

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ANDRÉ MARCELO SCHNEIDER

**Abordagem Evolucionária com Idades
para Construção de Conhecimento
Aplicado à Robótica Móvel**

Tese apresentada como requisito parcial
para a obtenção do grau de
Doutor em Ciência da Computação

Prof. Dr Dante Augusto Couto Barone
Orientador

Porto Alegre, 23 de janeiro de 2006

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Schneider, André Marcelo

Abordagem Evolucionária com Idades para Construção de Conhecimento Aplicado à Robótica Móvel / André Marcelo Schneider. – Porto Alegre: PPGC da UFRGS, 2006.

118 f.: il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2006. Orientador: Dante Augusto Couto Barone.

1. Inteligência Artificial. 2. Robótica. 3. Computação Evolucionária. 4. Robótica Evolucionária. I. Barone, Dante Augusto Couto. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Flávio Rech Wagner

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Muitas pessoas colaboraram de forma decisiva para a conclusão deste trabalho, especialmente minha esposa, Daniela, que, além de suportar meu meu "mau" humor e comportamento "não muito amistoso" nos períodos de maior dificuldade, também acreditou em mim e não permitiu que eu desistisse no meio do caminho. Também meu filho, Guilherme, que não teve toda a atenção que merecia nesse tempo e ainda teve de aguentar um pai ranzinza - espero que agora possa compensar.

Meu orientador, prof. Dante, que oportunizou o ingresso no doutorado. Devo dizer que relação de orientador-orientando se transformou numa parceria: que continuemos a desenvolver novos projetos.

O Instituto de Informática da UFRGS, que disponibilizou o equipamento necessário para os experimentos, bem como o espaço físico para a montagem da arena de testes.

Os amigos que acompanharam essa caminhada, Adrovane e Ana Paula, Luis Gustavo e Andreuza; os colegas do Laboratório de Robótica Inteligente e também os da techROBOT Sistemas de Informática.

Dentre estes amigos, também estão o Rodrigo, que muito colaborou na codificação dos programas; o Felipe, que acompanhou boa parte dos experimentos realizados com o NOMAD 200, colaborando para que ele não se "auto-destruísse"; e o Edson, com suas, às vezes um tanto malucas, sugestões.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Motivação	12
1.2 Objetivos	13
1.3 Estrutura do Trabalho	13
2 COMPUTAÇÃO EVOLUCIONÁRIA	14
2.1 Algoritmos Genéticos	15
2.1.1 Características dos Algoritmos Genéticos	17
2.1.2 Formas de Representação	18
2.1.3 Avaliação dos Indivíduos	19
2.1.4 Operadores Genéticos	19
2.1.5 Efeitos dos Operadores Genéticos	20
2.1.6 Inserção dos Novos Indivíduos na População	21
2.1.7 O Algoritmo Genético Simples de Holland	21
2.1.8 Sistemas Híbridos	22
2.2 Algoritmos Genéticos com Idades (AGI)	22
2.2.1 Conceituação	23
2.2.2 Vantagens em Relação ao Algoritmo Genético Convencional	23
2.2.3 Atribuição da Idade	24
2.3 Sistemas Classificadores	25
2.3.1 Funcionamento e Caracterização	26
2.3.2 Regras	27
2.3.3 Subsistema de Tratamento de Regras e Mensagens	27
2.3.4 Subsistema de Apropriação de Créditos	28
2.3.5 Subsistema de Descoberta de Novas Regras	29
2.3.6 Algoritmo Simplificado	30

3	ROBÓTICA EVOLUCIONÁRIA	31
3.1	Caracterização	32
3.2	Agentes Robóticos Evolucionários	33
3.2.1	Simuladores e Robôs Reais	33
3.3	Trabalhos Correlatos	34
4	PROPOSTA DE SISTEMA: APRENDIZADO EVOLUCIONÁRIO	40
4.1	Descrição do Problema	41
4.2	Alternativa Desenvolvida	41
4.3	Padrões de funcionamento do Robô	43
4.4	Descrição da Arquitetura Proposta	44
4.4.1	Memória	45
4.4.2	Módulo de Reprodução e Evolução	49
4.4.3	Módulo de Controle Populacional	50
4.4.4	Módulo de Execução	53
4.4.5	Critérios para Encerramento da Execução do Sistema	58
4.4.6	Fluxo de Execução do Sistema	59
5	EXPERIMENTOS E RESULTADOS	60
5.1	Ambientes - Cenários de Manobras	60
5.2	O Robô NOMAD 200	63
5.3	Identificação e Configuração do Parâmetros	65
5.3.1	O Simulador do NOMAD 200	66
5.4	Experimentos Simulados	68
5.4.1	Experimentos Simulados - Ambiente sem Obstáculos	68
5.4.2	Experimentos Simulados - Ambiente com Obstáculos	76
5.5	Experimentos com o Robô NOMAD 200	83
5.5.1	Experimentos Reais - Ambiente sem Obstáculos	83
5.5.2	Experimentos Reais - Ambiente com Obstáculos	90
5.5.3	Experimentos Reais - Ambiente com Obstáculos e Beco	93
5.5.4	Experimentos Reais - Execução Contínua	95
5.6	Discussão dos Resultados	102
5.6.1	Diferenças entre os Métodos de Cruzamento	104
5.6.2	Comparação com Trabalhos Similares	105
6	CONSIDERAÇÕES FINAIS	109
6.1	Contribuições	111
6.2	Possibilidades de Trabalhos Futuros	112
	REFERÊNCIAS	113

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
RNA	Redes Neurais Artificiais
AG	Algoritmo Genético
AGS	Algoritmo Genético Simples
AGI	Algoritmo Genético com Idades
IAD	Inteligência Artificial Distribuída
SC	Sistemas Classificadores
AE	Algoritmos Evolucionários
CE	Computação Evolucionária
RE	Robótica Evolucionária

LISTA DE FIGURAS

Figura 2.1: Algoritmo Genético Simples (AGS) de Holland.	22
Figura 2.2: Estrutura Geral de um Sistema Classificador	26
Figura 2.3: Subsistema de Tratamento de Regras e Mensagens	28
Figura 2.4: Subsistema de Apropriação de Crédito	29
Figura 2.5: Subsistema de Descoberta de Novas Regras	29
Figura 2.6: Algoritmo Simplificado de Sistemas Classificadores.	30
Figura 3.1: Arquitetura do Sistema Implementado Por Grefenstette (1994)	36
Figura 3.2: Arquitetura do Sistema Implementado Por Grefenstette (1996)	37
Figura 3.3: Arquitetura do Sistema proposto por Cazangi (2005)	38
Figura 4.1: Arquitetura de Agente Padrão	44
Figura 4.2: Arquitetura do Sistema.	45
Figura 4.3: Forma de Representação das Regras.	46
Figura 4.4: Situação representada pela regra	46
Figura 4.5: Valores considerados satisfatórios para cada um dos sensores	56
Figura 4.6: Código Procedimento de Avaliação.	57
Figura 4.7: Código Básico do Sistema Robótico.	59
Figura 5.1: Exemplos de Cenários para os Experimentos Simulados	61
Figura 5.2: Imagens dos Cenários para os Experimentos com o Robô Real	63
Figura 5.3: O Robô Nomad 200	64
Figura 5.4: Sensores do Robô NOMAD 200	65
Figura 5.5: Simulador do NOMAD 200	67
Figura 5.6: Interface de Comunicação com o NOMAD 200	68
Figura 5.7: Variação da População - Experimento 1.1	71
Figura 5.8: Variação da Idade Média da População - Experimento 1.1	71
Figura 5.9: Comportamento Padrão do robô - Experimento 1.1	72
Figura 5.10: Conjunto de Regras Consolidadas - Experimento 1.1	72
Figura 5.11: Evolução da População - Experimento 1.1	73
Figura 5.12: Variação da População - Experimento 1.2	75
Figura 5.13: Variação da Idade Média da População - Experimento 1.2	76
Figura 5.14: Variação da População - Experimento 2.1	78
Figura 5.15: Variação da Idade Média da População - Experimento 2.1	78
Figura 5.16: Comportamento Padrão do robô - Experimento 2.1	79
Figura 5.17: Conjunto de Regras Consolidadas - Experimento 2.1	80
Figura 5.18: Variação da População - Experimento 2.2	81
Figura 5.19: Variação da Idade Média da População - Experimento 2.2	82

Figura 5.20: Variação da População - Experimento 3.1	85
Figura 5.21: Variação da Idade Média da População - Experimento 3.1	86
Figura 5.22: Comportamento Padrão do robô - Experimento 3.1	86
Figura 5.23: Conjunto de Regras Consolidadas - Experimento 3.1	87
Figura 5.24: Variação da População - Experimento 3.2	88
Figura 5.25: Variação da Idade Média da População - Experimento 3.2	88
Figura 5.26: Conjunto de Regras Consolidadas - Experimento 3.2	89
Figura 5.27: Evolução da População - Experimento 3.2	90
Figura 5.28: Variação da População - Experimento 4.1	91
Figura 5.29: Variação da Idade Média da População - Experimento 4.1	91
Figura 5.30: Comportamento Padrão do robô - Experimento 4.1	92
Figura 5.31: Variação da População - Experimento 4.2	92
Figura 5.32: Variação da Idade Média da População - Experimento 4.2	93
Figura 5.33: Variação da População - Experimento 5.1	94
Figura 5.34: Variação da Idade Média da População - Experimento 5.1	95
Figura 5.35: Comportamento Padrão do robô - Experimento 5.1	96
Figura 5.36: Variação da População - Experimento 5.2	97
Figura 5.37: Variação da Idade Média da População - Experimento 5.2	97
Figura 5.38: Execução Contínua - Experimento 6.1a	98
Figura 5.39: Execução Contínua - Experimento 6.1b	98
Figura 5.40: Execução com Modificação do Cenário - Experimento 6.2a	99
Figura 5.41: Evolução da População - Experimento 6.2a	100
Figura 5.42: Execução com Modificação do Cenário - Experimento 6.2b	101
Figura 5.43: Evolução da População - Experimento 6.2b	102
Figura 5.44: Comportamento de Sub-População - Experimento 2.3	103
Figura 5.45: Comportamento de Sub-População - Experimento 4.3	104
Figura 5.46: Resultado de Reprodução utilizando Cruzamento Médio	105
Figura 5.47: Resultado de Reprodução utilizando Cruzamento Uniforme	106

LISTA DE TABELAS

Tabela 4.1: Intervalos de classificação da leitura dos Sensores	43
Tabela 4.2: Intervalos de classificação da leitura dos Sensores de Navegação .	53
Tabela 4.3: Intervalos de classificação da leitura dos Sensores de Segurança . .	56
Tabela 5.1: Configuração dos Parâmetros do Sistema	65
Tabela 5.2: Limites para os intervalos de leitura dos sensores	67
Tabela 5.3: Resultados do Experimento 1.1	69
Tabela 5.4: Totalização do Experimento 1.1	70
Tabela 5.5: Resultados do Experimento 1.2	74
Tabela 5.6: Totalização do Experimento 1.2	75
Tabela 5.7: Totalização do Experimento 1.3	76
Tabela 5.8: Totalização do Experimento 2.1	77
Tabela 5.9: Totalização do Experimento 2.2	81
Tabela 5.10: Totalização do Experimento 2.3	83
Tabela 5.11: Resultados do Experimento 3.1	84
Tabela 5.12: Totalização do Experimento 3.1	84
Tabela 5.13: Resultados do Experimento 3.2	87
Tabela 5.14: Totalização do Experimento 3.2	87
Tabela 5.15: Totalização do Experimento 4.1	90
Tabela 5.16: Totalização do Experimento 4.2	92
Tabela 5.17: Totalização do Experimento 5.1	94
Tabela 5.18: Totalização do Experimento 5.2	95

RESUMO

Este trabalho apresenta e discute uma proposta de estratégia inédita para o problema de aprendizado de regras através de Sistemas Classificadores, aplicado à robótica móvel, utilizando um robô NOMAD 200. Esta estratégia tem como base, teorias de Algoritmos Genéticos e de Sistemas Classificadores, que são os paradigmas constituintes do núcleo da arquitetura implementada para o controle do robô. O aspecto diferencial desta abordagem é a inspiração em Algoritmos Genéticos com Idades, para permitir o uso e controle de uma população de tamanho variável.

O sistema foi modelado observando-se características físicas do robô NOMAD 200 e sendo constituído por módulos de gerenciamento de memória, reprodução, controle da população e execução. A memória se apresenta como uma base de regras de produção; o módulo de reprodução incorpora um AG tradicional, com operadores de seleção, cruzamento e mutação; o controle populacional permite o uso de população de tamanho variável, através do de índices de usabilidade e similaridade das regras com as situações confrontadas pelo robô; por fim, o módulo de execução é responsável pela interação do robô com o ambiente, realizando leitura dos sensores e ações pelos atuadores e, quando necessário, ativar funções de segurança para preservar a integridade física do robô.

Para dar sustentabilidade à proposta, esta foi validada através de vários experimentos, realizados em ambientes simulados e em um ambiente real, com um robô NOMAD 200, em diferentes cenários. Os ambientes testados variam desde ambientes esparsos até labirintos com obstáculos e paredes ortogonais entre si. Para cada experimento são apresentados os resultados e respectiva análise de dados. Foram realizadas análises criteriosas no comportamento da população, observando seu crescimento e idade média, bem como os eventos ocorridos no processo de aprendizado, para certificar as características a que se propõe esta abordagem.

A principal contribuição deste trabalho é o uso da "IDADE" e "USABILIDADE" em um sistema baseado em SC. A usabilidade substitui o atributo de energia e respectivos cálculos do SC tradicional, no processo de escolha das regras, simplificando a implementação. Além disso, pode ser utilizado como índice de ajuste, para que possam ser usadas técnicas convencionais de seleção. A idade é responsável por preservar ou eliminar os indivíduos da população, através de estratégias de penalização e recompensa, possibilitando manter uma população de regras de tamanho variável, permitindo, ainda, manter a diversidade genética na população e evitar a sua homogeneização, bem como isentar o modelador do sistema da definição destes parâmetros.

Palavras-chave: Inteligência Artificial, Robótica, Computação Evolucionária, Robótica Evolucionária.

An evolutionary approach with ages to knowledge building applied to mobile autonomous robotics

ABSTRACT

In this work, we propose a new strategy to the problem of learning rules in a Evolutionary System that is applied for mobile robotics using a NOMAD 200 robot. This strategy is based on Genetic Algorithms and Classifier Systems theories, which are the paradigms of the implemented architecture core for robot control. The unique feature of this approach is the inspiration on Genetic Algorithms with Ages. This feature allows the algorithm to make use of a controlled variable size population.

The system was designed respecting the physical features of the NOMAD 200 robot. It is composed by modules of memory, reproduction, populational control and execution. The memory is the base for production rules. The reproduction module is a conventional GA, with operators for selection, crossover and mutation. The population control allows the use of a variable size population, based on the usability and the similarity of the rules on the situations presented to the robot. Finally, the execution module is responsible for the interaction between the robot and the environment, making the sensors reading and action application from the actuators and, if necessary, activating the security functions to preserve the physical integrity of the robot.

To give support to the proposal, it was validated through several experiments, performed both in a simulated environment and in a real NOMAD 200 robot, in several scenarios. The environments used in the experiments ranged from open spaces to labyrinths with obstacles and orthogonal walls. We present the results and data analysis for each one of the experiments. Also, the population behavior is analysed, by the observation of his growing and average age and the events occurred during the learning process, to confirm the features of these approach.

The main contribution of this work is the use of "AGE" and "USABILITY" in a CS based system. The usability replaces the strength attribute and respective calculations necessary in the process of choosing rules in traditional CS. Because of this change, our solution is simpler to implement than traditional CS systems. Besides that, the usability can be used as fitness value, making possible the use of conventional selection techniques. The Age is responsible for the decision of to preserve or to eliminate individuals from the population. The choose of individuals is done by a penalty and reward strategy, which permits a variable size population of rules with genetic diversity and avoid the population's homogenization. The use of the age for decision making also preserves the system developer from the task of defining these parameters.

Keywords: Artificial Intelligence, Robotics, Evolutionary Computation, Evolutionary Robotics.

1 INTRODUÇÃO

Os últimos anos testemunharam um grande esforço no desenvolvimento de modelos computacionais inteligentes inspirados em processos naturais, para a solução de problemas complexos. Um dos paradigmas comumente utilizados neste desenvolvimento é a Computação Evolucionária, que tenta mimetizar os processos de evolução natural descritos pela teoria de Charles Darwin.

E no que se refere ao estudo da inteligência, existem muitas hipóteses, mas ainda não existe um consenso. Permanecem muitas dúvidas do que realmente pode ser classificado como "inteligência" ou "comportamento inteligente". Uma entidade inteligente é aquela que apresenta capacidade de aprendizagem e adaptação. Portanto, pode-se dizer que a evolução, que gera comportamento adaptativo, é um ponto chave no processo de desenvolvimento de comportamento inteligente.

Existe uma grande necessidade de se desenvolver sistemas capazes de executar tarefas em ambientes desconhecidos e não estruturados. Estes sistemas devem possuir e construir habilidades para adaptar-se continuamente aos mesmos. Isto justifica a realização do grande número de projetos de pesquisa na área de sistemas adaptativos. E, dentre estes, estão os sistemas evolucionários.

Um movimento mais recente que investiga alternativas nestas teorias evolucionárias concentra seus esforços para sua aplicação em robótica. O objetivo central desta área, denominada Robótica Evolucionária, é capacitar máquinas, a partir da evolução de sistemas adaptativos evolucionários, para que sejam capazes de construir sua própria interpretação do mundo, de forma totalmente autônoma, sem a interferência direta de um projetista, a fim de executarem as tarefas a que foram designadas da forma mais eficiente possível.

1.1 Motivação

A motivação principal para o desenvolvimento deste trabalho é investigar a viabilidade de criar um sistema evolucionário, capaz de construir uma base de conhecimento, representado na forma de regras explicitamente declarativas, que habilita o robô, através de uma contínua interação com o ambiente, a realizar tarefas determinadas em um ambiente desconhecido.

Tradicionalmente, os robôs são programados para resolver determinadas tarefas e permanecem com seu conhecimento e suas habilidades configuradas de forma estática, o que não possibilita um comportamento adaptativo para ambientes desconhecidos ou não estacionários.

Considerando que robôs móveis necessitam, muitas vezes, atuar em ambientes desconhecidos ou, em casos extremos, em ambientes que sofrem constantes modi-

ficações em sua configuração, faz-se necessário que estes robôs desenvolvam a capacidade de adaptar-se continuamente às novas condições. Para isto, eles devem construir seu conhecimento a respeito do mundo através da interação com este. O processo de interação do robô com o ambiente se efetua a partir de seus sensores, que captam informações a respeito do ambiente para que ele possa executar alguma tarefa, a partir da ativação de seus atuadores.

1.2 Objetivos

O objetivo principal desta Tese de Doutorado é desenvolver uma proposta de arquitetura inteligente para capacitar um robô móvel a construir conhecimento e habilidade de locomoção, levando em consideração que este conhecimento deverá estar representado na forma de um conjunto de regras generalistas explicitamente declarativas, permitindo que a aprendizagem aconteça de forma autônoma e evolutiva, num contínuo processo de interação sensorio-motor com o ambiente, bem como, que um supervisor especialista possa instruir o sistema através da inserção de regras anteriormente conhecidas.

Esta arquitetura deve ser modelada a partir das atividades sensoriais do robô e nas tarefas que este deve executar. A partir destas informações, o sistema deve ser capaz de navegar pelo ambiente, evoluindo continuamente, inserindo e retirando regras de sua base de conhecimento. A partir deste processo de aprendizagem evolucionária, o robô deve ser capaz de executar as tarefas a que foi designado.

O número de regras que compõe o conjunto gerado deverá ser resultado do processo de aprendizagem, isentando o usuário do sistema das tarefas de modelagem e definição dos parâmetros relativos a este aspecto.

Inicialmente, o sistema deverá ser testado em simulação, com diferentes ambientes, para comprovar sua eficácia e garantir que, sendo executado em um robô real, não ocorram situações que possam colocar em risco ou, até mesmo, danificar o equipamento.

Posteriormente o sistema será validado no robô NOMAD 200 em ambientes reais, para confirmar a estratégia desenvolvida.

