalidades do sistema especificadas nos casos de uso estejam gerando os resultados esperados
Técnica	Executar os testes de unidade usando dados válidos e inválidos para verificar se: <ul style="list-style-type: none"> Os resultados esperados ocorrem quando dados válidos são usados Se as mensagens de erro ou aviso são exibidos quando dados inválidos são usados
Critério de Finalização	Verificar se todos os testes propostos foram executados Defeitos identificados foram tratados.
Considerações	Estes testes serão executados para o EA proposto para o presente projeto

6.2.3.2 Estratégia para o Teste de Sistema

Como dito anteriormente, o teste de sistema é subdividido em testes que exercitam por completo o sistema baseado em computador. As tabelas a seguir, descrevem os objetivos, as técnicas, os critérios de finalização e as considerações para o Teste de Carga (Tabela 6-2), o Teste de Estresse (Tabela 6-3), o Teste de Banco de Dados (Tabela 6-4) e o Teste de Validação (Tabela 6-5).

Tabela 6-2 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Carga

Objetivo do Teste	Garantir que o software execute adequadamente para as carga de dados executadas nos testes.
Técnica	Executar testes com cargas de modo crescente (com aumento progressivo de carga)
Critério de Finalização	Finalização dos testes com ou sem falhas do sistema e com ou sem travamento do computador.
Considerações	O software será executado no computador descrito no tópico ferramentas.

Tabela 6-3 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Estresse

Objetivo do Teste	Garantir que o software execute adequadamente para a cargas máxima de dados.
Técnica	Configurar o Sistema Evolutivo, com campos com valores máximos.
Critério de Finalização	Finalização dos testes com ou sem falhas do sistema e com ou sem travamento do computador.

Considerações O software será executado no computador descrito no tópico ferramentas.

Tabela 6-4 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para o Teste de Banco de Dados

Objetivo do Teste	Garantir que os acessos ao banco de dados funcional apropriadamente e sem corrupção dos dados.
Técnica	Inspeccionar o banco de dados para garantir que os campos foram preenchidos como pretendido e que todos os eventos do banco de dados ocorreram apropriadamente (criações, consultas, alterações e exclusões).
Critério de Finalização	Todos os acessos a base de dados funcionam como projetado e sem nenhuma corrupção dos dados.
Considerações	Processos são invocados manualmente e os registros criados são pequenos para aumentar a visibilidades de eventos não aceitáveis.

Tabela 6-5 – Descrição dos objetivos, das técnicas, dos critérios de finalização e das considerações para os Testes de Validação

Objetivo do Teste	Garantir que os todos os campos configuráveis do EA tenham a funcionalidade e o impacto esperado para os mesmos. Nesse sentido, pretende-se analisar qual o impacto de cada um dos campos configuráveis na convergência do EA.
Técnica	Realizar configurações semelhantes, alterando somente o parâmetro que se quer verificar e assim analisar o impacto daquele parâmetro na convergência do EA.
Critério de Finalização	Todos os campos configuráveis tenham sido testados.
Considerações	Uma planilha com todas as configurações testadas está em anexo logo após o capítulo 8.

6.3 Testes Realizados

Para o presente projeto, 2 tipos de testes foram implementados. Os Testes de Unidade e os testes de Sistema foram executados a fim de garantir que o software corresponda às especificações para o qual ele foi concebido.

Nos tópicos a seguir, serão apresentados os referidos testes realizados para o *EvolSys*.

6.3.1 Execução dos Testes de Unidade

No sistema construído os métodos relacionados ao EA foram testados. Para isso, utilizou-se a extensão da plataforma de desenvolvimento *Visual Studio* chamada de *Unit Test Generator* (tópico 6.2.1 - Recursos - página 117).

Neste processo um bloco de sistema é construído para englobar os testes unitários elaborados.

Como exemplo, tem-se o método relacionado à geração da População inicial do EA.

Para esse método um teste unitário foi construído e 100 entradas diferentes foram testadas. Desse total de entradas, 50 usaram valores que deveriam resultar em testes sem falhas e as outras 50 em testes que resultariam em falhas.

Esses testes foram executados a fim de verificar se o tamanho do cromossomo estipulado para os indivíduos estava de acordo com o esperado, se os alelos estavam dentro dos limites da gramática e se o número de indivíduos gerados correspondia com o estipulado para o teste. Para os 50 testes executados que resultariam em falhas, todos realmente falharam e para os 50 testes que resultariam sucesso, todos os testes passaram satisfatoriamente.

As entradas testadas foram executadas a partir de um arquivo “.csv” que é um arquivo de texto com entradas diferentes separadas por um caractere especial que, neste caso, foi a virgula (,).

A Figura 6-1 exibe o resultado do teste para entradas que resultam em resposta positiva da ferramenta do testes unitário.

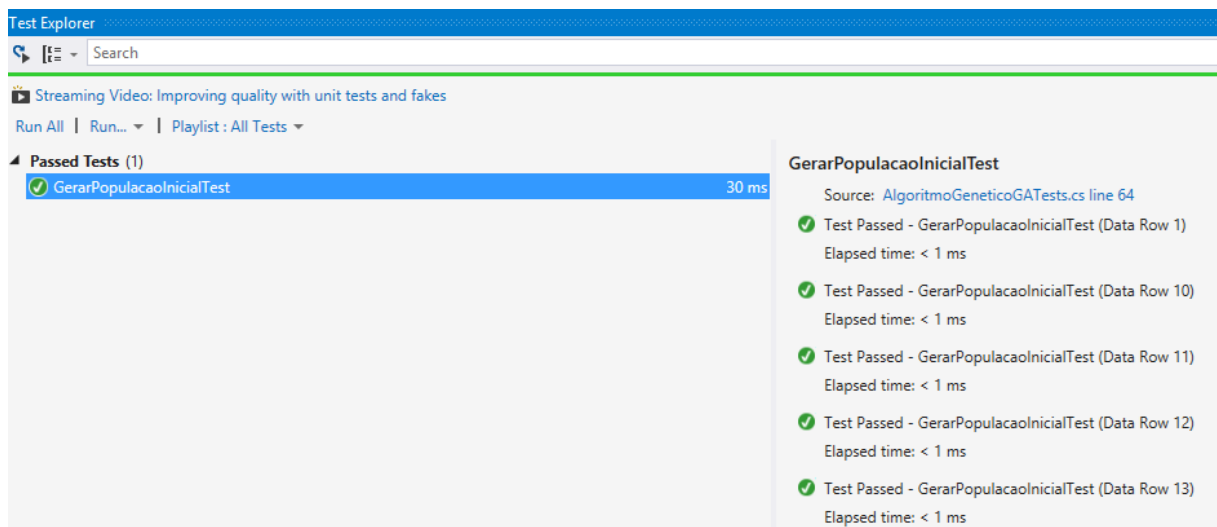


Figura 6-1 - Parte da tela referente à execução do teste com entradas corretas para avaliação do método GerarPopulacaoInicial

A Figura 6-2 exibe o resultado do teste para entradas que resultam em resposta negativa da ferramenta do testes unitário.



Figura 6-2 - Parte da tela referente à execução do teste com entradas incorretas para avaliação do método GerarPopulacaoInicial

A Figura 6-3 mostra o arquivo referente às entradas usadas no teste do método GerarPopulacaoInicial. Do lado esquerdo estão as entradas que resultam em respostas positivas do sistema e do lado direito estão as entradas que resultam em respostas negativas do sistema.

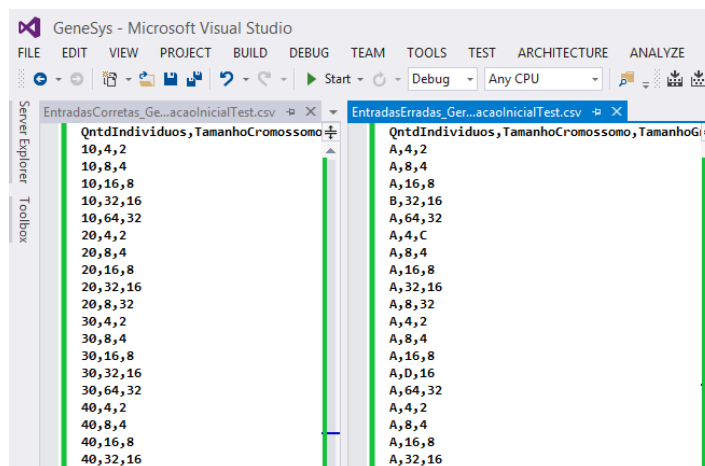


Figura 6-3 - Parte da tela referente à entradas utilizadas para avaliação do método GerarPopulacaoInicial

6.3.2 Execução dos Testes de Sistema

Os Testes de Sistema, como dito anteriormente, foram divididos em teste de Carga, Teste de Estresse e Teste de Banco de Dados. Os tópicos a seguir descreverão cada um dos Testes de Sistema executados para o presente projeto.

6.3.2.1 Execução dos Testes de Carga

Foram executados 5 níveis de teste de carga que são apresentados na Tabela 6-6.

Tabela 6-6 - Execuções do teste de carga

Configuração	Teste de Carga 01	Teste de Carga 02	Teste de Carga 03	Teste de Carga 04	Teste de Carga 05
Quantidade de Variáveis de Entrada	2	2	2	3	4
Quantidade de Valores Possíveis para Cada Variável de Entrada	2,2	2, 2	3, 3	3, 3, 3	3, 3, 4, 4
Quantidade de Variáveis de Saída	1	1	2	3	4
Quantidade de Valores Possíveis para Cada Variável de Saída	2	2	3, 3	3,3,3	3,3,4,4
Quantidade de Indivíduos	10	20	50	100	500
Cota de Elitismo	10%	10%	20%	20%	25%
Cota de Cruzamento	80%	80%	60%	60%	50%
Cota de Mutação	10%	10%	20%	20%	25%
Quantidade de Grupos de Progenitores	1	1	1	2	3
Proporção dos Grupos de Progenitores	100%	100%	100%	70%, 30%	50%, 30% e 20%
Método de Seleção	Roleta	Roleta	Roleta	Roleta	Roleta
Taxa de Seleção	10%	10%	50%	50%	80%
Quantidade de Progenitores por Cruzamentos	2	2	3	3	4
Quantidade de Pontos de Corte	1	2	2	2	3
Taxa de Mutação	1%	5%	10%	20%	20%
Número Máximo de Gerações	100	200	500	1000	10000
Tempo =>	1 segundos e 91 miliseg.	1 segundo e 98 miliseg.	2 segundos e 14 miliseg.	20 segundos e 70 miliseg.	12 minutos, 16 segundos e 98 miliseg.

Vale ressaltar que o aumento dos valores de entrada para ocasionar o aumento da carga para o EA do presente projeto pode ser contraproducente, uma vez que a depender da configuração, o EA pode executar de uma forma mais rápida e eficiente em relação a um teste com valores de entrada menores. Por exemplo, ao aumentar a Taxa de Mutação de uma

execução, pode-se fazer com que o algoritmo execute mais rapidamente uma vez que a diversidade da população irá aumentar o que pode ocasionar em uma convergência mais rápida. Entretanto, podem ocorrer situações onde uma alta taxa de mutação pode fazer com que o algoritmo se comporte de uma forma não esperada e a convergência demore muito para ser atingida, ou, em algumas situações, a convergência não é detectada e a busca se torna aleatória (*Randon Walk* – ver Capítulo 2 - Fundamentação Teórica – tópico 2.4.9 - Parâmetros – página 59).

6.3.2.2 Execução dos Testes de Estresse

Na Tabela 6-7 são apresentadas as configurações para o Teste de Estresse executado no *EvolSys*.

Para o teste de estresse com a configuração acima ocorreu um erro de estouro de memória quando o mesmo foi executado a 32 bits, ou seja, todos os blocos (*namespaces*) do projeto estavam sendo executados na plataforma x86. Já quando o teste foi executado na plataforma x64, ou seja, executado a 64 bits, a execução ocorreu tranquilamente.

Tabela 6-7 - Execuções do teste de carga

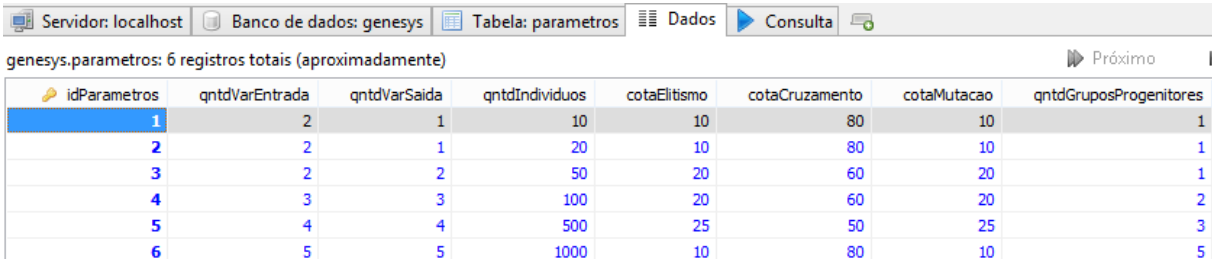
Configuração	Teste de Estresse
Quantidade de Variáveis de Entrada	5
Quantidade de Valores Possíveis para Cada Variável de Entrada	5, 5, 5, 5, 5
Quantidade de Variáveis de Saída	5
Quantidade de Valores Possíveis para Cada Variável de Saída	5, 5, 5, 5, 5
Quantidade de Indivíduos	1000
Cota de Elitismo	10%
Cota de Cruzamento	80%
Cota de Mutação	10%
Quantidade de Grupos de Progenitores	5
Proporção dos Grupos de Progenitores	35%, 25%, 20%, 15%, 5%
Método de Seleção	Roleta
Taxa de Seleção	100%
Quantidade de Progenitores por Cruzamentos	4
Quantidade de Pontos de Corte	3
Taxa de Mutação	50%
Número Máximo de Gerações	1000000
Tempo =>	30 minutos, 6 segundos e 78 miliseg.

6.3.2.3 Execução dos Testes de Banco de Dados

Os testes do banco de dados foram executados e os dados gravados dos testes anteriores foram conferidos e foi constatado que os mesmos foram executados a contento. Entretanto, como os métodos de exclusão dos dados secundários (Quantidade de Valores para Variáveis de Entrada, Quantidade de Valores para Variáveis de Saída e Quantidade de Grupos de Progenitores) não foram implementados, não é possível fazer a exclusão desses dados a partir do sistema.

Nesta versão a exclusão se dá por meio do acesso ao software HeidiSQL para realizar o gerenciamento do banco de dados. Este fato, não prejudica a execução dos testes uma vez que os dados, não sendo excluídos ficam disponíveis para serem resgatados, mas não são usados em outras configurações realizados.

A Figura 6-4 apresenta alguns dos valores inseridos no banco de dados para os testes anteriores.



idParametros	qntdVarEntrada	qntdVarSaida	qntdIndividuos	cotaElitismo	cotaCruzamento	cotaMutacao	qntdGruposProgenitores
1	2	1	10	10	80	10	1
2	2	1	20	10	80	10	1
3	2	2	50	20	60	20	1
4	3	3	100	20	60	20	2
5	4	4	500	25	50	25	3
6	5	5	1000	10	80	10	5

Figura 6-4 - Configurações inseridas no banco de dados para a tabela Parâmetros.

6.3.2.4 Execução dos Testes de Validação

Os testes de validação visam garantir que todos os requisitos estabelecidos para sistema evolutivo tenham a funcionalidade e o impacto esperado para os mesmos. Nesse sentido testes foram realizados para cada um dos campos configuráveis do EA e os mesmos se encontram no apêndice desse trabalho. Neste tópico será apresentado um dos testes realizados e uma análise será realizada a partir dos resultados alcançados.

Nos exemplos a seguir, tem-se 3 gráficos representando a execução de 3 configuração com espaço de busca diferentes.

Tabela 6-8 - Configurações com espaços de busca variados - Cromossomos de tamanho 9 (Código 02), 27 (Código 03) e 81 (Código 4) para uma gramática de tamanho 9

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada		Quantidade de Variáveis de Saída		Quantidade de Valores Possíveis para	Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento		
02	2	3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
03	3	3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
04	5	3, 3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

Para a primeira execução (Código 2 - Tabela 6-8 - Figura 6-5), tem-se um espaço de busca representado por um cromossomo de tamanho 9 tendo 9 valores possíveis em cada um dos genes do respectivo indivíduo. Para a segunda execução (Código 3 - Tabela 6-8 - Figura 6-6) tem-se um cromossomo de tamanho 27 tendo 9 valores possíveis em cada um dos genes do respectivo indivíduo. A terceira execução (Código 4 - Tabela 6-8 - Figura 6-7) corresponde a um cromossomo de tamanho 81 com 9 valores possíveis para cada gene do indivíduo.

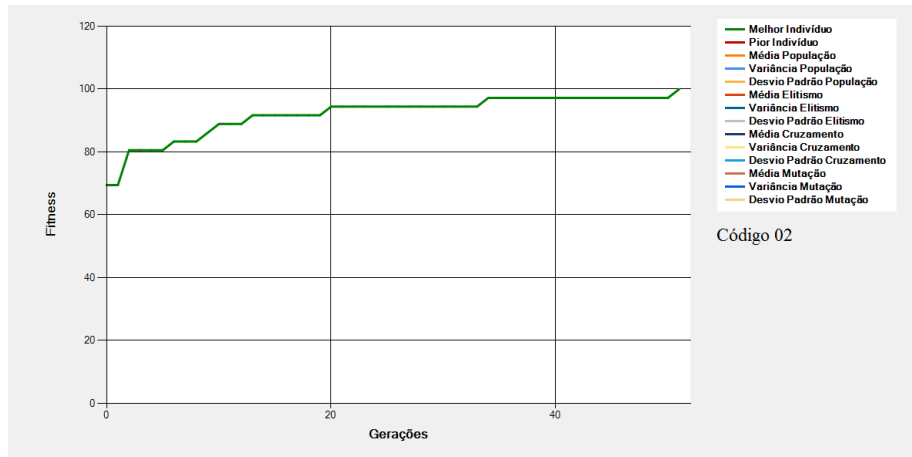


Figura 6-5 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 9 com 9 valores possíveis para cada gene.

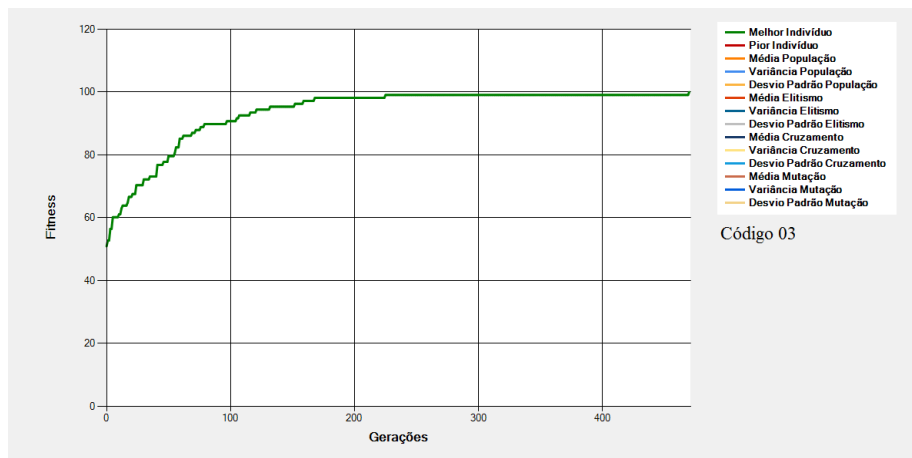


Figura 6-6 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 27 com 9 valores possíveis para cada gene.

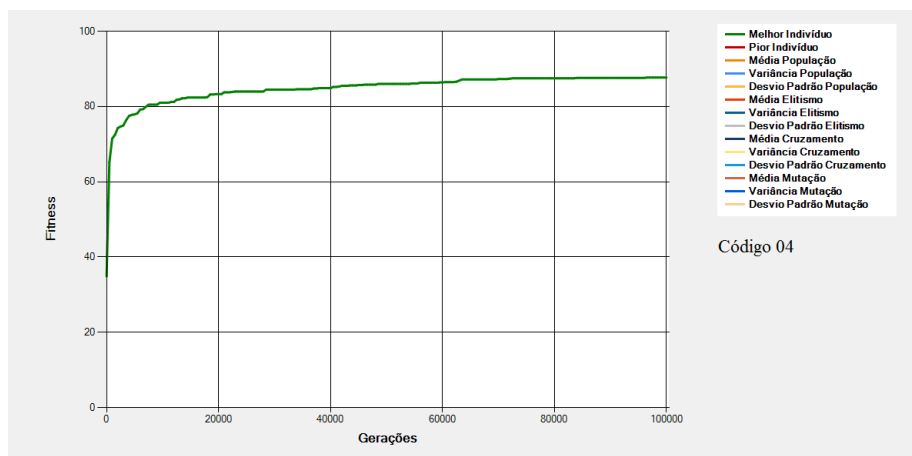


Figura 6-7 - Gráfico do Melhor Indivíduo para cromossomo de tamanho 81 com 9 valores possíveis para cada gene.

Percebe-se que, para a execução da Figura 6-5, como o espaço de busca é menor (cromossomo de tamanho 9 com 9 valores possíveis em cada gene) com cerca de 50 gerações foi possível alcançar o ótimo. Já na execução da Figura 6-6, como o espaço de busca é maior (cromossomo de tamanho 27 com 9 valores possíveis em cada gene) mais gerações foram necessárias para se encontrar o ótimo (cerca de 500 gerações).

No entanto, na execução da Figura 6-7, o número máximo de gerações configurado, que foi de 100 mil gerações, não foram suficientes para que o ótimo fosse alcançado. Isso se deve ao tamanho do espaço de busca que foi maior para o exemplo da Figura 6-7. Nesse caso o cromossomo tinha o de tamanho 81 genes com 9 valores possíveis em cada um deles. Esse fato, leva a um problema de explosão combinatória que justifica de fato o uso do EA para problemas dessa magnitude.

Com esses gráficos em vista, surge a pergunta:

Tendo em vista a não convergência para o ótimo global na execução da Figura 6-7, quais parâmetros do EA devem ser alterados para que se obtenha uma convergência mais rápida para o ótimo global?

Neste caso, é necessário analisar outras variáveis do EA como pode ser visto na Figura 6-8. Nela, analisando mais profundamente o exemplo da Figura 6-7 para que seja possível pondera-se entre alterar cota de elitismo, cota de cruzamento, cota de mutação, taxa de mutação, método de seleção dentre outros parâmetros.

Na Figura 6-8 são apresentado os gráficos do Melhor Indivíduo, da Média da População, do Pior Indivíduo, da Variância da População e do Desvio Padrão da População.

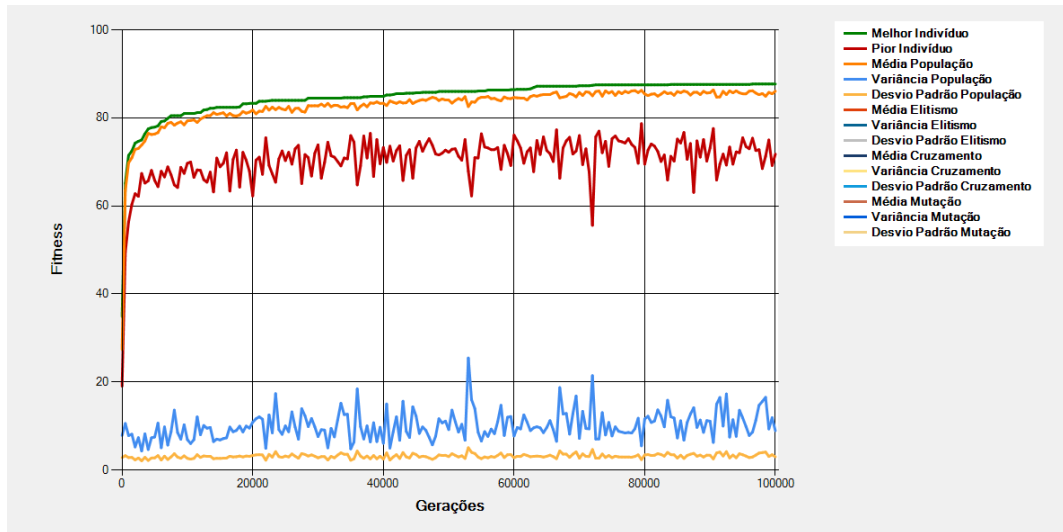


Figura 6-8 - Gráfico do Melhor Indivíduo, Média da População, Pior Indivíduo, Variância da População e Desvio Padrão da População.

No caso, da Figura 6-8 a variância da população está baixa demonstrando que os indivíduos da população não estão explorando de forma abrangente o espaço de soluções. Outro fato observado é que o algoritmo converge para soluções melhores muito rapidamente nas gerações iniciais (coeficiente angular da curva do melhor indivíduo é alto nas gerações iniciais) o que demonstra que Cotas de Elitismo e Cruzamento (que conduzem à convergência) estão altas para essa execução. Além disso, como a variância e o desvio padrão estão baixos, percebe-se que a população está bastante homogênea e a tendência é que, mesmo com um número maior de gerações, o gráfico se mantenha linear por mais tempo conduzindo a execução para um ótimo local.

Tendo em vista que os operadores que conduzem à convergência (elitismo e cruzamento) somados atuam em 90% da população, e que a mutação está baixa (10% de Cota e 10% de taxa de mutação) é de se esperar que, para uma execução em um espaço de busca maior, a convergência aconteça rapidamente, tornando alta, a possibilidade do EA ser conduzido à um ótimo local.

6.4 Considerações Finais

Neste capítulo foram apresentados os testes realizados para verificação das funcionalidades do ambiente criado para análise e configuração do Algoritmos Evolutivos.

Foram realizados Testes de Unidade e de Sistema que ocorreram como o esperado para esta versão do software.

Dentre os testes de Sistema realizados, destacam-se os testes de validação, que foram realizados com o intuito de verificar os requisitos do sistema de acordo com o comportamento do EA quando da alteração em cada um dos campos configuráveis do sistema.

Nesse sentido, análises foram feitas por intermédio dos dados de melhor indivíduo, variância da população, desvio padrão da população, número de gerações, etc. apresentados quando da execução das configurações testadas. Além dos gráficos presentes nesse capítulo, encontram-se, no apêndice desse trabalho, outros gráficos e suas respectivas análises.

Por fim, conclui-se que os testes do sistema são requisitos essenciais para verificação das funcionalidades do software criado e a análise do impacto de cada um dos campos configuráveis mostrou-se ferramenta importante para demonstrar como se dá a atuação de cada um dos parâmetros na convergência do EA.

CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

O desafio dessa pesquisa foi evoluir nas questões relacionadas ao uso dos algoritmos evolutivos para a sintonia da FRB do Sistema de Controle *Fuzzy* de um FMS.

Neste sentido tem-se a ideia de construção de um ambiente de configuração e análise de desempenho de Algoritmo Evolutivos para propiciar a definição da FRB sintonizada do Sistema de Controle de um FMS. A ideia é que trabalhos futuros utilizem o ambiente para definição da FRB sintonizada para um FMS.

Por intermédio do ambiente criado, é possível configurar variáveis do EA como quantidade de indivíduos por cruzamento, tamanho da população inicial, cota de cruzamentos, taxa de mutação dentre outros parâmetros. Também é possível configurar as regras do Sistema *Fuzzy* variando o número de variáveis de entrada e saída bem como os valores possíveis para cada uma delas.

O ambiente de configuração criado foi desenvolvido baseado no conjunto de atividades para geração de software orientados a objetos chamado de Processo Unificado e a linguagem orientada a objeto C# foi a usada para realizar a implementação do sistema. Várias ferramentas foram utilizadas como o ambiente de programação Visual Studio, o ambiente para modelagem de Banco de Dados chamado de DBDesigner, dentre outras. Além disso, testes foram realizados para investigar os componentes do software e detectar possíveis problemas que pudessem existir.

Teve-se como resultado um sistema de análise e configuração de EA para sintonia da FRB do Sistema de Controle de um FMS que permite configurar diversos parâmetros do EA.

Neste sentido, pode-se inferir também que, futuramente, com a colaboração entre a Lógica *Fuzzy* e os Algoritmos Evolutivos, será possível, em projetos futuros do Laboratório Tear, realizar pesquisas voltadas para a otimização do Sistema de Sequenciamento de Operações de um FMS. Desse modo, com o sistema disponível para configuração e análise de EA, uma configuração mais apropriada poderá ser usada para a geração do EA que leva a obtenção da FRB mais adequada a um problema específico aumentando desse modo, o desempenho do sistema de acordo com uma dada varável de otimização (*makespan, due date, tardiness, etc.*).

Portanto, com esse estudo, foi possível realizar avanços na área de desenvolvimento de sistemas que auxiliem os processos produtivos. Cumpre-se, portanto, a etapa de construção do ambiente para configuração e análise do desempenho de EAs para seleção da FRB sintonizada de um dado FMS.

7.2 Trabalhos Futuros

O trabalho desenvolvido até este ponto faz parte do projeto de desenvolvimento de um sistema que utilizará EA para gerar as bases de regras do Sistema de Controle *Fuzzy* de um FMS. Neste sentido, algumas inclusões podem ser desenvolvidas para este projeto.

Um ponto a ser desenvolvido está relacionado diretamente ao algoritmo proposto. Neste caso, pode-se aplicar a ideia de subpopulações aos grupos criados para seleção dos possíveis progenitores para realização dos cruzamentos (Figura 4-10). Neste caso, estes grupos, passariam a ser desenvolvidos como subpopulações onde o desempenho dos ancestrais seriam armazenados para verificação do desempenho do algoritmo.

Outro avanço pode ser realizado quando da seleção de regras e de variáveis. Nessa situação, regras e variáveis irrelevantes poderiam ser excluídas tornando a FRB menor.

Pode-se citar também o uso de um Sistema Evolutivo Adaptativo onde os parâmetros do Algoritmo Evolutivo poder ser alterados no decorrer e de acordo com a resposta da evolução corrente, tornando o processo evolutivo mais eficiente. Outro ponto importante do uso dessa

técnica é que a mesma propicia a investigação de quais variáveis devem ser usadas para conduzir para um melhor critério de otimização e com isso, avaliar o impacto de cada uma delas no desempenho do sistema de manufatura.

Outro avanço, pode ocorrer no uso de um sistema real, ou seja, a FRB usada na fábrica pode ser substituída em tempo real, por uma que está em um Banco de Dados de situações previamente verificadas no simulador. Isso propicia ao sistema uma adequação rápida às condições do sistema o que acarreta em um aumento de desempenho do sistema de manufatura.

Pode-se também utilizar outras técnicas de inteligência artificial, como as redes neurais, para realizar o ajuste do sistema de sequenciamento de operações e assim verificar o impacto causado por outra técnica na sintonia da FRB do Sistema de Controle de um FMS.

Além disso, pode-se utilizar a técnica de Aprendizado de Máquina para melhorar o processo de busca pelo EA que gera a FRB sintonizada para um FMS.

REFERÊNCIAS

ABDELHAK, B. J.; NAJIB, E.; ABDELAZIZ, H.; HNAIEN, F.; YALAOUI, F. Optimum sizing of hybrid PV/wind/battery using Fuzzy-Adaptive Genetic Algorithm in real and average battery service life. **International Symposium on Power Electronics, Electrical Drives, Automation and Motion**, p. 871–876, 2014. Ieee.

AHMED, L.; ADEEL, S. Neuro-Fuzzy Hybrid Intelligent System Using Grid Computing. **International Conference on Emerging Technologies**, p. 145–147, 2007. Islamabad - Pakistan: IEEE.

ANGSANA, A.; PASSINO, K. M. Distributed Fuzzy Control of Flexible Manufacturing Systems. **Transactions on Control Systems Technology, IEEE**, v. 2, n. 4, p. 423–435, 1994.

ARAI, Y.; WATANABE, T. Professor Ebrahim H . Mamdani Memorial Issue. **Journal of Advanced Computational Intelligence and Intelligent Informatics**, v. 16, n. 5, p. 591, 2012.

ARAÚJO, R. **Um Método para Modelagem de Controle de AMS Usando Redes de Petri Virtuais**, 2006. Universidade Federal de São Carlos - UFSCar.

ARDESTANI, M. M.; MOAZEN, M.; JIN, Z. Gait modification and optimization using neural network–genetic algorithm approach: Application to knee rehabilitation. **Expert Systems with Applications**, v. 41, n. 16, p. 7466–7477, 2014. Elsevier Ltd.

BARBOZA, A. O. **Simulação e Técnicas da Computação Evolucionária Aplicadas a Problemas de Programação Linear Inteira Mista**, 2010. Universidade Tecnológica Federal do Paraná - UTFPR.

BERNARDINI, F. C. **Combinação de Classificadores Simbólicos Utilizando Medidas de Regras de Conhecimento e Algoritmos Genéticos**, 2006. Universidade de São Paulo - USP.