1.3 Estrutura do Trabalho

Esta tese está organizada da seguinte forma: no Capítulo 2 são descritas as técnicas de Computação Evolucionária utilizadas como base para o desenvolvimento deste trabalho. Seguindo a revisão de literatura, no Capítulo 3 é apresentada a Robótica Evolucionária, esta área de investigação científica, seus objetivos, tecnologias envolvidas e trabalhos relacionados à esta tese. No Capítulo 4 são mostrados a estrutura do sistema desenvolvido, as estratégias de controle e aprendizagem utilizadas, bem como as variações e estratégias incorporadas e adaptadas. No Capítulo 5 são descritos a configuração do robô e dos cenários utilizados, a metodologia de condução dos experimentos, mostrando a realização dos mesmos e analisando os resultados obtidos. No último capítulo são apresentadas as conclusões gerais e propostas de trabalhos futuros. Para finalizar, são apresentadas as referências do material bibliográfico utilizado.

2 COMPUTAÇÃO EVOLUCIONÁRIA

Inspirada na teoria da evolução natural, a classe de algoritmos conhecida como Computação Evolucionária (CE) é constituída por estratégias que surgiram apresentando habilidades e eficiência na solução de problemas que envolvem a necessidade de adaptação, auto-configuração, otimização e aprendizado de máquina (GOLDBERG, 1989) (WHITLEY, 2001). Sua fundamentação encontra-se nas teorias de Charles Darwin, que afirmou que os sistemas naturais evoluem através de sucessivas gerações dos indivíduos mais aptos a competir, sobreviver e reproduzir em determinado ambiente. (DARWIN, 1998).

Para que um sistema computacional possa ser desenvolvido a partir destas idéias, é necessário construir um modelo do sistema, onde exista uma população de indivíduos que representem possíveis soluções para o problema em questão e, a partir de sucessivas iterações destes indivíduos com o ambiente (problema), o sistema evolui para encontrar uma solução satisfatória - mimetizando os processos naturais.

Na área de CE foram propostas e desenvolvidas várias técnicas para implementar a solução de problemas, o que resultou em um conjunto de categorias de Algoritmos Evolucionários (AE). Cada categoria possui seus aspectos particulares, mas todas buscam mimetizar a evolução natural em processos computacionais para a resolução de problemas. A semelhança existente entre elas faz com que os limites que diferenciam cada categoria sejam um tanto quanto nebulosos (FOGEL, 2000). As técnicas mais conhecidas são:

- Algoritmos Genéticos, criados por Holland, buscam evoluir uma população de indivíduos, que representam soluções para um problema em questão, mimetizando processos naturais de seleção, cruzamento e mutação (GOLDBERG, 1989) (HOLLAND, 1975).
- Estratégias Evolucionárias, desenvolvidas por Rechenberg e Schwefel, utiliza a metodologia onde um indivíduo é gerado e este dá origem a um único descendente, que virá a substituí-lo, caso seja mais adaptado ao problema que seu progenitor (MITCHELL, 1996).
- Programação Evolucionária, criada por Fogel, o algoritmo se propõe a operar sobre estruturas que não são somente representações numéricas de soluções de problemas. Inicialmente, foi utilizada na evolução de máquinas de estado finitos (FOGEL, 2000).
- Programação Genética, foi desenvolvida por Koza e objetiva evoluir estruturas do tipo árvore, que podem representar programas (KOZA, 1991).

- Sistemas Classificadores, criados por Holland, são uma variação dos AGs, onde cada indivíduo representa uma regra do tipo "situação-ação" (BULL, 2004).

Este capítulo tem como objetivo descrever os AE que serviram de base para a modelagem do sistema proposto nesta tese, os quais são: Algoritmos Genéticos, sua variante, Algoritmos Genéticos com Idades, e Sistemas Classificadores.

2.1 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são a forma mais conhecida de AE, o que se deve, provavelmente, à sua simplicidade de modelagem e implementação (MITCHELL, 1996) e, principalmente, pela possibilidade de aplicação em diferentes áreas do conhecimento (SERRADA, 1996).

John Holland dedicou-se ao estudo de processos naturais adaptáveis, tendo proposto originalmente os AGs em conjunto com seus alunos e colegas da Universidade de Michigan nos anos 1960 e 1970, com o objetivo de estudar formalmente o fenômeno da adaptação, tal como ocorre na natureza, e desenvolver modelos visando identificar quais mecanismos da adaptação natural pudessem ser importados para os sistemas computacionais (GOLDBERG, 1989).

Os AGs são técnicas de busca baseadas na Teoria da Evolução, nos quais as variáveis são representadas como genes em um cromossomo (indivíduo). Combinam a sobrevivência dos mais aptos com a troca de informação de uma forma estruturada, mas aleatória. O AG apresenta um grupo de soluções candidatas (população) na região de soluções. Por seleção natural e utilização de operadores genéticos, mutação e cruzamento, os cromossomos com melhor aptidão são encontrados; a seleção natural garante que os cromossomos mais aptos recebam maiores chances de gerar descendentes nas populações futuras. Usando um operador de cruzamento, o AG combina genes de dois cromossomos, criando dois novos cromossomos, os quais têm uma grande possibilidade de serem mais aptos que os seus genitores. São algoritmos adaptáveis e capazes, em teoria, da inovação perpétua. São métodos robustos de procura, até mesmo para funções ruidosas.

Enquanto que com as técnicas tradicionais existe o problema de ser muito provável que ocorra a convergência para um ótimo local, quando estejam na sua proximidade, os AGs conduzem a procura, a cada ciclo do algoritmo, em muitos pontos no espaço de busca, simultaneamente, sendo muito menos provável que fiquem presos num máximo local.

Holland acreditava que a incorporação das características naturais de evolução em um computador poderia produzir uma técnica para solucionar problemas da mesma maneira como funcionam na natureza os processos de seleção e adaptação. A forma de codificação dos cromossomos biológicos ainda não é totalmente compreendida pela comunidade científica, mas existem algumas características gerais da teoria que são aceitas, tais como: (DAVIS, 1991)

- evolução;
- seleção natural;
- reprodução;
- ausência de memória.

Os AGs ocupam lugar de destaque entre os paradigmas de CE devido a uma série de razões, dentre as quais: (SERRADA, 1996)

1. apresentam-se como o paradigma mais completo da CE, visto que englobam de forma simples e natural as suas principais características, sendo ainda, o que possui maior base teórica; a qual além de tudo, é de fácil compreensão e com grande capacidade de expansão;
2. apresentam resultados aceitáveis, com relação à precisão e recursos empregados (fáceis de implantar em computadores domésticos de porte médio), para uma ampla gama de problemas de difícil resolução por outros métodos;
3. são muito flexíveis, pois aceitam sem grandes dificuldades diversas alterações na sua implementação e permitem fácil hibridização (vantagem importante no caso de aprendizagem) inclusive com técnicas não relacionadas à CE;
4. em relação aos outros paradigmas da CE, são os que exigem menor conhecimento específico do problema em questão para o seu funcionamento, o que os torna altamente versáteis e, além disso, agregam conhecimento específico com pouco esforço;
5. são o paradigma mais usado dentro da CE;
6. apresentam tolerância à informação ruidosa ou incerta.

Na prática, um AG pode ser facilmente implementado com o simples uso de strings de bits ou caracteres para representar os cromossomos e, com simples operações de manipulação de bits podemos implementar cruzamento, mutação e outros operadores genéticos.

A idéia apresentada pelos AGs é utilizar o poder da evolução e seleção natural para resolver problemas de otimização. Os AGs, embora muito distantes ainda do modelo natural de evolução, vem demonstrando ter um grande potencial para solucionar problemas de otimização, classificação e adaptação.

Estes algoritmos simulam a sobrevivência da aptidão para solucionar um problema, entre os indivíduos de uma determinada população sobre consecutivas gerações. Cada geração consiste de uma população de vetores de caracteres que são análogos ao cromossomo que se observa no DNA humano. Cada indivíduo representa um determinado ponto no espaço de busca e uma possível solução para o problema em questão. Os indivíduos da população são então criados e constituídos com o claro propósito de buscar uma solução ótima, ou próxima desta, através de um processo de evolução.

Os AGs são considerados melhores que técnicas de IA convencional em problemas de otimização e adaptação porque são mais robustos. Diferentemente de sistemas convencionais de IA, baseados em paradigmas simbólicos, eles não falham facilmente, mesmo se as entradas mudam ou na presença de ruídos. Então, em processos de busca em espaços abrangentes ou n-dimensionais, um AG pode oferecer benefícios significativos sobre muitas técnicas de busca de otimização tradicionais (MITCHELL, 1996) (GOLDBERG, 1989).

2.1.1 Características dos Algoritmos Genéticos

Embora "estocásticos", AGs não podem ser considerados, de forma alguma, processos de busca "randômicos" ou "aleatórios". Em vez disso, eles exploram informações históricas do processo de resolução do problema em questão, para direcionar a busca para a região de melhor desempenho dentro do espaço de busca. As técnicas básicas dos AGs são designadas para simular processos encontrados em sistemas naturais que são necessários para que ocorra um processo evolutivo, especialmente aqueles que seguem os princípios introduzidos por Charles Darwin de "sobrevivência da adaptação". Já que, na natureza, a competição entre indivíduos por recursos escassos resulta em indivíduos mais bem adaptados dominando sobre os indivíduos mais fracos, também em processos computacionais isto apresenta resultados satisfatórios.

Portanto, os AGs são baseados numa analogia com a estrutura genética natural e comportamento de cromossomos dentro de uma população de indivíduos, usando os seguintes fundamentos:

- indivíduos numa população competem constantemente por recursos e por parceiros para reprodução;
- aqueles indivíduos com mais sucesso em cada "competição" produzirão mais descendentes do que aqueles indivíduos que obtiveram um desempenho menos satisfatório;
- genes de bons indivíduos propagam-se mais facilmente através das consecutivas populações, porque dois bons pais têm maior probabilidade de produzir descendentes que serão ainda melhores;
- desta forma, cada sucessiva geração será mais adaptada ao seu ambiente.

Portanto, um AG pode ser descrito como um sistema adaptativo evolutivo. A evolução é caracterizada pela interação dos componentes do sistema (indivíduos) com o problema (meio ambiente) e sua posterior reprodução, transmitindo o resultado desta interação para novos indivíduos. Estes, por sua vez, estarão mais adaptados a este ambiente que seus antecessores, conduzindo o sistema em um processo de convergência em direção a um resultado esperado, ou seja, a adaptação dos indivíduos ao meio ambiente.

As principais características dos AGs, que os diferem de outros algoritmos de busca, são:

1. trabalham com a representação e codificação das variáveis e não com as variáveis em si;
2. realizam, a cada ciclo, uma busca sobre uma população de pontos e não somente sobre um único ponto no espaço de soluções;
3. utilizam funções de custo e não derivadas ou outro tipo de conhecimento auxiliar;
4. utilizam combinações de regras de transição probabilísticas e determinísticas.

Tanto no processo de seleção natural como nos AGs, existem conjuntos de características, também chamadas de fenótipo. Estas características são passadas de pais para filhos na seqüência das populações por hereditariedade através da reprodução, pela transmissão de segmentos do genótipo. À medida em que as gerações vão passando, os indivíduos vão continuamente se adaptando cada vez mais ao meio ambiente, demonstrando com isso o processo de evolução da espécie. Esta evolução se deve ao processo de seleção natural, que permite aos indivíduos mais aptos sobreviverem e reproduzirem mais que os outros indivíduos que não conseguem se adaptar. Conseqüentemente, os filhos herdam as boas características dos pais, tornando-os igualmente ou mesmo mais bem adaptados ao ambiente que seus pais.

Desta forma, como cada indivíduo representa uma solução no espaço de soluções, também o propósito do AG é realizar uma seleção das soluções existentes para produzir soluções ainda melhores que as anteriores, pela combinação da informação dos cromossomos destes. O AG mantém uma população de n cromossomos (soluções) com valores de ajuste (adaptabilidade) associados. Os indivíduos desta população são então selecionados para reproduzir, baseado numa estratégia de seleção, produzindo descendentes por uma estratégia de reprodução.

Indivíduos da população "morrem" e são substituídos pelos novos indivíduos, criando uma nova geração. Desta forma é esperado que através de sucessivas gerações, as melhores soluções prosperem e permaneçam; enquanto as soluções menos aptas devem desaparecer. Normalmente, novas gerações de soluções são produzidas contendo, em média, uma quantidade maior de bons genes que uma solução típica encontrada nas gerações anteriores.

Considerando que a população convergiu e não está mais ocorrendo a produção de descendentes notavelmente diferentes daqueles existentes nas gerações anteriores, deve ser acionado o mecanismo de condição de parada do algoritmo, para que este identifique que o AG convergiu para um conjunto de soluções no qual alguma seja satisfatória para o problema em questão.

2.1.2 Formas de Representação

Quando Holland retirou dos processos naturais de evolução as idéias para criar um AG, ele se deparou com um fato de grande importância: as soluções para qualquer problema precisam estar representadas e codificadas na forma de cromossomos, compreensíveis e manipuláveis computacionalmente. Para tanto, uma população de indivíduos é mantida num espaço de busca para um AG. Cada um destes indivíduos representa uma possível solução ou, dependendo da aplicação e modelagem do algoritmo, uma solução intermediária para o problema. Assim, cada indivíduo é representado por um cromossomo, codificado como um vetor finito de componentes, em termos de algum alfabeto. Usualmente é utilizado o alfabeto binário $\{0,1\}$ em razão da simplicidade de compreensão e implementação.

Utilizando-se o alfabeto binário, tem-se duas alternativas de representação: posicional, que implica que a posição de cada gene influencia na obtenção do fenótipo, tal como a conversão de um valor binário para decimal; e não-posicional, onde a localização de cada um dos genes no cromossomo não é relevante, sendo possível obter-se o mesmo fenótipo a partir de diversos genótipos distintos.

Existem ainda outras formas de representação, utilizando outros alfabetos e também vetores com valores reais, que aproximam os AGs das Estratégias Evolutivas (FOGEL, 2000).

2.1.3 Avaliação dos Indivíduos

Um grau de adaptabilidade ao problema, denominado no contexto de AGs como "ajuste" deve ser atribuído a cada solução existente, representando a habilidade deste indivíduo para "sobreviver" e "competir" no ambiente em que está inserido. Quanto maior for o valor deste ajuste para um indivíduo, mais perto ele estará da solução do problema. Desta forma, o indivíduo que, na sucessão de gerações, alcançar um grau de ajuste ótimo (ou, geralmente, um grau satisfatório, próximo do ótimo) é então escolhido, para que a solução nele codificada possa ser então aplicada para solucionar o problema.

Este grau de ajuste é determinado por uma função de avaliação ou por um julgamento subjetivo. Esta função de avaliação é denominada "função de ajuste", que é usada para testar cada elemento da população, verificando o quão apto está para solucionar ou não o problema. Isso representa avaliar o quanto este indivíduo está adaptado ao meio ambiente para saber se ele tem condições de competir, sobreviver e gerar descendentes. O resultado desta avaliação é a atribuição do grau de ajuste a cada um dos indivíduos.

Em resumo, a função de avaliação pode ser considerada como um elo de ligação entre o AG e o problema a ser resolvido, pois ela retorna um valor ou uma lista de valores que apresentará o desempenho do indivíduo avaliado, frente ao problema (MITCHELL, 1996).

2.1.4 Operadores Genéticos

Um algoritmo genético padrão evolui, em suas sucessivas gerações, através de três operadores básicos: Seleção, Cruzamento e Mutação. (GOLDBERG, 1989)

2.1.4.1 Operador de Seleção

A idéia principal do operador de seleção é oferecer aos melhores indivíduos da população, preferência para o processo de reprodução. Sabendo que os descendentes herdam as características de seus progenitores, pode-se dizer que é naturalmente necessário que as melhores características existentes em uma população devam ter maiores possibilidades de reproduzir e ser preservadas no ambiente.

Uma das técnicas de seleção mais conhecida e amplamente utilizada é a "roleta". Existem várias adaptações, a partir da proposta original apresentada por Goldberg. (GOLDBERG, 1989) (DAVIS, 1991), que descreve seu funcionamento, inspirado nas roletas utilizadas em jogos de cassino.

Cada indivíduo da população tem um espaço na roleta, calculado proporcionalmente ao seu grau de ajuste ao problema. Desta forma, os indivíduos mais aptos terão uma área maior nesta roleta, tendo maiores chances de serem "sorteados" para o processo de reprodução. Os sorteios são realizados e os pares de reprodutores vão sendo selecionados. Esta ordem de seleção pode ser efetuada pela simples seqüência dos sorteios.

2.1.4.2 Operador de Cruzamento

O processo de cruzamento é o primeiro fator de distinção do AG em relação a outras técnicas de otimização. Trata-se de uma operação de cruzamento e recombinação entre os cromossomos (MITCHELL, 1996) (DAVIS, 1991). A idéia deste operador é permitir que, no processo de reprodução, os pares de indivíduos sele-

cionados gerem descendentes que recebem características de ambos os progenitores. Explicando a seqüência dos passos que envolvem o processo, pode-se dizer que:

- dois indivíduos da população são escolhidos para o processo de reprodução, utilizando-se o operador de seleção;
- é escolhido aleatoriamente um "ponto de corte" na extensão dos cromossomos, separando o cromossomo de cada progenitor em dois segmentos;
- dois novos indivíduos são criados a partir da recombinação destes segmentos: um indivíduo recebe o primeiro segmento do primeiro progenitor concatenado com o segundo segmento do segundo progenitor; e o outro indivíduo recebe o primeiro segmento do segundo progenitor concatenado com o segundo segmento do primeiro progenitor.

Existem ainda outras alternativas para o cruzamento, como utilizar dois ou mais pontos de corte. Ou ainda, o cruzamento uniforme, onde, para cada uma das posições dos cromossomos dos progenitores é realizado um sorteio para designar qual descendente recebe o gene de qual progenitor (GOLDBERG, 1989) (MITCHELL, 1996).

2.1.4.3 Operador de Mutação

Nos processos naturais, os indivíduos podem sofrer modificações em suas características, para que possam adaptar-se ainda melhor ao ambiente em que estão inseridos. Da mesma forma, pode ser utilizado nos AGs um operador de mutação que faz com que, com muito baixa probabilidade, uma porção dos novos indivíduos terá alguns de seus genes invertidos, cujo propósito é manter diversidade dentro da população e inibir convergências prematuras.

A mutação é realizada aplicando-se uma probabilidade de alteração para cada gene do cromossomo. Realizado o sorteio, a mutação pode ocorrer de duas formas: a primeira executa a inversão do valor binário do gene em questão, e a outra, realiza outro sorteio, para definir qual o valor a ser atribuído ao gene (que poderá, desta forma, permanecer inalterado).

2.1.5 Efeitos dos Operadores Genéticos

Normalmente, uma implementação de AG faz uso dos três operadores genéticos. O modelador do sistema tem a responsabilidade de definir a intensidade com que vai utilizar cada um. Seu uso deve ser balanceado, porque a predominância de um ou outro pode causar um declínio no desempenho do algoritmo. Devem ser observados alguns aspectos relevantes:

- o uso desnecessário de seleção tenderá a encher a população com cópias do melhor indivíduo da população;
- o uso de operadores de seleção e crossover (sem mutação) tenderá a causar que o algoritmo convirja a uma boa mas sub-ótima solução, que pode não ser tolerante a modificações significativas no ambiente;
- o uso de somente o operador de mutação induz um caminho aleatório através do espaço de busca;

2.1.6 Inserção dos Novos Indivíduos na População

Após ter gerado os novos indivíduos, através do processo de reprodução e aplicação dos operadores genéticos, é necessário que estes sejam agregados à população, ou seja, inseridos no ambiente para também competir e reproduzir.

Existem várias formas para executar esta tarefa. Em primeiro lugar, a população pode ser definida com tamanho fixo ou variável. Normalmente, as implementações citadas na literatura são modeladas com população de tamanho fixo, baseadas no AGS, onde é realizado um processo de substituição de gerações (?). Assim, segue uma breve descrição do funcionamento desta técnica.

Utilizar a técnica de substituição de gerações significa que, como pais produzem descendentes e deve ser feito espaço para estes; e como a população mantém um tamanho estático, os progenitores devem desaparecer (fazendo uma analogia com processos naturais, devem "morrer") para fornecer o espaço necessário para os novos indivíduos.

Existem outras variações para efetuar a inserção dos descendentes gerados nas próximas gerações, dependendo da modelagem do algoritmo genético. Um exemplo conhecido recomenda avaliar previamente o grau de ajuste de cada novo indivíduo antes deste ser agregado à nova população. Isso garante ao algoritmo genético que, a cada nova geração, não haja indivíduos menos aptos ao ambiente do que em gerações anteriores.

2.1.6.1 *Elitismo*

O conceito de elitismo foi inicialmente introduzido por De Jong (JONG, 1975) (JONG, 1988a), e representa uma adição aos métodos de seleção que força o AG a reter uma parcela dos melhores indivíduos existentes em cada geração, considerada a "elite" da população corrente. Desta forma, é possível eliminar-se o risco de que estes indivíduos - os melhores adaptados até o momento - ou suas características pudessem ser tragicamente perdidos no caso de que eles não fossem selecionados para o processo reprodutivo ou se as características neles contidas fossem destruídas pelos processo de crossover ou de mutação. A literatura sobre o assunto mostra que muitos pesquisadores tem achado que elitismo incrementa o desempenho do algoritmo genético significativamente (MITCHELL, 1996) (DAVIS, 1991).

2.1.7 O Algoritmo Genético Simples de Holland

Basicamente, o AG pioneiro criado por Holland possuía uma população de indivíduos de tamanho fixo e cada um destes indivíduos possuía somente um único cromossomo. Ainda hoje é amplamente utilizada esta modelagem básica. No entanto deve-se salientar que existem diversas variações entre os modelos. Tais variações são encontradas, normalmente, na forma de codificação dos cromossomos e na obtenção do grau de ajuste, não modificando o restante do algoritmo.

A Figura 2.1 apresenta as etapas de um típico AG, onde a população inicial é criada e posteriormente avaliada; se já existe uma solução razoável, o algoritmo é encerrado. Caso contrário, inicia-se um processo onde, a cada ciclo, são aplicados os operadores genéticos, o que resulta em novos indivíduos, os quais substituem os antigos na população e passam pela etapa de avaliação. Assim sucessivamente, até que venha a surgir um indivíduo que apresente uma solução considerada satisfatória. (HOLLAND, 1975) (GOLDBERG, 1989) (DAVIS, 1991) (MITCHELL, 1996).

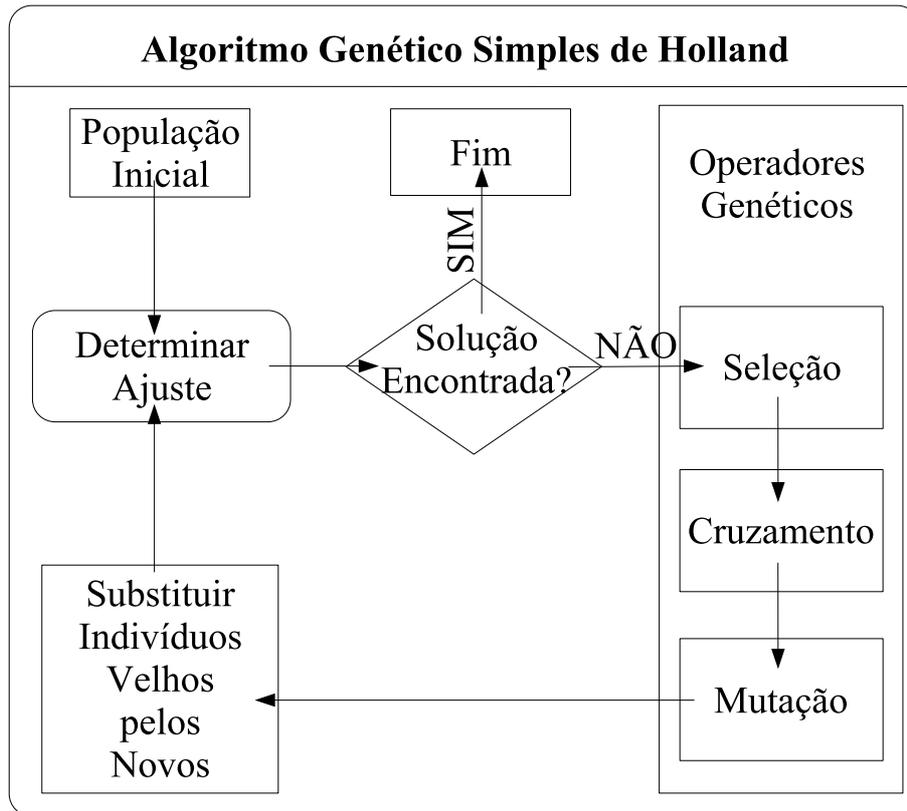


Figura 2.1: Algoritmo Genético Simples (AGS) de Holland.

2.1.8 Sistemas Híbridos

Um AG pode ser combinado com técnicas convencionais de solução de problemas formando uma solução híbrida (DAVIS, 1991). A hibridização permite a incorporação de heurísticas otimizadoras ao conjunto de operadores genéticos que passam portanto a ser dependentes do domínio. Nesse sentido, o AG passa a ser muito mais um paradigma de otimização do que um método pronto para utilização.

2.2 Algoritmos Genéticos com Idades (AGI)

A definição do tamanho da população é uma das mais importantes escolhas enfrentadas por todos os usuários de AGs, pois, como já foi visto anteriormente, esse fator está diretamente relacionado com o seu desempenho. Ou seja, se o tamanho da população for muito pequeno, ela pode ter problemas de convergência prematura e, se for muito grande, pode-se tornar não utilizável na prática por dispendar muitos recursos computacionais.

Já foi visto também como a pressão seletiva e a diversidade da população são importantes no processo de evolução dos AGs. Evidentemente, esses dois fatores são influenciados pelo tamanho da população. Muitas pesquisas têm sido desenvolvidas visando a investigação do tamanho da população sob diferentes perspectivas, entre as quais, o Algoritmo Genético com Idades (AGI) (MICHALEWICZ, 1996).

2.2.1 Conceituação

O AGI é uma proposta de implementação de um AG, onde é inserido um fator o qual, em termos biológicos, seria de importância incontestável: todos os organismos naturais nascem, se reproduzem e cedo ou tarde morrem. Desta forma é então introduzido o conceito de IDADE nos AGs e, de uma forma bastante simples, pois, no momento da criação é atribuído a cada indivíduo um tempo, em gerações, para sua permanência na população. Esse tempo é resultado de algum cálculo proporcional à sua aptidão ao problema, para que, quanto mais adaptado, mais tempo esse indivíduo permaneça na população e, conseqüentemente, maior a sua probabilidade de participar de processos reprodutivos e, desta forma gerar descendentes que perpetuem suas características.

Além do atributo de idade, existe também um contador de idade para cada indivíduo. Ao final de cada geração do AE, este contador é incrementado. Quando o contador atinge o valor estabelecido na idade, significa que este indivíduo chegou ao final de sua vida e que, portanto, será eliminado da população.

Visto que no AGI um indivíduo só será eliminado da população após ter permanecido na população durante seu tempo de vida, é evidente que o tamanho desta população é variável, ao contrário do AG convencional, no qual é normalmente estipulada uma população de tamanho fixo. Desta forma, este algoritmo propõe dispensar o uso dos mecanismos de seleção vistos anteriormente pois a idade do cromossomo substitui o conceito de seleção e, desde que ela depende da aptidão do indivíduo, influencia no tamanho da população em todo o processo (YEPES, 1999).

2.2.2 Vantagens em Relação ao Algoritmo Genético Convencional

A introdução da idade nos indivíduos componentes da população nos traz duas grandes vantagens (MICHALEWICZ, 1996):

- controle da pressão seletiva;
- tamanho variável da população.

O AGI proposto por Michalewicz (MICHALEWICZ, 1996) controla a pressão seletiva, tornando supérflua, parte da fase de seleção de progenitores, evitando assim os inconvenientes a ela ligados. Por exemplo, no caso de seleção feita por roleta, essa etapa seria desnecessária, pois a seleção seria feita levando em conta toda a população e não apenas uma pequena amostra, já que o mecanismo favorecedor dos organismos mais aptos será a própria idade (os mais aptos têm maior tempo de permanência e, conseqüentemente, estando mais tempo na população, têm mais chances de reproduzir do que os menos aptos).

Como na fase de substituição o AGI incorpora todos os novos indivíduos gerados na população e em vez de ocorrer substituição, há simplesmente a eliminação dos indivíduos "velhos", tornando-se o tamanho da população variável, o que traz mais vantagens do que inconvenientes, pois a idade funciona como um mecanismo auto-regulador do tamanho da população. Tem como grande vantagem o fato de manter um bom nível de diversidade na população, pois os indivíduos menos aptos (em determinado instante) não serão eliminados tão drasticamente como no AG tradicional, e sim, com já foi dito, ao extrapolarem seu limite de vida.

Ao ser executado um AGI, a evolução do tamanho da população durante o processo apresenta uma forma oscilante bastante peculiar, a qual pode ser explicada da seguinte forma (SERRADA, 1996):

"No início, quando a variedade de aptidões é relativamente alta, o tamanho da população cresce. Isto quer dizer que o AGI realiza uma busca pelo ótimo em amplitude. Uma vez que tenha sido localizada a vizinhança do ótimo, o algoritmo começa a convergir e o tamanho da população se reduz. Sem dúvida, ainda continua buscando uma melhoria. Sempre que se apresente uma ocasião para obter melhor resultado, outra 'explosão demográfica' tem lugar, a qual é seguida por outra etapa de convergência. Ao que parece, os AGIs incorporam um processo de auto-ajuste adaptando o tamanho da população a cada etapa do processo de evolução."

Portanto, o AGI difere do AG tradicional unicamente nos componentes referentes à fase de seleção/substituição, sendo, assim, amplamente maleável, aceitando diversas alternativas de variações e implementações sobre o seu conceito.

2.2.3 Atribuição da Idade

Normalmente, o parâmetro de idade é dado a cada indivíduo durante a fase de avaliação e permanece constante através de todo o processo de evolução. Ou seja, ao ser criado um indivíduo, seja na fase de criação da população inicial, seja na fase de cruzamento da população auxiliar, ele é avaliado e, de acordo com a sua aptidão, é atribuído este tempo máximo para sua vida, o qual será fixo, não variando no decorrer das etapas evolutivas. Em outras palavras, a idade do cromossomo determina o número de gerações durante as quais ele fará parte da população.

No entanto, existem muitas outras estratégias possíveis de atribuição da idade e, logicamente, se for atribuído a cada novo indivíduo gerado em um processo reprodutivo, o mesmo valor de duração, constante maior que 1, independente de todas as estatísticas da pesquisa, poderia causar um crescimento exponencial do tamanho da população. Além disso, como a participação na população estaria uniformemente distribuída, não haveria pressão seletiva nenhuma, o que acarretaria uma baixa performance do algoritmo. De forma a introduzir pressão seletiva, deve ser desenvolvida uma forma mais sofisticada de calcular a idade. Então, as formas de cálculo de idade, segundo Michalewicz (MICHALEWICZ, 1996) devem:

- Reforçar a presença dos indivíduos mais aptos e, conseqüentemente, restringir a presença de indivíduos com menos aptidão;
- controlar o tamanho da população de forma a mantê-lo bem adequado, particularmente evitando o crescimento exponencial da população e desperdício de esforço computacional;

Como já foi dito anteriormente, a idade de um indivíduo está diretamente ligada à sua aptidão e, conseqüentemente, a probabilidade do tamanho de sua descendência será diretamente proporcional à sua permanência na população. Por isso, o primeiro critério citado acima força a atribuição de um tempo de vida maior para os indivíduos com maior aptidão pois, assim terão maiores chances de reproduzir e gerar mais descendentes. Note-se que não é conveniente atribuir aos indivíduos uma duração estritamente proporcional a sua aptidão, pois isso pode ir contra o segundo critério.

A prática tem mostrado que, para os cálculos sobre a duração de um indivíduo, deve-se considerar, além da aptidão, o estado presente da busca (aptidão média da população, aptidão do melhor indivíduo, tamanho atual da população, etc...).

Deve-se observar que o cálculo da idade deve ser computacionalmente fácil de forma a evitar desperdício de recursos. Tendo em mente as observações acima, Michalewicz (1996) destaca três formas de cálculo de tempo de vida, dentre as quais, as quais são: Distribuição proporcional, Distribuição linear e Distribuição bilinear.

A primeira estratégia traz a idéia de seleção por roleta: o valor do tempo de vida para um determinado indivíduo é proporcional ao valor da sua aptidão. Contudo, esta estratégia apresenta sérias desvantagens, visto que não utiliza nenhuma informação sobre a "aptidão objetivo" do indivíduo a qual pode ser estimada relacionando a aptidão do indivíduo com a melhor aptidão encontrada até o momento.

Isto motivou a criação da estratégia linear, na qual o valor do tempo de vida é calculado de acordo com a aptidão individual em relação à melhor aptidão encontrada até o presente momento no processo. Entretanto, em muitos casos, um grupo de indivíduos pode ter valores de aptidão iguais ou aproximadamente iguais ao melhor valor encontrado e, com esta estratégia são atribuídos altos valores de tempo de vida a todos eles, aumentando perigosamente o tamanho da população e o esforço computacional exigido.

Finalmente, a distribuição bilinear tenta criar uma junção entre as duas primeiras. Ela acentua as diferenças entre média dos valores de aptidão, levando em consideração também os valores máximo e mínimo de aptidão encontrados durante todo o processo.

Em Michalewicz(1996) é realizada uma comparação do AG clássico em relação ao AGI. Nas experiências realizadas com otimização numérica, demonstrou-se que a distribuição linear é a que apresenta maior precisão, sendo ,em contrapartida, a que mais exige esforço computacional. A que exigiu menos esforço foi a bilinear, mas a sua precisão deixou a desejar. Já a distribuição proporcional apresentou um elo forte entre esforço computacional e precisão. Nos testes, Michalewicz tomou como parâmetros de medida o esforço computacional exigido pelo algoritmo e o desempenho apresentado, e usou como critério de parada o fato de não ocorrer melhora na procura do melhor valor num intervalo de 20 gerações.

O conhecimento sobre os parâmetros de seleção dos AGs está ainda bastante fragmentado. Entre esses parâmetros, o tamanho da população é visto como o mais importante, pois ele tem forte influência sobre o esforço computacional exigido por um AG. Pode ser que a melhor opção para este parâmetro seja torná-lo dinâmico, ou seja, que o seu tamanho varie de acordo com a necessidade atual do AG. Esta é a idéia central do AGI, o qual, em diferentes fases do processo de busca, trabalha com tamanhos diferentes de população, procurando manter sempre um bom conjunto de possíveis soluções disponível, mas sem exigir demais em termos de esforço computacional (MICHALEWICZ, 1996).

2.3 Sistemas Classificadores

A categoria de AE denominada Sistemas Classificadores foi originalmente proposta por Holland Holland(1975) , com o objetivo de criar um sistema de aprendizagem baseado na AE. Eles surgiram da combinação dos AGs com uma técnica amplamente utilizada na época, e ainda presente em nossos dias: os Sistemas Espe-

cialistas (SE)(BULL, 2004). A idéia era justamente eliminar o problema existente nos SE, a geração do conhecimento, que deve ser realizada manualmente por um especialista, passando a ser executada automaticamente pelo AE.

Resumidamente, um SC é uma técnica para criação e atualização evolucionária de conhecimento, representado por regras - também chamadas de classificadores - em um sistema que codifica alternativas de ações a serem executadas em um ambiente, quando este apresenta determinadas características. Exemplos de aplicações para essa metodologia são amplamente encontrados na literatura, como a minimização da perda em redes de distribuição de energia elétrica (VARGAS; LYRA; ZUBEN, 2004), entre outros (KOVACS; LANZI, 1999).

2.3.1 Funcionamento e Caracterização

Um SC realiza uma interação com o ambiente através de sensores e atuadores, que são responsáveis por um processo de troca de mensagens. As mensagens provenientes do ambiente, captadas pelos sensores do SC retratam o estado corrente do ambiente e, as mensagens do SC acontecem na forma de ações a serem aplicadas sobre o ambiente. As consequências geradas pelas ações do SC, detectadas pelos sensores numa etapa de retroalimentação, determinam a "recompensa" ou "punição" adequada ao classificador responsável pela ação. A Figura 2.2 apresenta os componentes que formam um SC e as relações existentes entre eles.

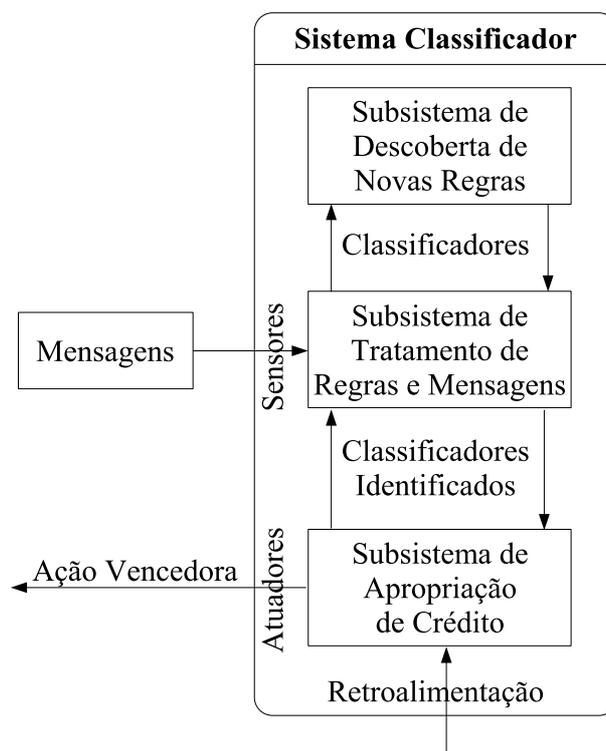


Figura 2.2: Estrutura Geral de um Sistema Classificador

O aprendizado baseado em SC possui duas abordagens distintas, na primeira, denominada abordagem "Pittsburgh"¹, um único indivíduo representa a solução

¹Esta abordagem é denominada desta forma, por ter sido desenvolvida na Universidade de Pittsburgh.

para o problema (MICHALEWICZ, 1996) (JONG, 1988b), enquanto na segunda, chamada abordagem "Michigan"², a solução é gerada pela configuração de toda a população. Esta abordagem surgiu pela necessidade de se trabalhar com cromossomos de tamanho fixo (MICHALEWICZ, 1996). Como o trabalho desenvolvido nesta tese utiliza os fundamentos desta abordagem, é apresentada uma breve descrição de suas características e componentes.

2.3.2 Regras

Cada regra é representada por um indivíduo da população e serve para executar uma ação vinculada à ocorrência de uma situação específica do ambiente, de modo que o desempenho do SC será sempre determinado pela atuação de toda a população. A forma de representação destas regras é uma composição antecedente-conseqüente, tal como regras de produção, seguindo o modelo:

$$SE \langle \textit{condição} \rangle \textit{ENTÃO} \langle \textit{ação} \rangle$$

O antecedente é um vetor de tamanho fixo, formado utilizando-se o alfabeto ternário {"0", "1" e "#"}, representando uma situação possível do ambiente; e o conseqüente também é um vetor de tamanho fixo, normalmente codificado com o alfabeto binário {"0", "1"}, representa a ação a ser executada em tal situação. O caractere "#" utilizado na parte antecedente é conhecido como o símbolo "não importa". Sua função é atuar como um "curinga", podendo assumir o valor "1" ou "0", durante a fase de comparação.

Nos SC, cada regra possui um atributo de "energia", utilizada para expressar sua força durante o processo evolutivo.

A comparação da parte antecedente das regras com a mensagem caracterizando a situação corrente do ambiente é que definirá quais regras irão competir ou não. A competição é uma forma de avaliação das regras, sendo que as melhores adaptadas disputarão pelo direito de atuar sobre o ambiente. Para tanto, existe o conceito de "especificidade" de cada regra, que é a medida inversamente proporcional à quantidade de símbolos "#" (não importa) na parte antecedente do classificador.