BOULIF, M.; ATIF, K. A New Fuzzy Genetic Algorithm for the Dynamic Bi-Objective Cell Formation Problem Considering Passive and Active Strategies. **International Journal of Approximate Reasoning**, v. 47, n. 2, p. 141–165, 2008.

CASTRO, P. A. D.; CAMARGO, H. A. Learning and optimization of fuzzy rule base by means of self-adaptive genetic algorithm. **2004 IEEE International Conference on Fuzzy Systems**, v. 2, p. 1037–1042, 2004. Budapest - Hungary.

CATARINA, A. S.; BACH, S. L. Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binária e real. **Acta Scientiarum. Technology**, v. 25, n. 2, p. 147–152, 2003.

CHAN, F. T. S.; CHAN, H. K.; LAU, H. C. W. The State of the Art in Simulation Study on FMS Scheduling: A Comprehensive Survey. **The International Journal of Advanced Manufacturing Technology**, v. 19, n. 11, p. 830–849, 2002.

- CHI, S.-C.; YAN, M.-C. A Fuzzy Genetic Algorithm for High-Tech Cellular Manufacturing System Design. **Annual Meeting of the Fuzzy Information**, v. 2, p. 907–912, 2004. IEEE.
- CHOWDHURY, B. H.; RAHRNAN, S. A review of recent advances in economic dispatch. **IEEE Transactions on Power System**, v. 5, n. 4, p. 1248–1259, 1990.
- CINTRA, M. E.; CAMARGO, H. A. Geração de Regras Fuzzy com Pré-Seleção de Regras Candidatas. **VI Encontro Nacional de Inteligência Artificial–ENIA**, v. 1, p. 1341–1350, 2007. Rio de Janeiro - Brazil.
- CINTRA, M. E.; CAMARGO, H. DE A.; HRUSCHKA, E. R.; NICOLETTI, M. DO C. Fuzzy Rule Base Generation through Genetic Algorithms and Bayesian Classifiers A Comparative Approach. **Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)**, 2007.
- COELHO, L. DOS S.; MARIANI, V. C. Otimização de despacho econômico com ponto de válvula usando estratégia evolutiva e método quase-newton. **Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN)**, v. 4, n. 1, p. 1–12, 2006.
- COELHO, T. S.; FERNANDES, M. A. R.; MIOT, H. A.; YORIYAZ, H. Uso do método fuzzy c-means para segmentação de imagens dermatoscópicas de lesões de pele. **Revista Brasileira de Física Médica**, v. 6, n. 2, p. 99–102, 2012.
- COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. VAN. **Evolutionary algorithms for solving multi-objective problems**. New York - NY - USA: Springer Science, Business Media, 2007.
- CORDÓN, O.; HERRERA, F. Hybridizing Genetic Algorithms with Sharing Scheme and Evolution Strategies for Designing Approximate Fuzzy Rule-Based Systems. **Fuzzy Sets and Systems**, v. 118, n. 2, p. 235–255, 2001.
- CORDÓN, O.; HERRERA, F.; HOFFMANN, F.; MAGDALENA, L. Recent Advances in Genetic Fuzzy Systems. **Journal of Information Sciences**, v. 136, n. 1-4, p. 1–5, 2001.
- COSENZA, H.; VILLELA, L. E.; MORE, J. Aplicação de Um Modelo de Hierarquização Como Instrumento para Tomada de Decisão: Caso de uma Multinacional. **XXVI Encontro Nacional de Engenharia de Produção - ENEGEP**, p. 1–8, 2006.
- DARWIN, C. **On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life**. London - England: London : John Murray, Albemarle Street, 1859.
- DEPEW, D. J.; WEBER, B. H. The Fate of Darwinism : Evolution After the Modern Synthesis. **Biological Theory**, v. 6, p. 89–102, 2011.
- DESOUKY, S. F.; SCHWARTZ, H. M. Hybrid Intelligent Systems Applied to the Pursuit-Evasion Game. **2009 IEEE International Conference on Systems, Man and Cybernetics**, p. 2603–2608, 2009. San Antonio - TX - USA: IEEE.

DIAS, J. P.; CORRÊA, R. M. Multiobjective Optimization of Multibody Systems with Genetic Algorithms. (C. A. Motasoaes, J. A. C. Martins, H. C. Rodrigues, et al., Eds.) **III European Conference on Computational Mechanics**, p. 437, 2006. Dordrecht - Netherlands: Springer Netherlands.

DORAN, R. W. The Gray Code. **Journal of Universal Computer Science**, v. 13, n. 11, p. 1573–1597, 2007.

FANTI, M. P.; MAIONE, B.; TURCHIANO, B. Comparing digraph and Petri net approaches to deadlock avoidance in FMS. **IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics**, v. 30, n. 5, p. 783–98, 2000.

FANTI, M.; ZHOU, M. Deadlock control methods in automated manufacturing systems. **IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans**, v. 34, n. 1, p. 5–22, 2004.

FAZZOLARI, M.; ALCALA, R.; NOJIMA, Y.; ISHIBUCHI, H.; HERRERA, F. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems : Current Status and Further Directions. **IEEE Transactions on Control Systems Technology**, v. 21, n. 1, p. 45–65, 2013.

FORREST, S. Genetic algorithms: principles of natural selection applied to computation. **Science**, v. 261, n. 5123, p. 872–878, 1993.

GABRIEL, P.; DELBEM, A. **Fundamentos de Algoritmos Evolutivos**. São Carlos - SP - Brazil: ICMC - SP, 2008.

GAFFNEY, J.; GREEN, D. A.; PEARCE, C. E. M. Binary Versus Real Coding for Genetic Algorithms: A False Dichotomy? **ANZIAM Journal**, v. 51, p. 347–359, 2010.

GARRIDO, A. A Brief History of Fuzzy Logic. **BRAIN. Broad Research in Artificial Intelligence and Neuroscience**, v. 3, n. 1, p. 71–77, 2012.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. New York - NY - USA: Addison-Wesley, 1989.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic Algorithms and Machine Learning. **Machine Learning**, v. 3, n. 2-3, p. 95–99, 1988.

GOMES, L. On conflict resolution in Petri nets models through model structuring and composition. **3rd IEEE International Conference on Industrial Informatics**, p. 489–494, 2005.

GOMIDE, F. A. C.; GUDWIN, R. R.; TANSCHKEIT, R. Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações. **Proceedings of 6th International Fuzzy Systems Association World Congress-IFSA95, Tutorials**, 1995.

HAN, X.; OUYANG, M.; LU, L.; LI, J. A comparative study of commercial lithium ion battery cycle life in electric vehicle: Capacity loss estimation. **Journal of Power Sources**, v. 268, p. 658–669, 2014. Elsevier B.V.

HASOON, S. O.; JASIM, Y. A. Diagnosis Windows Problems Based on Hybrid Intelligence Systems. **Journal of Engineering Science and Technology**, v. 8, n. 5, p. 566–577, 2013.

HERRERA, F.; LOZANO, M. **Fuzzy Genetic Algorithms : Issues and Models**. Granada - Spain: Department of Computer Science and A.I. - University of Granada, 1999.

HERRERA, F.; LOZANO, M.; VERDEGAY, J. Applying genetic algorithms in fuzzy optimization problems. **Fuzzy Systems & Artificial Intelligence Reports and Letters**, v. 1, p. 39–52, 1994.

HERRERA, F.; LOZANO, M.; VERDEGAY, J. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. **Artificial Intelligence Review**, p. 265–319, 1998.

HINTERDING, R.; MICHALEWICZ, Z.; EIBEN, A. E. Adaptation in evolutionary computation: A survey. **Transactions on Evolutionary Computation**, v. 3, n. 2, p. 124–141, 1997.

HIRAMA, K. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro - RJ - Brazil: Elsevier, 2012.

HOLLAND, J. H. Nonlinear environments permitting efficient adaptation. **Computer and Information Sciences II**, 1967. Academic Press.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Michigan - USA: University of Michigan Press, 1975.

IVAN, I.; CIUREA, C. Quality Characteristics of Collaborative Systems. **Second International Conferences on Advances in Computer-Human Interactions**, p. 164–168, 2009. Cancun - México: IEEE.

JAMSHIDI, O.; PILEVAR, A. H. Automatic Segmentation of Medical Images Using Fuzzy c-Means and the Genetic Algorithm. **Journal of Computational Medicine**, v. 2013, p. 1–7, 2013.

JONG, K. A. DE. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems**, 1975. University of Michigan.

JONG, K. A. DE. **Evolutionary Computation: A Unified Approach**. Cambridge - MA - USA: The MIT Press, 2006.

KATO, E. R. R.; MORANDIN JR., O.; SGAVIOLI, M. A Conflict Solution Manufacturing System Modeling Using Fuzzy Coloured Petri Net. **International Conference on Systems, Man and Cybernetics**, p. 3983–3988, 2010. Istanbul - Turkey: IEEE.

KEUNG, K. W.; IP, W. H.; YUEN, D. An intelligent hierarchical workstation control model for FMS. **Journal of Materials Processing Technology**, v. 139, n. 1-3, p. 134–139, 2003.

KHOO, L. P.; ANG, C. L.; ZHANG, J. A Fuzzy-Based Genetic Approach to the Diagnosis of Manufacturing Systems. **Engineering Applications of Artificial Intelligence**, v. 13, n. 3, p. 303–310, 2000.

KOEHN, P. **Combining Genetic Algorithms and Neural Networks : The Encoding Problem** Thesis ANN p 67, 1994.

KUMAR, A. Encondig Schemes in Genetic Algorithm. **International Journal of Advanced Research in IT and Engineering**, v. 2, n. 3, p. 1–7, 2013.

KUMAR, M.; HUSIAN, M.; UPRETI, N.; GUPTA, D. Genetic Algorithm: Review and Application. **International Journal of Information Technology and Knowledge Management**, v. 2, n. 2, p. 451–454, 2010.

LI, C.; WU, W.; RONG, G. Heuristic search and concurrency strategy based on Petri net for FMS scheduling. **Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control**, p. 80–85, 2014. Miami - FL - USA.

LIAO, T. W.; CELMINS, A. K.; HAMMELL, R. J. A fuzzy c-means variant for the generation of fuzzy term sets. **Fuzzy Sets and Systems**, v. 135, n. 2, p. 241–257, 2003.

LINDEN, R. **Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional**. 3rd ed. Rio de Janeiro - Brazil: Brasport, 2012.

LINGALA, M.; STANLEY, R. J.; RADER, R. K.; et al. Fuzzy logic color detection: Blue areas in melanoma dermoscopy images. **Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society**, v. 38, n. 5, p. 403–10, 2014. Elsevier Ltd.

LUSTOSA, L.; MESQUISA, M. A.; QUELHAS, O.; OLIVEIRA, R. **Planejamento e Controle da Produção**. Rio de Janeiro - RJ - Brazil: Elsevier, 2008.

MAMDANI, E. H. Application of fuzzy algorithms for control of simple dynamic plant. **Proceedings of the Institution of Electrical Engineers**, v. 121, n. 12, p. 1585–1588, 1974.

MAMDANI, E. H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. **IEEE Transactions on Computers**, v. C-26, n. 12, p. 1182–1191, 1977.

MEDEIROS, S. C. D. A. **Inversão de Parâmetros em Dados Sísmicos por Algoritmos Genéticos**, 2005. Pontifícia Universidade Católica do Rio de Janeiro.

MENDEL, G. Versuche über Pflanzenhybriden. **Verhandlungen des naturforschenden Vereins Bruenn**, 1866.

MITCHELL, M. **An introduction to genetic algorithms**. Fifth ed. London - England: MIT Press, 1998.

MORANDIN JR, O.; KATO, R. R. E.; ARAUJO, G. R.; SASSO, V. L. A modeling strategy for control and interlocking of an AMS using Virtual Petri Nets. **International Conference on Systems, Man and Cybernetics**, p. 3493–3498, Oct. 2007. Montreal - Que: IEEE.

- MORANDIN JR., O.; KATO, E. R. R. Virtual Petri nets as a modular modeling method for planning and control tasks of FMS. International Conference on Systems, Man and Cybernetics. **Anais...** . v. 2, p.1521–1527, 2003. IEEE.
- MUKAIDONO, M. **Fuzzy Logic for Beginners**. Singapore: World Scientific, 2001.
- MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541–580, 1989.
- NANVALA, H. Use of Fuzzy Logic Approaches in Scheduling of FMS : A Review. **International Journal on Computer Science and Engineering**, v. 3, n. 4, p. 1734–1739, 2011.
- NAUCK, D.; KRUSE, R. A neuro-fuzzy method to learn fuzzy classification rules from data. **Fuzzy Sets and Systems**, v. 89, n. 3, p. 277–288, 1997.
- NGUYEN, H. T.; WALKER, E. A. **A First Course in Fuzzy Logic**. Third ed. Boca Raton - FL - USA: Chapman & Hall/CRC, 2006.
- OUIDDIR, R.; RHALI, M.; ABDELHAKEM-KORIDAK, L. Economic Dispatch using a Genetic Algorithm: Application to Western Algeria’s Electrical Power Network. **Journal of Information Sciences and Engineering**, v. 21, p. 659–668, 2005.
- PACHECO, M. A. C. **Algoritmos genéticos: princípios e aplicações**. 1999.
- PARK, J.; BAE, H.; DINH, T.-C.; RYU, K. Operator allocation in cellular manufacturing systems by integrated genetic algorithm and fuzzy data envelopment analysis. **The International Journal of Advanced Manufacturing Technology**, 2014.
- PRAKASH, A.; CHAN, F. T. S.; DESHMUKH, S. G. FMS scheduling with knowledge based genetic algorithm approach. **Expert Systems with Applications**, v. 38, n. 4, p. 3161–3171, 2011. Elsevier Ltd.
- PRESSMAN, R. S. **Engenharia de Software**. 6th ed. São Paulo - SP - Brazil: MacGraw-Hill, 2010.
- QIN, Z.; BAI, M.; RALESCU, D. A fuzzy control system with application to production planning problems. **Information Sciences**, v. 181, n. 5, p. 1018–1027, 2011.
- RABÊLO, R. A. L.; FERNANDES, R. A. S.; CARNEIRO, A. A. F. M.; BRAGA, R. T. V. Uma abordagem baseada em sistemas de inferência Fuzzy Takagi-Sugeno aplicada ao planejamento da operação de sistemas hidrotérmicos de geração. **Sba: Controle & Automação**, v. 22, n. 1, p. 49–64, 2011.
- REIS, F. B. DOS. **Estratégia de Modelagem para Controle de FMS Combinando Redes de Petri e um Sistema Fuzzy**, 2012. Universidade Federal de São Carlos - UFSCar.
- REIS, F.; CARIDÁ, V.; MORANDIN JR., O.; CASTRO, R.; TUMA, C. A Colaborative Fuzzy CPN System for Conflict Solution of Flexible Manufacturing System. **Industrial**

Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, p. 3216–3221, 2013. Vienna - Austria.

REZENDE, S. O. **Sistemas Inteligentes: Fundamentos e Aplicações**. 1st ed. Barueri - SP - Brazil: Manole, 2005.

REZNIK, L. **Fuzzy Controllers**. United Kingdom: Newnes, 1997.

ROYES, G. F.; BASTOS, R. C. Fuzzy sets in political science. **IFSA World Congress and 20th NAFIPS International Conference**, v. 2, p. 935–940, 2001.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd ed. New Jersey - USA: Prentice-Hall, 2009.

SGAVIOLI, M. **Modelagem de Sistemas de Manufatura Usando Redes de Petri Coloridas Fuzzy Focando a Solução de Conflitos**, 2010. Universidade Federal de São Carlos - UFSCar.

SHLEEG, A. A.; ELLABIB, I. M. Comparison of Mamdani and Sugeno Fuzzy Interference Systems for the Breast Cancer Risk. **International Journal of Computer, Information, Systems and Control Engineering**, v. 7, n. 10, p. 695–699, 2013.

SINGH, H.; GUPTA, M. M.; MEITZLER, T.; et al. Real-Life Applications of Fuzzy Logic. **Advances in Fuzzy Systems**, p. 1–3, 2013.

SOUIER, M.; SARI, Z.; HASSAM, A. Real-time rescheduling metaheuristic algorithms applied to FMS with routing flexibility. **The International Journal of Advanced Manufacturing Technology**, v. 64, n. 1-4, p. 145–164, 2012.

SRIVASTAVA, P. R.; KIM, T. Application of Genetic Algorithm in Software Testing. **International Journal of Software Engineering and Its Applications**, v. 3, n. 4, p. 87–96, 2009.

TABASSUM, M.; MATHEW, K. A Genetic Algorithm Analysis towards Optimization solutions. **International Journal of Digital Information and Wireless Communication**, v. 4, n. 1, p. 124–142, 2014.

TAKAGI, T.; SUGENO, M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. SMC-15, n. 1, p. 116–132, 1985.

TEIMOURZADEH, S.; ZARE, K. Application of binary group search optimization to distribution network reconfiguration. **International Journal of Electrical Power and Energy Systems**, v. 62, p. 461–468, 2014.

TOSATTI, M. A.; FILHO, J. C. A. G.; FORBECK, F. R.; et al. Algoritmo Híbrido Genético-Fuzzy aplicado em Câncer de Próstata. **Hifen**, v. 32, n. 62, p. 164–171, 2008.

TUNCEL, G. A Heuristic Rule-Based Approach for Dynamic Scheduling of Flexible Manufacturing Systems. **Multiprocessor Scheduling, Theory and Applications**. p.315–323, 2007. Vienna - Austria.

YUAN, Y.; ZHUANG, H. A genetic algorithm for generating fuzzy classification rules. **Fuzzy Sets and Systems**, v. 84, n. 1, p. 1–19, 1996.

YUFENG, C.; ZHIWU, L.; AL-AHMARI, A. Nonpure Petri Net Supervisors for Optimal Deadlock Control of Flexible Manufacturing Systems. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 43, n. 2, p. 252–265, 2013.

ZADEH, L. A. Fuzzy Sets. **Information and Control**, v. 353, p. 338–353, 1965.

ZADEH, L. A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. **Systems, Man and Cybernetics, IEEE Transactions**, p. 28–44, 1973.

Apêndice A

TESTES DE VALIDAÇÃO

8.1 Considerações Iniciais

Neste apêndice são apresentadas as tabelas e os gráficos referentes ao desempenho do EvolSys para cada uma das configurações executadas. A objetivo da execução desses testes é garantir que todos os requisitos do sistema evolutivo tenham a funcionalidade e o impacto na convergência esperado para os mesmos.

É importante ressaltar que o cálculo do *fitness* é realizado comparando o cromossomo de cada um dos indivíduos com a FRB Ideal para verificar o quão semelhante é esse cromossomo em relação à própria FRB Ideal. Desse modo, quanto maior for a semelhança entre o indivíduo e a FRB Ideal maior será o valor do *fitness* e quanto menor for a semelhança entre o indivíduo e a FRB Ideal menor será o valor do *fitness* (mais detalhes no tópico 4.2.4 - Cálculo do *Fitness* – página 88).

Desse modo, se o indivíduo gerado for exatamente igual à FRB Ideal, então tem-se 100% de semelhança e o *fitness* é igual a 100 (*fitness* máximo). Neste caso, a execução para, pois o critério de parada foi atingido (*fitness* igual a 100 o que representa 100% de semelhança entre o cromossomo do melhor indivíduo e a FRB Ideal).

A seguir, serão exibidas as configurações executadas durante os testes com os respectivos gráficos. Vale lembrar que, a partir do tópico 8.7, o espaço de busca será o mesmo em todos os casos (tamanho e gramática dos cromossomos sempre iguais), pois não ocorrerão alterações referentes às Regras *Fuzzy*.

8.2 Execução Padrão

Tabela 8-1 - Execução Padrão

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção de Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
01	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

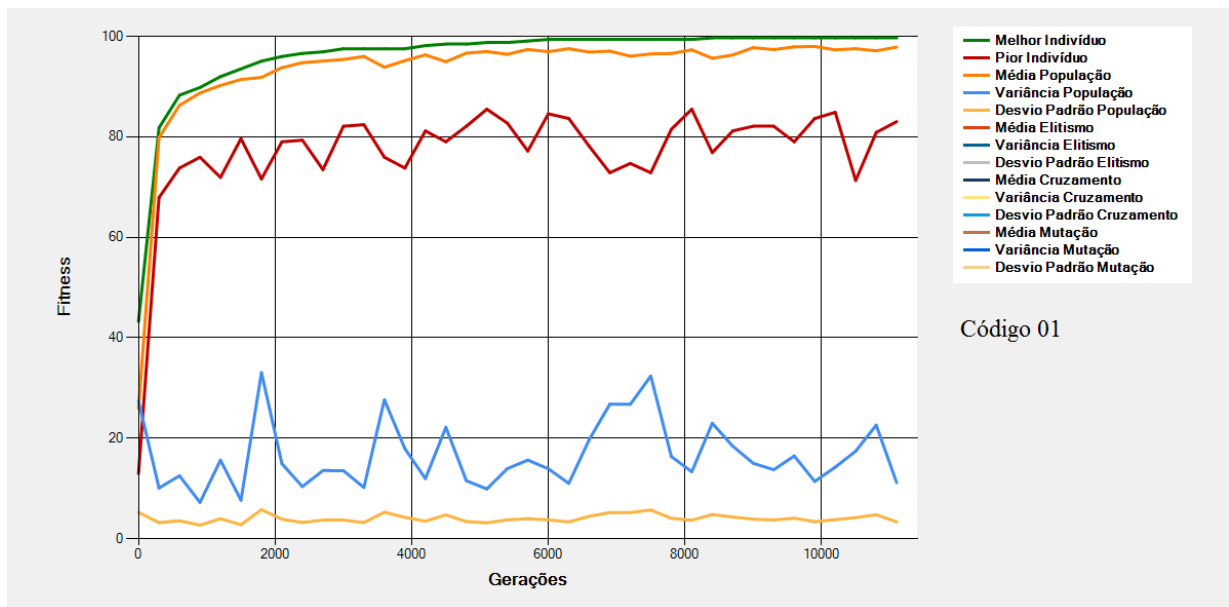


Figura 8-1 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-1 - Execução Padrão (Código 01)

É mostrado na Figura 8-1 o gráfico do Melhor Indivíduo, Média da População, Pior Indivíduo, Variância da População e Desvio Padrão da População de acordo com os dados da Tabela 8-1. Nessa tabela tem-se a configuração de execução padrão a qual será a base para realização das próximas execuções. Alguns dos campos serão alterados a cada execução para que seja possível verificar os comportamentos dos gráficos quando da alteração de cada um dos campos. Na execução acima a parada do algoritmo se deu com cerca de 11 mil gerações de um total de 100 mil possíveis de acordo com a configuração da Tabela 8-1.

8.3 Regras Fuzzy - Quantidade de Variáveis de Entrada

Tabela 8-2 - Regras Fuzzy - Quantidade de Variáveis de Entrada

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutaç�o	Execuç�o
	Quantidade de Vari�veis de Entrada	Quantidade de Valores Poss�veis para Cada Vari�vel	Quantidade de Vari�veis de Sa�da	Quantidade de Valores Poss�veis para Cada Vari�vel		Cota de Elitismo	Cota de Cruzamento	Cota de Mutaç�o	Quantidade de Grupos de	Proporç�o dos Grupos de	M�todo de Seleç�o	Taxa de Seleç�o	Quantidade de Progenitores por	Quantidade de Pontos de Corte		
02	2	3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
03	3	3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
04	5	3, 3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

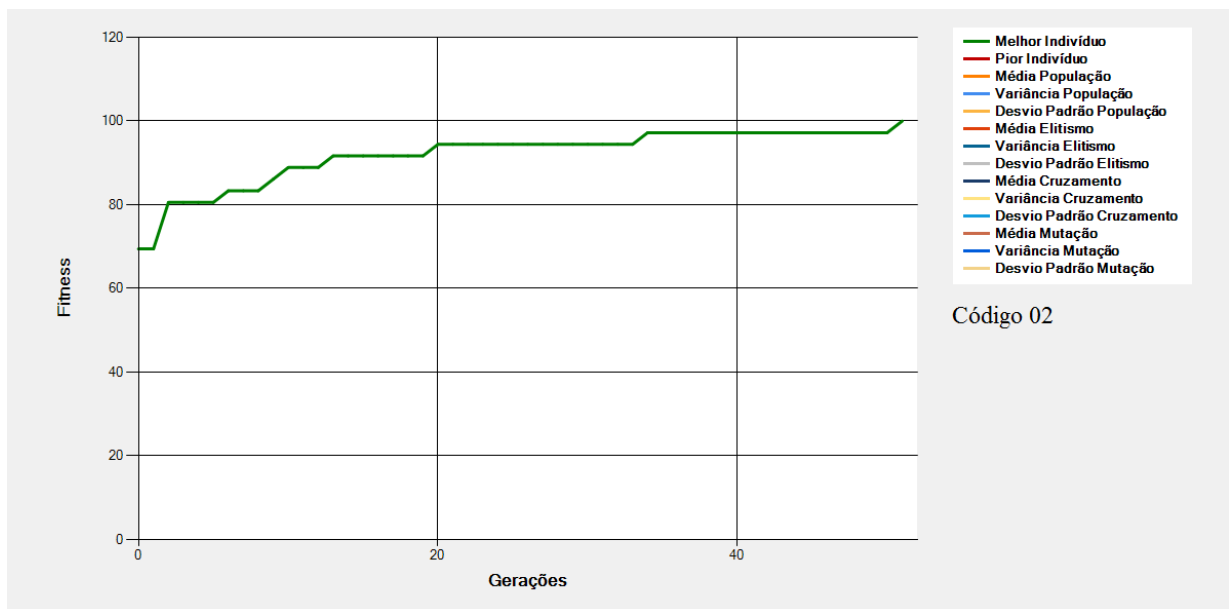


Figura 8-2 - Gr fico do Melhor Indiv duo de acordo com os dados da Tabela 8-2 - Regras Fuzzy (C digo 02)

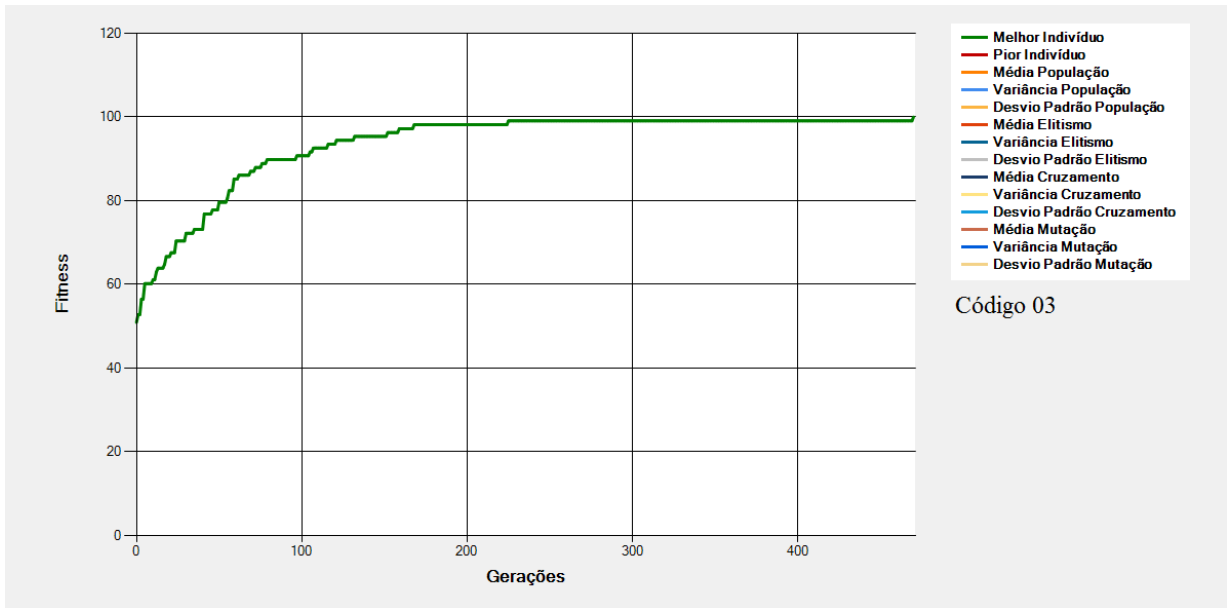


Figura 8-3 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-2 - Regras Fuzzy (Código 03)

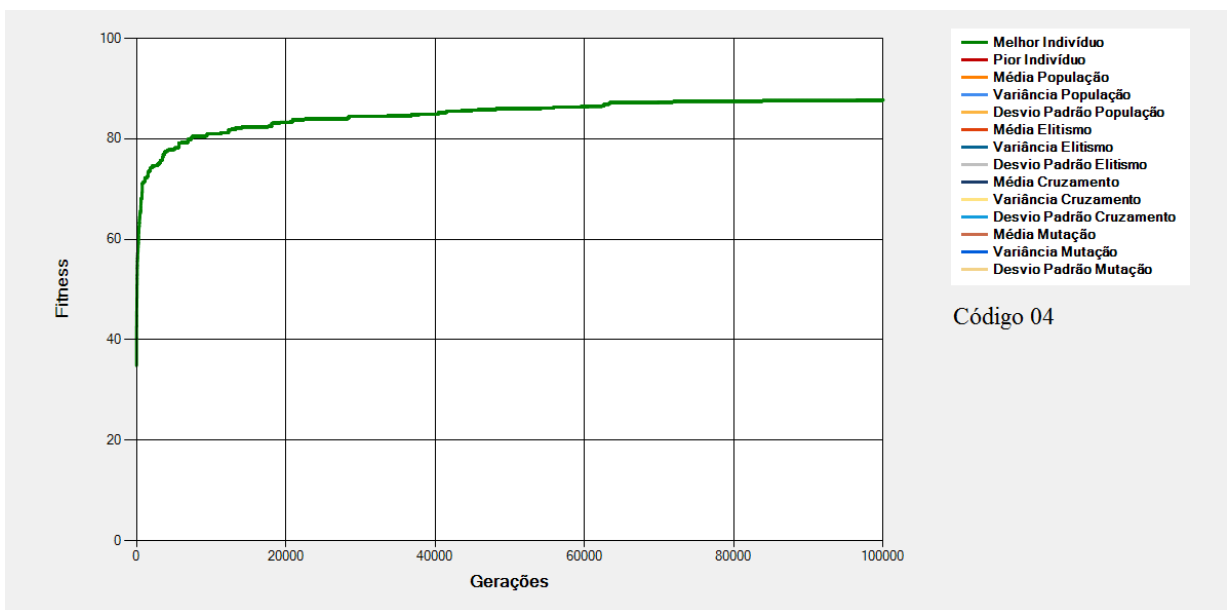


Figura 8-4 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-2 - Regras Fuzzy (Código 04)

São mostrados na Figura 8-2, na Figura 8-3 e na Figura 8-4 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-2. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Variáveis de Entrada” e nos “Valores Possíveis para Cada Variável de Entrada”.

Para as execuções de códigos 02, 03 e 04 tem-se cromossomos de tamanhos **9** ($3 * 3$), **27** ($3 * 3 * 3$) e **243** ($3 * 3 * 3 * 3 * 3$) respectivamente³⁰. Nestes casos, os valores referentes a “Quantidade de Variáveis de Entrada” e aos “Valores Possíveis para Cada Variável de Entrada” alteram o espaço de busca do EA fazendo com que o número de gerações máximo e a convergência sejam bem distintos em cada um dos gráficos.

No gráfico da Figura 8-2 por exemplo, foram necessárias cerca de 50 gerações para que o EA atingisse o critério de parada. No gráfico da Figura 8-3 foram necessárias cerca de 500 gerações para que o critério de parada fosse atingido e na Figura 8-4 o algoritmo parou pois o número máximo de gerações (100 mil gerações) definidos na configuração foi atingido. Percebe-se que na Figura 8-4, 100mil gerações não foram suficientes para que o EA chegasse ao ótimo.

³⁰ Tamanho dos cromossomos obtidos por intermédio da multiplicação entre os valores possíveis para cada variável de entrada.