Os SC são constituídos por três subsistemas: o Subsistema de Tratamento de Regras e Mensagens, o Subsistema de Apropriação de Crédito e o Subsistema de Descoberta de Novas Regras, descritos na sequência:

2.3.3 Subsistema de Tratamento de Regras e Mensagens

Os sensores recebem as mensagens do ambiente e enviam a mesma para o Subsistema de Tratamento de Regras e Mensagens, onde é codificada para um formato compreensível pelo SC, tal como apresenta a Figura 2.3. Assim, inicia um processo de identificação e comparação desta mensagem com a parte antecedente de todas as regras existentes na população.

Este processo de identificação pode ser feito pela comparação de cada elemento dos vetores, de acordo com regras especificadas pelo projetista do sistema, ou apenas pelo cálculo de uma variante da Distância de Hamming. Segundo Booker (BOOKER, 1985), este cálculo permite que indivíduos mais específicos em cromossomos muito longos tenham chance de competir com indivíduos menos específicos.

²Denominação empregada devido ao trabalho de Holland na Universidade de Michigan

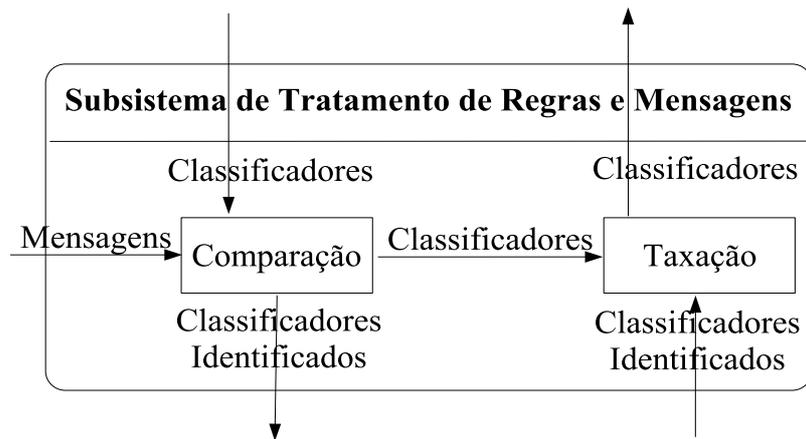


Figura 2.3: Subsistema de Tratamento de Regras e Mensagens

Fazendo a comparação explícita, cada posição da mensagem é comparada com os valores, exatamente da mesma posição, diferentes de "#" (não importa) da regra.

Aquelas regras que possuem todos os valores não conflitantes com as da mensagem (mesmo valor ou "#") são identificadas para a fase da "competição", sendo enviados ao Subsistema de Apropriação de Crédito. Caso não existam regras totalmente identificadas com a mensagem do ambiente, deve existir alguma estratégia que trate o índice de especificidade, a fim de eleger a ação que deverá ser executada pelo sistema.

A cada iteração do sistema, todas as regras, incluindo aquelas selecionadas para a competição, sofrerão um decréscimo em sua energia, correspondente à uma taxa de vida.

2.3.4 Subsistema de Apropriação de Créditos

No subsistema de Apropriação de Créditos, todas as regras identificadas com a mensagem do ambiente participam de uma "competição" ou "leilão", onde o vencedor é a regra que apresentar maior "lance" efetivo. O lance é calculado em função da energia e da especificidade de cada regra. A regra vencedora paga seu lance, descontando de sua energia a quantia ofertada e terá o direito de atuar sobre o ambiente. Complementarmente, uma nova taxa de participação é cobrada de cada indivíduo que participou da "competição", inclusive do vencedor. A Figura 2.4 mostra o funcionamento deste subsistema.

Uma vez definida a regra vencedora, que será responsável pela ação sobre o ambiente, uma nova mensagem é fornecida pelo próprio ambiente, caracterizando o seu novo estado, o qual é consequência da ação indicada pela regra vencedora. Este é o ponto de partida para se definir o valor da recompensa ou punição que esta regra receberá.

Por fim, o processo volta novamente ao tratamento de mensagens e apropriação de créditos, gerando um ciclo durante uma "época" de iterações. Ao final de cada época, o SC participará de outra fase de sua evolução, a fase de descoberta de novas regras.

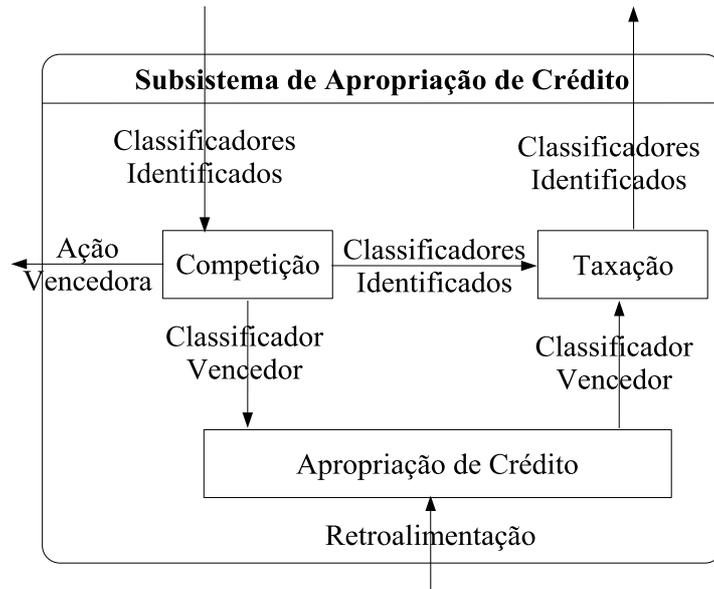


Figura 2.4: Subsistema de Apropriação de Crédito

2.3.5 Subsistema de Descoberta de Novas Regras

Concluindo um período de iterações, recebendo mensagens do ambiente e executando as respectivas ações, o SC realizou um devido ajuste na energia dos classificadores. Assim, como indica a Figura 2.5, na fase de Descoberta de Novas Regras, aplica-se os procedimentos intrínsecos aos AGs na população de regras corrente, fazendo uso de todas as vantagens desta técnica na busca de uma população de regras cada vez mais adaptada ao ambiente, de acordo com os objetivos a serem alcançados.

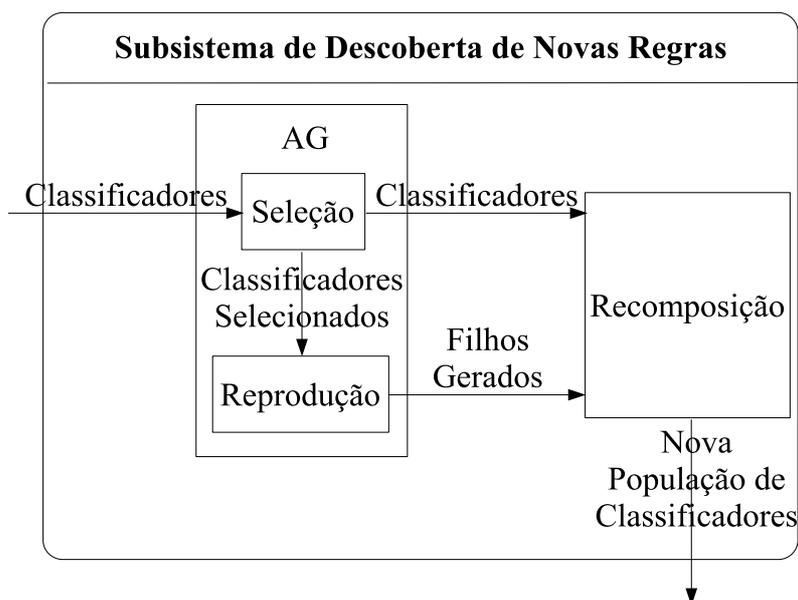


Figura 2.5: Subsistema de Descoberta de Novas Regras

Desta forma, cada regra é tratada como um indivíduo, ou seja, um "cromossomo" e, basicamente, o AG produz uma nova geração de regras pela aplicação dos operadores de seleção, cruzamento e mutação. As estratégias utilizadas serão as mesmas disponíveis para qualquer outra aplicação envolvendo AG's, inclusive, vale ressaltar que também aqui, normalmente, a população inicial é criada aleatoriamente e a única característica própria é a medida de adequação de cada indivíduo, a qual, em SC, é representada pela energia, que é inicializada em todos os classificadores com o mesmo valor.

Considerando que o valor da energia está diretamente associado à adequação do classificador ao ambiente, com o qual ele interage e, a ação indicada pelo classificador vencedor pode ir contra ou a favor dos objetivos, o classificador poderá diminuir ou aumentar sua energia ao longo das iterações. Através desta metodologia de medição, é possível avaliar cada um dos classificadores durante o processo evolutivo.

2.3.6 Algoritmo Simplificado

A Figura 2.6 apresenta uma visão global de um SC, com seus mecanismos e componentes:

1. Gerar População Inicial Regras
2. Para uma Época de Iterações
 - Receber mensagem do ambiente
 - Iniciar processo de Competição
 - Comparar todos os classificadores com a mensagem
 - Ativar os classificadores selecionados
 - Definir os vencedores por leilão
 - Cobrar taxas de energia de todos os Classificadores
 - Executar ação do classificador vencedor no ambiente
 - Cobrar taxa de energia do vencedor
 - Receber mensagem do ambiente e detectar realimentação
 - Recompensar ou punir classificador atuante
3. Executar o Algoritmo Genético
 - Selecionar pares para a reprodução
 - Cruzar os pares e gerar novos indivíduos
 - Aplicar mutação nos novos indivíduos
 - Recompor a população inserindo os novos classificadores
4. Se o critério de parada não foi atendido, ir para (2)

Figura 2.6: Algoritmo Simplificado de Sistemas Classificadores.

3 ROBÓTICA EVOLUCIONÁRIA

Evolução e aprendizagem são conceitos básicos que geram o que conhecemos como "Comportamento Adaptativo", cada qual em uma dimensão temporal diferente. Evolução diz respeito à adaptação através de sucessivas gerações e aprendizagem é a adaptação de um indivíduo durante a sua existência (NOLFI; FLOREANO, 1999). Nos últimos anos, biólogos, psicólogos, filósofos e neurologistas não estão mais sozinhos focando atenção para estes tópicos, mas estão sendo acompanhados também por cientistas que desenvolvem trabalhos em outras áreas, tais como engenharia, computação e robótica. Seu interesse está em descobrir e implementar formas de mimetizar o comportamento adaptativo observado nos seres vivos em seus projetos. Assim, construindo dispositivos robóticos e sistemas controladores capazes de desenvolver e apresentar comportamento adaptativo, estes pesquisadores criaram um novo campo de pesquisa, conhecido como "Robótica Evolucionária" (RE).

O termo "Robótica Evolucionária" foi apresentado inicialmente por Cliff, Harvey e Husbands (CLIFF; HARVEY; HUSBANDS, 1992) e Floreano e Mondada (FLOREANO; MONDADA, 1994) em seus trabalhos que simulavam populações de robôs evoluindo suas habilidades. O objetivo da RE é desenvolver controles para robôs através de evolução artificial. (CLIFF; HARVEY; HUSBANDS, 1993) (NOLFI et al., 1994)

A RE surgiu como uma proposta de metodologia de projeto, ou seja, uma forma de automatizar o projeto de sistemas de controle para agentes autônomos, a partir de processos auto-organizados que utilizam algoritmos inspirados na evolução natural (MEYER; HUSBANDS; HARVEY, 1998) (NOLFI; FLOREANO, 2000). Com o passar do tempo, vários dos trabalhos desenvolvidos passaram a abordar a investigação e testes de hipóteses relacionadas à filogenia do comportamento em sistemas naturais (TUCI, 2004) (QUINN, 2001). Embora produzindo robôs com habilidades para executar certas tarefas pré-determinadas, os pesquisadores em RE conseguem realizar evoluções com componentes de baixo-nível e examinar os mecanismos e as condições para que, sob as quais, uma certa condição consiga evoluir. Além disso, considerando a construção destes componentes de baixo-nível, o comportamento esperado é que o sistema evolua gradualmente, permitindo uma potencial avaliação dos estágios intermediários, através de como o agente consegue adaptar-se ao ambiente. (TUCI, 2004)

Pode-se dizer que a RE, no que tange a várias outras novas técnicas biologicamente inspiradas, representa uma revolução conceitual nas engenharias e ciência da computação semelhante à uma revolução na ciência. Tanto que atualmente já está sendo apresentada uma ainda mais nova denominação, originária deste movimento: a "Nova Robótica" (PFEIFER; IIDA; BONGARD, 2004), considerada uma

abordagem na qual os princípios partem da inspiração biológica.

Mesmo sendo conceitos distintos e apresentando suas particularidades, evolução e aprendizagem, quando aplicados cooperativamente, propiciam um incremento significativo no desempenho de sistemas computacionais. Segundo Nolfi (NOLFI; FLOREANO, 1999), sob uma perspectiva evolucionária, a aprendizagem tem várias funções, entre elas:

- permitir adaptar às mudanças do ambiente que ocorrem durante a vida de um indivíduo ou através de poucas gerações;
- permitir que a evolução utilize a informação extraída do ambiente pela canalização da busca evolucionária;
- poder ajudar e guiar a evolução.

Entretanto, também existem custos, tais como:

- o atraso na habilidade de aprimorar o grau de adaptabilidade;
- incremento irreal.

Este capítulo apresenta uma breve descrição da área, metodologias de implementação e uma relação de alguns trabalhos relevantes já desenvolvidos.

3.1 Caracterização

Uma das características mais importantes desta abordagem é o fato de ela ser um processo completamente automático, no qual a participação do projetista do robô ou do sistema de controle é limitada praticamente à especificação de um critério para avaliar até que ponto a evolução dos indivíduos consegue desempenho satisfatório ao realizar a tarefa desejada. Ou seja, especificar o "quão boa" pode ser considerada uma solução representada por um determinado indivíduo da população. Uma grande dificuldade a ser superada é o fato de que a probabilidade de um indivíduo pertencente às gerações iniciais poder realizar a tarefa desejada, pelo menos em parte, é inversamente proporcional à complexidade da tarefa proposta. Isto faz com que o sistema possa, muito provavelmente, iniciar sua operação apresentando um desempenho bastante deficiente. (FLOREANO, 1997)

Assim, pode-se afirmar que as principais questões normalmente abordadas pela RE são (SIMÕES, 2003):

1. sintetizar automaticamente comportamentos mais complexos do que aqueles que podem ser produzidos manualmente (BULLOCK, 1995)
2. explorar amplamente as características do ambiente e dos indivíduos, mesmo que algumas delas sejam obscuras ao projetista (POLLACK et al., 2000)
3. produzir o comportamento esperado especificando-se o que o robô deve fazer, e não como ele deve operar (THOMPSON, 1996)
4. mostrar que técnicas evolucionárias podem reduzir o esforço humano necessário para produzir um sistema de controle, em comparação com métodos manuais de projeto (THOMPSON, 1997)

E, para alcançar estes objetivos, como já foi dito anteriormente, a RE trabalha com técnicas de aprendizagem e evolução, a fim de gerar comportamento adaptativo. As mais conhecidas técnicas de Inteligência Computacional que representam estes dois conceitos são, respectivamente, RNA e AG.

3.2 Agentes Robóticos Evolucionários

Na RE, assim como em várias áreas relacionadas com a IA, existem diversos objetivos diferentes. Enquanto alguns pesquisadores estão voltados principalmente à construção de máquinas que "executem bem" determinadas tarefas, outros estão preocupados em descobrir "como" estas tarefas conseguem ser executadas pelos robôs, num processo evolucionário. (MEYER; HUSBANDS; HARVEY, 1998)

Portanto, com estas alternativas, pode-se concluir que a robótica representa um domínio de aplicação interessante para a computação evolucionária, porque, tradicionalmente, a otimização de parâmetros de robôs frequentemente implica em realizar buscas em espaços multi-dimensionais, caracterizados por relações não-lineares e não-diferenciáveis entre variáveis, o que dificulta a obtenção de soluções analíticas (FLOREANO, 1997). Com esta nova abordagem, os robôs deverão ter capacidade e autonomia para que se auto-configurem seus parâmetros, de forma a otimizar seu desempenho na execução das tarefas a que foram destinados.

Mas, apesar de se apresentar como um poderoso método, ainda envolve muito tempo de treinamento para que os cromossomos artificiais possam codificar potencialmente e evoluir vários aspectos do robô, tais como controle de arquitetura, aspectos morfológicos, desenvolvimento de regras, o aprendizado e formas de aplicação destas regras.

3.2.1 Simuladores e Robôs Reais

Uma preocupação constante e considerada de importância fundamental em estudos em RE é se o processo de evolução do sistema de controle, é desenvolvido em ambientes simulados ou em robôs reais.

É evidente que ao se utilizar um processo de simulação, por mais que sejam inseridos ruídos e algum comportamento aleatório ao ambiente, este jamais se apresentará tal como o mundo real. Se o pesquisador recorre à simulação, o perigo é que a evolução pode explorar generalizações e abstrações da simulação e assim o sistema não poderá ser satisfatoriamente transferido para o mundo real ou, se for, poderá não levar total vantagem na dinâmica específica e interações do mundo real.

Assim, pode-se dizer que os testes realizados em simulação fazem com que se limite consideravelmente a análise do desempenho do robô e a perspectiva de funcionamento no mundo real.

Portanto, os testes em RE deveriam todos ser executados com robôs reais, em seu ambiente operacional. Isto se justifica, porque um processo computacional evolucionário, frequentemente produz soluções específicas que combinam, de forma otimizada, características do mundo físico onde o organismo evoluiu, da mesma maneira que acontece na natureza.

Infelizmente, embora a maioria das pessoas que desenvolvem trabalhos neste campo concorde que os resultados obtidos através de processos simulados devem ser considerados com certo cuidado, de fato, somente alguns pesquisadores executam suas experiências evolucionárias completamente em robôs físicos. Isto se deve a

vários fatores, tais como a inexistência de robôs com características adequadas às técnicas evolucionárias ou custo e risco de operação de um robô real em ambientes inóspitos.

Outro ponto em questão é que, quando é utilizado um processo em simulação, uma vez que o AE apresenta uma solução satisfatória, ela é normalmente transferida para o robô real e, desta forma, este controlador encerra sua evolução. No caso de o ambiente apresentar alguma modificação ou mesmo, características não previstas pelo projetista do sistema, este robô não teria mais condições de se adaptar. Assim sendo, para um sistema ser considerado verdadeiramente evolucionário, ele deve ser capaz de apresentar um sistema evolucionário continuado, evoluindo durante toda a existência do robô (WERGER; MATARI’C, 1999).

Desta forma, recentemente foram apresentando alguns trabalhos onde a população de indivíduos do AE é um conjunto de robôs reais, ou seja, a população existe fisicamente (SIMÕES; DIMOND, 1999) (WATSON; FICICI; POLLACK, 1999). Nesses experimentos, a população evolui sem interferência computacional externa ou de intervenções do projetista.

Assim, para concluir, pode-se distinguir as seguintes metodologias diferentes de execução:

1. evolução e execução de uma população somente em simulação computacional;
2. evolução completamente realizada em uma população simulada e o melhor indivíduo é então executado em um robô real;
3. evolução de uma população simulada e depois transferida para uma população de robôs reais;
4. evolução de uma população iniciada em simulação, até atingir um determinado desempenho, para depois ser transferida para uma população de robôs reais, onde é dada continuidade ao processo evolutivo;
5. evolução completamente executada em uma população de robôs físicos, sem intervenção humana.

3.3 Trabalhos Correlatos

Os primeiros eventos relacionados à RE aconteceram no início dos anos 1990, tornando esta área de pesquisa uma disciplina bem estabelecida, com vários grupos no mundo todo, informando continuamente novos avanços teóricos e práticos. A partir de então, a RE vem se apresentando como uma abordagem nova e promissora para o desenvolvimento de, principalmente, robôs móveis autônomos, capazes de agir com rapidez e robustez em ambientes reais, inóspitos e que apresentem um alto grau de ruído ou aleatoriedade.

Mesmo sendo um dos objetivos da RE, a evolução em robôs reais, enfrentando condições adversas, uma parcela significativa dos trabalhos vem sendo realizados somente em simulação.