8.4 Regras Fuzzy - Quantidade de Valores Possíveis para Cada Variável de Entrada

Tabela 8-3 - Regras Fuzzy - Quantidade de Valores Possíveis para Cada Variável de Entrada

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por	Quantidade de Pontos de Corte		
05	4	2, 2, 2, 2	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
06	4	4, 4, 4, 4	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
07	4	5, 5, 5, 5	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

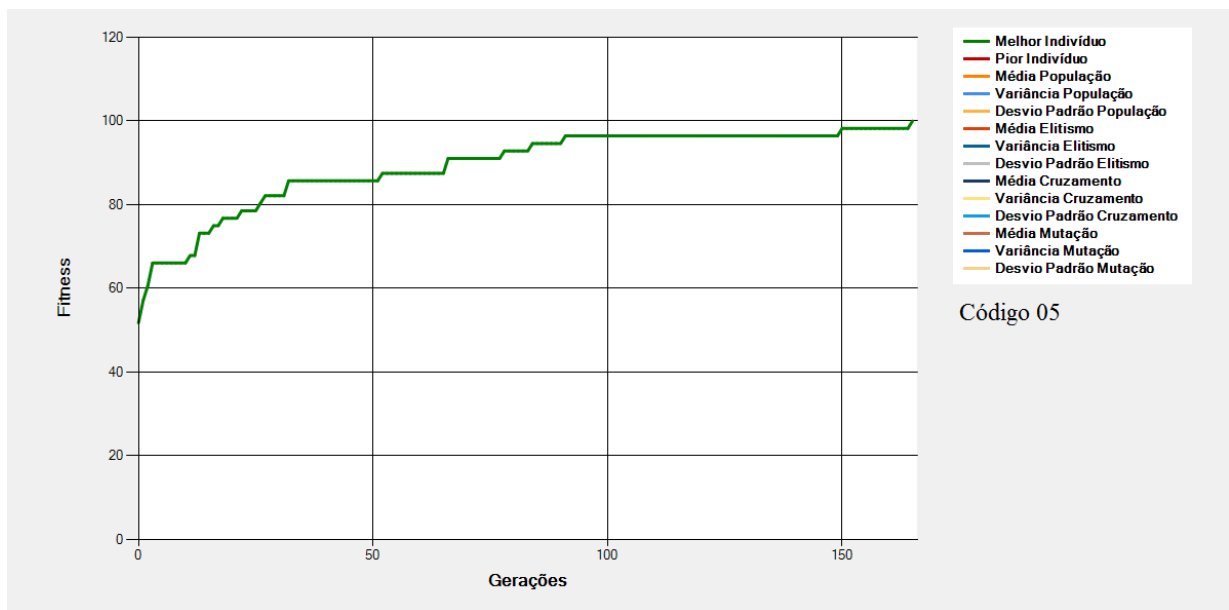


Figura 8-5 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras Fuzzy (Código 05)

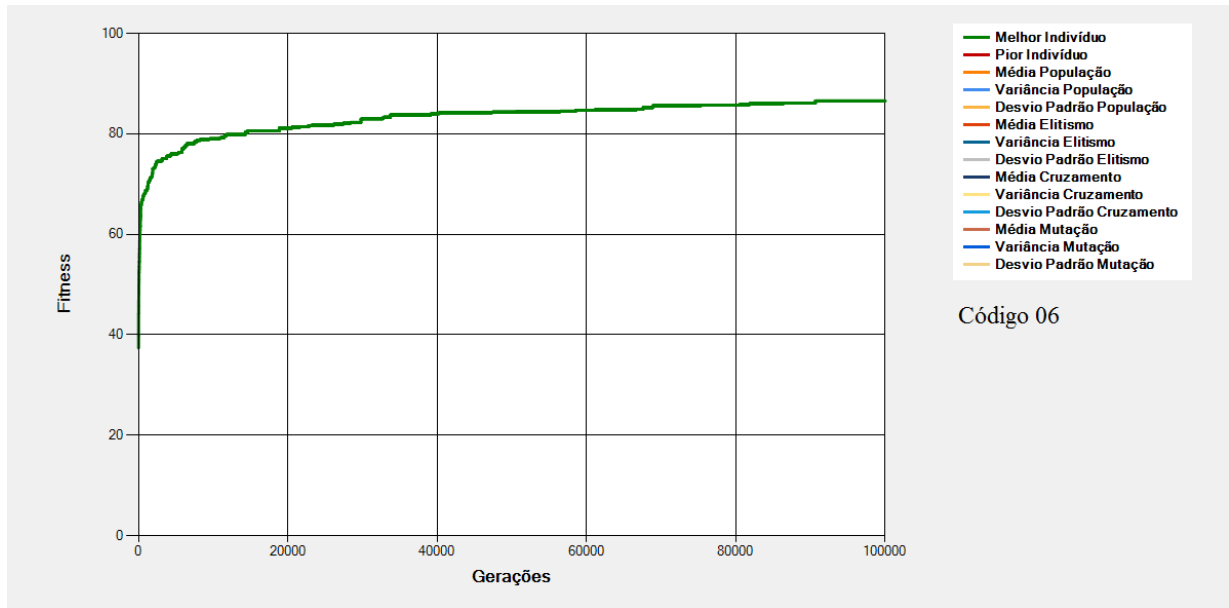


Figura 8-6 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras Fuzzy (Código 06)

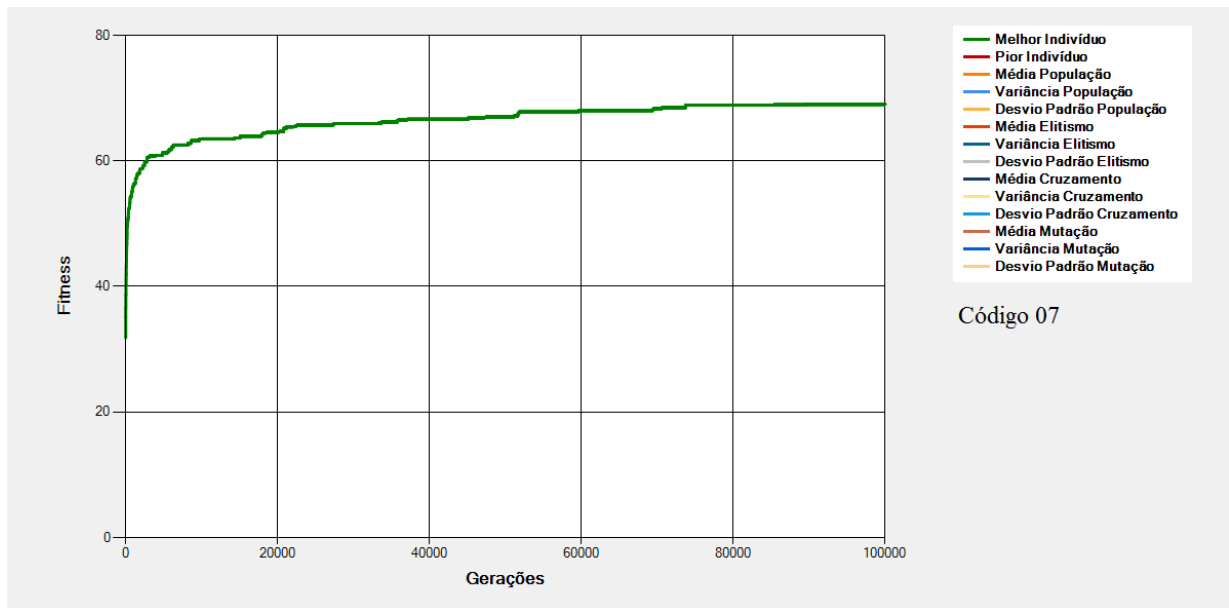


Figura 8-7 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-3 - Regras Fuzzy (Código 07)

São mostrados na Figura 8-5, na Figura 8-6 e na Figura 8-7 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-3. Nessas configurações tem-se alterações (em relação à Execução Padrão) nos “Valores Possíveis para Cada Variável de Entrada”.

Para as execuções de códigos 05, 06 e 07 tem-se cromossomos de tamanhos **16** ($2 * 2 * 2 * 2$), **256** ($4 * 4 * 4 * 4$) e **625** ($5 * 5 * 5 * 5$) respectivamente³¹. Nestes casos, as configurações referentes

³¹ Tamanho dos cromossomos obtidos por intermédio da multiplicação entre os valores possíveis para cada variável de entrada.

aos “Valores Possíveis para Cada Variável de Entrada” alteram o espaço de busca do EA fazendo com que o número de gerações máximo e a convergência sejam bem distintos em cada um dos gráficos.

No gráfico da Figura 8-5 por exemplo, foram necessárias cerca de 180 gerações para que o EA atingisse o critério de parada. Já nos gráficos das Figura 8-6 e Figura 8-7 as 100mil gerações (número máximo de gerações configurados) não foram suficientes para que o EA atingisse o critério de parada.

Na Figura 8-6 a execução parou quando o melhor indivíduo estava com *fitness* em torno de 85. Já na Figura 8-7 a execução parou com o melhor indivíduo com *fitness* em torno de 65. Isso se explica pelo fato do espaço de busca da configuração de código 06 (cromossomos de tamanhos 256) ser menor que o espaço de busca da configuração 07 (cromossomos de tamanhos 625) provocando (para o caso da configuração de código 07) uma convergência mais lenta e, conseqüentemente, a obtenção de indivíduos com menor valor de *fitness* em relação à configurações que levem a espaço de busca menores.

8.5 Regras Fuzzy - Quantidade de Variáveis de Saída

Tabela 8-4 - Regras Fuzzy - Quantidade de Variáveis de Saída

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Quantidade de Indivíduos	Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento		
08	4	3, 3, 3, 3	1	3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
09	4	3, 3, 3, 3	3	3, 3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
10	4	3, 3, 3, 3	5	3, 3, 3, 3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

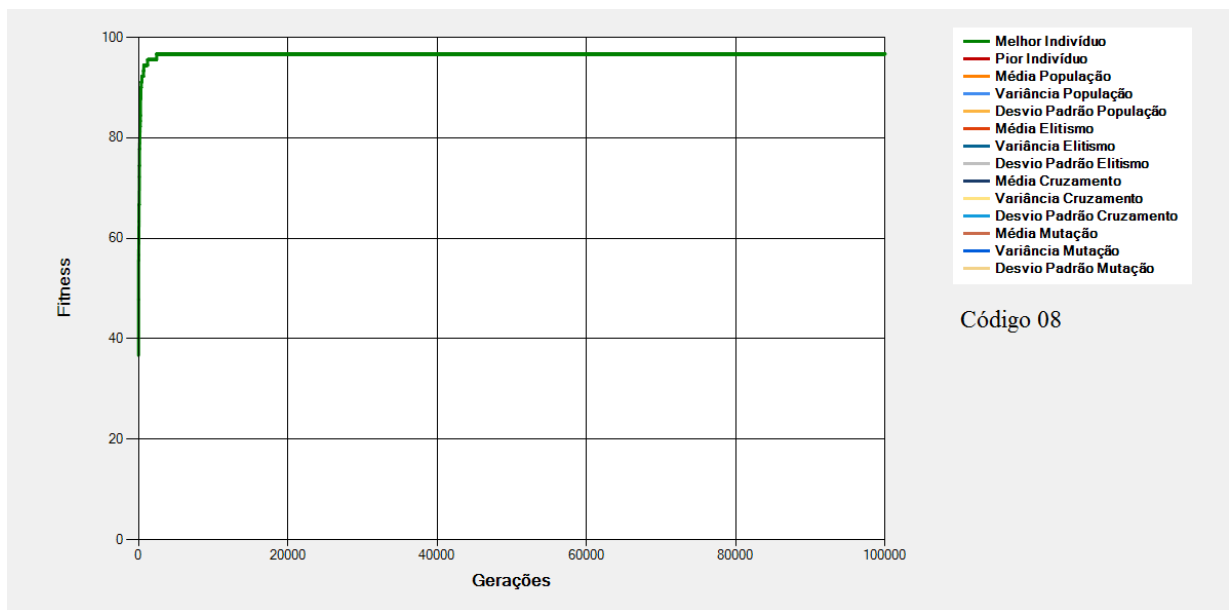


Figura 8-8 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras Fuzzy (Código 08)

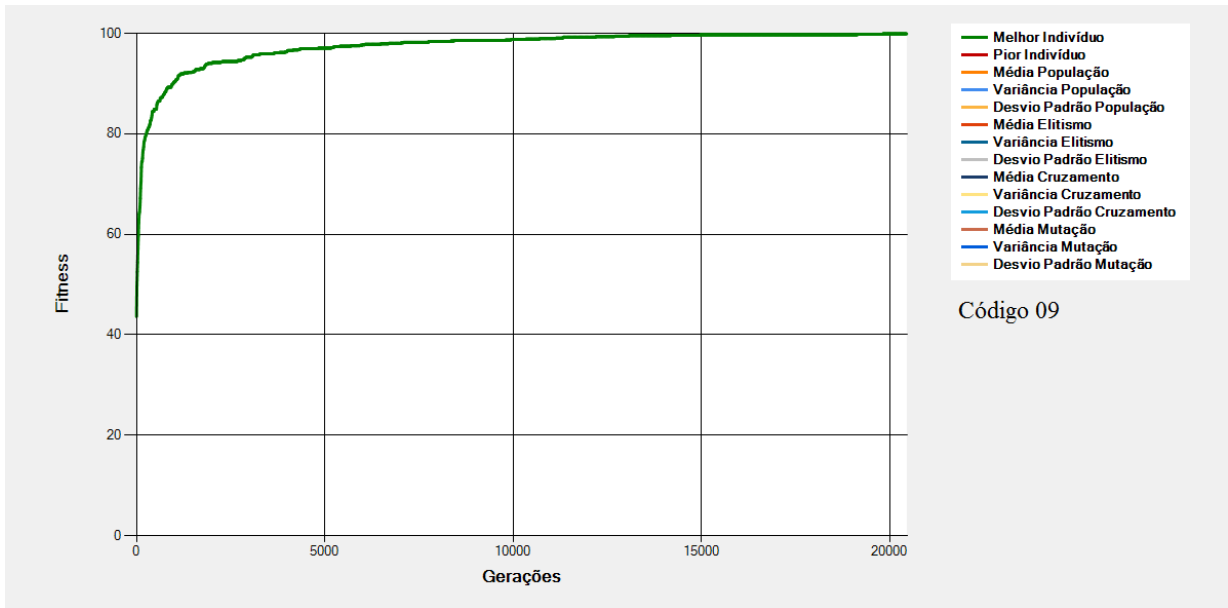


Figura 8-9 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras *Fuzzy* (Código 09)

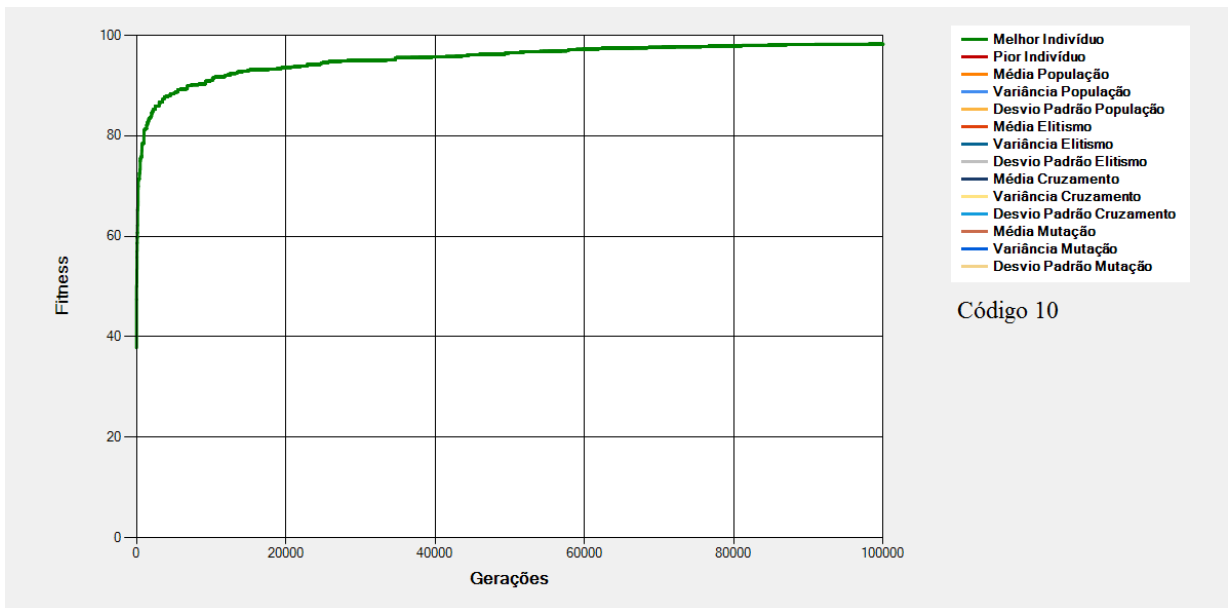


Figura 8-10 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-4 - Regras *Fuzzy* (Código 10)

São mostrados na Figura 8-8, na Figura 8-9 e na Figura 8-10 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-4. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Variáveis de Saída” e nos “Valores Possíveis para Cada Variável de Saída”.

Para as execuções de códigos 08, 09 e 10 tem-se as gramáticas (valores possíveis em cada gene) **3**, **27** ($3 * 3 * 3$) e **243** ($3 * 3 * 3 * 3 * 3$) respectivamente³².

A Figura 8-8 exibe o gráfico da execução de código 08. Nela percebe-se que a convergência do EA se dá de forma bastante acentuada e em seguida o gráfico fica constante. Isso indica a possibilidade da busca ter sido conduzida para um ótimo local.

Ainda na execução de código 08, o fato de os valores da gramática variarem em apenas 3 valores possíveis, é determinante para que o espaço de busca seja menor em relação às execuções de código 09 e 10. Neste caso, como os operadores de Elitismo e Cruzamento (que conduzem à convergência) estão atuando em intensidade bem maior que o operador de Mutação (que conduz a uma dispersão maior dos dados) a tendência é que, em um espaço de busca menor, a convergência se dê muito rapidamente o que pode acarretar em estacionamentos em ótimos locais.

Nas execuções de código 09 e 10, como o espaço de busca é maior (mais valores possíveis nas gramáticas) a convergência se dá mais lentamente o que possibilita o tempo necessário para que o operador de Mutação atue efetivamente de modo a buscar soluções em espaço de busca inexplorados fazendo com que seja diminuída a tendência de convergências prematuras para ótimos locais.

³² O conjunto de valores possíveis em cada um dos genes, aqui chamado de “gramática”, é obtido por intermédio da multiplicação entre os valores possíveis para cada variável de saída.

8.6 Regras Fuzzy - Quantidade de Valores Possíveis para Cada Variável de Saída

Tabela 8-5 - Regras Fuzzy - Quantidade de Valores Possíveis para Cada Variável de Saída

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
11	4	3, 3, 3, 3	2	2, 2	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
12	4	3, 3, 3, 3	2	4, 4	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
13	4	3, 3, 3, 3	2	5, 5	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

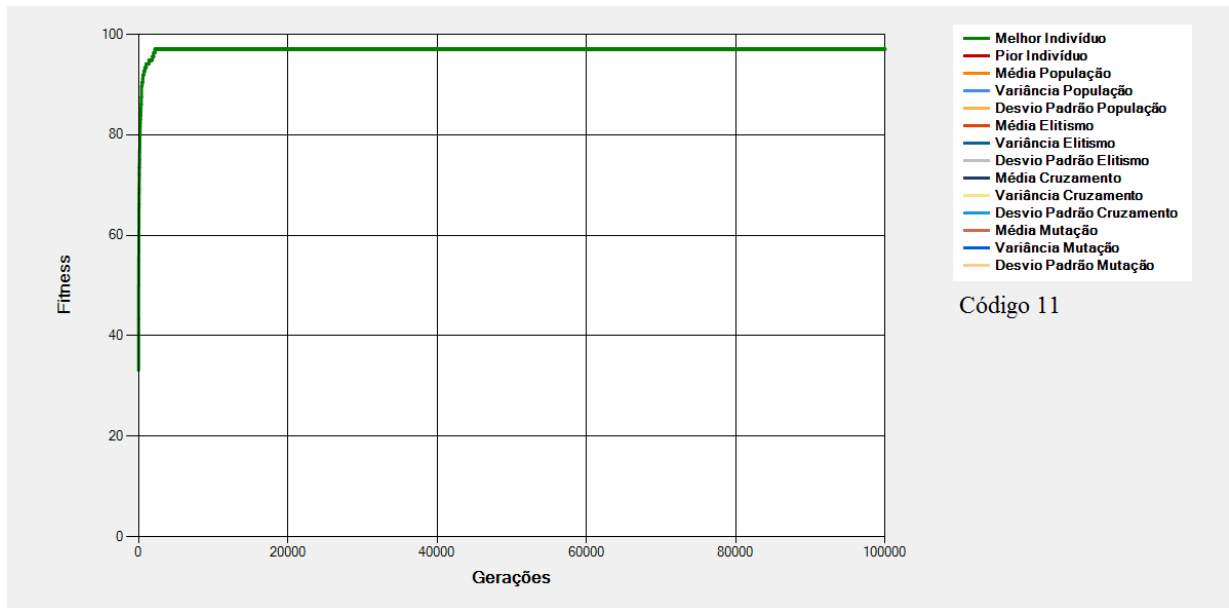


Figura 8-11 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras Fuzzy (Código 11)

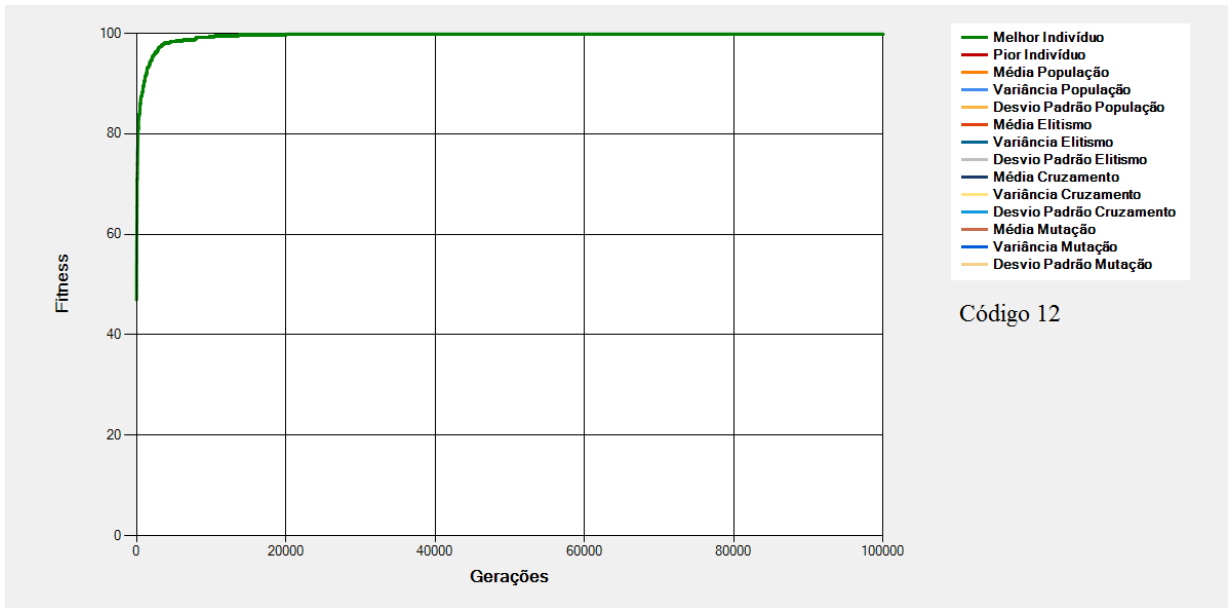


Figura 8-12 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras Fuzzy (Código 12)

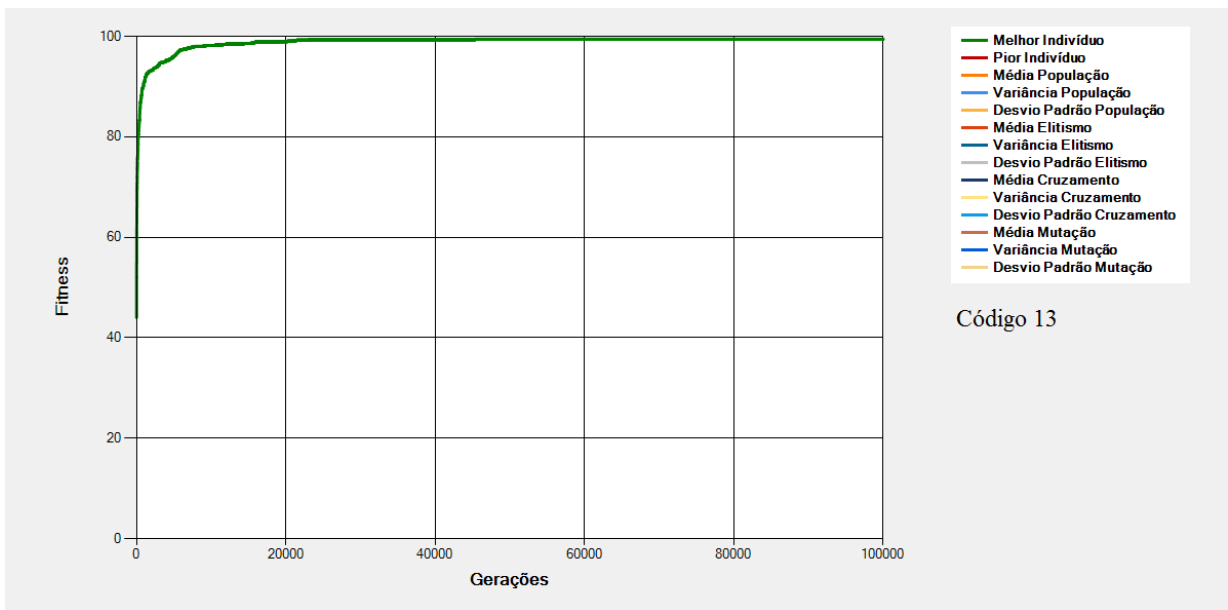


Figura 8-13 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-5 - Regras Fuzzy (Código 13)

São mostrados na Figura 8-11, na Figura 8-12 e na Figura 8-13 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-5. Nessas configurações tem-se alterações (em relação à Execução Padrão) nos “Valores Possíveis para Cada Variável de Saída”.

Para as execuções de códigos 11, 12 e 13 tem-se as gramáticas (valores possíveis em cada gene) **4** ($2 * 2$), **16** ($4 * 4$) e **25** ($5 * 5$) respectivamente³³.

Nos gráficos acima percebe-se que há a tendência do EA ter conduzido a um estacionamento em um ótimo local. Nestes casos, como os espaços de buscas são relativamente pequenos, tem-se tendências a convergências prematuras para ótimos locais haja vista os valores dos operadores que conduzem à convergência que, somados, correspondem a 90% da população. (Elitismo 30 % e Cruzamento 60%). Observa-se também o fato de que, à medida que o espaço de busca aumenta, a convergência tende a ser mais lenta, o que dá o tempo necessário para que o operador de mutação atue, fazendo com que a busca seja realizada em outros locais, resultando assim, em indivíduos mais aptos.

³³ O conjunto de valores possíveis em cada um dos genes, aqui chamado de “gramática”, é obtido por intermédio da multiplicação entre os valores possíveis para cada variável de saída.

8.7 População - Quantidade de Indivíduos

Tabela 8-6 - População - Quantidade de Indivíduos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
14	4	3, 3, 3, 3	2	3, 3	10	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
15	4	3, 3, 3, 3	2	3, 3	500	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000
21	4	3, 3, 3, 3	2	3, 3	1000	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	100000

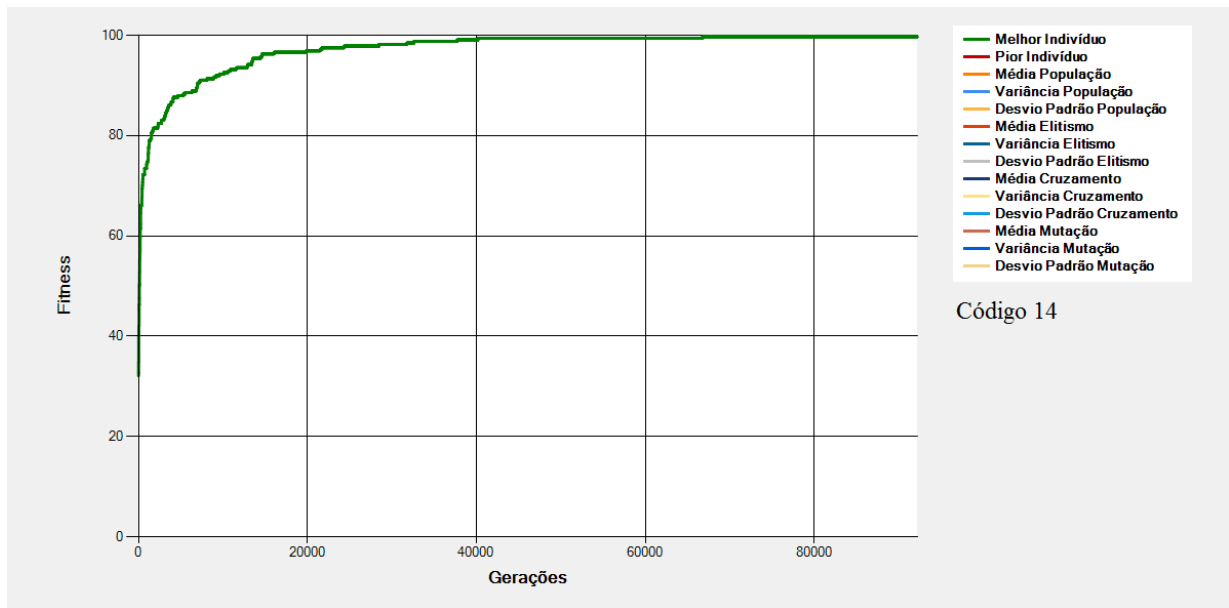


Figura 8-14 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 14)

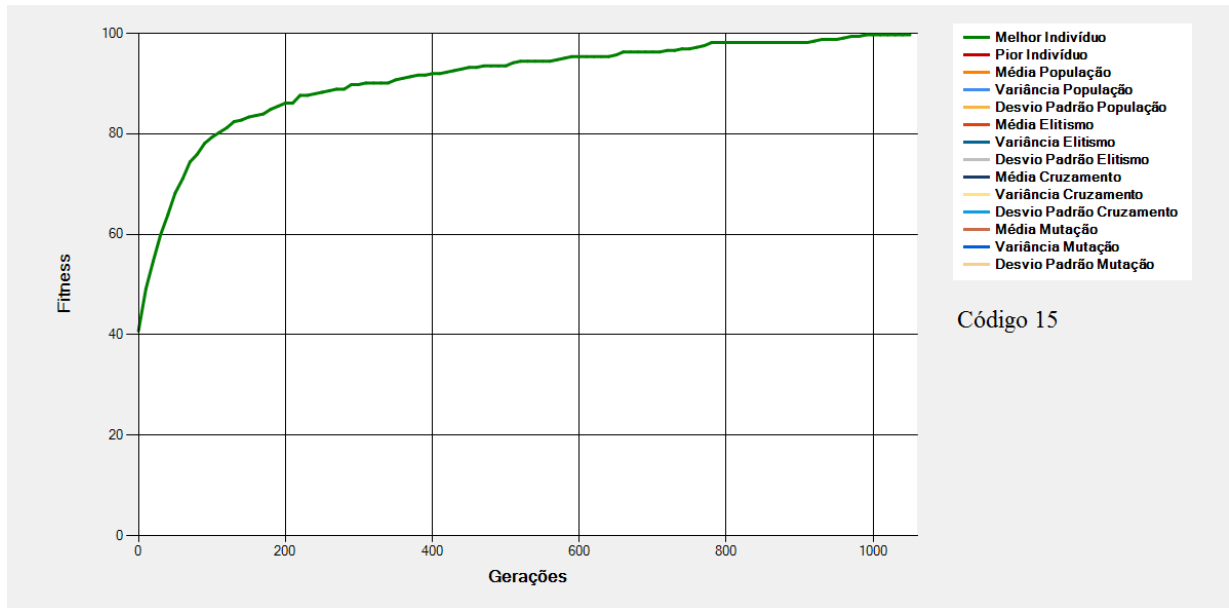


Figura 8-15 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 15)

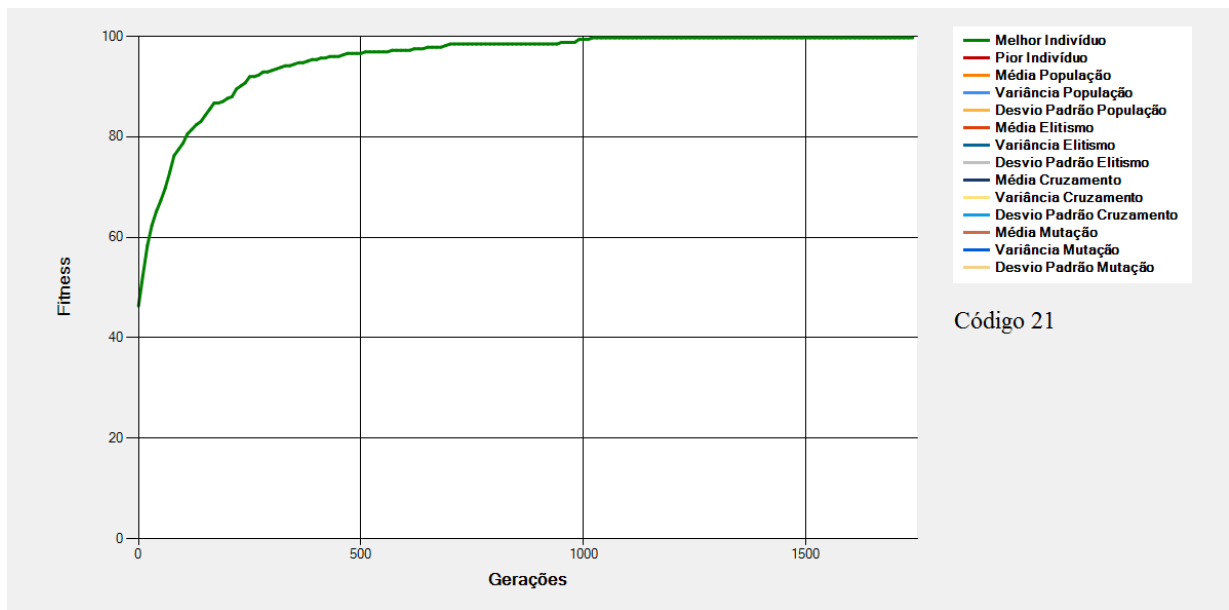


Figura 8-16 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-6 - População (Código 21)

São mostrados na Figura 8-14, na Figura 8-15 e na Figura 8-16 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-6. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Indivíduos”.