Um dos trabalhos pioneiros na área de RE, foi desenvolvido por Floreano e Mondada, (FLOREANO; MONDADA, 1994), onde foram utilizados AGs para evoluir

controladores neurais para robôs autônomos. Os controles evoluíam em um mini-robô Khepera (MONDADA; FRANZI; IENNE, 1993)¹, objetivando realizar movimentos livres de colisão. O sistema foi modelado com oito neurônios de entrada, ligados diretamente aos sensores infravermelhos do robô, e dois neurônios de saída, ligados diretamente aos motores. Este trabalho alcançou seu objetivo com sucesso, originando outros, tais como (FLOREANO; MONDADA, 1996), onde foi novamente utilizado um AG para evoluir uma RNA, mas desta vez, para que o Khepera encontrasse seu local de recarga de baterias - um ponto luminoso no ambiente. Neste modelo, foram usadas mais entradas para a RNA, acrescentando a necessidade de sensoriar o nível de energia corrente, além dos sensores de aproximação, que necessitavam registrar a distância de objetos e simultaneamente, a distância do objetivo (identificado por um ponto luminoso). Em ambos os trabalhos, a evolução acontece totalmente no robô real.

Outro trabalho utilizando um AG para evoluir os pesos de uma RNA, foi desenvolvido por Miglino (MIGLINO; NAFASI; TAYLOR, 1994). A evolução acontece em simulação e, a rede resultante é então transferida para o robô real. O robô utilizado foi construído com blocos de Lego e controlado por um hardware microprocessado. Os resultados também foram satisfatórios, gerando novos trabalhos utilizando essa mesma metodologia, onde controladores neurais foram desenvolvidos utilizando AGs, com o objetivo de limpar um cenário delimitado por paredes (NOLFI; PARISI, 1995)(NOLFI, 1997). Nestes experimentos, depois de ter os controladores evoluídos em simulação, também foram executados fisicamente em um robô Khepera, que, para tanto, foi equipado com uma garra para coletar objetos espalhados no cenário.

Partindo do paradigma de arquitetura de subordinação proposto por Brooks (BROOKS, 1986) (BROOKS, 1990) (BROOKS, 1991), onde os comportamentos, previamente modelados, são hierarquizados, um sistema de controle foi implementado por Koza (KOZA, 1992) (KOZA, 1994), utilizando Programação Genética. Neste trabalho, robôs simulados realizavam tarefas de acompanhar paredes e movimentar objetos. Utilizando a mesma técnica, Reynolds (REYNOLDS, 1993) implementou comportamentos de desvio de obstáculos, implementando em experimentos simulados. Por fim, utilizando a mesma arquitetura de comportamentos, Nordin e Banzhaf (NORDIN; BANZHAF, 1995) (NORDIN; BANZHAF, 1996) desenvolveram um sistema similar, mas executado fisicamente em um robô Khepera.

Gu (GU et al., 2003) utilizou um AG para evoluir comportamentos modelados em lógica nebulosa, em uma arquitetura hierarquizada de comportamento, similar à arquitetura de subordinação. O objetivo é conseguir que um robô quadrúpede alcance algumas habilidades necessárias para jogar futebol, dominando os comportamentos de perseguir a bola e alcançar uma posição na quadra. Neste trabalho, a metodologia utilizada foi de evoluir os controladores somente no robô real, para que este realizasse o aprendizado de forma completamente autônoma.

Além dos trabalhos que utilizam AG como uma ferramenta de otimização para outras técnicas de IA, como os citados anteriormente, também existem trabalhos que procuraram focar a hibridização de AE com sistemas simbólicos tradicionais, tais

¹ O Robô Khepera é uma plataforma robótica em miniatura, com um diâmetro de 55 mm e altura de 30 mm. Possui duas rodas controladas por motores independentes e encoders incrementais. Seu conjunto básico de sensores é composto por oito sensores infravermelhos de proximidade, que medem a reflexão da luz enviada pelo robô, bem com a luminosidade do ambiente.

como bases de conhecimento constituídas por regras de produção. Esta alternativa se caracteriza pelo uso dos Sistemas Classificadores (SC).

Os pioneiros na aplicação de SC aplicados à RE são Dorigo, Schnepf e Colombetti (DORIGO; SCHNEPF, 1993) (DORIGO; COLOMBETTI, 1994) (DORIGO, 1995). Eles iniciaram os estudos para investigar e comprovar a factibilidade do uso de SC como ferramenta para implementar aprendizado e controle adaptativo para robôs. O principal problema enfrentado na época era que, utilizando um SC de modo convencional, o tempo de aprendizagem se tornava proibitivo, tendo em vista o custo computacional dessa categoria de aplicação. Desta forma, a solução foi encontrada a partir de três ações:

- paralelizar o SC, passando a ter um sistema composto por vários SC, cada qual, trabalhando com um determinado comportamento;
- reduzir a complexidade do sistema, criando módulos que representassem comportamentos em uma arquitetura distribuída hierarquicamente; e, por fim,
- usar supervisão no aprendizado, com reforço positivo (e, em alguns experimentos, também reforço negativo para ações indesejadas) para as ações realizadas satisfatoriamente.

Seguindo pelo mesmo caminho, Grefenstette e Schultz utilizaram o sistema baseado em SC, chamado SAMUEL (GREFENSTETTE; RAMSEY; SCHULTZ, 1990), e implementaram um sistema para navegação de um robô real, livre de colisões (GREFENSTETTE; SCHULTZ, 1994). A arquitetura deste sistema possuía dois módulos distintos, que operavam paralelamente: um módulo simulado para testar as regras candidatas a representar os comportamentos; e outro módulo que executava os comportamentos já testados em um Nomad 200 (seção 5.2). Esta arquitetura está representada na Figura 3.1 (GREFENSTETTE; SCHULTZ, 1994).

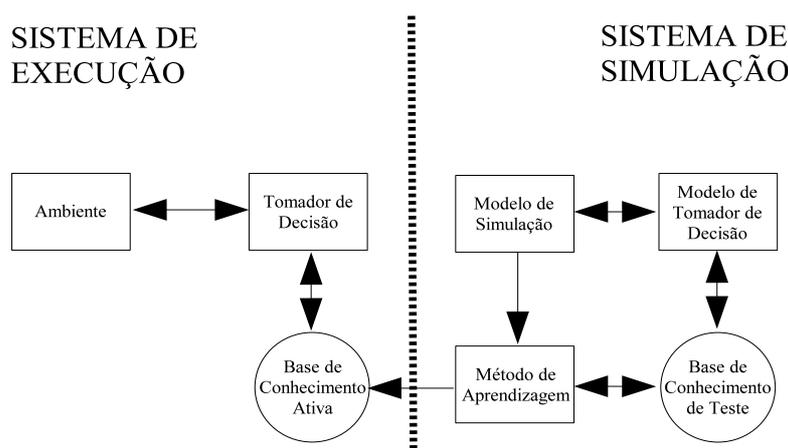


Figura 3.1: Arquitetura do Sistema Implementado Por Grefenstette (1994)

A dificuldade do custo computacional para a implementação também foi enfrentada. Para diminuir a complexidade das regras, os sensores do robô foram agregados em grupos, reduzindo para somente quatro sinais de entrada, o que implica

em diminuir significativamente o poder de sensoriamento do robô, mas, em contrapartida, reduz drasticamente a complexidade e o tamanho do espaço de busca. As principais características desse AE são:

- cada indivíduo da população representava um conjunto de regras de navegação;
- esta população tinha tamanho fixo durante toda a execução;
- a evolução do sistema se dava através de um AE (baseado em SC) no módulo de aprendizagem em simulação;
- cada conjunto de regras (indivíduo) era testado;
- as novas gerações substituíam as antigas;
- o melhor indivíduo era depois transferido ao módulo de execução no robô real.

As dificuldades enfrentadas com as diferenças entre as execuções em simulação e no mundo real motivaram o desenvolvimento de uma nova proposta, apresentada na Figura 3.2 onde o módulo de aprendizagem procurava detectar possíveis mudanças no ambiente real para então representá-las no ambiente simulado para garantir a geração de regras ainda mais adaptadas ao estado atual do mundo real. (GREFENSTETTE, 1996) (MORIARTY; SCHULTZ; GREFENSTETTE, 1999)

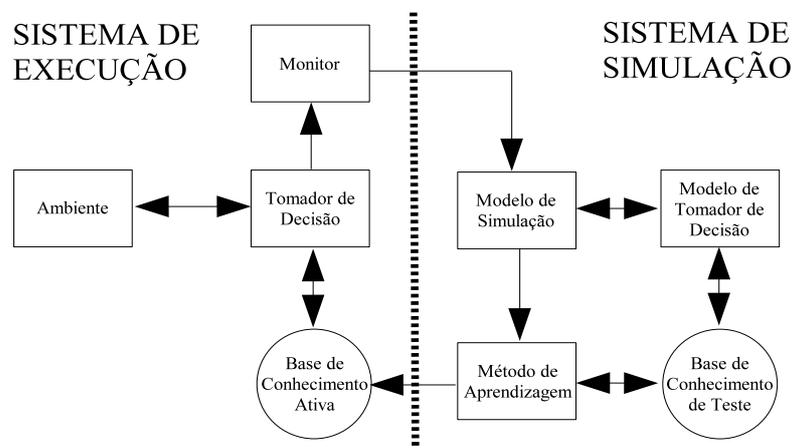


Figura 3.2: Arquitetura do Sistema Implementado Por Grefenstette (1996)

Katagami e Yamada (KATAGAMI; YAMADA, 2000) apresentaram uma abordagem alternativa, propondo um método de aprendizagem com SC onde os robôs adquirem conhecimento pela interação com humanos. Os autores denominaram essa abordagem de "Sistemas Classificadores Interativos". Sua principal característica é que a função de avaliação é substituída por uma interface onde um observador humano deve indicar qual (ou quais) indivíduo (s) apresenta (m) desempenho satisfatório. O restante do sistema evolutivo funciona de acordo com as abordagens tradicionais dos SC, produzindo novos controladores a partir daqueles que foram classificados como satisfatórios pelos observadores.

Mais recentemente, outros trabalhos utilizando SC para a geração de regras para navegação de robôs foram realizados, mas apresentam somente a replicação das estratégias anteriormente desenvolvidas. Chamindra (CHAMINDRA, 2003) desenvolveu um pequeno sistema de navegação autônoma de um robô simulado, a partir de um SC convencional, utilizando populações de tamanho fixo, onde cada indivíduo representava uma regra de navegação. O sistema apresentou bons resultados para o simulador desenvolvido e os objetivos para trabalhos futuros é transferir o sistema para alguma plataforma robótica real.

Dentre os trabalhos mais recentes, que apresentaram contribuições e novas abordagens utilizando SC em robótica móvel, está o sistema desenvolvido por Cazangi (CAZANGI; FIGUEIREDO, 2001), (CAZANGI; FIGUEIREDO; ZUBEN, 2003), (CAZANGI; ZUBEN; FIGUEIREDO, 2003), (CAZANGI, 2004), que utilizou um mini-robô Khepera II para executar um sistema de aprendizagem cujo objetivo era encontrar pontos luminosos em um ambiente com obstáculos. A Figura 3.3 mostra o funcionamento e a interação das partes componentes deste sistema.

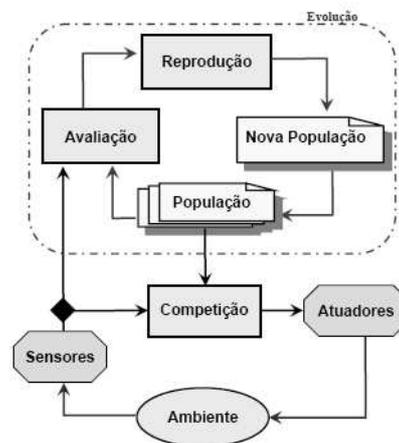


Figura 3.3: Arquitetura do Sistema proposto por Cazangi (2005)

O sistema foi modelado inspirado nos SC, mas não utilizando todas suas características, procurando simplificar a forma de descoberta de novas regras. Assim como o trabalho de Chamindra, a população também é de tamanho fixo e composta por indivíduos que representam, cada um, uma única regra. Alguns pontos foram considerados críticos no desenvolvimento do trabalho, dentre os quais:

- necessidade de avaliar a factibilidade de cada regra, visto que é composta por um segmento antecedente formado pelo sensoriamento de obstáculos juntamente com a proximidade do objetivo;
- tamanho da população, considerando que é de tamanho fixo;
- taxa de reprodução, para não substituir drasticamente a população;
- capacidade de generalização das regras;

A forma de disparo do processo evolutivo acontece quando determinados eventos ocorrem com o robô, os quais são:

- colisão;
- captura de alvo, e
- monotonia (período prolongado sem ocorrer nenhum dos eventos anteriores).

Em cada um destes eventos, é realizada uma varredura na população, para selecionar as regras candidatas à geração de novas regras. Essa seleção é realizada por critérios de semelhança de cada uma das regras com a situação sensoriada no evento ocorrido. Desta forma, as novas são geradas e uma parcela da população é sumariamente substituída, caracterizando a substituição de gerações. O problema identificado foi que, após sucessivos processos evolutivos gerados pelo evento de captura, a população acaba se especializando neste tipo de atividade e passa a ter seu conhecimento sobre desvio de obstáculos deteriorado, podendo apresentar instabilidade em seu comportamento. A solução encontrada para minimizar o problema foi limitar o número de descendentes gerados em cada processo evolutivo disparado pelas capturas, proporcionalmente ao histórico do desempenho do robô. Ou seja, caso ele tenha realizado sucessivas capturas, menos descendentes serão gerados.

Como se trata de evoluir a população, para que esta, como um todo, represente a solução para o problema, isto acarreta que pode ser observada a formação de categorias de regras. Desta forma, ao final do experimento, a população é formada por vários conjuntos (categorias) que proporcionam determinados comportamentos.

Os testes foram realizados em simulação, obtendo bons resultados, que depois foram então transferidos para o robô real, onde confirmou sua eficácia, atingindo o objetivo proposto. Confrontado com um cenário mais complexo, o robô necessitou ajustar sua base de conhecimento, o que aconteceu pela característica de aprendizado contínuo, ou seja, o sistema continuou o processo de evolução das regras. Também foram realizados experimentos onde o robô efetuou todo seu aprendizado com o robô real, ou seja, iniciou o processo sem conhecimento prévio. Neste caso, os resultados obtidos também foram satisfatórios e muito semelhantes aos simulados².

Apesar dos trabalhos iniciais de Cazangi terem sido desenvolvidos utilizando um robô real, o mais recente artigo (CAZANGI; ZUBEN; FIGUEIREDO, 2005) apresenta uma nova característica para o sistema: o uso de informação inspirada nos feromônios que as formigas liberam no ambiente, a fim de possuir informação sobre caminhos já trilhados no cenário. Como se trata de um elemento que não possui sensores e atuadores correspondentes no robô real, este trabalho é executado somente em simulação.

Por fim, outros trabalhos trazem à comunidade científica uma nova abordagem na RE, que foca uma metodologia baseada na "evolução embarcada", a qual é representada por (FICICI; WATSON; POLLACK, 1999) (SIMÕES; DIMOND, 1999) (SIMÕES; DIMOND, 2001) (SIMÕES; BARONE, 2002) (SIMÕES, 2003) (WATSON; FICICI; POLLACK, 1999) (WATSON; FICICI; POLLACK, 2002). A principal característica desta abordagem é que a população do AE existe fisicamente, ou seja, cada indivíduo é embarcado em um robô real. Os experimentos realizados abordam tarefas ainda bem simples, tais como exploração do ambiente, desvio de obstáculos, recuperação de objetos e sistemas predador-presa.

²deve se levar em consideração que, pelo tamanho reduzido do robô Khepera, o ambiente real utilizado para ele pode ser considerado parcialmente controlado, porque sofre menos com eventos como diferenças de luminosidade e outras interferências externas

4 PROPOSTA DE SISTEMA: APRENDIZADO EVOLUCIONÁRIO

Sistemas de aprendizagem autônoma também são conhecidos como "Sistemas de Aprendizado de Máquina" e representam uma sub-área da IA que considera o aprendizado como sendo a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Assim, Aprendizado de Máquina (AM) pode ser definido como um programa de computador que aprende a partir da experiência, com respeito a alguma classe de tarefa e medida de desempenho (MITCHELL, 1997).

Já existem alguns sistemas desenvolvidos, que utilizam abordagens evolucionárias para o aprendizado dos robôs, como pode ser visto no capítulo 3. No entanto, como pode ser observado, o principal caminho trilhado pelos pesquisadores em RE foi utilizar RNAs em conjunto com algum AE, agregando as capacidades de aprendizado e evolução, respectivamente, construindo, desta forma, um algoritmo híbrido. A opção pelas RNAs se dá pelas suas atrativas características de aprendizagem e adaptação, eficiente aquisição de conhecimento, independência do domínio, tolerância a ruídos e falhas, entre outras (WANG, 2002). No entanto, estes sistemas podem apresentar desvantagens, relacionadas à característica de que o conhecimento em RNAs não é explicitamente declarado e o tempo de aprendizado pode ser consideravelmente longo, fazendo com que o sistema não tenha a garantia de que vá convergir para uma solução (WANG, 2002).

A Lógica Nebulosa também é uma alternativa interessante para se implementar um AE, devido a grande capacidade de modelar relacionamentos entre sinais de entrada e saída, incorporando heurísticas na forma de regras e, portanto, sendo uma interessante alternativa para sistemas que não podem ser modelados com precisão (WANG, 2002).

Este capítulo descreve o sistema desenvolvido, apresentando seus módulos, características e funcionalidades. O sistema foi projetado tendo como base, idéias provenientes, principalmente, de três técnicas evolucionárias: Algoritmos Genéticos, Algoritmos Genéticos com Idades e Sistemas Classificadores. Como o objetivo é construir um sistema completamente autônomo, onde o robô evolua seu conhecimento sem interferência externa, procurou-se criar uma alternativa simplificada. Assim, o sistema não pode ser classificado exclusivamente como pertencente a alguma destas categorias da computação evolucionária, mas como uma nova forma de implementação, que utiliza conceitos e estratégias que apresentaram bons resultados em cada uma delas.

4.1 Descrição do Problema

O aprendizado em robótica móvel é um problema que consiste, basicamente, na aquisição de conhecimento pertinente ao conjunto de tarefas que este robô deve realizar, possibilitando que ele as execute de forma satisfatória ou, ao menos, razoável. Várias técnicas são utilizadas com este propósito, desde as clássicas metodologias de IA, para sistemas deliberativos, até abordagens de IC, como Fuzzy, RNAs e CE.

Uma característica desejável para um sistema de aprendizado é que ele seja adaptativo, caracterizando um campo de aplicação pertinente às técnicas de IC. RNAs são largamente utilizadas com este propósito, no entanto não permitem a declaração explícita do conhecimento gerado, e apresentam dificuldades expressivas para inserção de novos comportamentos ou conhecimento em um sistema já treinado. Para minimizar estes problemas, existem trabalhos com RNA incrementais, buscando contornar este inconveniente. Os AE também são utilizados, normalmente como mecanismos de otimização de RNAs (NOLFI, 1997).

Navegação autônoma em robótica móvel configura um domínio bastante apropriado para a investigação da aprendizagem para robôs. É um domínio, apesar de sua complexidade, bastante conhecido e que possibilita a realização de análises e avaliações efetivas do desempenho das técnicas aplicadas. Outro ponto é que o interesse em navegação autônoma é grande, por existir muitas aplicações práticas para um sistema desta natureza, tais como: exploração de ambientes, limpeza de superfícies e dutos, sistemas de vigilância, etc.

Desta forma, o problema abordado para estudo e aplicação da estratégia de aprendizagem proposta neste trabalho é habilitar um robô móvel a navegar em um ambiente desconhecido, evitando colisões e, preferencialmente, acompanhando as paredes, para não vagar aleatoriamente.

4.2 Alternativa Desenvolvida

O objetivo inicial deste trabalho é o desenvolvimento de um sistema de aprendizado autônomo, onde o robô possa construir e manter uma base de regras do tipo "situação-ação", tais como as apresentadas em 2.3.2. Este processo de construção e manutenção do conhecimento deve acontecer a partir da interação direta com o ambiente e sem a necessidade de interferência externa.