Nessas execuções, quanto maior a quantidade de indivíduos maior será a dispersão dos dados no espaço de soluções na busca pelo ótimo. Na Figura 8-14 por exemplo, cerca de 90 mil gerações são necessárias para que o ótimo seja atingido. Com o aumento de indivíduos na execução da Figura 8-15 percebe-se uma diminuição no número de gerações necessárias para que o ótimo seja alcançado

(Cerca de 1100 gerações) pois a busca pelo ótimo se dá em pontos mais dispersos do gráfico, o que tende a evitar convergências prematuras para ótimos locais.

No entanto, algo curioso acontece na execução referente ao gráfico da Figura 8-16. Neste caso o número de indivíduos dobrou em relação à execução da Figura 8-15. Entretanto, neste caso, o número de gerações para se chegar ao ótimo aumentou ao invés de diminuir. Isso ocorre porque uma quantidade grande de indivíduos tende a fazer com que bons indivíduos sejam perdidos durante os processos de seleção e cruzamento fazendo com que a convergência se torne mais lenta e o número de gerações aumente.

8.8 Cotas - 100% de Elitismo

Tabela 8-7 - Cotas - 100% de Elitismo

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
22	4	3, 3, 3, 3	2	3, 3	100	100%	0%	0%	1	100%	Roleta	50%	2	1	10%	100000

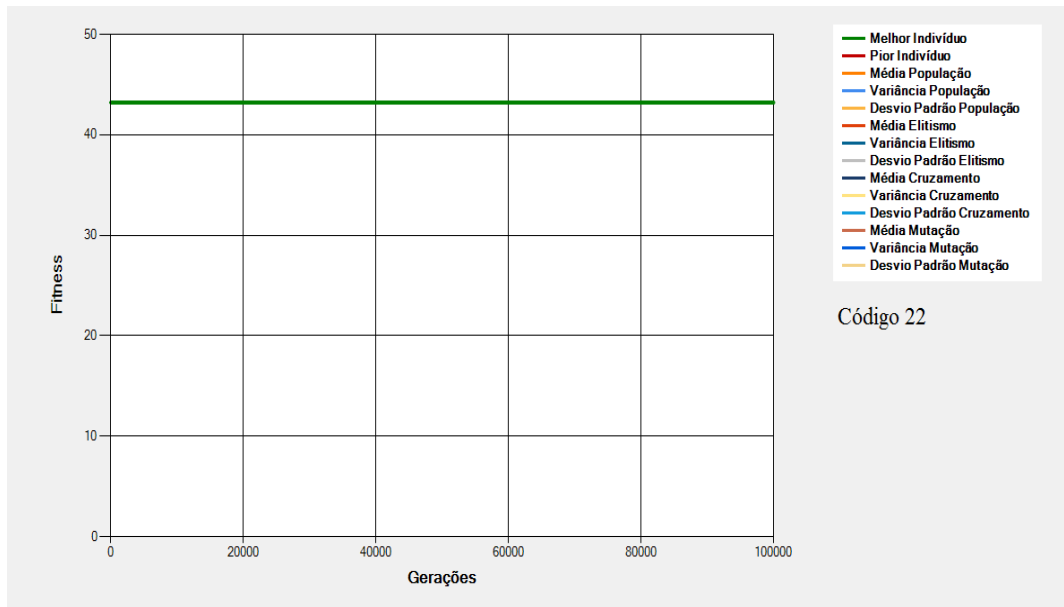


Figura 8-17 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-7 - Cotas (Código 22)

São mostrados na Figura 8-17 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-7. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Elitismo” na “Cota de Cruzamento” e na “Cota de Mutação”.

Nesta execução, percebe-se que, com 100% de elitismo, todos os indivíduos da geração inicial serão salvos e, desse modo não haverá mudanças em relação ao *fitness* do melhor indivíduo no decorrer das gerações (já que não há cruzamentos nem mutações). Com isso o gráfico da configuração fica uma constante no ponto do *fitness* do melhor indivíduo da geração inicial.

8.9 Cotas – 100% de Cruzamento

Tabela 8-8 - Cotas - 100% de Cruzamento

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
23	4	3, 3, 3, 3	2	3, 3	100	0%	100%	0%	1	100%	Roleta	50%	2	1	10%	100000

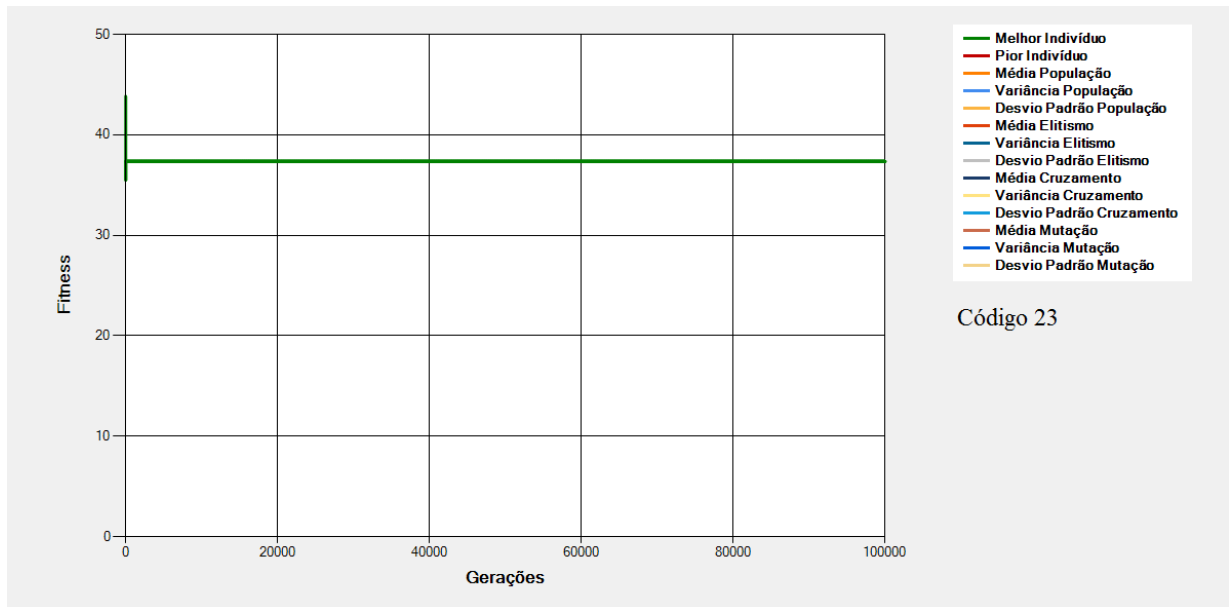


Figura 8-18 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-8 - Cotas (Código 23)

São mostrados na Figura 8-18 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-8. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Elitismo” na “Cota de Cruzamento” e na “Cota de Mutação”.

Nessa execução, percebe-se que, com 100% de Cota de Cruzamento, tem-se uma convergência prematura do EA. Nesse caso, como o operador de Cruzamento é o único operador atuante (já que não há elitismo nem mutações) e o mesmo é um operador que é usado para condução à convergência, essa se dá rapidamente e, logo nas primeiras iterações, apesar de uma oscilação inicial do Melhor Indivíduo, o EA fica rapidamente restrito a um ótimo local como pode ser visto na forma contínua que o gráfico assume no decorrer do processo.

Outro fator interessante é o fato de que, sem a presença do elitismo, o *fitness* do melhor indivíduo cai em determinado momento já que os melhores indivíduos das gerações não são salvos e, conseqüentemente podem ser perdidos provocando a oscilação vista no início do processo.

8.10 Cotas - 100% de Mutação

Tabela 8-9 - Cotas - 100% de Mutação

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção de Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
24	4	3, 3, 3, 3	2	3, 3	100	0%	0%	100%	1	100%	Roleta	50%	2	1	10%	100000

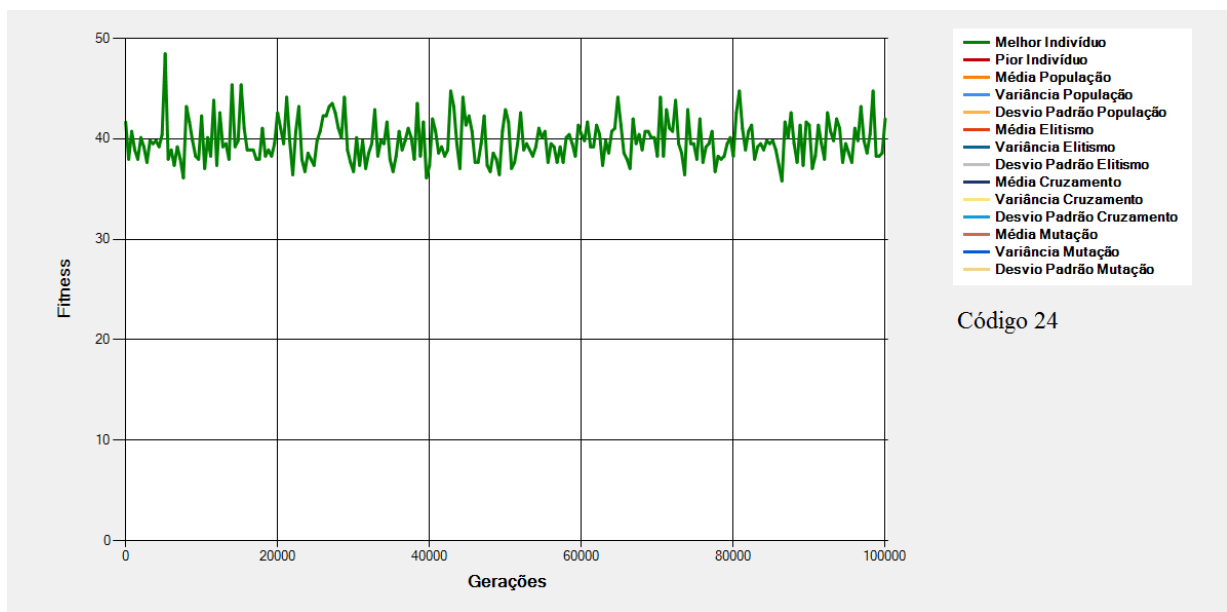


Figura 8-19 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-9 - Cotas (Código 24)

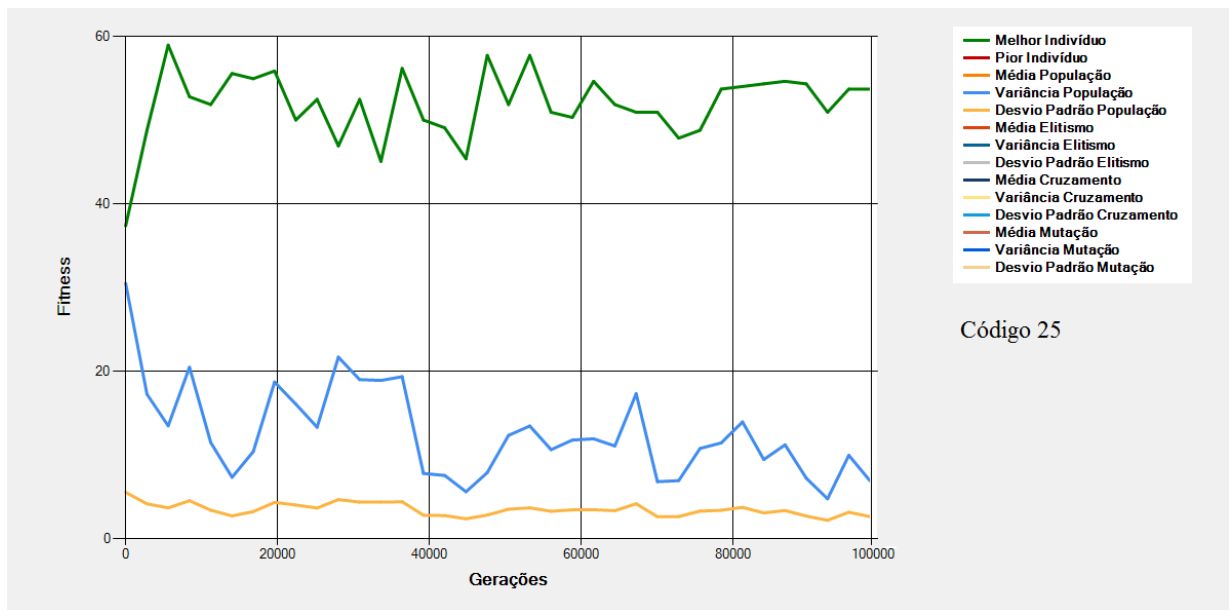
São mostrados na Figura 8-19 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-9. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Elitismo” na “Cota de Cruzamento” e na “Cota de Mutação”.

Nessa execução, percebe-se que, com 100% de Cota de Mutação, não há uma convergência do EA uma vez que todos os indivíduos terão seus genes sucessivamente alterados em as gerações e, por melhor que seja uma solução encontrada, ele é perdida durante a aplicação da mutação. Esse fato provoca uma oscilação do gráfico do melhor individuo em todo o processo o que faz com que o EA não atinja o ótimo global.

8.11 Cotas – Elitismo e Cruzamento

Tabela 8-10 - Cotas - Elitismo e Cruzamento

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutações	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
25	4	3, 3, 3, 3	2	3, 3	100	0%	90%	10%	1	100%	Roleta	50%	2	1	10%	100000
26	4	3, 3, 3, 3	2	3, 3	100	40%	50%	10%	1	100%	Roleta	50%	2	1	10%	100000
27	4	3, 3, 3, 3	2	3, 3	100	80%	10%	10%	1	100%	Roleta	50%	2	1	10%	100000



Código 25

Figura 8-20 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 25)

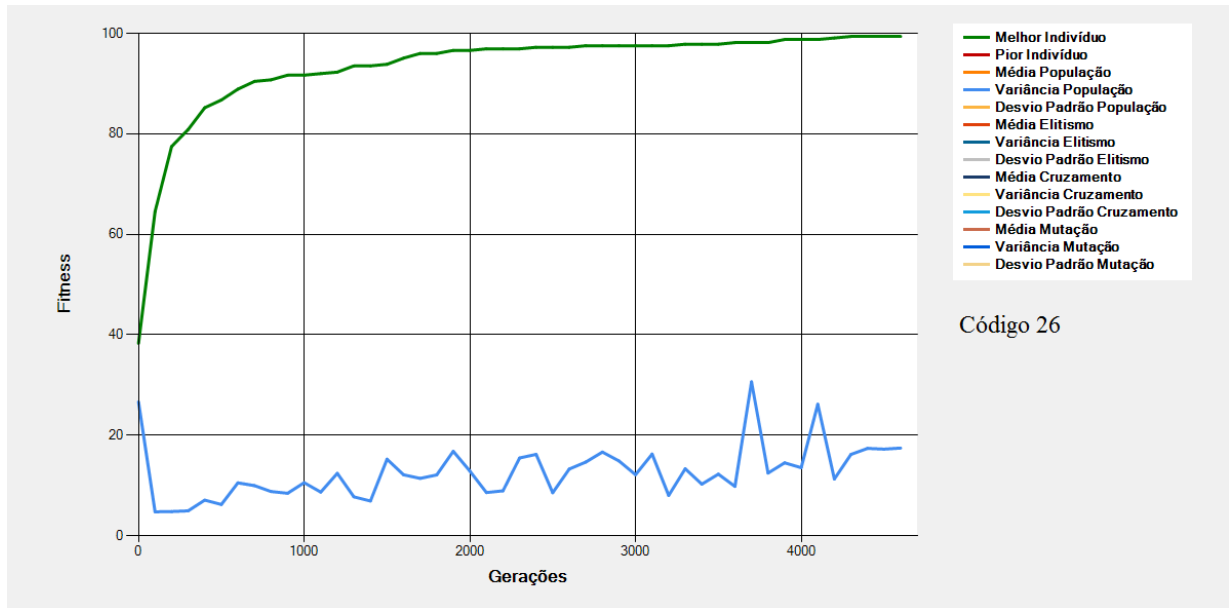


Figura 8-21 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 26)

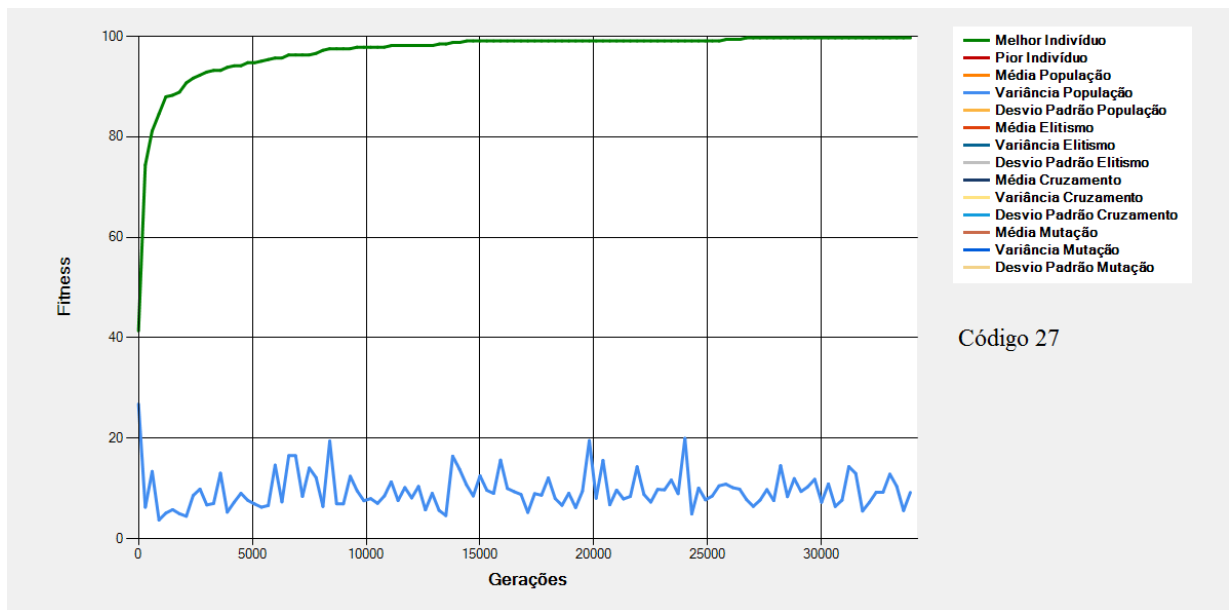


Figura 8-22 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-10 - Cotas (Código 27)

São mostrados na Figura 8-20, na Figura 8-21 e na Figura 8-22 os gráficos dos melhores indivíduos, variância da população e desvio padrão da população de acordo com os dados da Tabela 8-10. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Elitismo” e na “Cota de Cruzamento”.

Na Figura 8-20, a alta taxa de cruzamentos aliada a inexistência de elitismo, faz com que, soluções boas sejam perdidas no processo e a convergência para o ótimo global não é alcançada. Uma possível alternativa nesse caso, pode ser a divisão da população em grupos para que no processo de

seleção dos progenitores, indivíduos com *fitness* mais altos sejam majoritariamente escolhidos para realizarem os cruzamentos diminuindo as chances de se perderem bons genes durante o processo de seleção.

No gráfico da Figura 8-21 o EA atinge o ótimo global com cerca de 4,5 mil gerações. Já na Figura 8-22 a convergência se dá com cerca de 25 mil gerações o que dá mais de 20 mil gerações de diferença. Isso se dá pelo fato de que, na Figura 8-22 a configuração é de 80% de elitismo, o que resulta em uma baixa renovação dos indivíduos da população tornando a convergência mais lenta que a do gráfico da Figura 8-21.

8.12 Cotas – Elitismo e Mutação

Tabela 8-11 - Cotas - Elitismo e Mutação

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
28	4	3, 3, 3, 3	2	3, 3	100	0%	60%	40%	1	100%	Roleta	50%	2	1	10%	100000
29	4	3, 3, 3, 3	2	3, 3	100	15%	60%	25%	1	100%	Roleta	50%	2	1	10%	100000
30	4	3, 3, 3, 3	2	3, 3	100	35%	60%	5%	1	100%	Roleta	50%	2	1	10%	100000

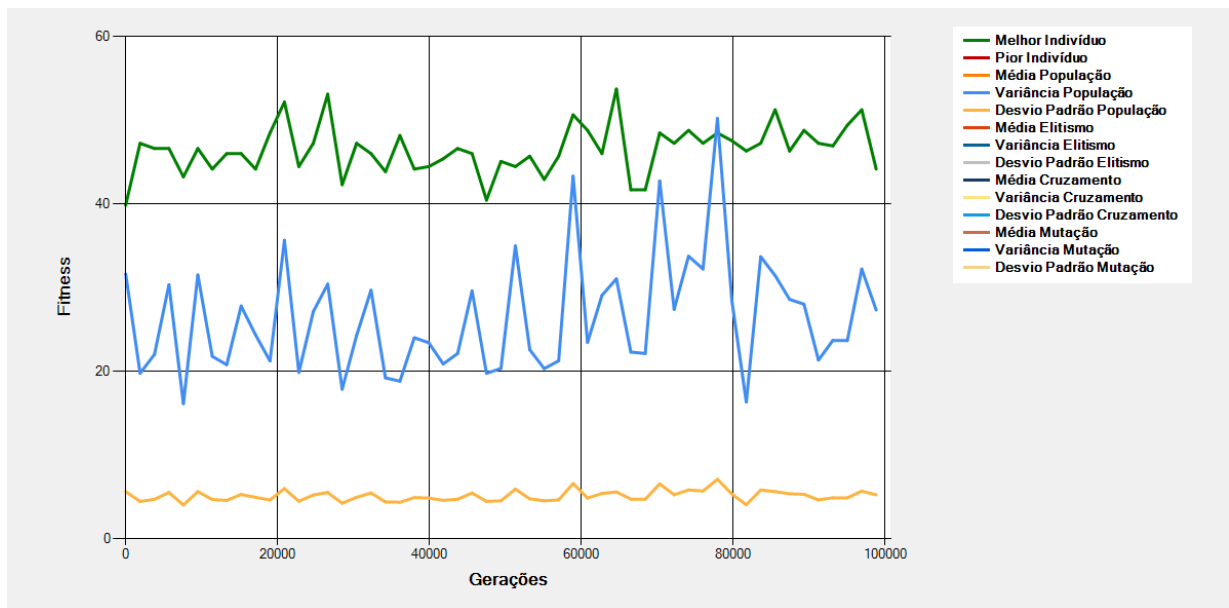


Figura 8-23 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 28)

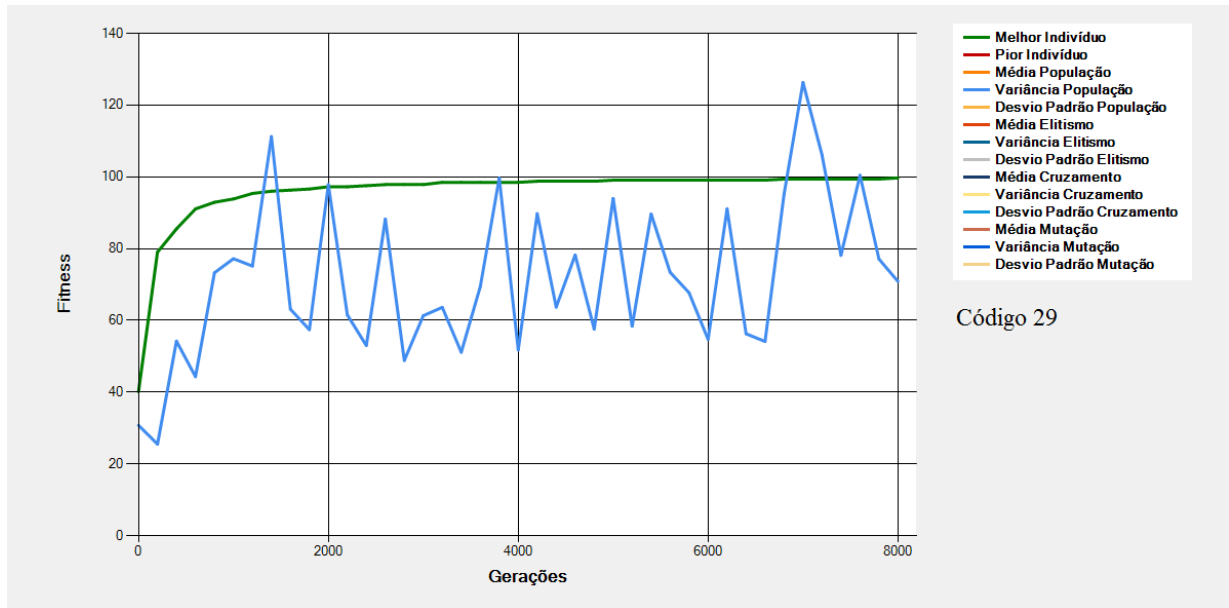


Figura 8-24 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 29)

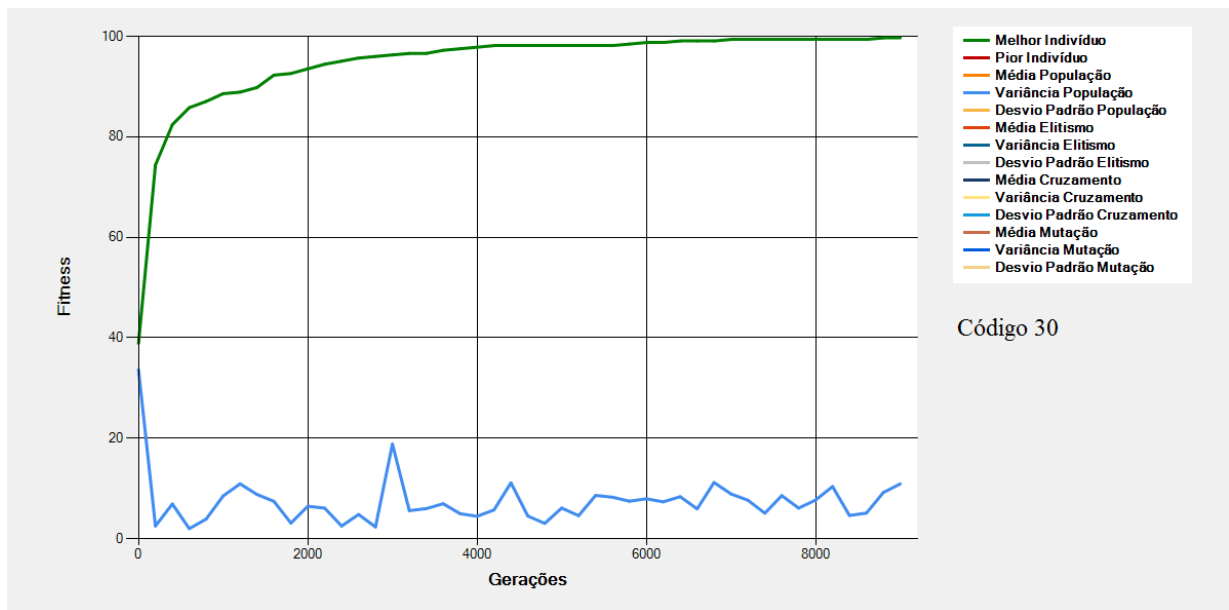


Figura 8-25 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-11 - Cotas (Código 30)

São mostrados na Figura 8-23, na Figura 8-24 e na Figura 8-25 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-11. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Elitismo” e na “Cota de Mutação”.

Assim como no caso da Figura 8-20, o exemplo da Figura 8-23 demonstra que a inexistência do elitismo faz com que soluções boas sejam perdidas durante os processos de cruzamento, acarretando na ausência de convergência para um ótimo local. Ainda em comparação com o exemplo da Figura 8-20, percebe-se na execução da Figura 8-23, uma alta grau de variância da população o

que pode ser explicado, nesse caso, com a configuração de um alto grau de Cota de Mutação (o que faz com que, a cada geração a população contenha indivíduos bem variados entre si).

Nos testes de código 29 e 30 (Figura 8-24 e Figura 8-25 respectivamente), o aumento do elitismo faz com que a convergência ocorra mais rapidamente e valores próximos do ótimo sejam alcançados com um número menor de gerações. Entretanto, a diminuição da cota de mutação, faz com que seja perceptível no gráfico, trechos onde o melhor indivíduo se mantém o mesmo por mais gerações além da baixa variância da população. Isso torna a convergência mais lenta, pois novas buscas por soluções em espaço de busca diferentes já não ocorrem com tanta frequência e, conseqüentemente, o EA demora algumas gerações a mais para chegar ao ótimo global.