A opção de trabalhar com regras explicitamente declarativas vem ao encontro de uma possibilidade importante, a de que o projetista humano possa inserir novas regras arbitrariamente, as quais teriam sido descobertas em outros processos de aprendizagem, passando estas a fazer parte da base de conhecimento. No entanto, mesmo estas regras tendo sido inseridas externamente, elas devem estar suscetíveis ao processo de manutenção da base de conhecimento, ou seja, elas podem sofrer mutações e também podem chegar a ser consideradas obsoletas, acarretando sua exclusão da base.

Uma característica limitante nos sistemas de aprendizagem baseados em AE, identificada nos trabalhos correlatos, é a definição de quantas regras compõem uma base razoável de conhecimento a ser gerada e, ainda, quantas são as regras que devem ser usadas no processo evolutivo para a descoberta deste conjunto final.

Desta forma, o ponto norteador deste trabalho é a busca por uma alternativa que minimize essa deficiência e, portanto, isente o projetista de sistemas de apren-

dizagem para robôs de definir os valores para esses parâmetros, os quais devem ser contemplados pela estratégia evolucionária inerente ao sistema.

Para desenvolver os experimentos, é necessário definir precisamente qual o problema que o sistema deve abordar. A opção foi pela navegação autônoma de robôs, que é um problema da robótica móvel que consiste, basicamente, em fazer com que um robô realize trajetórias satisfatórias, possibilitando ao robô realizar tarefas como deslocar-se, evitar colisões, desviar de obstáculos, capturar alvos, atingir locais objetivos no ambiente, entre outras. Robôs móveis autônomos devem ser capazes de aprender estratégias de navegação e, para confirmar sua característica de autonomia, devem conseguir adaptar-se a novas situações e alterações ocorridas no ambiente. Isso significa que, quanto mais diversificado e instável o ambiente, maior deve ser a autonomia do robô.

Mais especificamente, a tarefa a ser executada é a navegação livre de colisões, mas com o acompanhamento de paredes. A opção por esta tarefa se dá pelo fato de que as plataformas robóticas disponíveis no LRI/UFRGS possuem somente sensores de ultrassom e infravermelho, não possibilitando determinar diferenciação de características nos objetos sensorizados (identificação de alvos). Além disso, esta tarefa se configura como uma combinação de objetivos conflitantes, visto que o desvio de obstáculos requer uma ação de distanciamento dos objetos no ambiente, enquanto que, para efetuar o acompanhamento de uma parede, é necessário que o robô faça uma aproximação da mesma (o que, para o primeiro objetivo é uma situação de risco).

Os conceitos básicos que regem este processo evolucionário são a "satisfação" e/ou a "insatisfação" do robô em relação ao resultado obtido com um determinado movimento em relação à uma situação sensorizada por ele. Desta forma, aqueles movimentos que puderam gerar mais "satisfação" ao robô (andar livre de risco de colisão, não vagar aleatoriamente, etc...) acabam por ser incorporados definitivamente ao seu conhecimento, "consolidando" a regra. Todas as regras, já consolidadas ou ainda em apreciação pela satisfação do robô, são gravadas diretamente nos cromossomos, o que configura um sistema possuidor de uma "memória genética".

Inicialmente, é necessário saber que o sistema de controle e aprendizagem pode ser executado em um sistema simulado, diretamente embarcado no robô ou, ainda, em um computador que se comunica com o robô. A última alternativa é utilizada em situações onde o hardware, apesar de permitir sensoriamento e atuação, não possui a capacidade de processamento necessária para a construção autônoma do conhecimento¹.

A alternativa utilizada neste trabalho é realizar experimentos em simulação, para garantir que o sistema seja validado e apresente um desempenho satisfatório, não pondo em risco a estrutura eletromecânica do robô. Por fim, afastada a possibilidade destas situações críticas na execução e movimentação do robô, novos testes devem ser realizados no robô NOMAD 200, para verificar o desempenho em sistemas reais.

¹Um exemplo dessa limitação é o robô Khepera (LPFI), ou ainda, o conjunto de robôs construídos no LRI, que utilizam um pequeno processador de 2 MHz e somente 64 KB de memória, para uso em Futebol de Robôs.

4.3 Padrões de funcionamento do Robô

Para desenvolver um sistema de aprendizado robótico, é necessário definir qual a estrutura eletromecânica (sensores e atuadores) onde o sistema irá atuar, ou ao menos, quais suas características básicas. Assim, o sistema apresentado neste trabalho prevê que o robô funcione com as seguintes regras de sensoriamento e atuação:

- movimenta-se exclusivamente para a frente, em velocidade constante;
- realiza movimentos de curva aplicando uma variação de ângulo de direção do movimento;
- possui um procedimento para garantir sua sobrevivência em situações de risco;
- executa paradas somente em situações de risco e colisões;
- realiza giro sobre o próprio eixo somente após estar parado (situações de risco e colisões);
- efetua leitura dos sensores para navegação e para segurança de forma independente;
- possui leitura de sensores de navegação que permitem executar leituras de distâncias maiores, apesar de ineficientes para objetos próximos;
- possui leitura de sensores de segurança, que permitem obter informações de objetos que estejam próximos ao robô;
- usa somente os sensores localizados na parte frontal do robô, entre o mais à direita e o mais à esquerda;
- as leituras dos sensores são discretizadas em quatro intervalos para cada categoria (navegação e segurança), os quais são:

Tabela 4.1: Intervalos de classificação da leitura dos Sensores

Navegação	Segurança	Interpretação
0	-	Muito Distante / Sem sinal de obstáculo
1	0	Distante
2	1	Próximo
3	2	Muito Próximo
-	3	Perigosamente Próximo

onde a categoria de medida "Distante" para um sensor de segurança pode representar também "Sem sinal de obstáculo", que é identificada neste trabalho como "leitura nula".

4.4 Descrição da Arquitetura Proposta

Para que se possa construir agentes inteligentes e autônomos, é necessário que exista uma arquitetura para o agente. Uma arquitetura pode ser vista como uma metodologia para construir agentes, a qual uma vez especificada, apresenta os componentes dos mesmos e a forma como estes interagem. Arquiteturas diferentes têm sido propostas (WOOLDRIDGE; JENNINGS, 1995) (JENNINGS; WOOLDRIDGE, 1998) para assegurar que os agentes tenham um nível particular de inteligência e autonomia. Basicamente, estas arquiteturas se dividem em dois grandes grupos: Reativas (BROOKS, 1986), que não incluem modelos simbólicos do mundo e manifestam o conhecimento através de comportamentos; e Cognitivas (WOOLDRIDGE; JENNINGS, 1995), também denominadas deliberativas, que executam processos de planejamento e tomada de decisão. Basicamente, a arquitetura de um agente possui uma estrutura, como a apresentada na figura (4.1)

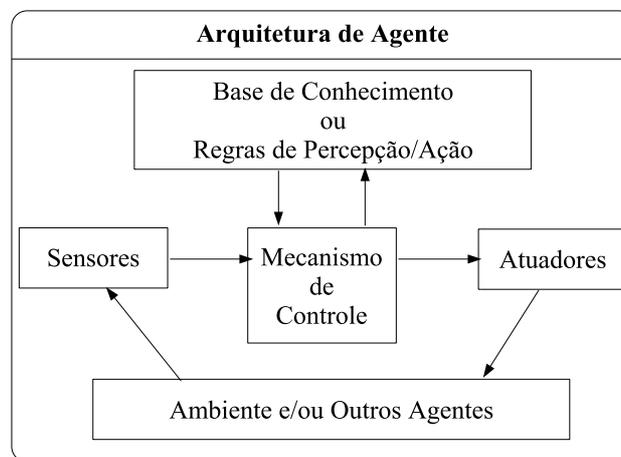


Figura 4.1: Arquitetura de Agente Padrão

O sistema se caracteriza como uma abordagem híbrida, construída a partir dos fundamentos de arquiteturas para agentes reativos, visto que para se trabalhar com abordagem deliberativa, é necessário possuir um modelo explícito do domínio e de todas as entidades nele existentes, o que não se enquadra no escopo deste trabalho. No entanto, esta arquitetura não pode ser considerada completamente reativa, porque foi incorporado um aspecto pertinente aos agentes deliberativos, que é uma forma de representação do conhecimento explicitamente declarativa.

O sistema de aprendizagem evolucionária foi originalmente projetado para habilitar um agente robótico - de forma individual e a partir da interação com o ambiente - a construir, evoluir e manter uma base de conhecimento necessária para executar uma determinada tarefa. Assim, a arquitetura do sistema pode ser vista como sendo constituída por quatro módulos básicos:

- **Memória:** armazena a população de regras;
- **Reprodução/Evolução:** responsável pela descoberta de novas regras;
- **Controle Populacional:** gerencia o tamanho e qualidade da população;
- **Execução:** responsável pela interação do robô com o ambiente

A Figura 4.2 apresenta representação gráfica da arquitetura, com a base de conhecimento e o módulo de controle, com todos os seus componentes. Na seqüência do texto, são descritas as características de cada módulo.

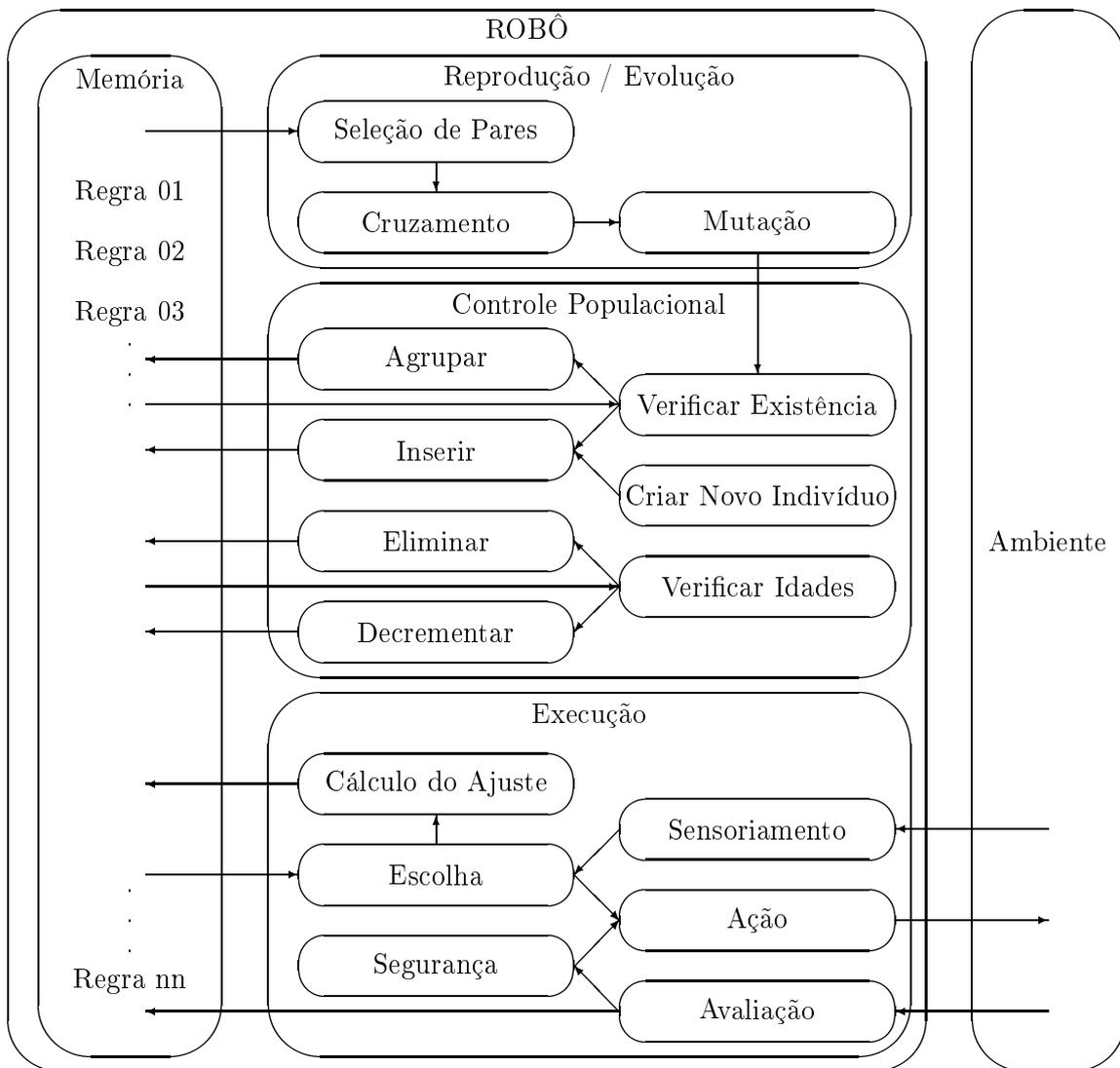


Figura 4.2: Arquitetura do Sistema.

4.4.1 Memória

O módulo de memória é responsável pelo armazenamento da população de regras. Como o objetivo é gerar um conjunto simplificado de regras, espera-se que este não venha a crescer demasiadamente, para não comprometer o desempenho do sistema. Assim, todo o conjunto é mantido em memória de trabalho, permitindo um acesso rápido durante todo o processo evolutivo. Durante o processo de execução, quando o acesso à memória é mínimo, todo o conteúdo pode ser atualizado em disco, a fim de armazenar o conhecimento em memória não-volátil.

4.4.1.1 Forma de Representação das Regras

Cada indivíduo da população representa uma única regra de movimentação e a forma de representação utilizada é a mesma da formação das regras nos Sistemas de Produção; sendo constituída por uma parte antecedente e outra conseqüente, com a seguinte estrutura lógica:

SE < antecedente > ENTÃO < conseqüente >

O antecedente da regra representa uma situação de leitura sensorial do ambiente, e, complementarmente, o segmento conseqüente, indica a ação correspondente a ser executada para a respectiva situação.

O segmento antecedente é formado por um vetor de caracteres, utilizando o alfabeto binário {0,1}, onde cada sensor é representado por um par de elementos. Isto possibilita representar e diferenciar as quatro situações possíveis de leitura para cada um dos sensores.² Como nos experimentos deste trabalho são utilizados somente os nove sensores frontais do robô, este segmento da regra é formado por 18 elementos. Para a parte conseqüente da regra, que designa a ação, tem-se outra seqüência de caracteres, composta por quatro unidades, representadas também pelo alfabeto binário {0,1}. O primeiro destes elementos informa o sinal - positivo ou negativo - e os outros três, a representação binária do valor correspondente para ação (detalhes na seção 4.4.4.4).

Para ilustrar, a estrutura (genótipo) da regra pode ser observada na Figura 4.3:

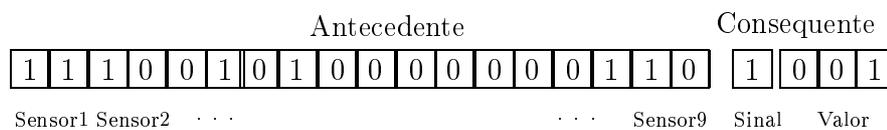


Figura 4.3: Forma de Representação das Regras.

Esta regra está graficamente representada na Figura 4.4, que apresenta os valores esperados para cada um dos sensores e o valor correspondente da ação a ser tomada em sua execução.

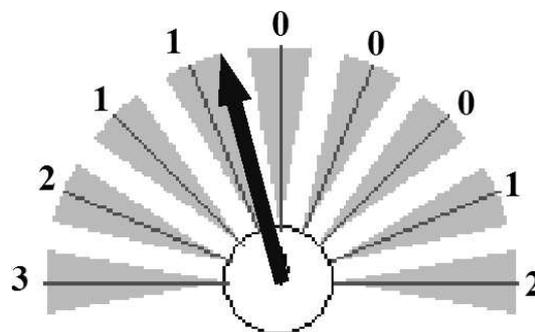


Figura 4.4: Situação representada pela regra

²Aqui deve se observar a diferença da representação usual dos SC, onde o segmento antecedente utilizaria o alfabeto ternário {0,1,#} para fazer uso da característica de generalização proporcionada pelo componente {#}.

Cada regra possui também dois atributos de extrema importância para esta implementação:

- idade, e;
- ajuste.

Idade se refere ao número de gerações que esta regra ainda permanecerá na base de conhecimento. Na verdade, não tem o mesmo perfil que a idade de um ser vivo, que é gradativamente incrementada com o passar do tempo. Trata-se do "TEMPO RESTANTE DE VIDA" do indivíduo, que, durante todo o processo, recebe incrementos e decrementos, conforme a importância da respectiva regra para o sistema. O uso desta estratégia permite que aquelas regras em desuso venham a ser eliminadas depois de ter passado um tempo pré-estabelecido.

O atributo de ajuste, ou grau de adaptabilidade, é um atributo bem conhecido em AGs, e aqui, representa o índice de utilidade desta regra, a partir da semelhança da mesma com as situações sensoriadas durante um período de execução.

O atributo de idade é inspirado nos Algoritmos Genéticos com Idades (AGIs), propostos por Michalewicz (MICHALEWICZ, 1996) (SERRADA, 1996) (YEPES, 1999). Deve-se notar que esta abordagem possui diferenças do AGI original descrito por Michalewicz, onde a idade era calculada no momento da criação do indivíduo, o qual permaneceria na população pelo número de gerações respectivo. Aqui, este atributo é dinâmico: pode ser incrementado e/ou decrementado constantemente, de acordo com o uso e desempenho do indivíduo em relação ao ambiente. Ou seja, ele também representa muito bem a adaptabilidade e usabilidade do indivíduo.³

4.4.1.2 Tamanho da População

A abordagem sugere que o sistema funcione com uma população de regras de tamanho variável, visto que o usuário não necessita ter ciência de qual é o número razoável de regras que formará o conjunto final. Desta forma, o controle do número de indivíduos acontece através do funcionamento dos mecanismos internos de controle populacional.

No início do processo, a população inicial é constituída por um conjunto de indivíduos definido arbitrariamente pelo programador. Durante o processo de interação com o ambiente e aprendizado, a população terá um crescimento acentuado no período inicial, para abranger mais pontos no espaço de busca e, posteriormente, alcançará um processo de estabilização, para depois reduzir seu tamanho gradativamente.

A partir do estágio de estabilização, já pode ser observada a existência de uma "sub-população", que é formada pelas regras que apresentam relevância para o desempenho da tarefa designada - são as chamadas "regras consolidadas". Desta forma, neste estágio do processo, a população total do sistema será formada por esta sub-população mais o restante dos indivíduos, dentre os quais, existem aqueles evoluindo para gerar novas regras ainda mais satisfatórias.

Por fim, a etapa de diminuição do tamanho da população acontece porque aquelas regras restantes - não relevantes para o objetivo do sistema - vão gradativamente

³É importante observar que neste trabalho, diferentemente da proposta original de Michalewicz, o atributo de idade é usado para o controle populacional, deixando o processo de seleção de reprodutores para o atributo de ajuste. Ou seja, o objetivo do trabalho original de AGI, que era eliminar a pressão seletiva não está sendo considerado.

sendo eliminadas e, como o sistema apresenta estabilidade, também não ocorre a geração de novas regras. No entanto, pelas características adaptativas do sistema, a qualquer momento, quando tornar-se necessário, o processo evolutivo reinicia e a quantidade de indivíduos torna a crescer.

4.4.1.3 *Representatividade da População Inicial*

Uma das grandes dificuldades enfrentadas em qualquer AE é definir a forma de criação da população inicial, afinal, isso influencia significativamente na escolha das regras mais adequadas no início do processo evolucionário. Como a primeira geração de indivíduos em um AG convencional é criada aleatoriamente, não existe representatividade destes indivíduos em relação às situações que o ambiente possa apresentar.

Para contornar esta deficiência, uma alternativa encontrada foi gerar os primeiros indivíduos de forma que estes representem uma distribuição uniforme no espaço de busca. O processo para isso é calcular a diferença entre o valor mínimo e o valor máximo possíveis de ser representados pelo segmento antecedente do cromossomo, quando convertidos para valores decimais. Esta diferença é então dividida pelo número de indivíduos da população inicial, resultando no valor de intervalo médio entre os indivíduos. Assim, o procedimento de geração da primeira população cria sequencialmente os indivíduos, intercalados pela valor do intervalo médio calculado, completando o segmento consequente com valores definidos aleatoriamente.

Outra alternativa é criar um conjunto mínimo de regras válidas e anexá-lo na população inicial, permitindo, desta forma, que o robô possa, ao menos, andar para frente quando não há sinal sendo detectado pelos sensores. Essa idéia também é utilizada na condução de experimentos onde a complexidade do ambiente é incrementada gradativamente e o robô permanece com o conhecimento gerado no experimento anterior. Essa alternativa vem ao encontro dos objetivos do trabalho, que visam possibilitar a um usuário ou projetista inserir no sistema, alguma regra que achar pertinente.

Nestes experimentos, ao invés de inserir um conjunto pré-definido de regras, definiu-se arbitrariamente inserir uma única regra, permanente e imutável (não sofre incremento ou decremento na idade), chamada no sistema de "Regra ZERO". É a regra cujo segmento situação é composto somente por "0s", o que significa não estar identificando qualquer objeto na proximidade. Sua ação é a translação com ângulo de variação nulo, ou seja, um deslocamento frontal em linha reta.

4.4.1.4 *Definição da Idade Inicial e Idade Máxima*

Como o sistema foi modelado para que cada indivíduo tenha sua permanência na população controlada pelo seu desempenho, a partir da sua idade (tempo de vida restante), foi definido arbitrariamente que este atributo terá um valor máximo de 1000 unidades. O valor inicial, no momento da criação de um indivíduo da população inicial não deve ser alto, para que todas as novas regras criadas que não venham a ser utilizadas pelo sistema, sejam descartadas em um tempo não muito prolongado; mas também não deve ser um tempo muito baixo, porque o sistema deve dar tempo suficiente para descobrir se estas regras podem ter relevância em alguma determinada situação. Para as regras geradas nos processos reprodutivos, a idade inicial é a média das idades dos respectivos progenitores. A justificativa desta estratégia é que, considerando que os novos filhos gerados possam vir a ser mais

adaptados e, conseqüentemente, poderiam até substituir os seus progenitores nas execuções das ações, então eles devem iniciar sua vida numa classificação similar a estes.

4.4.1.5 *Classificação de Categorias de Regras*

Considerando que a idade de cada indivíduo pode variar durante todo o processo, foram definidas 4 categorias:

1. Iniciante: a regra que ainda não atingiu 30% da idade máxima. É uma regra em suas primeiras utilizações.
2. Em Crescimento: regra que possui entre 30% e 50% da idade máxima. Já apresentou resultados satisfatórios, mostrando que pode ser relevante para o conjunto de regras do sistema.
3. Candidata a Consolidada: regra que atingiu 50% da idade máxima. Significa que já foi utilizada com sucesso diversas vezes, e que apresenta grande probabilidade de vir a se consolidar;
4. Consolidada: a regra que ultrapassou a marca de 90% do limite máximo para a idade. Confirmou sua eficácia e relevância para o sistema.

4.4.2 **Módulo de Reprodução e Evolução**

O módulo de reprodução e evolução é responsável pela descoberta de novas regras. Este processo acontece de forma evolucionária, considerando que a base de conhecimento é constituída por uma população de regras, que deve evoluir ao passar das gerações, adaptando este conjunto para que se tenha regras que representem satisfatoriamente as situações que ocorrem no ambiente, bem como, também levando-se em consideração que as ações correspondentes para cada situação também apresentem desempenho satisfatório. Desta forma, este módulo é constituído como um AE, contendo os principais operadores de um AG convencional, os quais são:

- **Seleção de Pares** : a seleção é de extrema importância para a geração de regras apropriadas, visto que indica quais os indivíduos que irão participar do processo reprodutivo e poderão produzir descendentes. Ter maiores chances de ser selecionado garante, automaticamente, maiores chances de gerar mais descendentes, transmitindo suas características às futuras gerações. Por isso optou-se pela conhecida estratégia da roleta, largamente utilizada em aplicações de AGs. Foi escolhida esta metodologia porque atribui maior probabilidade de seleção àqueles indivíduos considerados mais adaptados ao ambiente (o atributo "ajuste" maior), e o ajuste, no caso da população de regras, é representado por um índice de usabilidade da regra. Ou seja, indica quantas vezes e o quanto esta regra é similar às situações enfrentadas pelo robô durante sua execução. A cada processo reprodutivo são selecionados alguns indivíduos que gerarão os novos indivíduos. O número de reprodutores foi definido para ser fixo durante toda a execução do sistema, porque utilizando esta abordagem inspirada em AGI, o número de reprodutores fixo em cada geração permite manter relativa estabilidade no tamanho da população total.

- **Cruzamento** : a recombinação das características dos progenitores selecionados, gerando novos indivíduos, garante a condução do processo evolucionário, porque está criando novas regras a partir de segmentos de outras regras já consideradas satisfatórias em situações anteriores. Para esta tarefa, são usadas duas diferentes estratégias: médio e uniforme. No cruzamento médio, uma única regra é gerada a partir da média aritmética dos fenótipos dos progenitores, representando uma situação mediana entre estes. Para o cruzamento uniforme, é realizada uma recombinação gene-a-gene dos progenitores, gerando dois novos indivíduos.
- **Mutação** : garantir diversidade genética em um sistema adaptativo para um ambiente não estático é de extrema importância. Portanto, é necessário garantir que não aconteça uma homogeneização da população e que esta tenha indivíduos que representem os mais distantes pontos do espaço de busca, mesmo que estes pareçam momentaneamente sem importância. A mutação colabora com a manutenção da diversidade realizando inversões de valores no cromossomo dos novos indivíduos gerados.

Os elementos do AG implementados neste módulo são justamente os três operadores acima descritos. As outras funções necessárias para o funcionamento de um AG, a "avaliação" e "inserção dos novos indivíduos na população" são realizadas por outros módulos do sistema, visto que este possui muitas particularidades de funcionamento.

Um ponto de extrema importância é que o processo reprodutivo ocorre somente enquanto o sistema necessita evoluir, ou seja, enquanto ocorrem eventos insatisfatórios durante o processo de movimentação no ambiente. A partir do momento que o sistema estabiliza e o robô não mais apresenta situações de perigo ou de não acompanhamento de paredes, a sub-população de "regras consolidadas" controla a movimentação e as demais vão sendo descartadas. Desta forma, o sistema acaba por possuir a característica de, durante sua execução, apresentar "gerações sem reprodução".

4.4.3 Módulo de Controle Populacional

Deve-se ressaltar que um módulo de controle populacional se torna necessário para que a população não apresente crescimento descontrolado.

O sistema desenvolvido foi projetado para que sua população de regras possua tamanho variável. Para tanto, se faz necessário implementar estratégias que permitam controlar este tamanho, bem como a qualidade do conjunto de indivíduos existentes na população, a fim de que não ocorra um crescimento exagerado no número de indivíduos, bem como não aconteça uma situação de homogeneização da população. Estas duas situações inviabilizariam o funcionamento do sistema. Desta forma, o módulo de controle populacional executa as seguintes tarefas:

- verificar a existência de indivíduos com antecedentes iguais;
- inserir novos indivíduos;
- agrupar atributos de indivíduos iguais;
- verificação da idade de todos os indivíduos

- decrementar a idade de todos os indivíduos;
- eliminar indivíduos que chegaram à idade limite;
- criar novo indivíduo;

4.4.3.1 *Verificação de Existência*

Assim que o Módulo de Reprodução gera novos indivíduos e os transfere para o Controle Populacional, este efetua uma consulta exaustiva em toda a população, para verificar a existência, ou não, de um exemplar com o segmento antecedente idêntico ao que foi gerado. Isto acontece, porque o sistema prevê que não possam existir duas regras iguais para a mesma situação.

4.4.3.2 *Inserir*

Caso a consulta para verificação de existência tenha resultado negativo, o novo indivíduo é automaticamente inserido na população. Deve-se observar que, devido à abordagem de controle por idade, não existe uma "substituição" de indivíduos, porque os progenitores das novas regras continuam fazendo parte da população. O que acontece é que, após o processo reprodutivo, a população tem seu tamanho incrementado. No entanto, mesmo que todos os novos indivíduos criados, diferentes dos existentes, sejam automaticamente inseridos na população, esta não sofre um crescimento contínuo e descontrolado, porque a estratégia da idade para cada regra se encarrega de controlar o número de indivíduos da população corrente.

4.4.3.3 *Agrupar*

Se já existe na população um indivíduo com o segmento antecedente igual ao antecedente da regra que acaba de ser criada, este novo exemplar não é inserido, para prevenir a existência de duas ações distintas para uma mesma situação. O que ocorre nestes casos, é que o valor correspondente à idade que seria atribuída ao novo indivíduo é somada ao mesmo atributo do indivíduo já existente. Desta forma, a geração de regras com antecedente iguais às já existentes faz com que se prolongue ainda mais a permanência destas na base de conhecimento. A justificativa é que a probabilidade de serem regras relevantes é alta, visto que os progenitores são escolhidos a partir do índice de usabilidade.

Uma questão a ser abordada neste ponto é que, ao comparar as partes antecedentes e considerar regras iguais, não se está avaliando as ações correspondentes que as mesmas trazem consigo. A estratégia utilizada no sistema é que permaneça a ação que estiver vinculada à regra com a maior idade. Desta forma, se a regra já existente na população tem idade menor que o valor inicial atribuído à nova regra, a ação é substituída por aquela criada na reprodução. Caso contrário, se a regra existente já tiver sido utilizada e avaliada com sucesso, o que incrementa sua idade, tornando mais elevado do que em sua situação inicial, sua ação é que permanecerá.

4.4.3.4 *Verificação da Idade*

Após os períodos de Execução e Reprodução, deve ser verificada a idade de todos os indivíduos da população, pelo fato de que a estratégia da permanência das regras na base estar fundamentada no valor correspondente a este parâmetro.

Como a idade inicial para cada indivíduo é definida na sua concepção, isso implica que, se a regra não for utilizada nenhuma única vez no período correspondente à esta idade, em gerações, ela será automaticamente eliminada da população; no caso de ser utilizada, mas com insucesso, essa eliminação pode ocorrer num período ainda mais curto, devido ao decremento ocorrido nos processos de avaliação. Assim, com este tipo de controle, somente permanecem na população aquelas regras que realmente apresentam utilidade ao sistema.

Outra tarefa que acontece nesta verificação é atribuir valor nulo ao grau de adaptabilidade (ajuste) de cada regra, inicializando a verificação da usabilidade dos indivíduos da população para que esta seja novamente avaliada durante o próximo período de execução.

4.4.3.5 *Decrementar*

Todos os indivíduos que possuem o atributo idade com valores maiores do que ("um") têm este decrementado em uma unidade. Ou seja, a cada período de uma geração, a idade de todos diminui.

4.4.3.6 *Eliminar*

Aqueles indivíduos que alcançaram idade igual a somente uma unidade são retirados da população, pois o decremento faria com que atingissem idade igual a zero - o que significa final da vida.

4.4.3.7 *Criar Novo Indivíduo*

Esta é uma função criada para garantir a diversidade genética e o funcionamento do sistema em situações de modificação do ambiente. O sistema cria e insere um novo indivíduo na população sempre que ocorrer uma das seguintes situações:

- sempre que uma determinada leitura dos sensores for efetuada e o sistema não encontrar na população, uma regra com o índice de similaridade mínimo determinado, será criado um novo indivíduo, composto pela parte antecedente extraída da respectiva leitura e ao segmento conseqüente será atribuído o valor da ação da regra que alcançou a maior similaridade à leitura.
- quando ocorre um evento de perigo, será criado um novo indivíduo, composto pela parte antecedente extraída da leitura anterior ao evento e, ao segmento conseqüente será atribuído o valor inverso da ação da regra executada que gerou o perigo.
- quando ocorre um evento de perda da referência da lateral do cenário, será criado um novo indivíduo, composto pela parte antecedente extraída da leitura anterior ao evento e, ao segmento conseqüente será atribuído o valor inverso da ação da regra executada.

O uso desta estratégia permite que, mesmo quando o sistema já esteja funcionando há um longo período e, conseqüentemente, com uma pequena população bem especializada ao ambiente, ele possa ajustar-se a mudanças ocorridas, não dependendo somente das características genéticas existentes na população.

4.4.4 Módulo de Execução

O módulo de execução é responsável pela interação do robô com o ambiente e verificação da utilidade das regras já existentes. Essa interação acontece durante um período de tempo (Período de Execução), definido por um número máximo de iterações (percepção-ação) pré-estabelecido de iterações. Este deve ser um número que permita ao sistema atribuir um índice de usabilidade satisfatório e representativo para cada regra avaliada neste período. Para tanto, este módulo realiza as seguintes atividades:

- sensoriamento;
- escolha de regras;
- atribuição do ajuste;
- ação;
- avaliação;
- segurança.

4.4.4.1 Sensoriamento

A leitura dos sensores é efetuada para dar início à interação entre o robô e o ambiente. Os dados lidos são valores que representam a distância em polegadas entre a lateral do robô e algum objeto. Com estes valores, monta-se uma outra representação, discretizando a situação do ambiente, a partir dos intervalos apresentados na Tabela 4.2. Essa discretização acontece classificando os sinais em quatro categorias: sem sinal de obstáculo, obstáculo distante, obstáculo perto e, por fim, obstáculo muito perto. Esta informação resultante é atribuída a um vetor de caracteres com a mesma estrutura do segmento antecedente das regras da base, para servir como instrumento de comparação com as regras.

Tabela 4.2: Intervalos de classificação da leitura dos Sensores de Navegação

Valor	Interpretacao
0	Muito Distante / Sem sinal de obstáculo
1	Distante
2	Perto
3	Muito perto

4.4.4.2 Escolha de Regras

A informação recebida pelo sensoriamento, que representa a situação corrente do robô no ambiente é utilizada num processo de comparação, efetuando uma consulta na base de regras, testando exaustivamente todos os indivíduos da população, a fim de selecionar a regra mais adequada à situação obtida pelo sensoriamento.

Esta escolha não é a execução de uma consulta com resultado binário (*existe / não existe*), mas sim, uma consulta que permite encontrar/selecionar regras cujos segmentos relativos ao segmento <situação> possam também estar "próximos" da

configuração do ambiente. Isso acontece porque pode não existir na base, uma regra 100% idêntica. Desta forma, a cada comparação, é necessário atribuir um índice de similaridade para cada regra, considerando a informação da situação lida pelos sensores. Portanto, devem ser aplicadas estratégias que implementem características do poder de "generalização" da base de regras, permitindo que situações semelhantes sejam atendidas pela mesma regra.

Desta forma, para o cálculo de comparação utiliza-se a Distância Euclidiana entre os vetores do sinal de entrada e regra a verificar. Assim, tem-se, a partir do número de posições coincidentes e não-coincidentes que cada regra tem em relação à situação capturada do ambiente, o atributo "especificidade" de cada regra. Optou-se por esta alternativa, para criar um sistema bem simplificado, que não utilizasse a sistemática de competição e lance⁴ dos SC clássicos nem que fosse necessário utilizar o modo de generalização a partir do elemento $\{\#\}$ na forma de representação da regra. A equação (4.1) apresenta este cálculo,

$$Espec(\mathbf{Regra}) = \frac{\sum_{i=1}^{nS} (|\mathbf{x}_i - \mathbf{y}_i|)}{nS} \cdot 100 \quad (4.1)$$

onde:

nS representa o número de sensores a comparar,

x é o vetor de valores (decimais) lidos pelo sensores, e

y é o vetor de valores (decimais) correspondentes ao fenótipo dos sensores na regra.

Uma implicação de se utilizar um índice de similaridade, é que pode acontecer um momento que exista mais de uma regra com o mesmo valor ou valores muito próximos, para este índice e que, conseqüentemente, poderiam ser igualmente aplicadas à respectiva situação. Para contornar estes casos, optou-se em agregar mais uma informação ao cálculo de escolha da regra: o atributo de idade. Considerando que este valor representa o quanto esta regra já foi utilizada com desempenho satisfatório, ele pode contribuir de forma significativa para definir o quão boa é uma regra para uma dada leitura sensorial (equação (4.2)).

$$VlrUtil(\mathbf{Regra}) = 0,9 \cdot Espec(\mathbf{Regra}) \cdot M(Regra) \cdot Idade(\mathbf{Regra}) \quad (4.2)$$

onde M é um índice de proporcionalidade progressiva, variando de 0,0 a 0,1, que atribui maior peso quanto maior for a idade do indivíduo, utilizando a equação:

$$M(\mathbf{Regra}) = 10^{-5} \cdot T(Regra)^2 \quad (4.3)$$

onde T é tempo restante de vida da regra.

Deve-se observar que a participação da idade é de, no máximo, 10%, o que impede situações em que uma regra com baixa especificidade, mas já considerada consolidada, seja escolhida em detrimento de outra regra mais específica, mas recentemente criada e com idade mais baixa.

⁴Ver em 2.3.4

4.4.4.3 Atribuição do Ajuste

O índice de similaridade calculado pelo processo de escolha da regra mais apropriada é utilizado para incrementar, a cada iteração, o atributo "ajuste" de cada regra, que foi inicializado com valor nulo na última verificação de idade. Ao final do período de execução, esse atributo indica cumulativamente o quanto cada regra possui de similaridade com as situações sensoriadas, ou seja, é um índice que representa o grau de utilidade das regras durante esta fase, como mostra a equação (4.4):

$$Ajuste(\mathbf{a}) = \sum_{j=1}^{nI} \left(\frac{\sum_{i=1}^{nS} (|\mathbf{x}_i^j - \mathbf{y}_i^a|)}{nS} \cdot 100 \right) \quad (4.4)$$

onde

a representa a regra que está sendo avaliada.

nI representa o número de iterações realizadas durante o período de execução.

4.4.4.4 Ação

Com a regra mais apropriada escolhida, o sistema converte a representação da ação contida no segmento consequente desta regra e a transmite para os atuadores. A conversão é realizada aplicando-se a equação (4.5) sobre os três últimos elementos do segmento e o primeiro elemento identifica o sinal para o valor calculado. Desta forma, este valor está compreendido no conjunto $\{-7, -6, \dots, 6, 7\}$ e representa um multiplicador para a variação do ângulo de direção do robô durante seu deslocamento.

$$Angulo(\mathbf{Regra}) = S \cdot \left(\sum_{i=0}^2 (2^i \mathbf{x}_i) \right) \cdot \Delta\Theta \quad (4.5)$$

onde

$S \in \{-1, +1\}$, de acordo com o valor do primeiro elemento da ação;

$\Delta\Theta$ representa o intervalo de variação do ângulo.

4.4.4.5 Avaliação

Após o robô ter executado sua ação, o sistema realiza uma análise para identificar se o resultado foi satisfatório. Basicamente, ele verifica se continua com referência a algum objeto, se o robô entrou em uma situação perigosa e se realmente se encontra numa situação desejada ou não. Para isso, se faz necessária uma nova leitura dos sensores de navegação, que são os parâmetros para a movimentação, e também, uma leitura dos Sensores de Segurança (no caso do Nomad 200, são sensores de infravermelho). Assim como os sensores de navegação, esses retornam valores contínuos, que são classificados segundo as categorias apresentadas na tabela (4.3)⁵. Isso faz com que o sistema receba uma retro-alimentação, para verificar se a aplicação da ação da respectiva regra obteve êxito, ou seja, mede a Satisfação/Insatisfação do robô com relação à aplicação da regra.

De acordo com esta avaliação, a regra executada pode receber um incremento ou um decremento em sua idade, fazendo com que, desta forma, caso tenha tido sucesso

⁵Os limites dos intervalos Perto e Muito Perto dos Sensores de Segurança coincidem com os mesmos limites de intervalo dos Sensores de Navegação. A diferença básica é que para a segurança é realizado um sensoriamento em uma distância bem menor e com sensores mais adequados para esta distância, enquanto que para a navegação são atingidas distâncias maiores

Tabela 4.3: Intervalos de classificação da leitura dos Sensores de Segurança

Valor	Interpretação
0	Distante / Sem sinal de obstáculo
1	Perto
2	Muito perto
3	Perigosamente perto

na execução da ação, permaneça por mais tempo na base. Mas por outro lado, caso o resultado da ação não tenha sido satisfatório, colocando o robô em uma situação de risco, por exemplo, terá um decremento na idade, como forma de penalização. Isto significa que sucessivos insucessos podem conduzir à eliminação de uma regra da base de conhecimento.