8.13 Cotas - Cruzamento e Elitismo

Tabela 8-12 - Cotas - Cruzamento e Elitismo

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
31	4	3, 3, 3, 3	2	3, 3	100	90%	0%	10%	1	100%	Roleta	50%	2	1	10%	100000
32	4	3, 3, 3, 3	2	3, 3	100	50%	40%	10%	1	100%	Roleta	50%	2	1	10%	100000
33	4	3, 3, 3, 3	2	3, 3	100	10%	80%	10%	1	100%	Roleta	50%	2	1	10%	100000

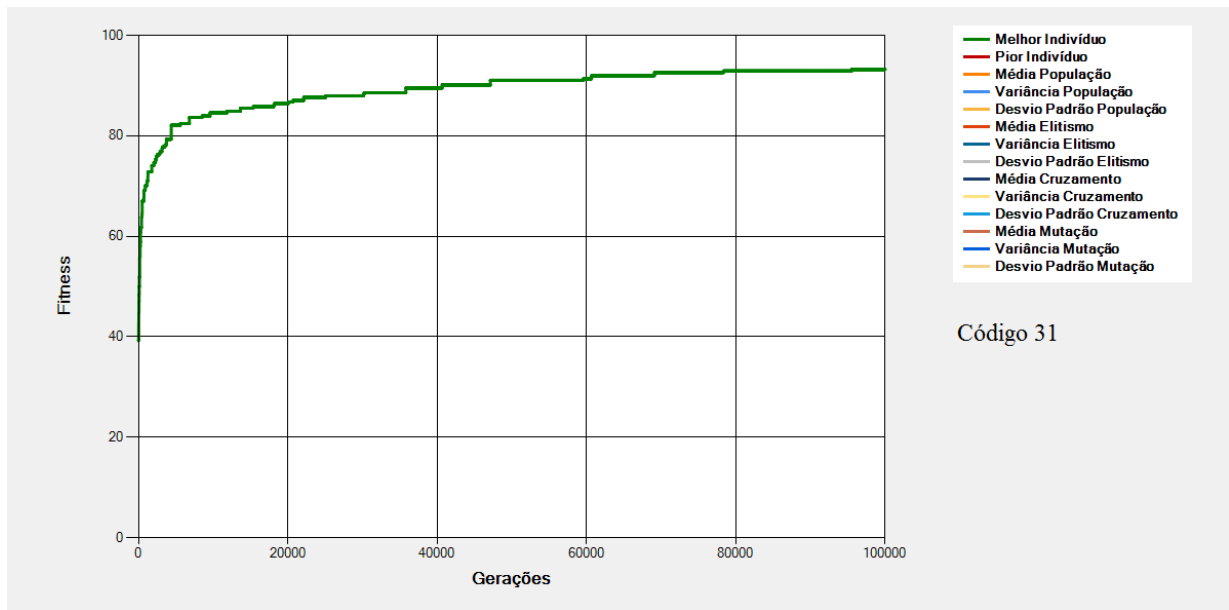


Figura 8-26 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 31)

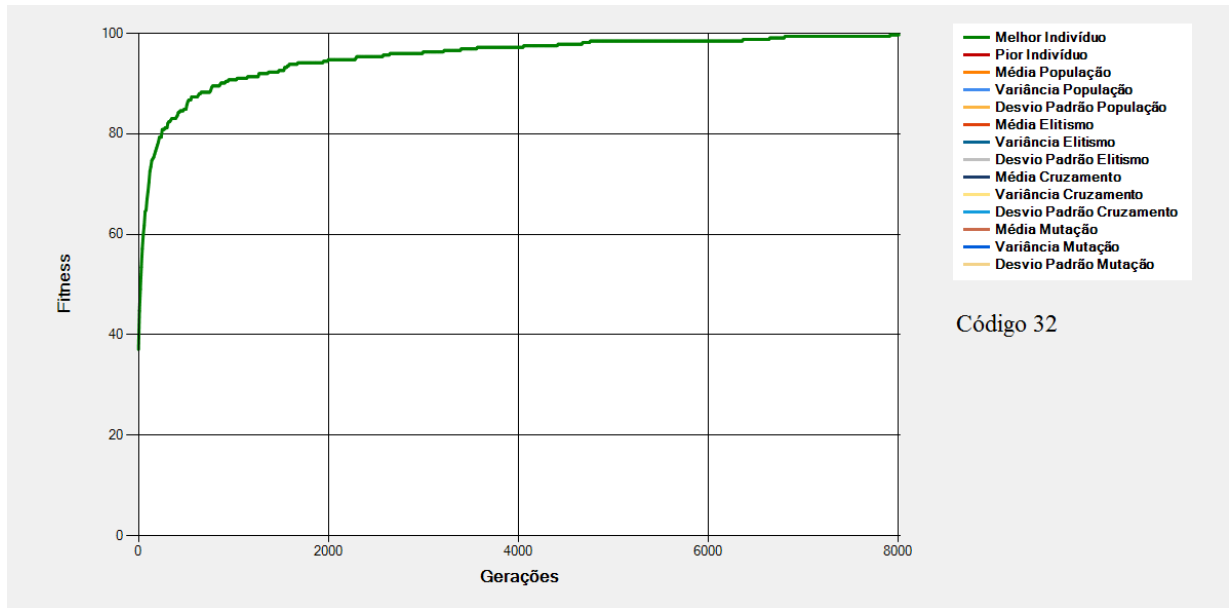


Figura 8-27 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 32)

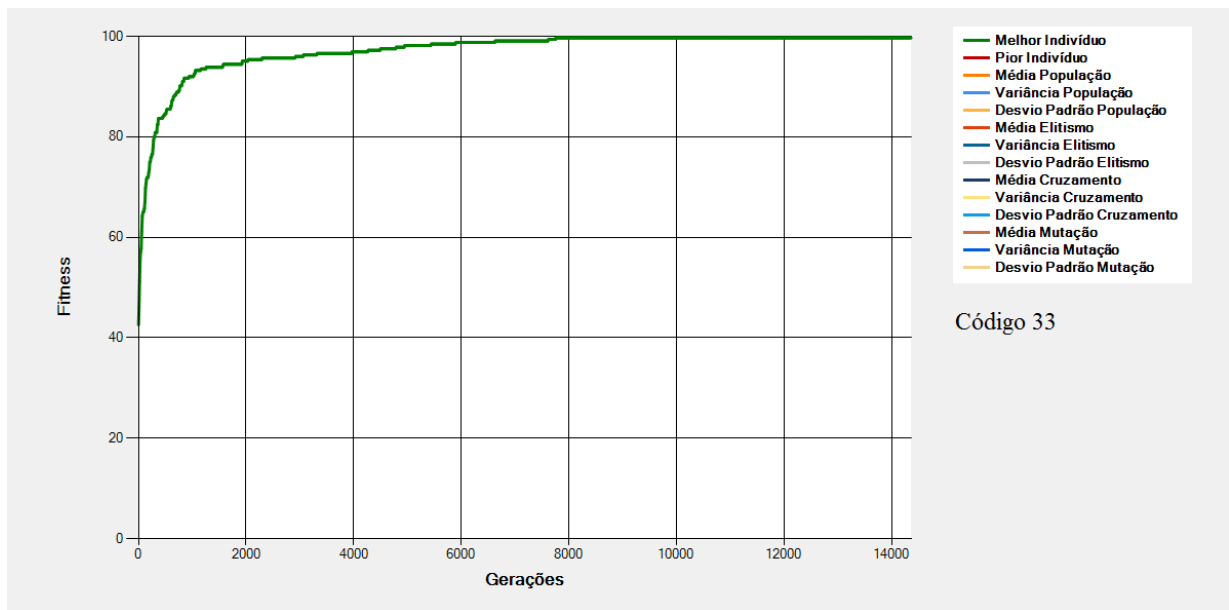


Figura 8-28 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-12 - Cotas (Código 33)

São mostrados na Figura 8-26, na Figura 8-27 e na Figura 8-28 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-12. Nessas configurações tem-se alterações (em relação à Execução Padrão) nas “Cota de Cruzamento” e na “Cota de Elitismo”.

Na Figura 8-26, a ausência de cruzamentos, faz com que o operador de mutação seja o responsável por buscar soluções dentro do espaço de busca. Neste caso, como a taxa de elitismo é de

90% e a mutação ocorre de forma aleatória, a convergência se dá muito lentamente (ausência de diversidade entre uma geração e outra) e 100mil gerações não foram suficientes para que se chegasse ao ótimo.

Na Figura 8-27 e na Figura 8-28 a redução do elitismo e o aumento do cruzamento impactam diretamente no número de gerações para a convergência do EA. No caso da Figura 8-28 por exemplo, a taxa de cruzamentos em 80% faz com que bons genes sejam perdidos acarretando na incapacidade do EA em encontrar soluções ainda melhores, o que pode ser percebido na forma linear exibida no gráfico entre as gerações 8mil e 14mil.

8.14 Cotas - Cruzamento e Mutação

Tabela 8-13 - Cotas - Cruzamento e Mutação

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
34	4	3, 3, 3, 3	2	3, 3	100	30%	0%	70%	1	100%	Roleta	50%	2	1	10%	100000
35	4	3, 3, 3, 3	2	3, 3	100	30%	30%	40%	1	100%	Roleta	50%	2	1	10%	100000
36	4	3, 3, 3, 3	2	3, 3	100	30%	50%	20%	1	100%	Roleta	50%	2	1	10%	100000

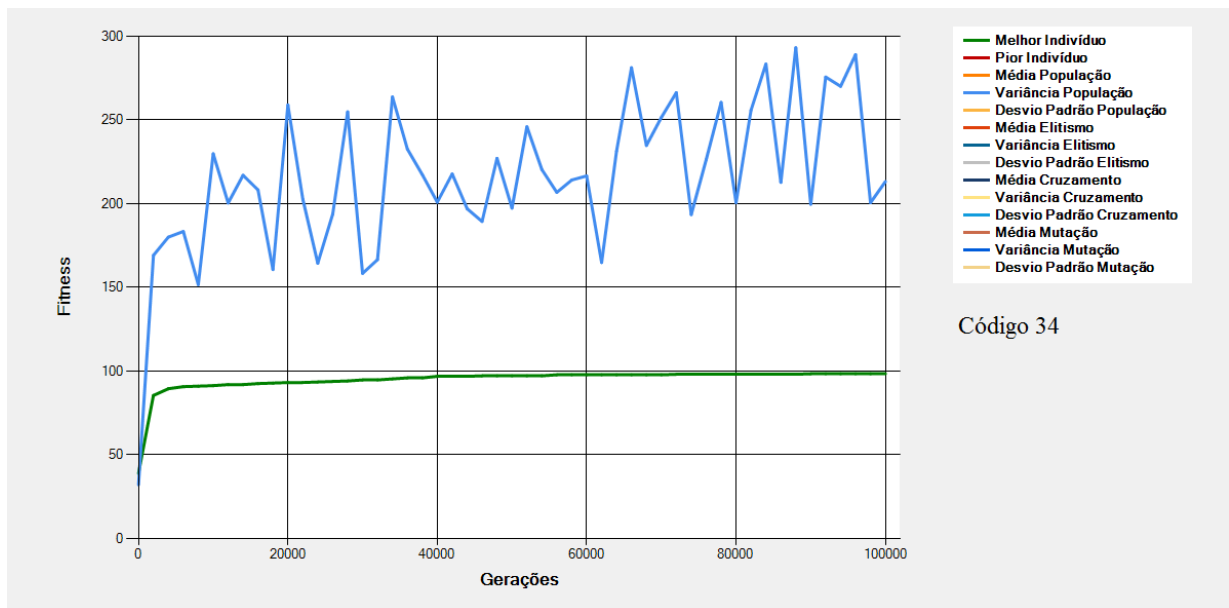


Figura 8-29 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 34)

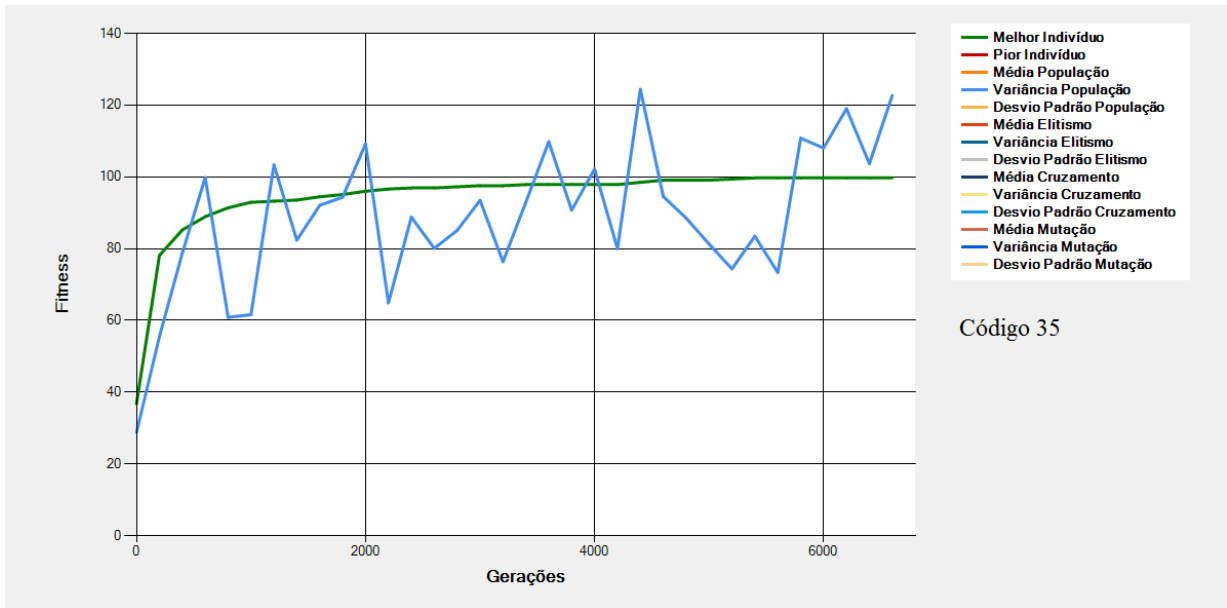


Figura 8-30 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 35)

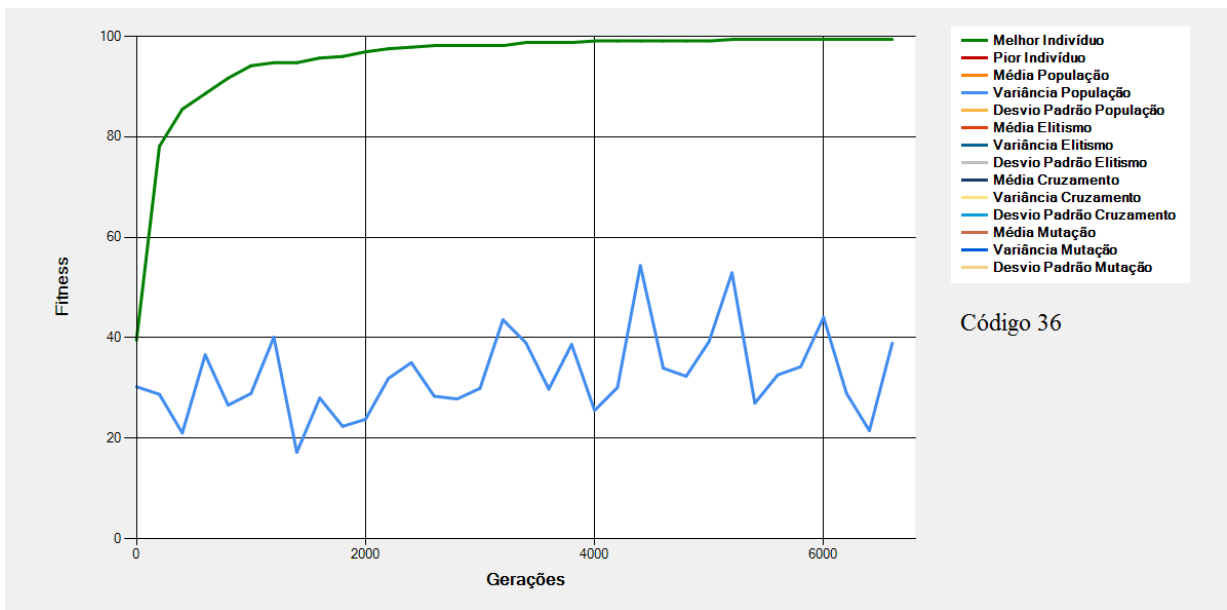


Figura 8-31 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-13 - Cotas (Código 36)

São mostrados na Figura 8-29, na Figura 8-30 e na Figura 8-31 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-13. Nessas configurações tem-se alterações (em relação à Execução Padrão) nas “Cota de Cruzamento” e na “Cota de Mutação”.

Na configuração da Figura 8-29, a ausência de cruzamentos provoca uma diminuição na velocidade de convergência do algoritmo fazendo com que 100mil gerações não sejam suficientes para que o ótimo global seja alcançado.

Na Figura 8-30 com a diminuição da cota de mutação e a presença de cruzamento, a convergência do EA se deu de forma mais acentuada e cerca de 7 mil gerações já foram suficientes para se alcançar o ótimo.

Em comparação, a Figura 8-30 e a Figura 8-31, apesar de apresentarem gráficos do melhor indivíduo semelhantes, apresentam gráficos de variância da população bem distintos. Enquanto na Figura 8-30 a variância da população é bastante alta, na Figura 8-31 a variância da população é mais baixa. Isso ocorre pelo fato de que a cota de mutação para a configuração da Figura 8-30 ser mais alta acarretando numa diversidade maior da população e, conseqüentemente, num maior valor da variância dos indivíduos.

8.15 Cotas - Mutação e Elitismo

Tabela 8-14 - Cotas - Mutação e Elitismo

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutação	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
38	4	3, 3, 3, 3	2	3, 3	100	40%	60%	0%	1	100%	Roleta	50%	2	1	10%	100000
39	4	3, 3, 3, 3	2	3, 3	100	25%	60%	15%	1	100%	Roleta	50%	2	1	10%	100000
40	4	3, 3, 3, 3	2	3, 3	100	5%	60%	35%	1	100%	Roleta	50%	2	1	10%	100000

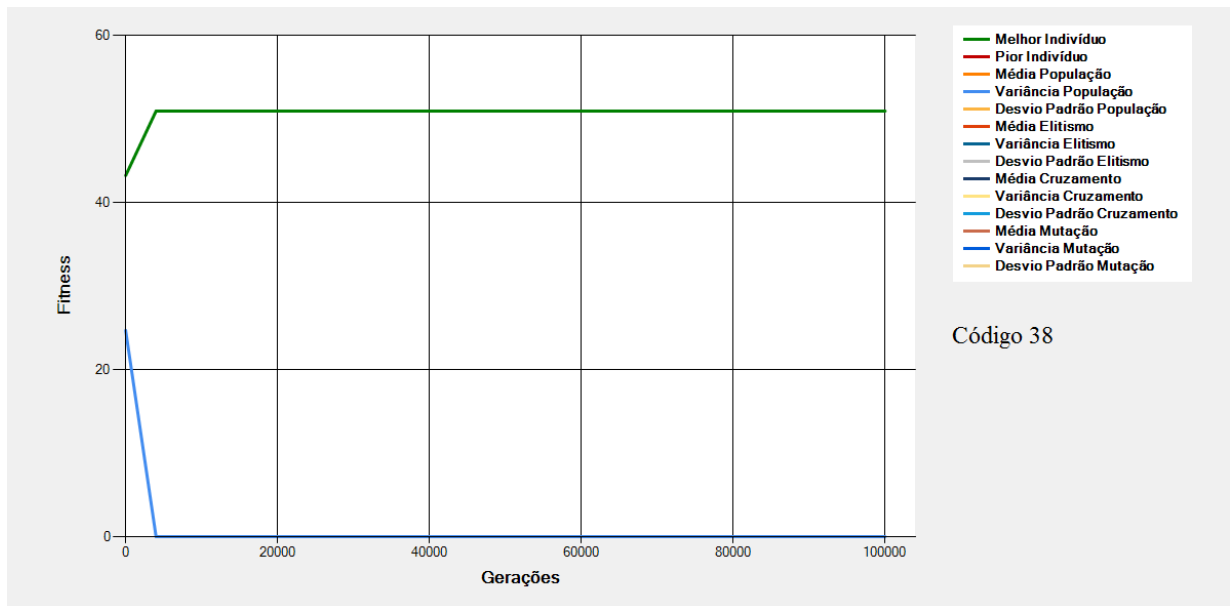


Figura 8-32 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 38)

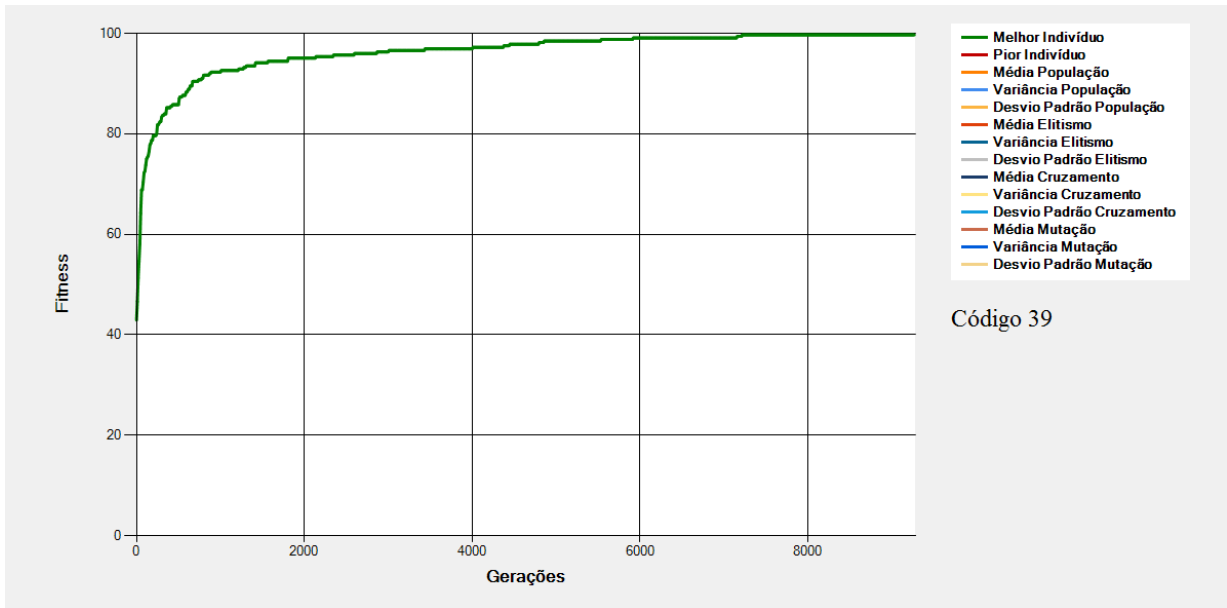


Figura 8-33 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 39)

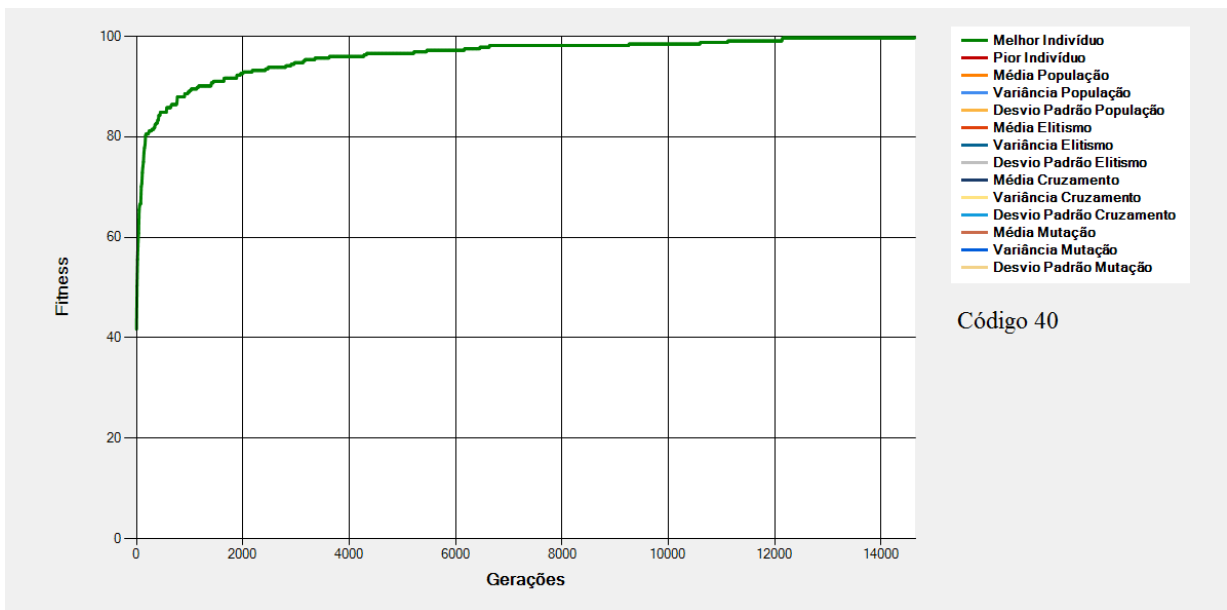


Figura 8-34 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-14 - Cotas (Código 40)

São mostrados na Figura 8-32, na Figura 8-33 e na Figura 8-34 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-14. Nessas configurações tem-se alterações (em relação à Execução Padrão) nas “Cota de Mutação” e na “Cota de Elitismo”.

Na Figura 8-32, a ausência de mutação faz com que haja uma diminuição gradativa na diversidade da população provocando um estacionamento, logo no início da execução, em algum ponto do espaço de soluções. O gráfico demonstra que, em poucas gerações a população já está

completamente homogênea (variância da população diminui até chegar a zero) e, não é possível, sem um operador que provoque divergência populacional, buscar novas soluções no espaço de busca.

Na Figura 8-33 e na Figura 8-34 a diminuição do elitismo e o aumento da mutação impactam no número de gerações necessárias para alcançar o ótimo. No caso da configuração da Figura 8-34 com a cota de elitismo mais baixa e a cota de mutação mais alta menos indivíduos bons são salvos e a convergência se dá mais lentamente, fazendo com que mais gerações sejam necessárias para se chegar ao ótimo (cerca de 15 mil da configuração de código 40 – Figura 8-34 – contra cerca de 9 mil da configuração de código 39 – Figura 8-33)

8.16 Cotas - Mutação e Cruzamento

Tabela 8-15 - Cotas - Mutação e Cruzamento

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutação	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
41	4	3, 3, 3, 3	2	3, 3	100	30%	70%	0%	1	100%	Roleta	50%	2	1	10%	100000
42	4	3, 3, 3, 3	2	3, 3	100	30%	40%	30%	1	100%	Roleta	50%	2	1	10%	100000
43	4	3, 3, 3, 3	2	3, 3	100	30%	20%	50%	1	100%	Roleta	50%	2	1	10%	100000

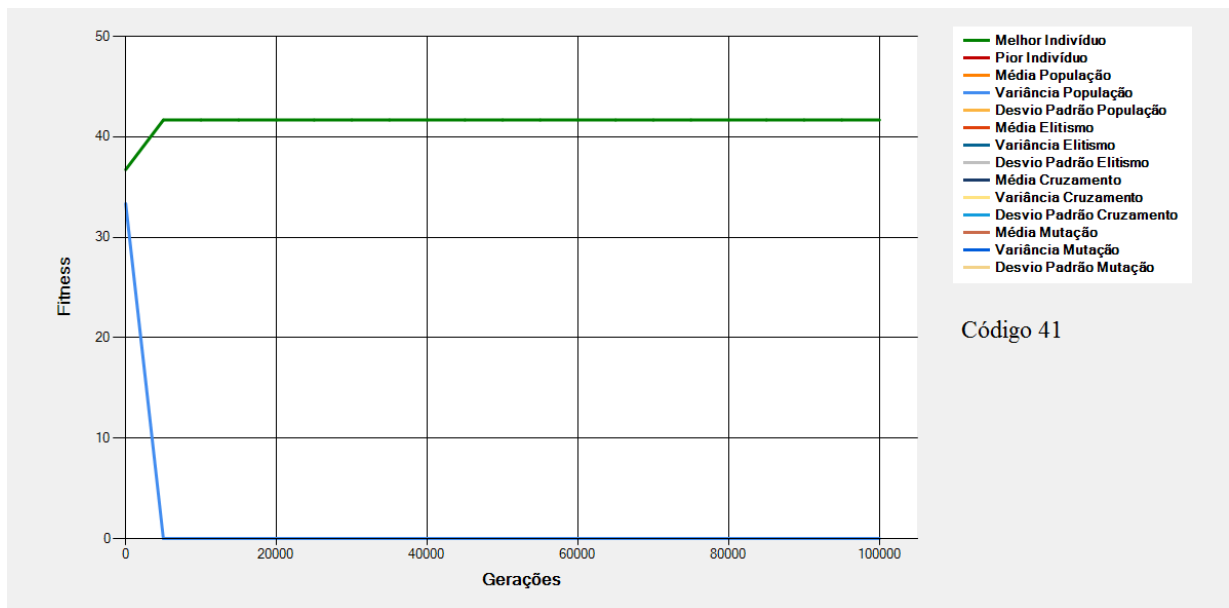


Figura 8-35 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 41)

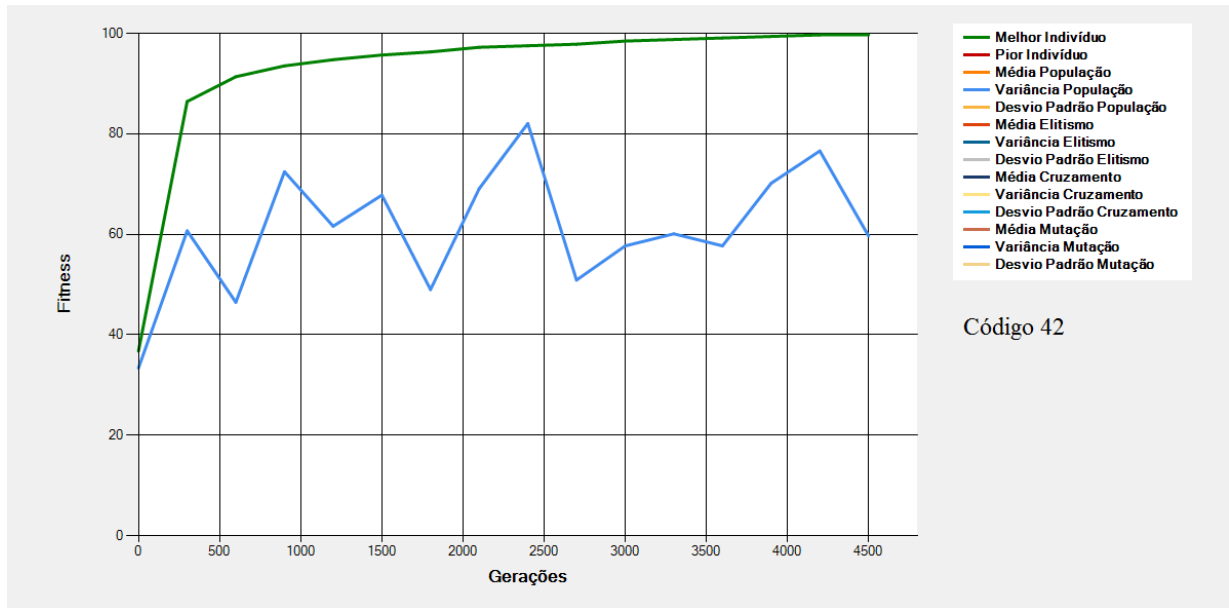


Figura 8-36 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 42)

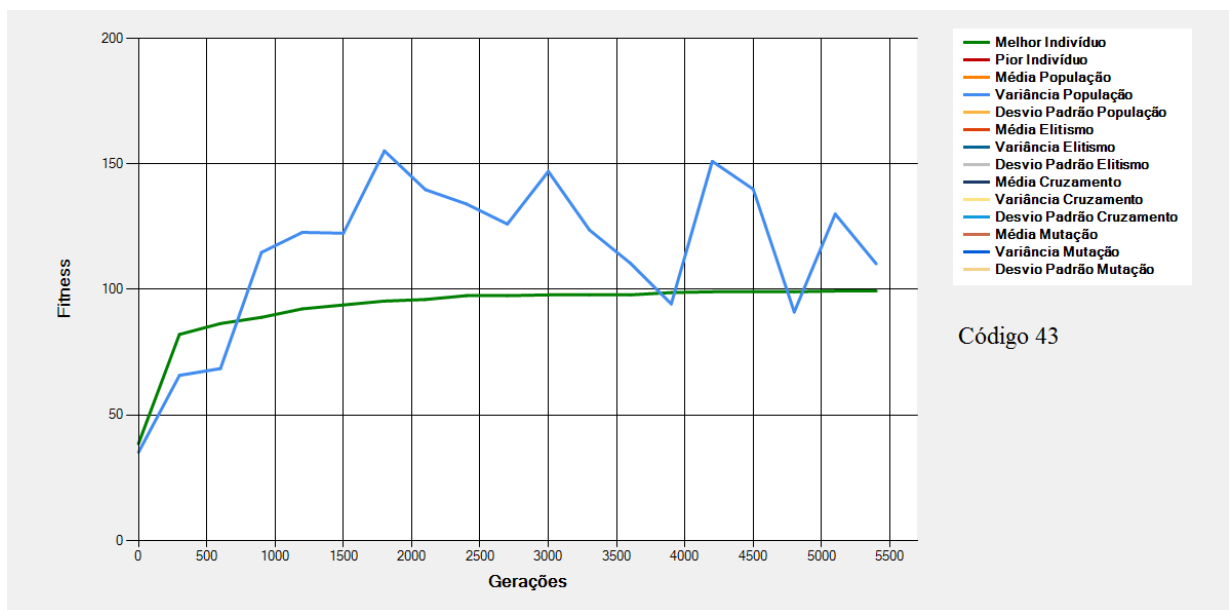


Figura 8-37 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-15 - Cotas (Código 43)

São mostrados na Figura 8-35, na Figura 8-36 e na Figura 8-37 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-15. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Cota de Mutação” e na “Cota de Cruzamento”.

No exemplo da Figura 8-35, ocorre o mesmo que o exemplo da Figura 8-32. Com o passar das gerações a população vai ficando cada vez mais homogênea impossibilitando o EA encontrar novas soluções o que faz com que a execução, logo no início, estacione em um ponto do espaço de soluções.

Na Figura 8-36 e na Figura 8-37 tem-se as execuções das configurações de código 42 e 43. Nelas, percebe-se que com o aumento da cota de mutação e a diminuição da cota de cruzamentos, aumenta-se a diversidade da população acarretando em um aumento no número de gerações necessárias para se alcançar o ótimo. Entretanto, essa diversidade diminui os riscos do EA ser conduzido a convergências para ótimos locais uma vez que, com essa diversidade, diferentes locais do espaço de busca estão sendo explorados a fim de se alcançar o ótimo global.

8.17 Grupos de Progenitores - 2 Grupos

Tabela 8-16 - Grupos de Progenitores - 2 Grupos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
44	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	2	70%, 30%	Roleta	50%	2	1	10%	100000
45	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	2	30%, 70%	Roleta	50%	2	1	10%	100000

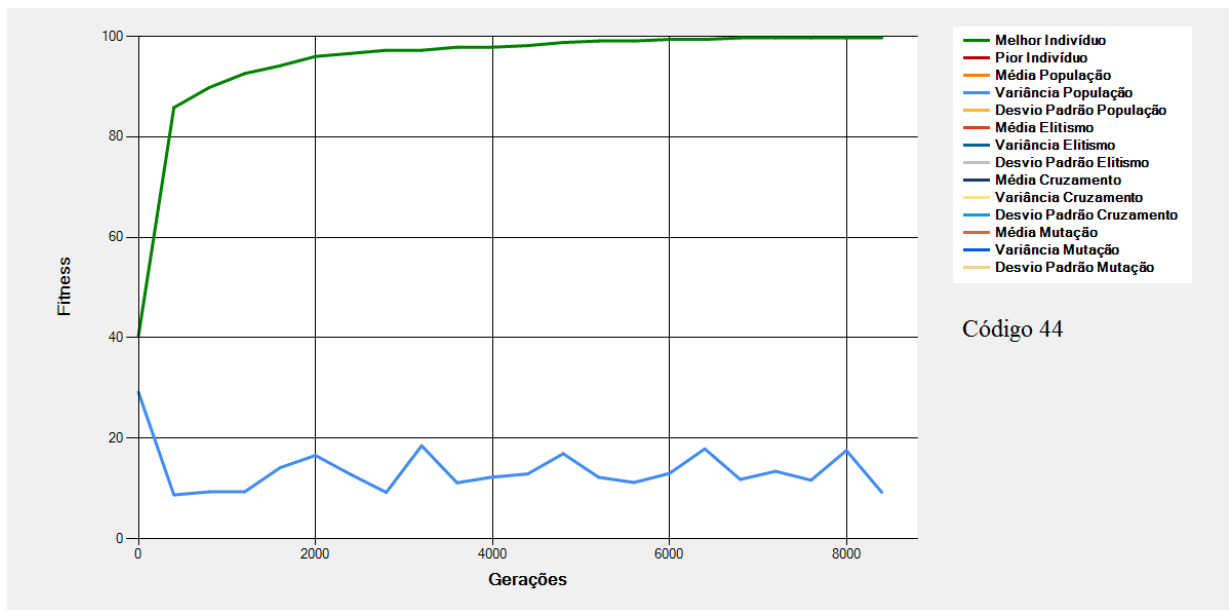


Figura 8-38 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-16 - Grupos de Progenitores (Código 44)

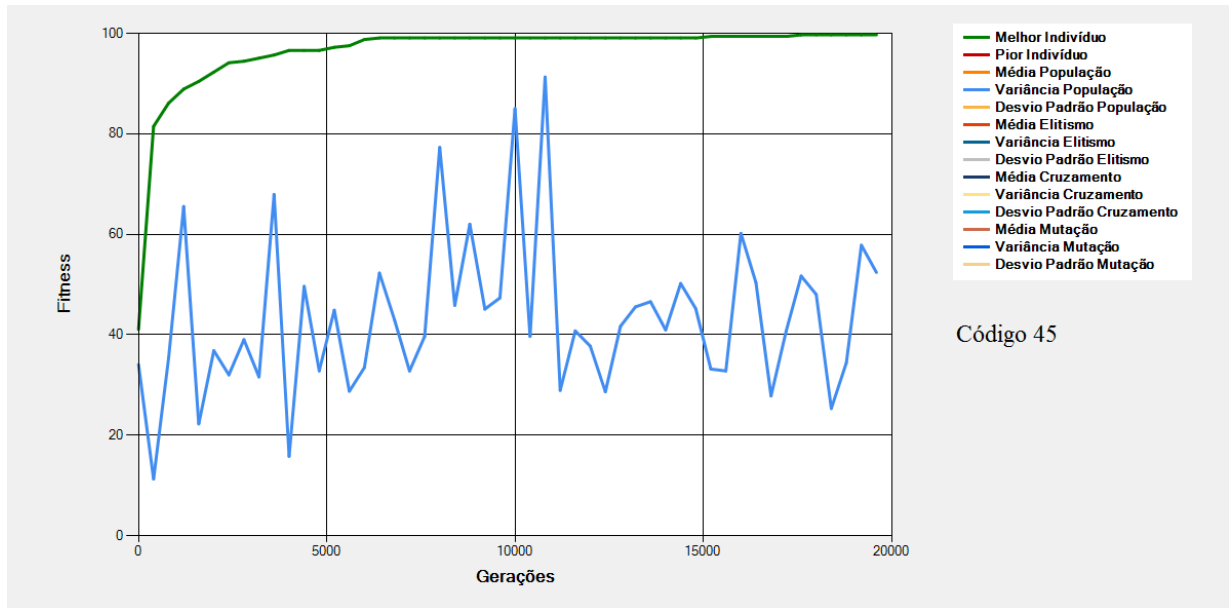


Figura 8-39 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-16 - Grupos de Progenitores (Código 45)

São mostrados na Figura 8-38 e na Figura 8-39 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-16. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Grupo de Progenitores” e na “Proporção dos Grupos de Progenitores”.

Como explicando no tópico 4.2.5- Operadores Evolutivos de Elitismo, Seleção, Cruzamento e Mutação (página 91), a população pode ser dividida em grupos para auxiliar o processo de seleção dos progenitores dos indivíduos da próxima população.

Nas configurações de código 48 e 49 a população foi dividida em 2 grupos - Melhores e Piores. Na configuração de código 48 dos 50% selecionados para serem os progenitores da próxima população (taxa de seleção), 70% eram oriundos do grupo dos Melhores e 30% do grupo dos piores. Essa divisão faz com que, nos cruzamentos, uma menor quantidade de genes bons sejam perdidos e os cruzamentos sejam direcionados para aumentar a velocidade da convergência do EA. Na Figura 8-38 (Código 48 – 70% Melhores e 30% Piores) a convergência se deu com pouco mais de 8 mil gerações. Já na Figura 8-39 (Código 49 – 30% Melhores e 70% Piores) o ótimo foi alcançado com quase 20 mil gerações.

8.18 Grupos de Progenitores - 3 Grupos

Tabela 8-17 - Grupos de Progenitores - 3 Grupos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
46	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	3	60%, 20%, 20%	Roleta	50%	2	1	10%	100000
47	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	3	20%, 60%, 20%	Roleta	50%	2	1	10%	100000
48	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	3	20%, 20%, 60%	Roleta	50%	2	1	10%	100000

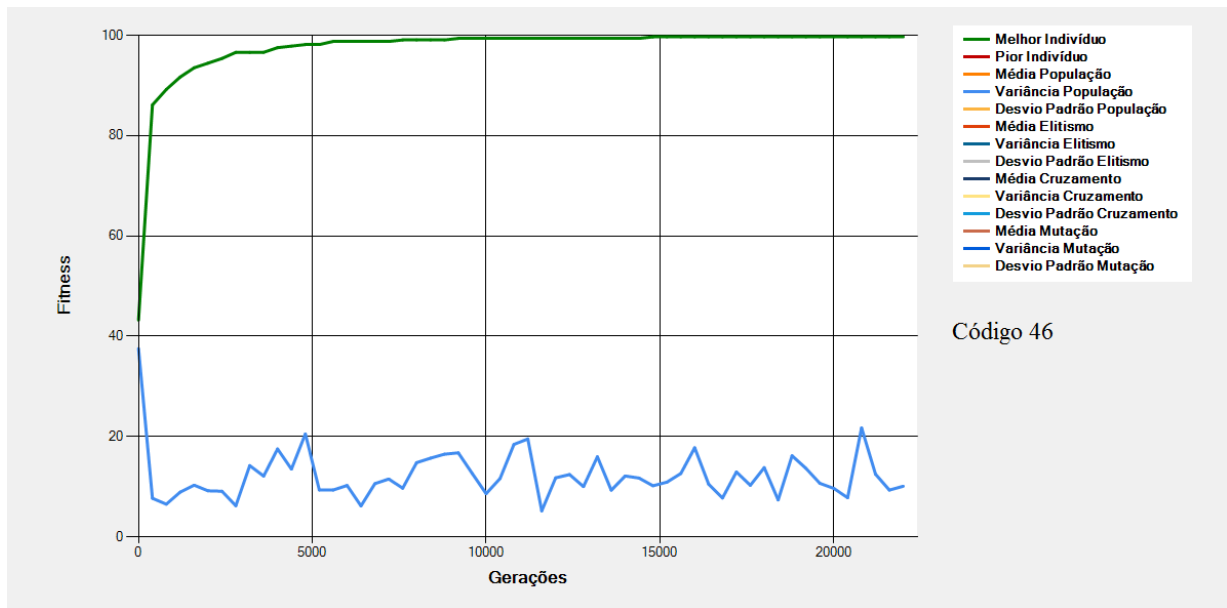


Figura 8-40 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 46)

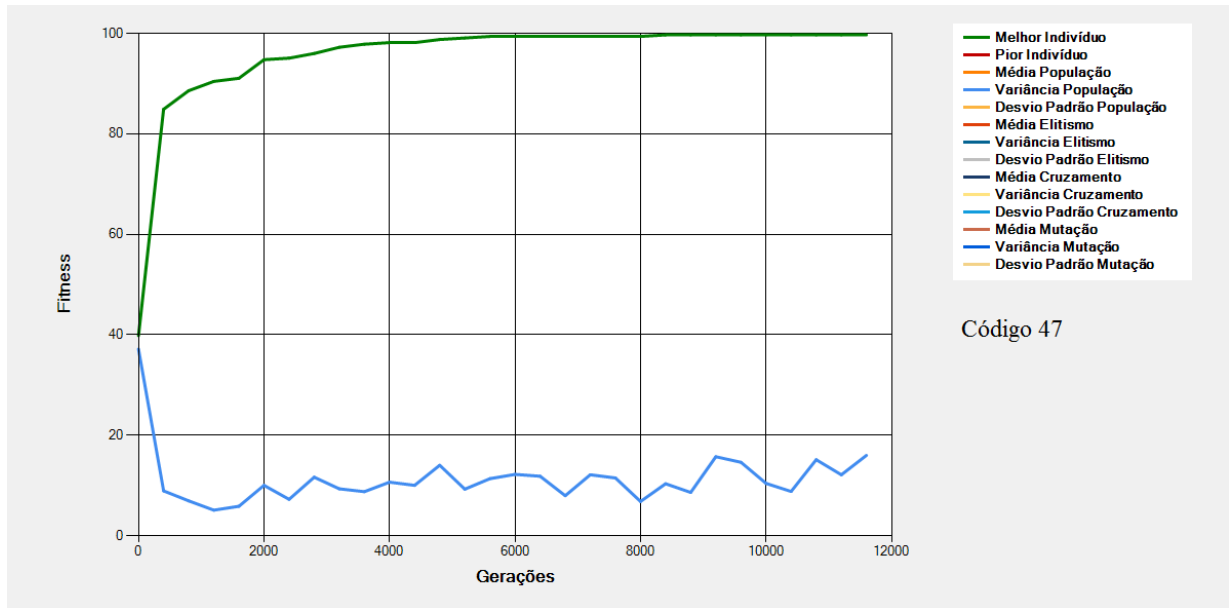


Figura 8-41 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 47)

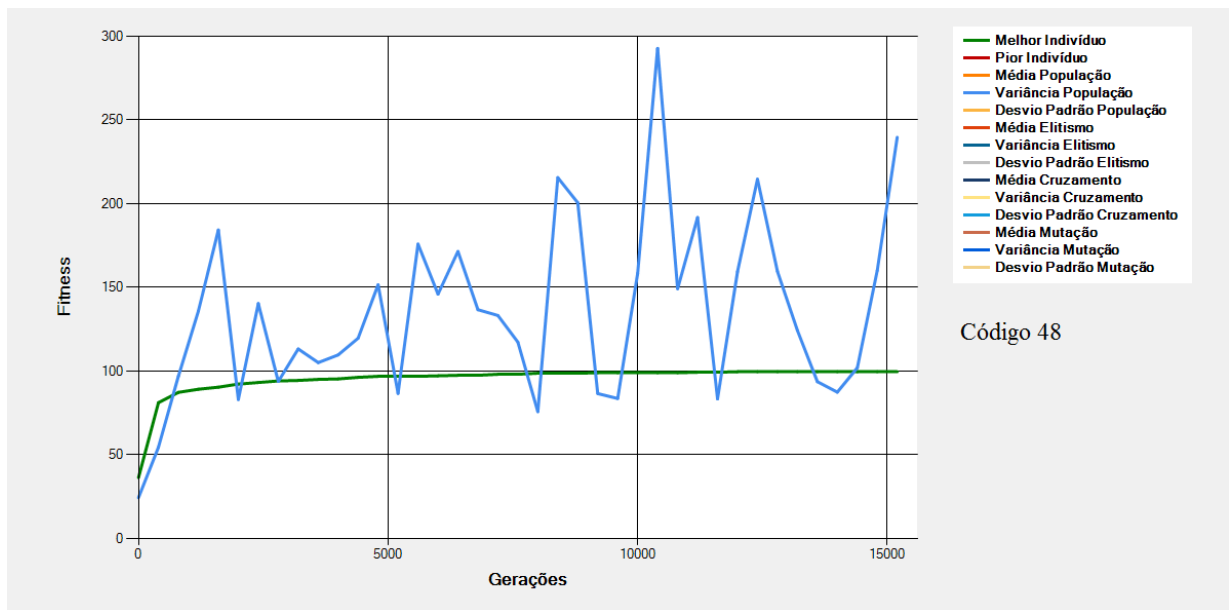


Figura 8-42 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-17 - Grupos de Progenitores (Código 48)

São mostrados na Figura 8-40, na Figura 8-41 e na Figura 8-42 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-17. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Grupo de Progenitores” e na “Proporção dos Grupos de Progenitores”.

Como explicando no tópico 4.2.5 - Operadores Evolutivos de Elitismo, Seleção, Cruzamento e Mutação (página 91), a população pode ser dividida em grupos para auxiliar o processo de seleção dos progenitores dos indivíduos da próxima população.

Nos exemplos acima as populações foram divididas em 3 grupos – Melhores, Intermediários e Piores. Percebe-se que, na execução da Figura 8-40, com uma divisão com porcentagem maior no grupo dos melhores (60% - 20% - 20%), tem-se uma convergência inicial levemente mais rápida que a execução da Figura 8-41, a qual privilegia os intermediários (20% - 60% - 20%). Entretanto, nesses casos, alcançar o ótimo se torna um processo mais demorado devido a uma menor diversidade da população o que acarreta numa baixa procura pontos diferentes dos atuais no espaço de soluções.

No teste de código 48 (Figura 8-42) percebe-se uma alta variância da população provocada pelo grande número de indivíduos de *fitness* baixo nos processos de seleção e cruzamento – Piores (20% - 20% - 60%).

8.19 Grupos de Progenitores - 4 Grupos

Tabela 8-18 - Grupos de Progenitores - 4 Grupos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutaçào	Execuçào
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutaçào	Quantidade de Grupos de Progenitores	Proporçào dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
49	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	4	70%, 10%, 10%, 10%	Roleta	50%	2	1	10%	100000
50	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	4	10%, 70%, 10%, 10%	Roleta	50%	2	1	10%	100000
51	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	4	10%, 10%, 70%, 10%	Roleta	50%	2	1	10%	100000
52	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	4	10%, 10%, 10%, 70%	Roleta	50%	2	1	10%	100000

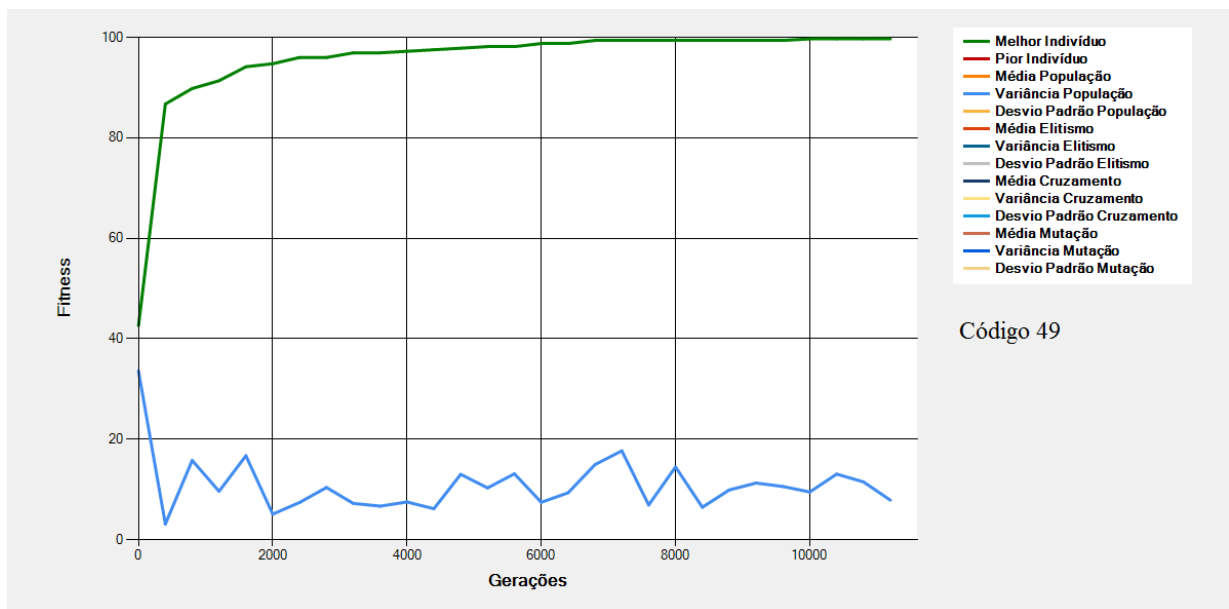


Figura 8-43 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 49)

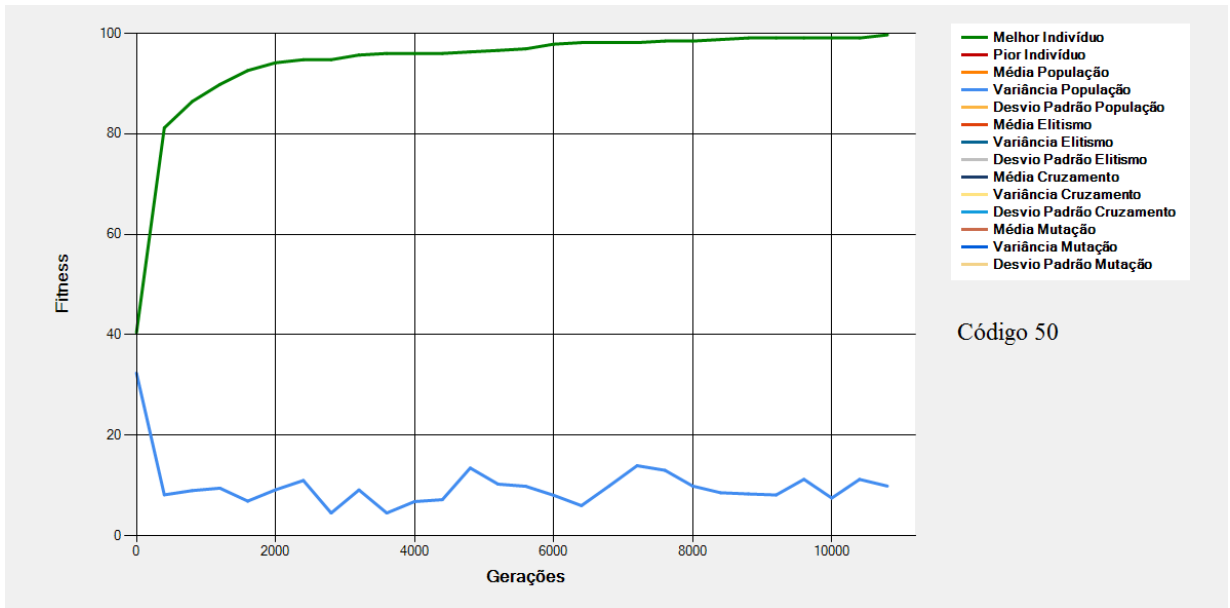


Figura 8-44 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 50)

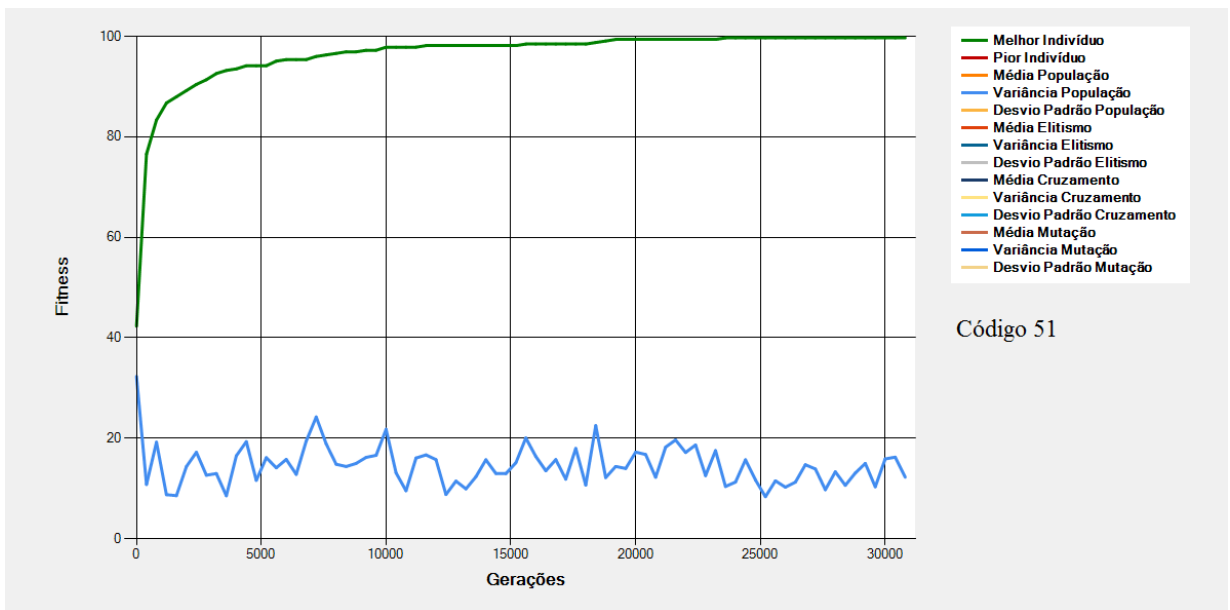


Figura 8-45 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 51)

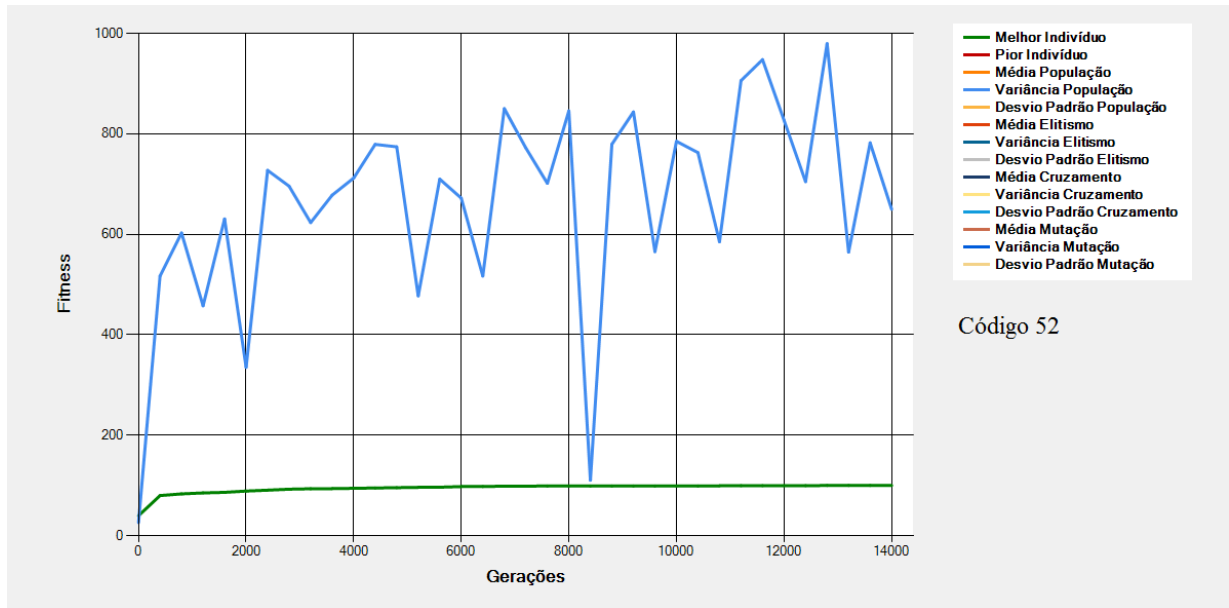


Figura 8-46 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-18 - Grupos de Progenitores (Código 52)

São mostrados na Figura 8-43, na Figura 8-44, na Figura 8-45 e na Figura 8-46 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-18. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Grupo de Progenitores” e na “Proporção dos Grupos de Progenitores”.

Nessas execuções a seleção foi feita baseada na divisão da população em 4 grupos – Melhores, Bons, Ruins e Piores. Nas execuções acima percebeu-se, na medida em que as proporções dos grupos foram sendo alteradas, modificações em relação ao número de gerações necessárias para a convergência e variância da população.

Na configuração de código de código 52 (Figura 8-46) houve uma alta variância da população ocasionada pela alta quantidade de indivíduos de *fitness* baixo nos cruzamentos, o que faz com que a diversidade da população aumente evitando quedas prematuras em ótimos locais.

8.20 Grupos de Progenitores - 5 Grupos

Tabela 8-19 - Tabela referente aos Grupos de Progenitores - 5 Grupos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
53	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	5	60%, 10%, 10%, 10%, 10%	Roleta	50%	2	1	10%	100000
54	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	5	10%, 60%, 10%, 10%, 10%	Roleta	50%	2	1	10%	100000
55	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	5	10%, 10%, 60%, 10%, 10%	Roleta	50%	2	1	10%	100000
56	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	5	10%, 10%, 10%, 60%, 10%	Roleta	50%	2	1	10%	100000
57	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	5	10%, 10%, 10%, 10%, 60%	Roleta	50%	2	1	10%	100000

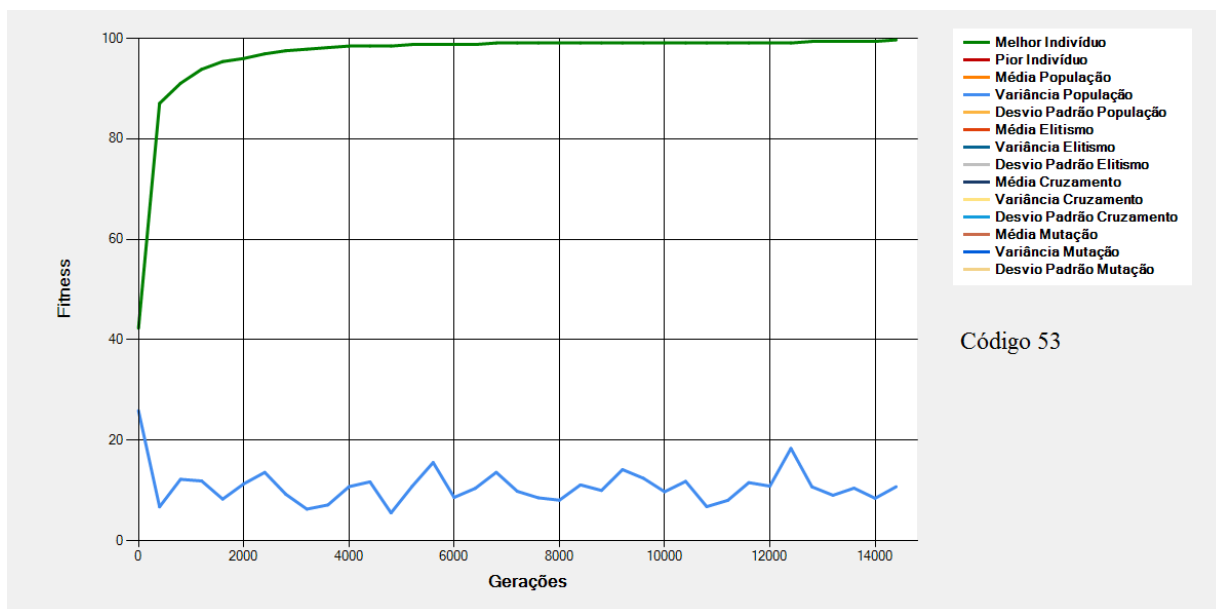
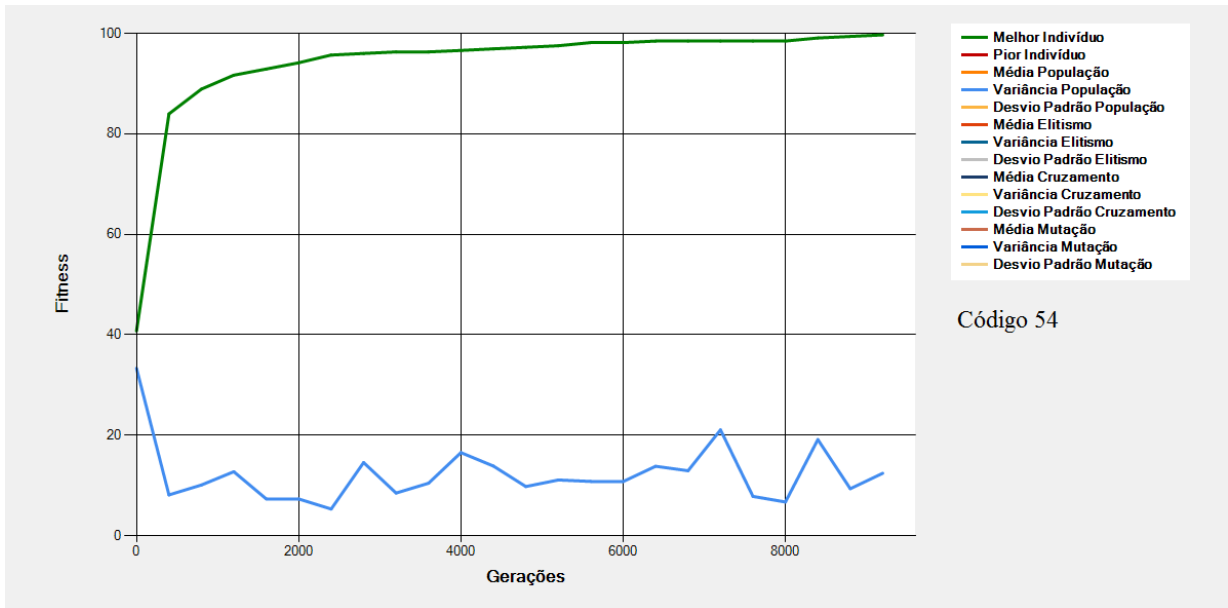
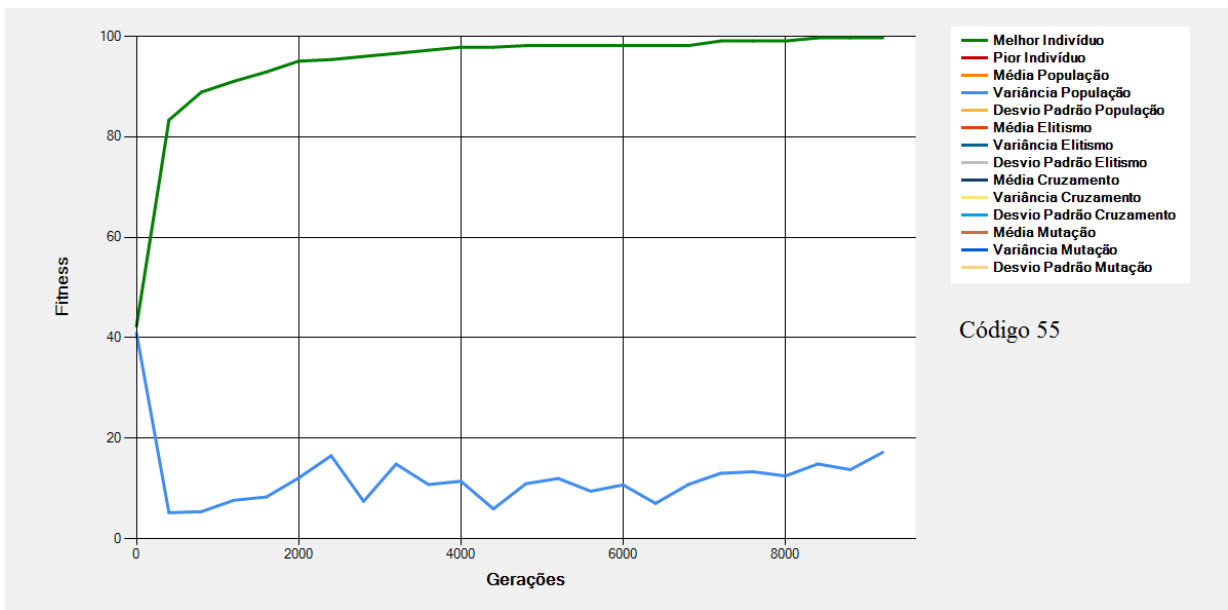


Figura 8-47 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 53)



Código 54

Figura 8-48 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 54)



Código 55

Figura 8-49 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 55)

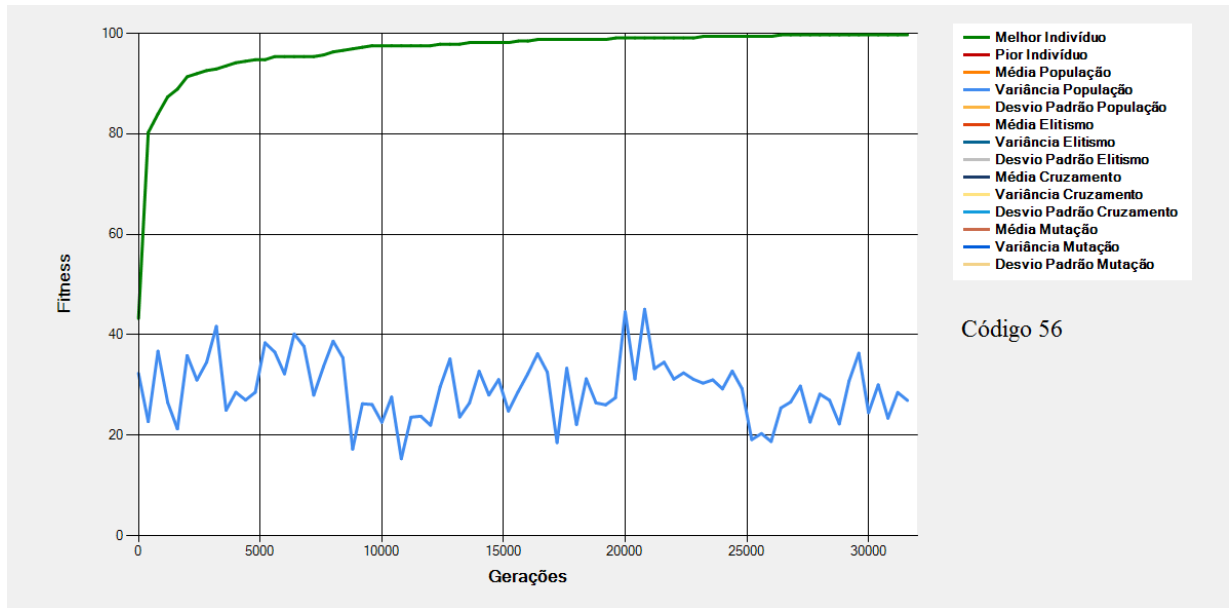


Figura 8-50 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 56)

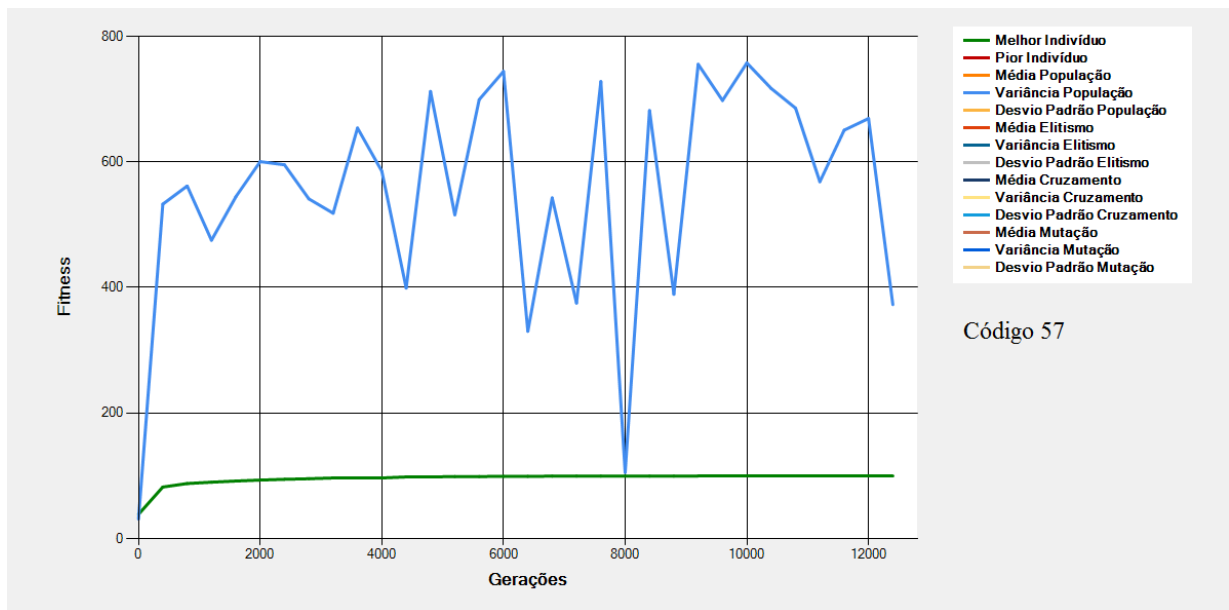


Figura 8-51 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-19 - Grupos de Progenitores (Código 57)

São mostrados na Figura 8-47, na Figura 8-48, na Figura 8-49, na Figura 8-50 e na Figura 8-51 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-19. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Grupo de Progenitores” e na “Proporção dos Grupos de Progenitores”.

Nas execuções acima, a população foi dividida em 5 grupos – Melhores, Bons, Intermediários, Ruins e Piores. A medida que a proporção foi sendo modificada para cada um dos grupos percebeu-se um aumento gradativo da variância da população o que impactou diretamente no número de

gerações necessárias para se alcançar o ótimo. As proporções maiores nos grupos dos Bons e Intermediários foram as que executaram com um menor número de gerações – próximo de 9 mil gerações.

8.21 Seleção - Métodos de Seleção

Tabela 8-20 - Seleção de Progenitores - Métodos de Seleção

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
58	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Torneio de 2	50%	2	1	10%	100000
59	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Torneio de 3	50%	2	1	10%	100000

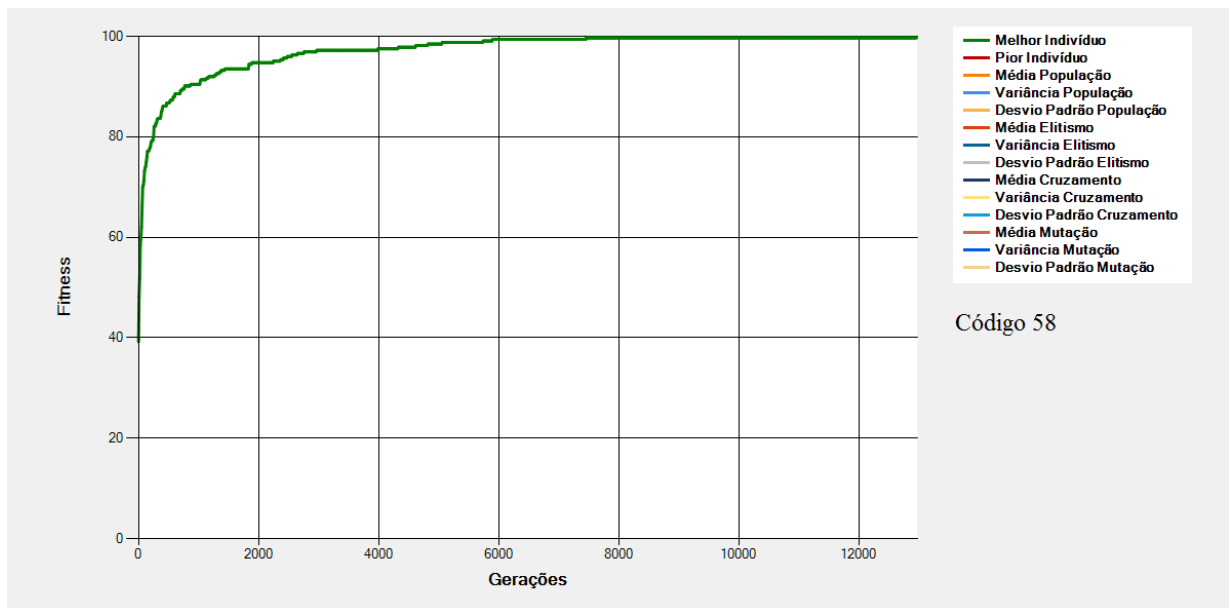


Figura 8-52 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-20 - Seleção de Progenitores (Código 58)

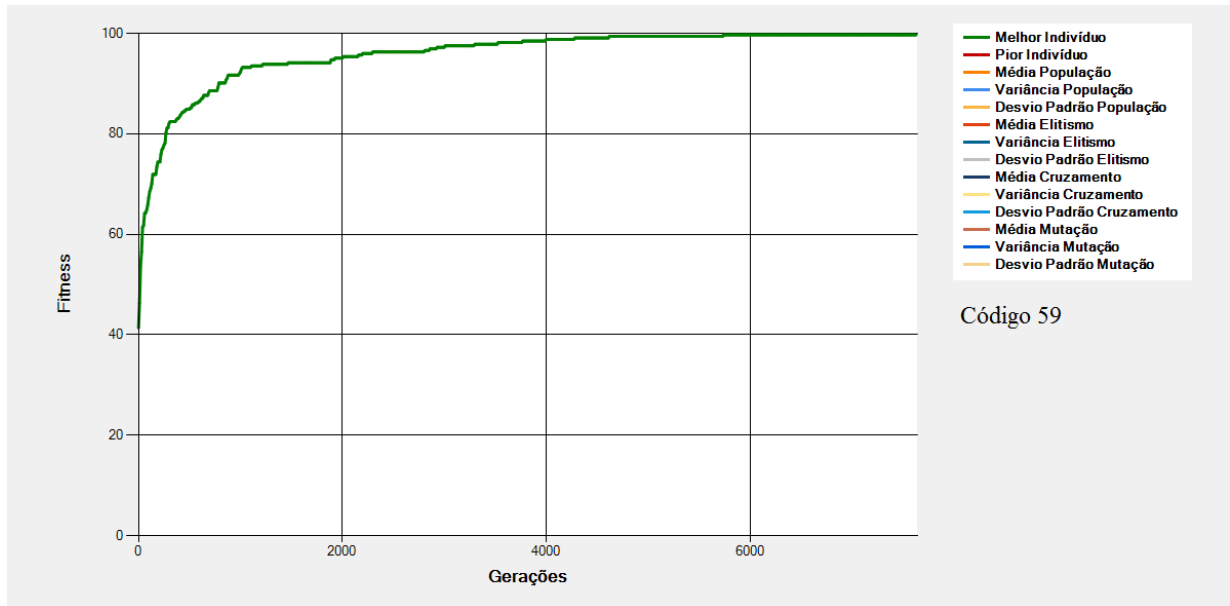


Figura 8-53 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-20 - Seleção de Progenitores (Código 59)

São mostrados na Figura 8-52 e na Figura 8-53 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-20. Nessas configurações tem-se alterações (em relação à Execução Padrão) no “Método de Seleção”.

Nas configurações acima percebe-se que com o uso do torneio de 3 menos gerações foram necessárias para se alcançar o ótimo. Isso se deve ao fato de que no torneio de 3, a probabilidade do indivíduo de menor *fitness* ser escolhido cai (2.4.5 - Operador Evolutivo de Seleção – página 50) e mais indivíduos bons são selecionados no processo aumentando a velocidade de convergência do EA.

8.22 Seleção - Taxa de Seleção

Tabela 8-21 - Seleção de Progenitores - Taxa de Seleção

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
60	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	10%	2	1	10%	100000
61	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	70%	2	1	10%	100000
62	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	100%	2	1	10%	100000

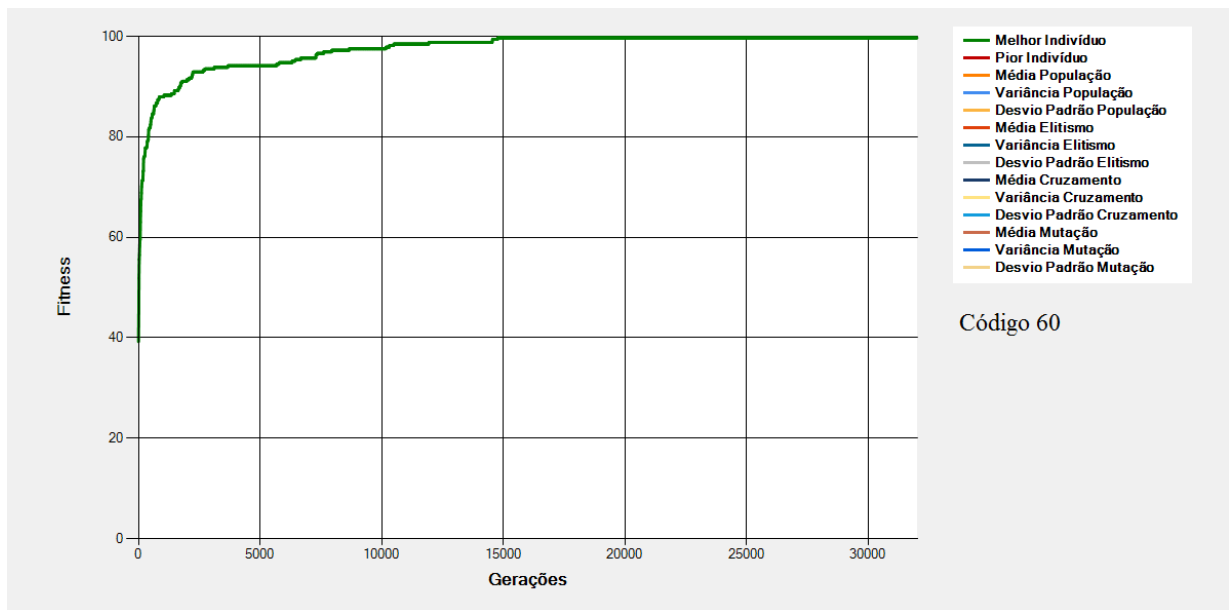


Figura 8-54 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 60)

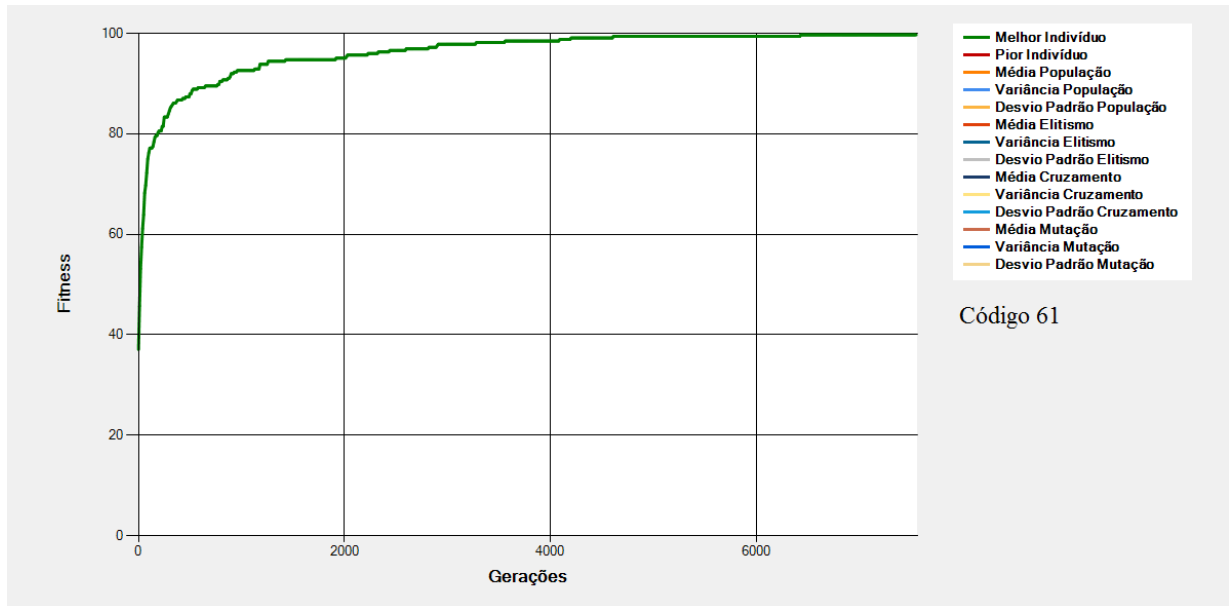


Figura 8-55 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 61)

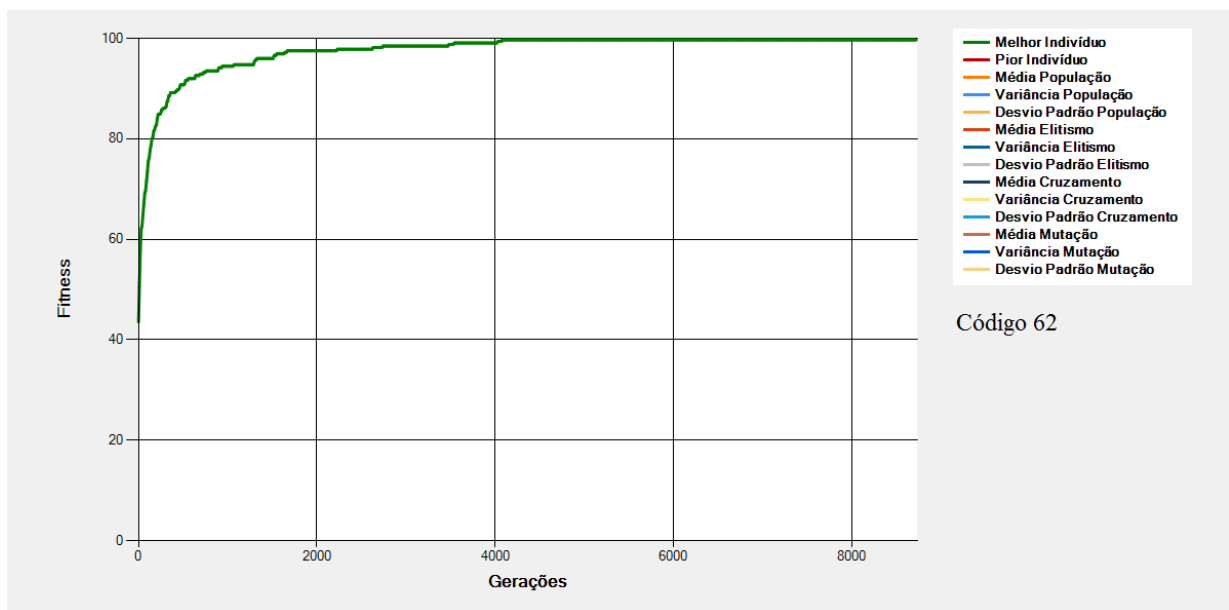


Figura 8-56 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-21 - Seleção de Progenitores (Código 62)

São mostrados na Figura 8-54, na Figura 8-55 e na Figura 8-56 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-21. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Taxa de Seleção”.

Nos exemplos acima a Taxa de Seleção vai sendo aumentada no decorrer do processo alterando com isso o número de gerações necessárias para se alcançar o ótimo global.

Na Figura 8-54, com a taxa em apenas 10%, poucos indivíduos serão os responsáveis por gerar a próxima população. Isso torna a população cada vez mais homogênea o que provoca a

diminuição da busca em outros pontos do espaço de soluções e, conseqüentemente, acarreta na diminuição da velocidade da convergência do EA.

Na Figura 8-55 é apresentado o gráfico da configuração com 70% de taxa de seleção. Essa taxa faz com que boa parte dos indivíduos sejam selecionados para serem progenitores o que provoca uma diversidade maior na população e conseqüentemente, aumenta a velocidade de convergência do EA.

Com a taxa de 100% apresentada na configuração de código 62, todos os indivíduos serão selecionados para serem progenitores o que provoca, como pode ser visto na Figura 8-56, uma diversidade muito grande da população, observada pelo aumento no número de gerações para se chegar ao ótimo. Essa diversidade muito alta faz com que genes bons sejam perdidos nos cruzamentos e isso a convergência demore um pouco mais para ser atingida.

8.23 Cruzamentos - Quantidade de Progenitores por Cruzamentos

Tabela 8-22 - Cruzamentos - Quantidade de Progenitores por Cruzamentos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
63	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	3	1	10%	100000
64	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	4	1	10%	100000

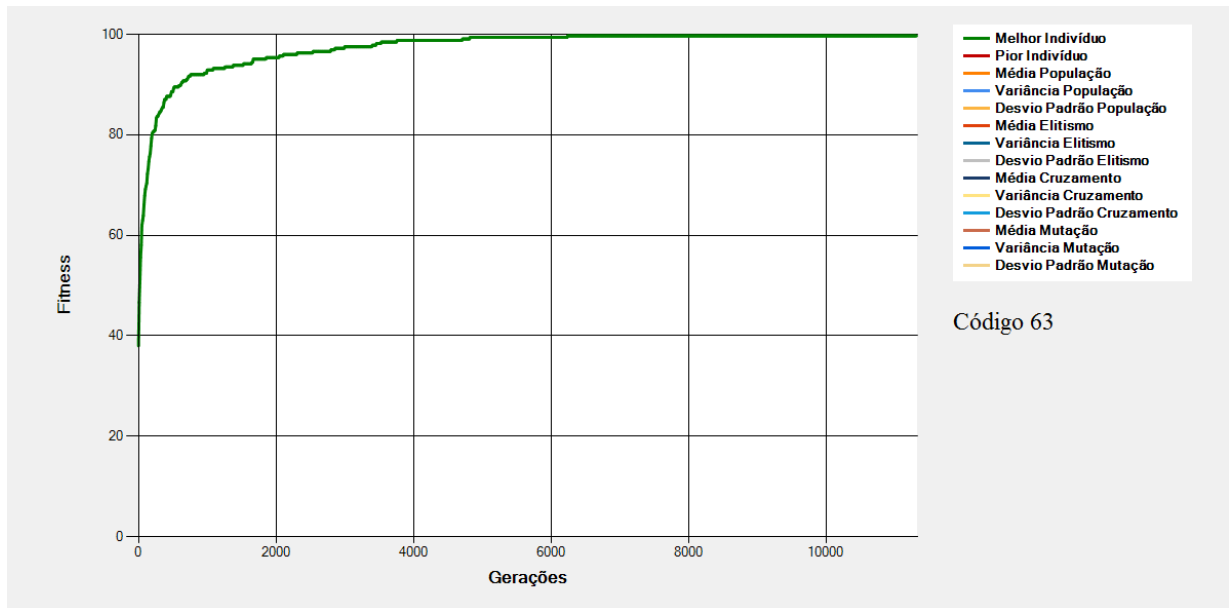


Figura 8-57 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-22 - Cruzamentos (Código 63)

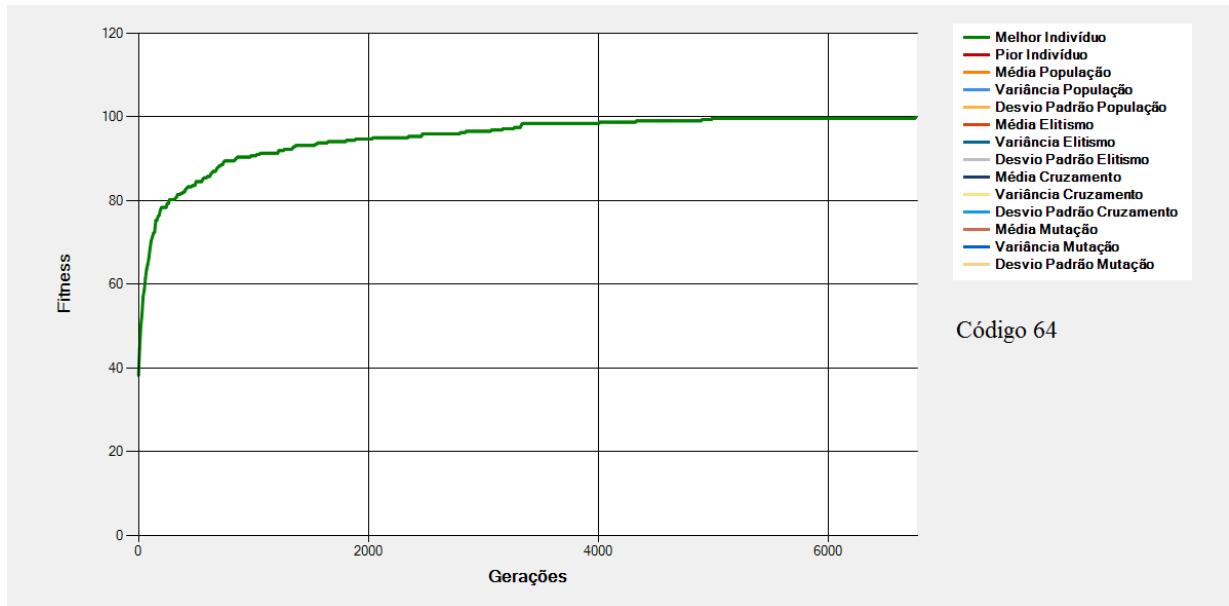


Figura 8-58 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-22 - Cruzamentos (Código 64)

São mostrados na Figura 8-57 e na Figura 8-58 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-22. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Progenitores por Cruzamento”.

A quantidade de progenitores por cruzamento provoca uma diversidade maior na população o que, por vezes, é benéfica ao processo de convergência do EA.

Nos exemplos acima, com mais progenitores, houve uma diversidade maior da população gerada o que fez com que a velocidade da convergência fosse mais acentuada e o ótimo fosse atingido com um número menor de gerações no caso da execução de código 64 em comparação com a execução de código 63.

8.24 Cruzamentos - Quantidade de Pontos de Corte

Tabela 8-23 - Cruzamentos - Quantidade de Pontos de Corte

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
65	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	2	10%	100000
66	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	3	10%	100000

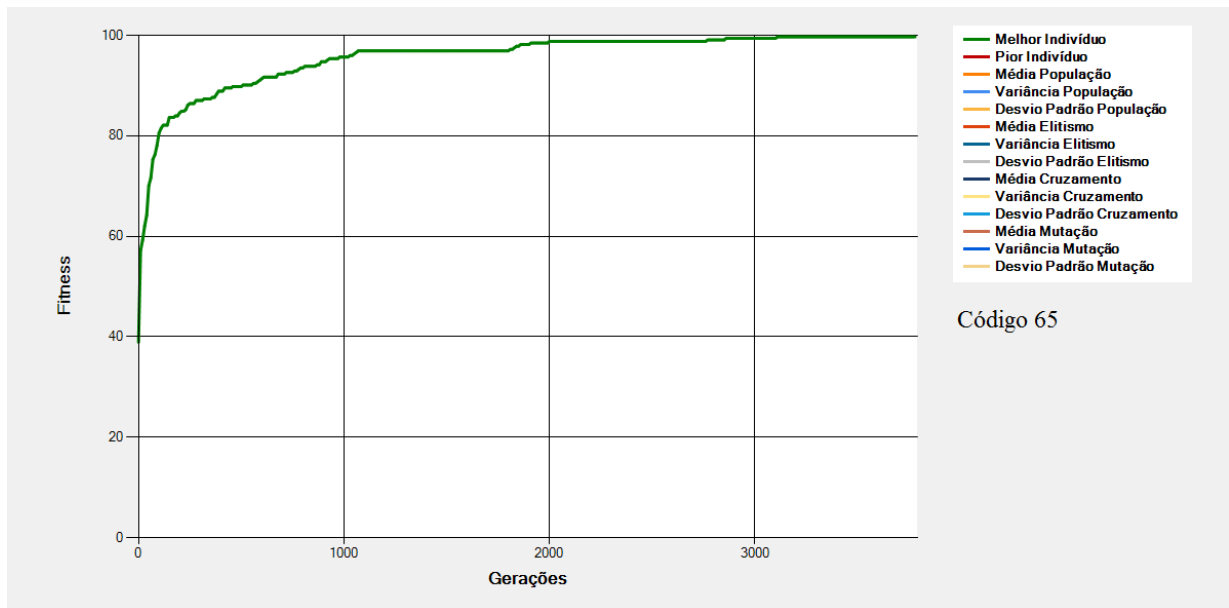


Figura 8-59 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-23 - Cruzamentos (Código 65)

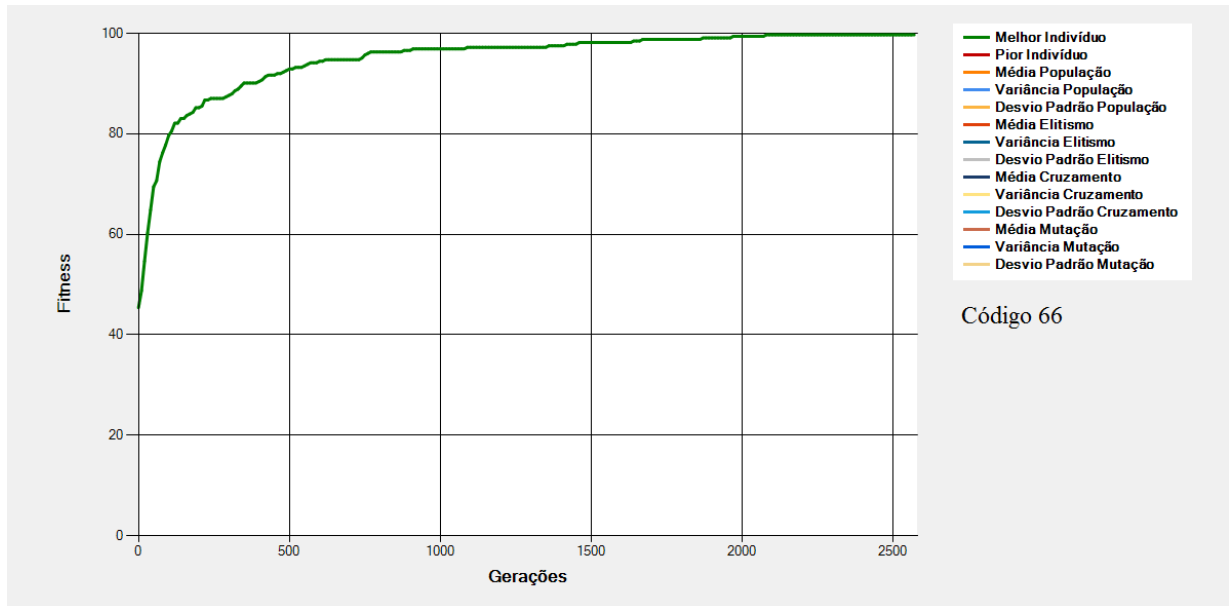


Figura 8-60 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-23 - Cruzamentos (Código 66)

São mostrados na e na Figura 8-60 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-23. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Pontos de Corte”.

O aumento no número de pontos de corte permite uma distribuição maior dos genes na população gerada. Essa distribuição pode, por vezes, acarretar no aumento da velocidade de convergência do EA como pode ser observado na execução de código 66 (Figura 8-60) em comparação com a execução de código 65 (Figura 8-59).

8.25 Cruzamentos - Quantidade de Progenitores e de Pontos de Corte por Cruzamento

Tabela 8-24 - Cruzamentos - Quantidade de Progenitores e Quantidade de Pontos de Corte por Cruzamentos

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
67	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	3	2	10%	100000
68	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	3	3	10%	100000
69	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	4	2	10%	100000
70	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	4	3	10%	100000

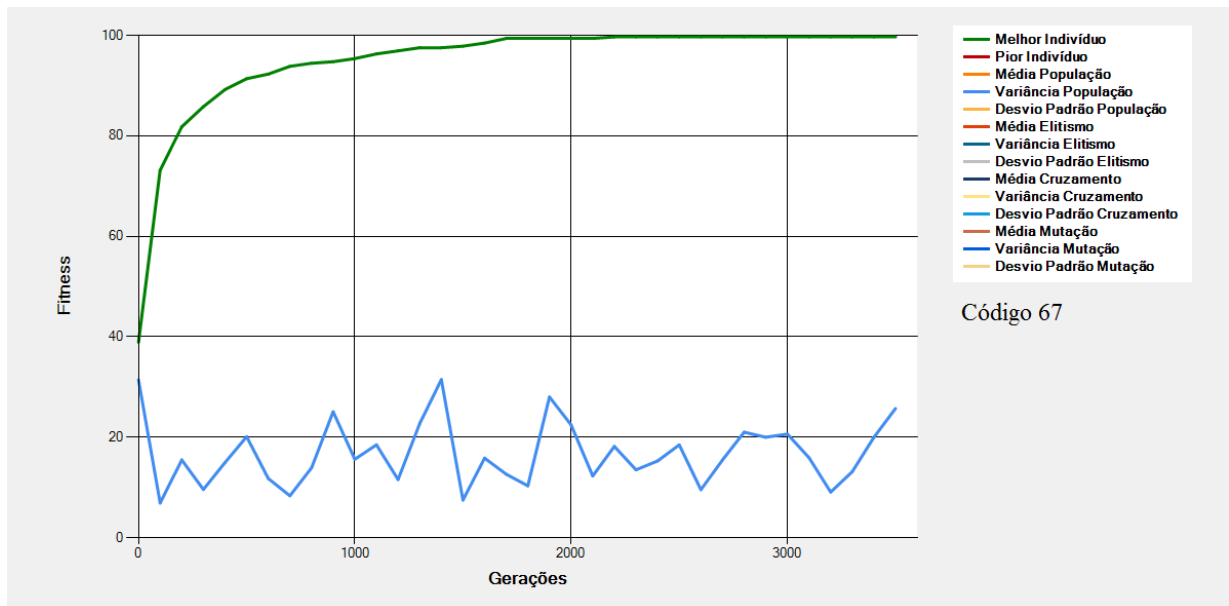


Figura 8-61 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 67)

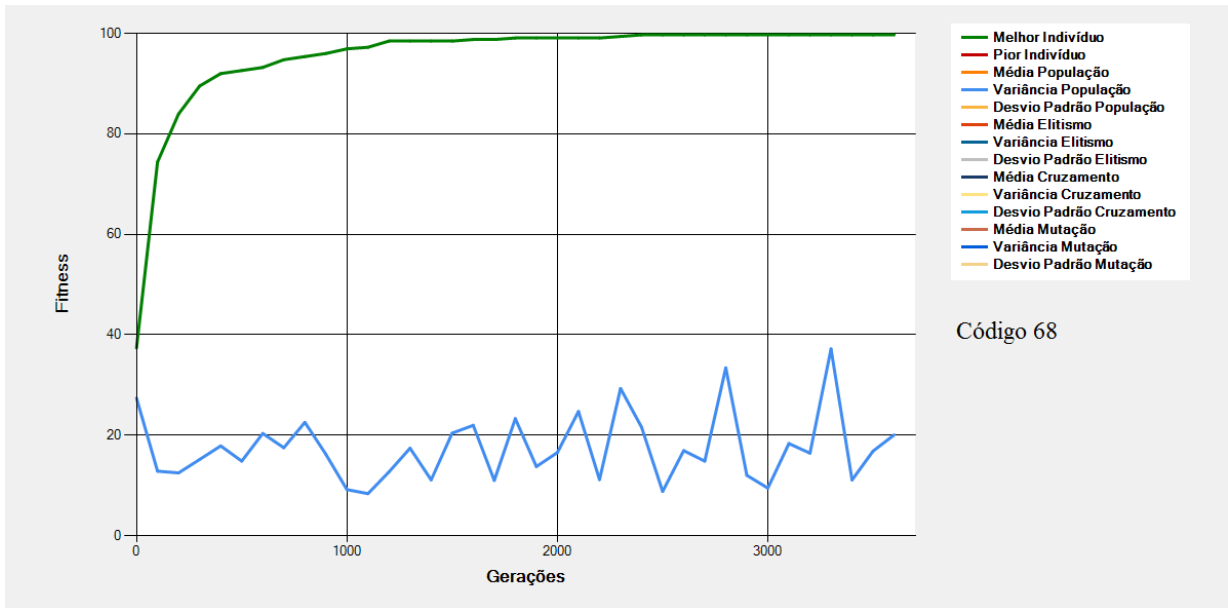


Figura 8-62 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 68)

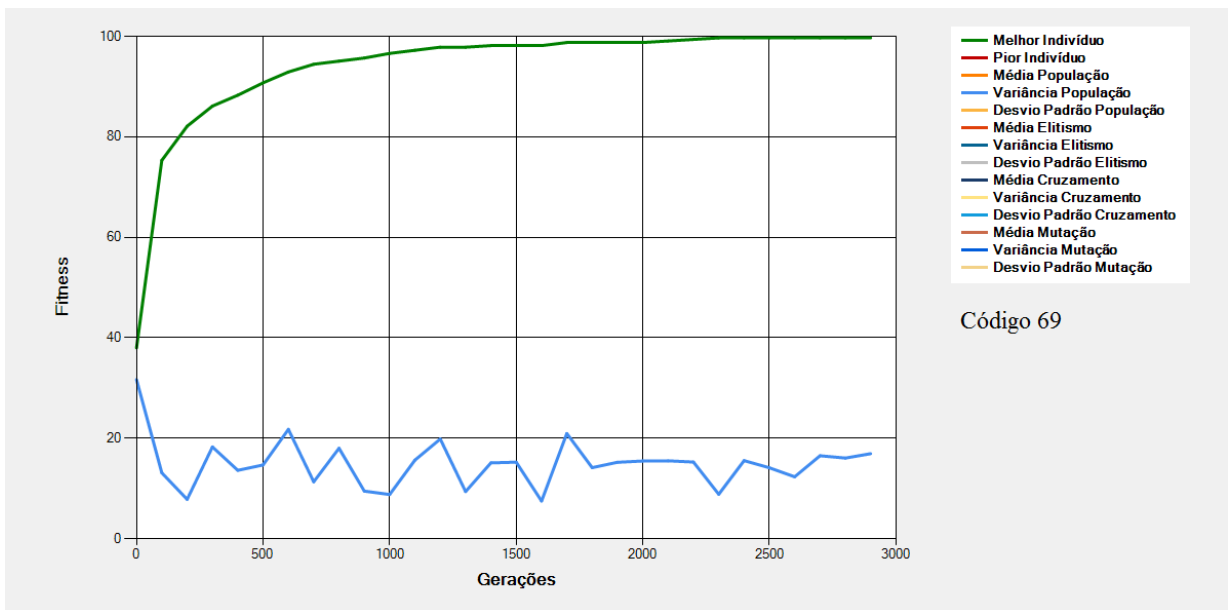


Figura 8-63 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 69)

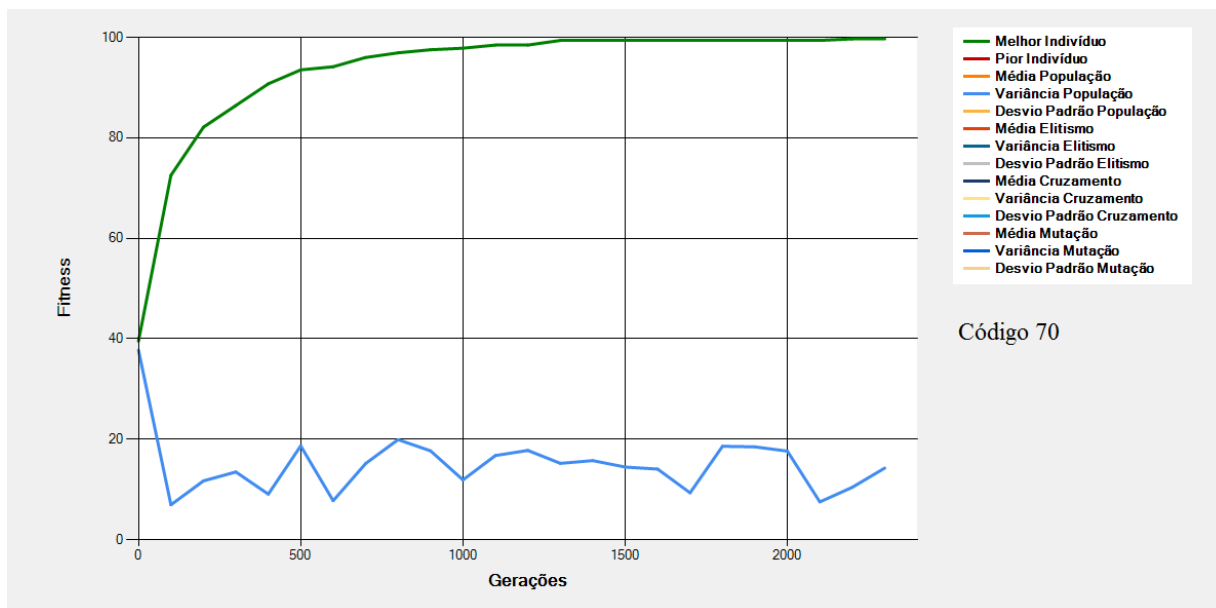


Figura 8-64 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-24 - Cruzamentos (Código 70)

São mostrados na Figura 8-61, na Figura 8-62, na Figura 8-63 e na Figura 8-64 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-24. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Quantidade de Progenitores por Cruzamento” e na “Quantidade de Pontos de Corte”.

Nas execuções acima, um número maior de progenitores por cruzamento fez com que a quantidade de gerações necessárias para se chegar ao ótimo global fosse diminuída. Além disso, o número maior de cortes fez com que os genes fossem dispersados em maior intensidade na população fazendo com que a diversidade da população aumentasse e o número de gerações para se alcançar o ótimo fosse diminuída.

Das execuções acima, a que chegou ao ótimo em um número menor de gerações foi a que combinou a maior quantidade de progenitores por cruzamento com o maior número de pontos de corte para realização dos cruzamentos.

8.26 Mutação - Taxa de Mutação

Tabela 8-25 - Mutação - Taxa de Mutação

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutação	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
71	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	1%	100000
72	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	5%	100000
73	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	20%	100000
74	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	50%	100000
75	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	70%	100000
76	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	100%	100000

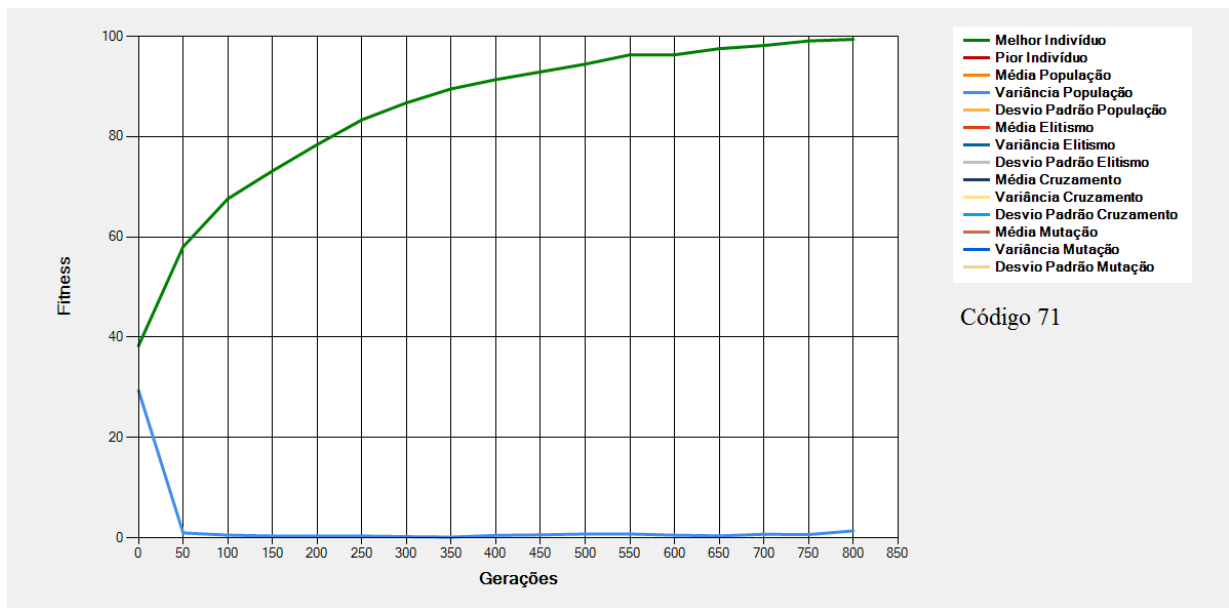
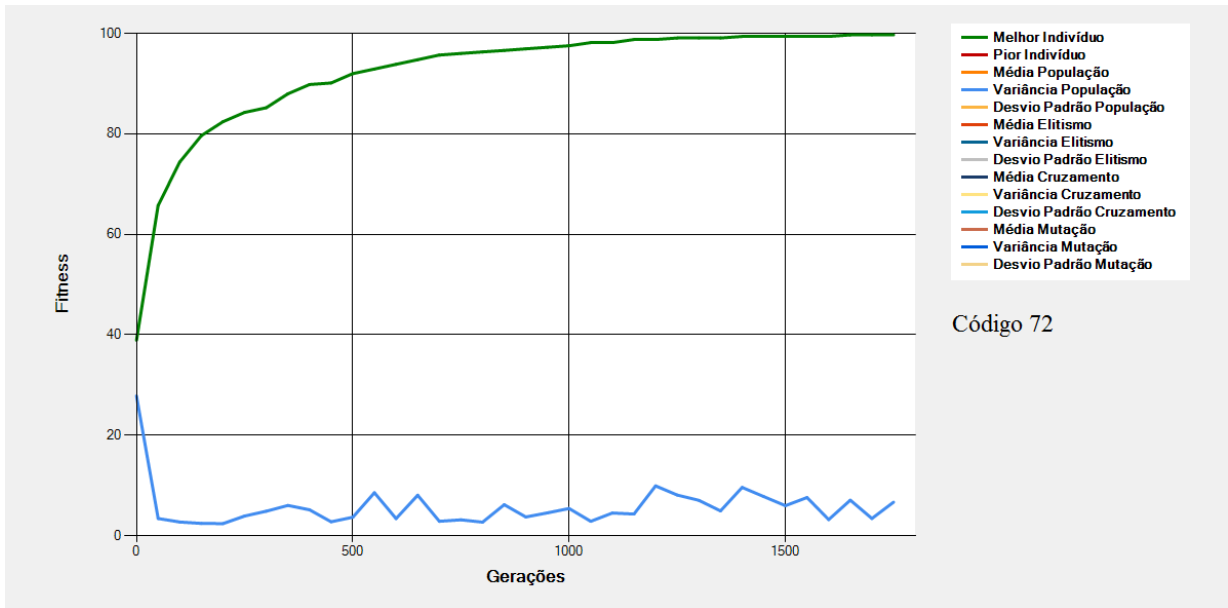
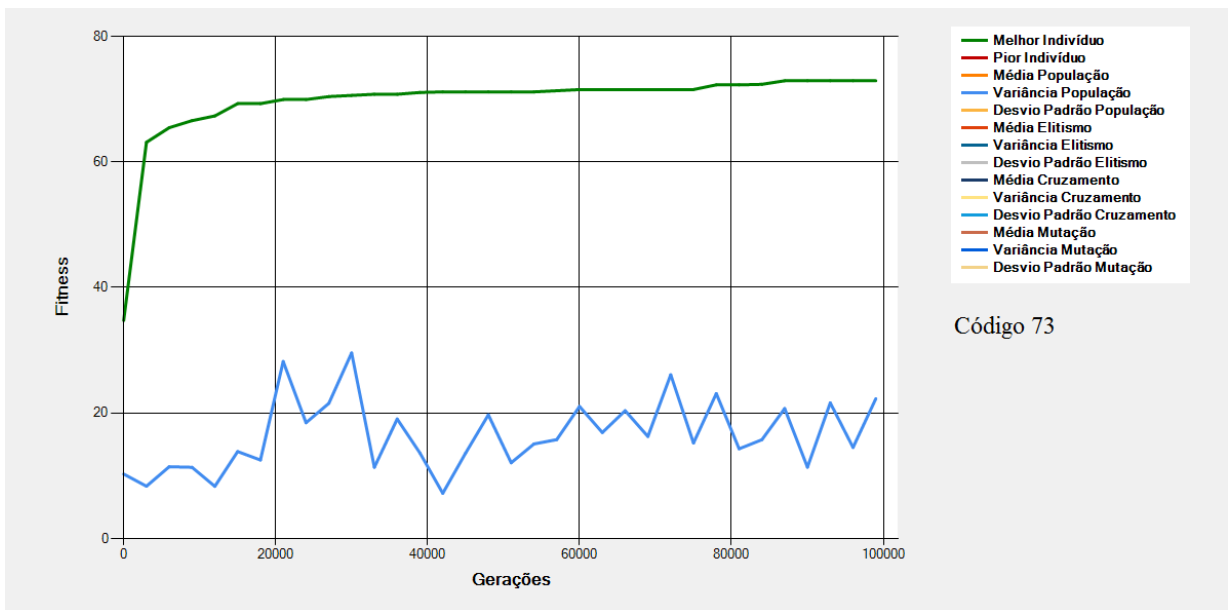


Figura 8-65 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 71)



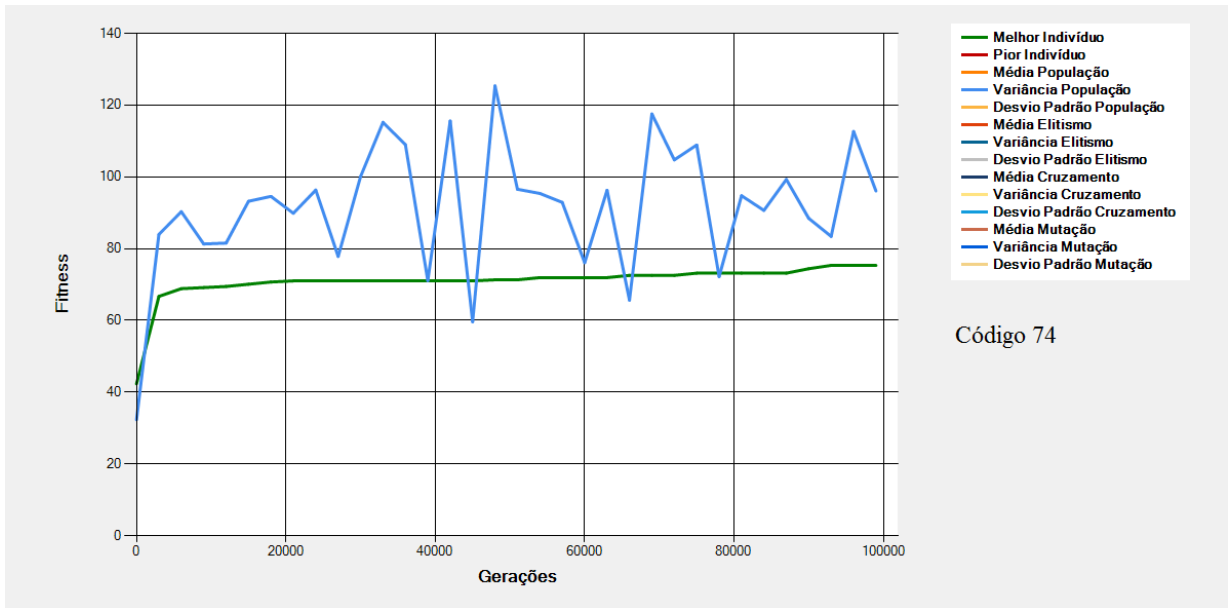
Código 72

Figura 8-66 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 72)



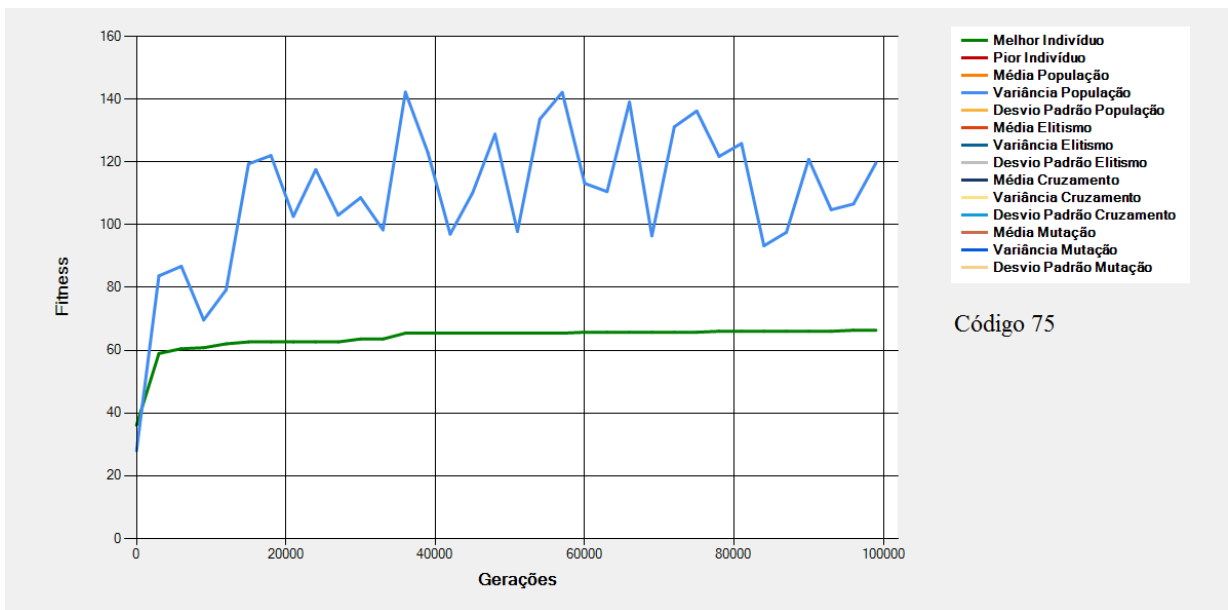
Código 73

Figura 8-67 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 73)



Código 74

Figura 8-68 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 74)



Código 75

Figura 8-69 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 75)

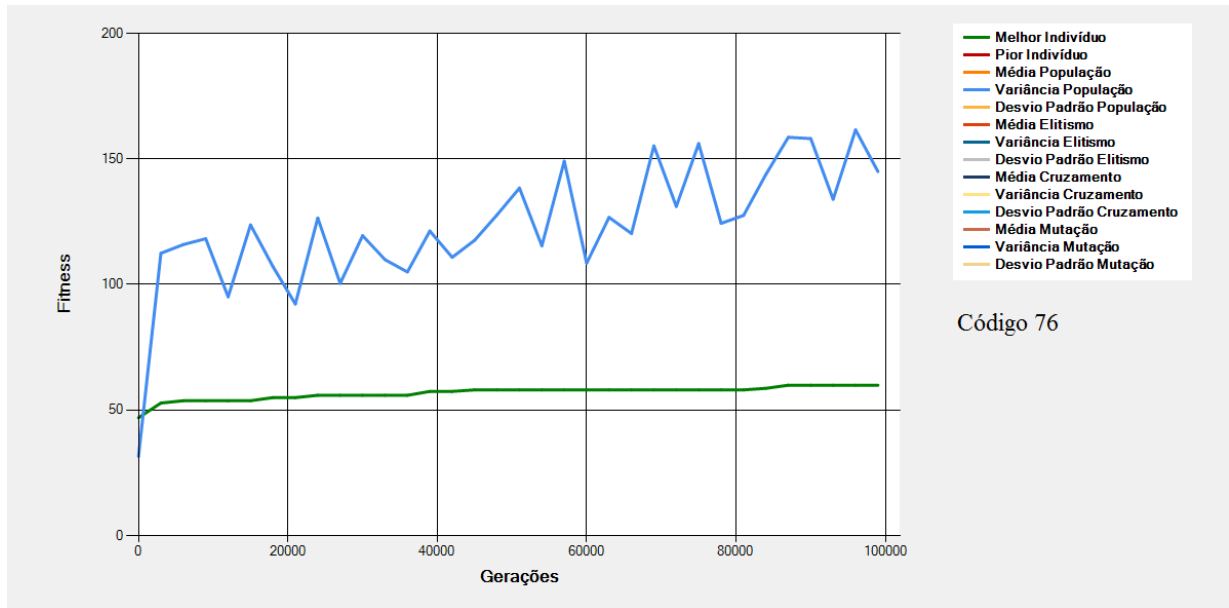


Figura 8-70 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-25 - Mutação (Código 76)

São mostrados na Figura 8-65, na Figura 8-66, na Figura 8-67, na Figura 8-68, na Figura 8-69 e na Figura 8-70 os gráficos dos melhores indivíduos e da variância da população de acordo com os dados da Tabela 8-25. Nessas configurações tem-se alterações (em relação à Execução Padrão) na “Taxa de Mutação”.

Nos exemplos acima percebe-se que a medida que a taxa de mutação aumenta, maior é a variância da população provocada pela dispersão dos indivíduos no espaço de soluções provocada pelo operador de mutação. Com a dispersão dos dados cada vez maior a cada geração percebe-se que a convergência não é atingida nas configurações de código 73, 74, 75 e 76 que correspondem a taxas de mutação de 20%, 50%, 70% e 100% respectivamente. Nestes casos ocorre o chamado *Randon Walk* ou busca aleatória e, desse modo, a convergência não é atingida.

8.27 Execução - Número Máximo de Gerações

Tabela 8-26 - Execução - Número Máximo de Gerações

Código	Regras Fuzzy				População	Cotas			Grupos de Progenitores		Seleção		Cruzamentos		Mutações	Execução
	Quantidade de Variáveis de Entrada	Quantidade de Valores Possíveis para Cada Variável de Entrada	Quantidade de Variáveis de Saída	Quantidade de Valores Possíveis para Cada Variável de Saída		Cota de Elitismo	Cota de Cruzamento	Cota de Mutação	Quantidade de Grupos de Progenitores	Proporção dos Grupos de Progenitores	Método de Seleção	Taxa de Seleção	Quantidade de Progenitores por Cruzamento	Quantidade de Pontos de Corte		
77	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	1000
78	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	10000
79	4	3, 3, 3, 3	2	3, 3	100	30%	60%	10%	1	100%	Roleta	50%	2	1	10%	1000000

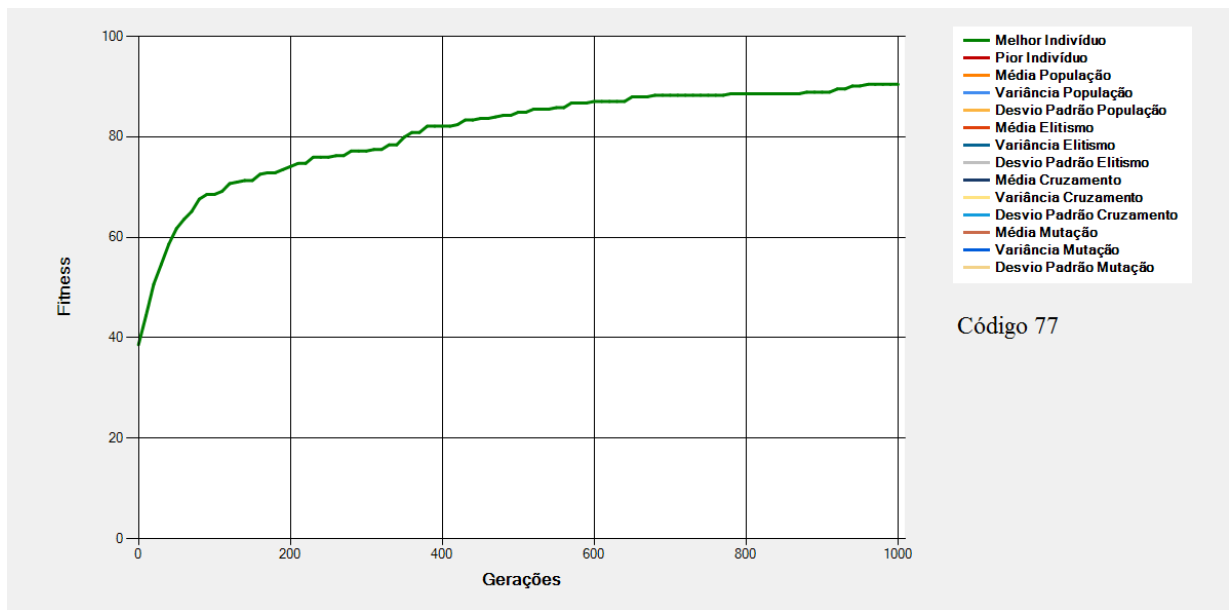


Figura 8-71 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 77)

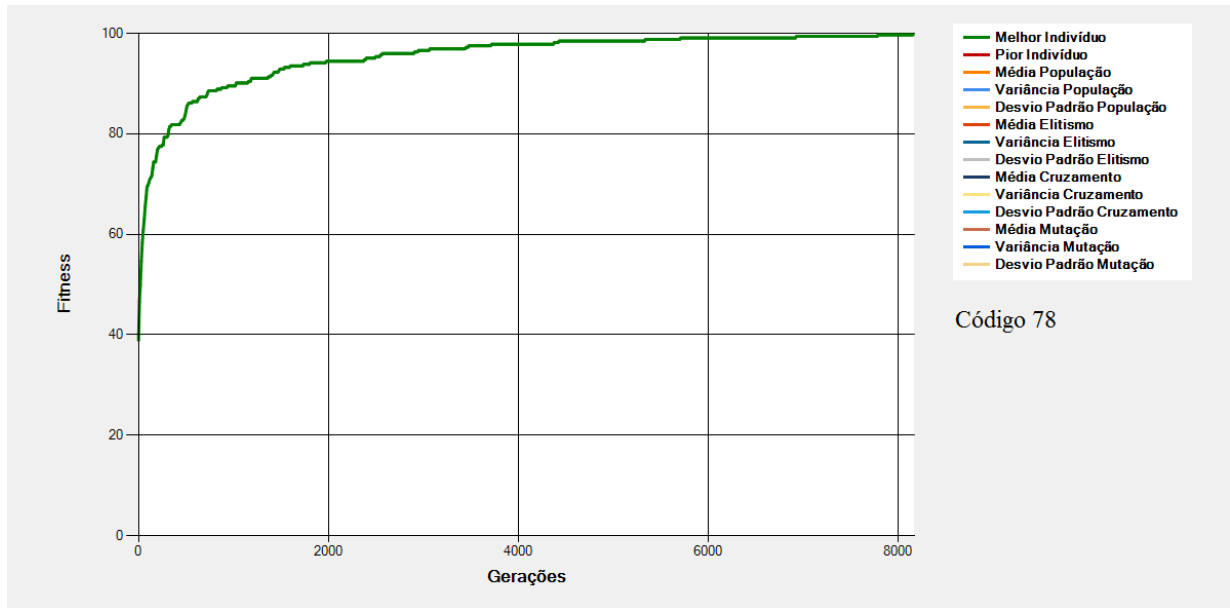


Figura 8-72 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 78)

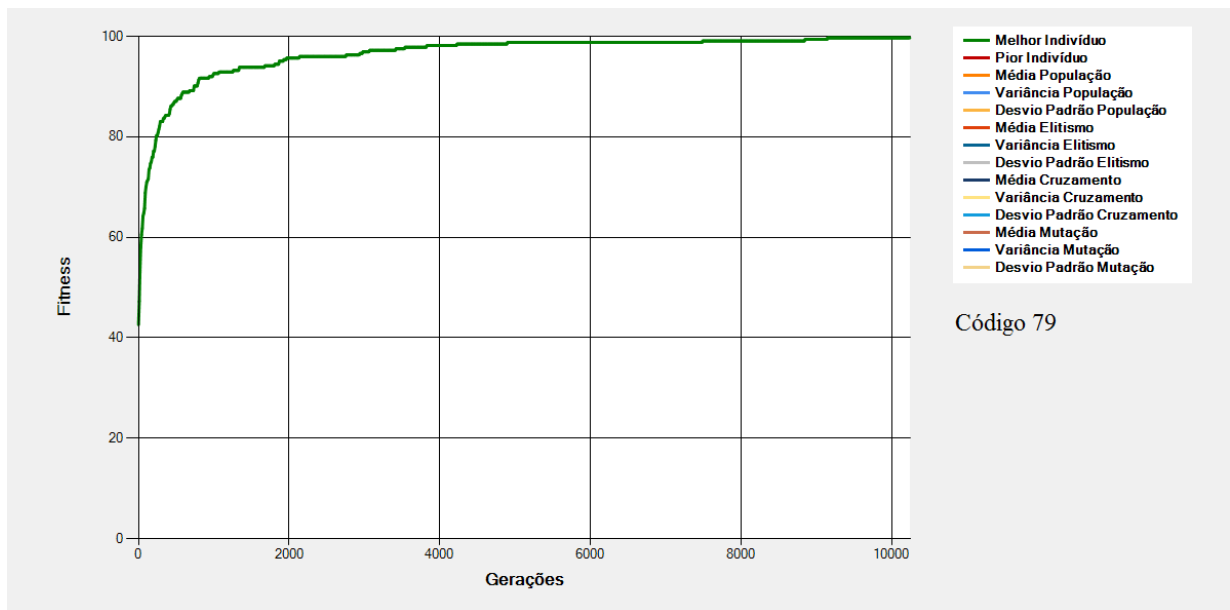


Figura 8-73 - Gráfico do Melhor Indivíduo de acordo com os dados da Tabela 8-26 - Execução (Código 79)

São mostrados na Figura 8-71, na Figura 8-72 e na Figura 8-73 os gráficos dos melhores indivíduos de acordo com os dados da Tabela 8-26. Nessas configurações tem-se alterações (em relação à Execução Padrão) no “Número Máximo de Gerações”.

Nos exemplos acima, a definição do número de gerações tem papel importante na execução do EA uma vez que um número baixo de gerações pode não dar o tempo suficiente para que o EA atinja o ótimo global. Entretanto, um número muito grande de gerações faz com que o tempo de processamento seja alto, o que pode ser um problema em situações que exigem respostas rápidas.