Os critérios utilizados para definir o índice de satisfação incluem:

- índice de similaridade da regra com a situação lida;
- valores da leitura anterior dos sensores de segurança (antes da ação);
- valores da leitura atual dos sensores de segurança (após a ação);
- situações não desejadas (dos sensores), informadas no sistema, que descritas como "perigosamente perto" e "muito distante ou sem atividade sensorial";
- repetição do uso de regras e/ou ações repetidas;
- situações desejadas para cada um dos sensores, também informadas no sistema, a partir dos valores para as leituras (avaliados individualmente, para cada sensor), mostrados na Figura 4.5:

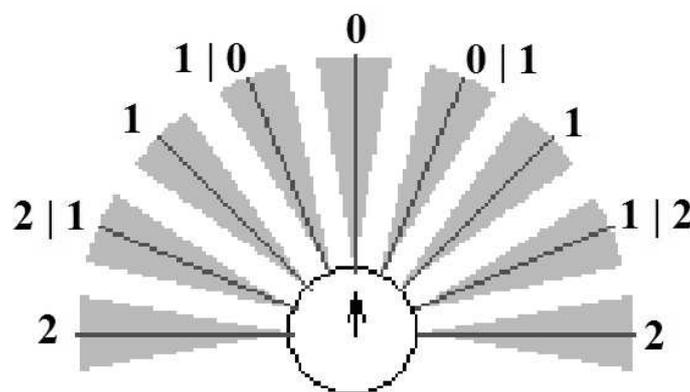


Figura 4.5: Valores considerados satisfatórios para cada um dos sensores

Assim como um AG convencional é diretamente dependente da função de fitness, neste trabalho também esta é a função mais crítica. É a partir desta avaliação que as regras satisfatórias permanecem na base ou, caso contrário, são eliminadas. Considerando que esta aplicação objetiva capacitar um robô a deslocar-se em um

ambiente, preferencialmente acompanhando as paredes, a função de avaliação foi codificada, a partir das variáveis acima citadas, com as etapas apresentadas na Figura 4.6.

1. Verificação de Regras Repetidas
 - Se Regra foi executada repetidas vezes além do limite estabelecido
 - A Regra é eliminada
 - Termina a avaliação
2. Verificação da Situação Atual
 - Leitura dos Sensores de Navegação
 - Verifica a semelhança com a situação desejada
 - Comparar com a situação anterior ao movimento executado
3. Verificação de Perda de Referência de Objetos
 - Se todos os sensores apresentam sinais de "Muito Distante"
 - Se Regra Anterior apresentava obstáculos
 - Se Regra Anterior foi executada repetidas vezes
 - Regra Anterior é eliminada
 - Se "Muito Distante" repete além do número limite de passos
 - Penaliza Regra anterior
 - Termina a avaliação
4. Verificação de Situação de Perigo
 - Leitura dos Sensores de Segurança
 - Se situação de perigo é detectada em algum dos sensores
 - Aciona o módulo de segurança
 - Se for uma regra consolidada
 - Seu tempo de vida é reduzido
 - Senão
 - A Regra é eliminada
 - Termina a avaliação
5. Atribuição dos Valores de Satisfação/Insatisfação
 - Calcula a Satisfação Atual pelas leituras desejadas
 - Atribui a soma destes valores ao totalizador de incremento da regra
 - Se Satisfação Atual for menor que Satisfação Anterior
 - Diminua valor do incremento;
 - Senão
 - Se for maior que a satisfação para o movimento anterior
 - Acrescentar valor ao incremento
 - Aplicar Incremento ao tempo de vida restante da Regra

Figura 4.6: Código Procedimento de Avaliação.

4.4.4.6 Procedimento de Segurança

Utilizando os dados obtidos pela leitura dos sensores de segurança, obtidos no processo de avaliação, o sistema executa um controle de segurança básico, para permitir que o robô não possa vir a se encontrar numa situação perigosa e vir a ser danificado ou, até mesmo, destruído por ter executado uma ação indevida. Este comportamento poderia ser visto como um instinto básico de sobrevivência - inerente ao robô - que serve para evitar estas situações de alto-risco, como uma

colisão violenta, por exemplo.

Quando o robô se encontrar em uma situação considerada perigosa, identificada pela leitura dos sensores de segurança, o sistema pára completamente todos os motores e avalia quais os sensores que apresentam valores de leitura de obstáculos, independentemente de sua distância, e calcula qual a lateral do robô que apresenta o menor valor de somatório de proximidade das leituras destes sensores. Com isso, executa um giro sob o próprio eixo para que, na próxima translação, possa se afastar destes obstáculos perigosamente próximos.

Em um caso extremo, como o de acontecer efetivamente uma colisão, o sistema encerra a execução imediatamente.

4.4.5 Critérios para Encerramento da Execução do Sistema

É interessante observar que a proposta deste trabalho é que o processo aconteça de forma contínua, ou seja, enquanto o robô existir e estiver interagindo com o ambiente, novas regras poderão surgir, de acordo com a necessidade vigente, e regras obsoletas estarão sendo eliminadas - de forma totalmente autônoma.

Um processo continuado implica na não utilização de uma "condição de parada" para o sistema. O que acontece, no entanto, para a validação dos experimentos deste trabalho, é uma avaliação do número de iterações satisfatórias realizadas pelo agente, para definir uma pausa no processo de descoberta de novas regras, introduzindo um período de execução prolongado, até que aconteça algum movimento insatisfatório, provocando o reinício da descoberta de novas regras.

Desta forma, definiu-se que o experimento é finalizado quando:

- for ultrapassado o limite de gerações estabelecido para o experimento;
- for executado um número mínimo de movimentos ótimos; ou
- for executado um número mínimo de movimentos bons (maior do que o de movimentos ótimos).

Para isso, considerou-se arbitrariamente a seguinte classificação:

- **um movimento ótimo** é aquele realizado com sucesso por uma regra consolidada (com tempo restante de vida superior a 900);
- **um bom movimento** é aquele realizado com sucesso por uma regra ainda não consolidada.

Durante os experimentos realizados tanto em simulação quanto com o robô real, que serão descritos no próximo capítulo, se observou algumas situações onde, ao contornar um obstáculo, ou realizar uma curva convexa, o robô acaba executando alguns poucos movimentos sem perceber objetos (leitura nula), mas, imediatamente, retoma a referência com a lateral do cenário ou o objeto que estava contornando. Assim, constatou-se que uma única leitura nula, ou até mesmo, uma breve seqüência destas, não pode ser considerada um movimento realizado sem sucesso. Para evitar uma avaliação negativa, definiu-se mais um parâmetro para o sistema, que especifica o limite para o número de iterações desse tipo entre as iterações que encontram objetos. Esta característica também foi incluída na função de avaliação, como pode ser visto acima.

4.4.6 Fluxo de Execução do Sistema

Por fim, a Figura 4.7 apresenta, de uma forma bastante simplificada, o fluxo lógico de todo o sistema de aprendizagem evolucionária implementado, relacionando as funções dos módulos anteriormente descritos, que constituem a arquitetura do sistema.

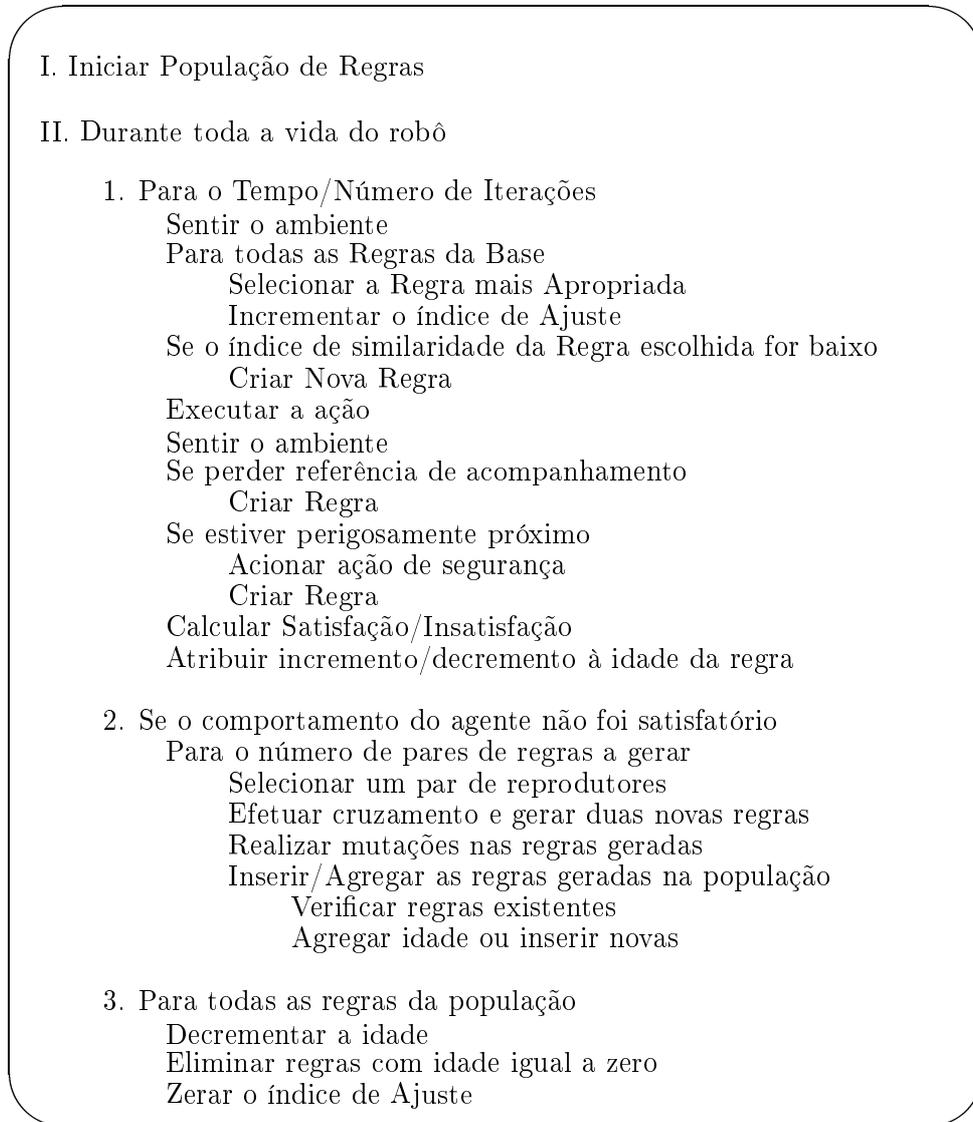


Figura 4.7: Código Básico do Sistema Robótico.

5 EXPERIMENTOS E RESULTADOS

Este capítulo descreve a metodologia utilizada nos experimentos realizados com o objetivo de validar o algoritmo, bem como para ajustar os parâmetros do sistema e do robô. Para cada conjunto de testes, simulados e reais, são apresentadas as configurações dos cenários, do sistema e dos parâmetros do robô. Na sequência, estão descritos os resultados obtidos nos experimentos, bem como a análise destes resultados.

A sequência de conjuntos de testes realizados foi a seguinte:

- testes em simulação, em ambientes simples, similares aos disponíveis para o robô real, para observar e avaliar o desempenho dos algoritmos;
- testes com o robô real, em ambiente esparsos, objetivando, confirmar o funcionamento do sistema no mundo real;
- testes com o robô real, inserindo obstáculos no ambiente;
- testes com o robô real, para observar o comportamento contínuo do sistema;
- testes com o robô real, para observar a validade do conhecimento prévio.

5.1 Ambientes - Cenários de Manobras

Os cenários de manobras para o robô, sejam eles simulados ou reais, devem ser projetados com o objetivo que este possa executar movimentos com diferentes graus de complexidade, tais como andar aleatoriamente, acompanhar paredes em uma e nas duas laterais, e, realizar curvas côncavas e convexas.

Os cenários reais foram montados a partir da disponibilidade de espaço físico no Instituto de Informática da UFRGS e, para os experimentos simulados, optou-se em recriar as configurações dos ambientes reais, os quais são apresentados na Figura 5.1.

Nestes cenários podem ser observadas as seguintes características, segundo sua identificação:

- a) Os cenários mais simples são formados por ambientes esparsos, os quais são constituídos somente por quatro paredes, apresentando, desta forma, baixa complexidade, bastando, para que o robô atinja seu objetivo, que ele simplesmente deixe apresentar situações de perigo e, de vagar aleatoriamente pela região central do cenário (onde os sensores não observam obstáculo algum) e aprenda a acompanhar as paredes. O acompanhamento é resultado da combinação de movimentos de aproximação e repulsão, fazendo que ele não esteja

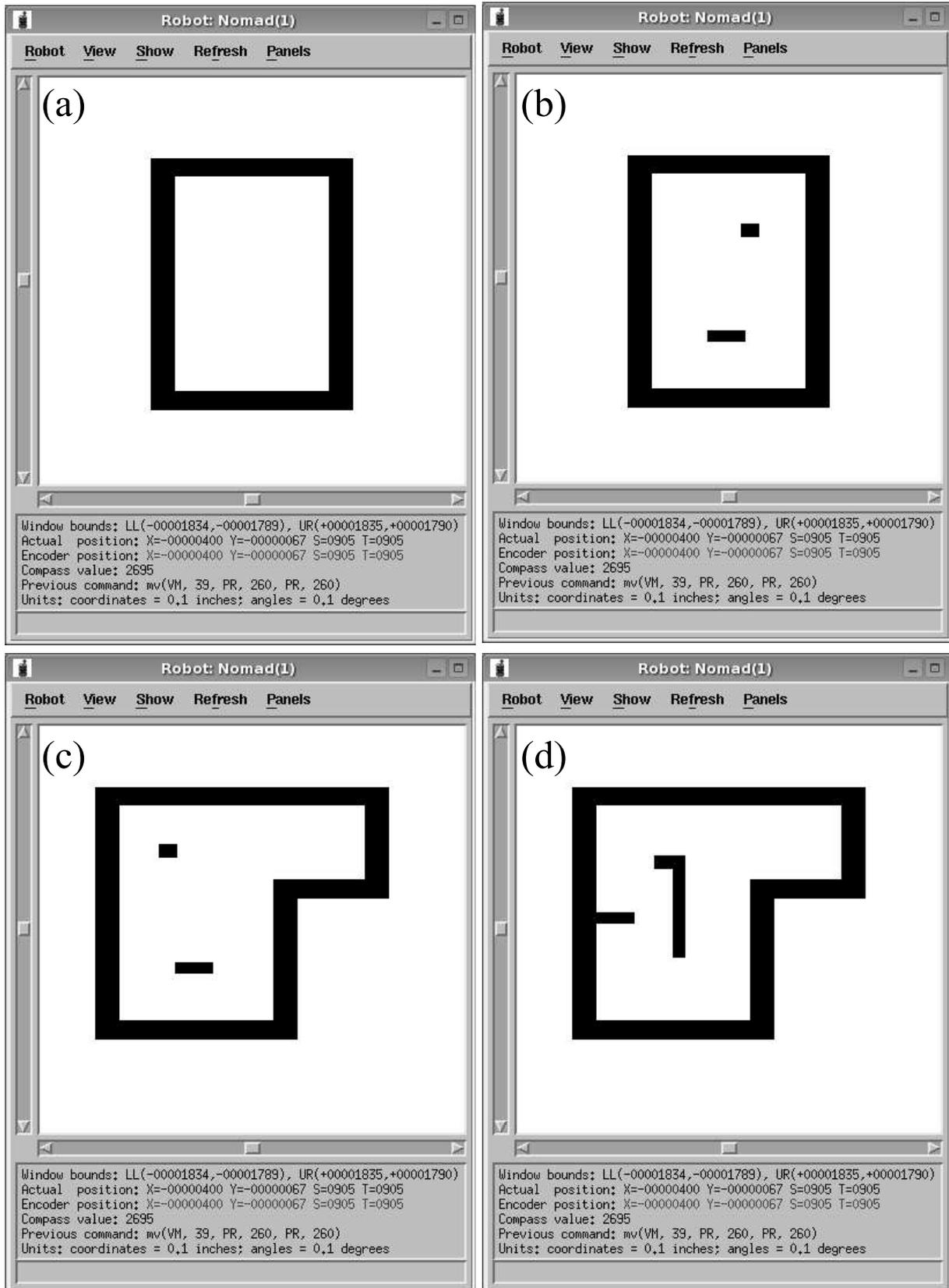


Figura 5.1: Exemplos de Cenários para os Experimentos Simulados

tão próximo da parede ao ponto de ativar o instinto de segurança e, também, não esteja tão distante que não consiga mais identificar a respectiva lateral. Nesta configuração, o robô executa, além do acompanhamento das laterais do cenário, com movimentos de aproximação-repulsão, somente curvas convexas. Estas curvas são caracterizadas por manobras de distanciamento das paredes. Outro ponto a observar neste tipo de cenário é que as regras geradas considerarão satisfatório manter os sensores ativos somente em um lado.

- b) Outra situação é quando ele se depara com um corredor, que mantém duas paredes paralelas muito próximas, permitindo que o o robô passe, mas tendo leitura de obstáculos nos sensores das duas laterais simultaneamente. Isto faz com que ele tenha de desenvolver um comportamento bem diferente das situações anteriormente descritas: necessidade de coordenar um movimento para a frente e considerar isto satisfatório quando tanto os sensores da direita quanto os da esquerda estão ativos. Neste exemplo também existe um obstáculo ainda mais próximo de um dos cantos do cenário, tal como uma coluna que deve ser contornada ao realizar o deslocamento pelo perímetro.
- c) Em situações onde existem, ou obstáculos perpendiculares à uma das paredes, ou ainda, como neste cenário, um canto aberto, a complexidade do ambiente é incrementada consideravelmente, em razão dos novos tipos de manobras necessários. A principal mudança é que neste cenário ele deve, além das curvas convexas, realizar também curvas côncavas, que são efetuadas com movimentos que aproximam o robô das paredes, mantendo uma distância segura para não gerar uma colisão e, ao mesmo tempo, não permitindo que a distância faça com que os sensores deixem de observar os obstáculos. Neste cenário ainda existe o problema do caminho sem saída (beco), que exige uma manobra de retorno.
- d) Podem ainda ser constituídos cenários densos com características de um labirinto, onde, além de alguns caminhos não ter saída, também exige manobras de acompanhamento de paredes paralelas em curvas (zigue-zague), aumentando ainda mais o grau de dificuldade de movimentação e, conseqüentemente, a complexidade para alcançar um conjunto de regras satisfatórias para gerar movimentos sem risco de colisão.

Em todos os experimentos simulados, o robô inicia o processo na posição central do cenário, o que pode ser visto como uma situação irreal, visto que, na maioria das vezes, quando um agente é inserido em um ambiente, isto ocorre a partir de alguma extremidade.

O aspecto final dos cenários reais construídos pode ser observado nas fotos da Figura 5.2. A primeira foto mostra a arena para o robô e ao lado, o computador onde é executado o sistema, juntamente com a base de rádio para a comunicação computador-robô. Nas fotos seguintes, são mostrados a disposição dos obstáculos, o espaço que o robô possui entre as laterais e os obstáculos e, por fim, um beco, onde o robô, não tendo saída, é obrigado a executar um movimento de retorno.



Figura 5.2: Imagens dos Cenários para os Experimentos com o Robô Real

5.2 O Robô NOMAD 200

Antes de apresentar a arquitetura de controle e os detalhes de cada componente, se faz necessário conhecer as características, funcionalidades e restrições do robô utilizado neste trabalho, porque estas informações servem de parâmetro para a definição de sensoriamento e atuação do agente robótico no ambiente.

O robô **NOMAD 200** (Figura 5.3), é um sistema robótico móvel integrado, equipado com módulos sensoriais de infravermelho e ultra-som, além de outros não

utilizados¹ neste trabalho, tais como câmera de vídeo e laser². Possui hardware dedicado ao controle dos sensores, dos motores e também para comunicação. Existe também um conjunto de software, desenvolvido pelo fornecedor do robô, o qual possui uma interface gráfica e um simulador, que foi utilizado neste trabalho. É uma plataforma bastante interessante para ensinar e pesquisar a Robótica e a Inteligência Artificial e a escolha se deu, principalmente, porque existe um exemplar no Laboratório de Robótica Inteligente da UFRGS.



Figura 5.3: O Robô Nomad 200

O sistema mecânico do NOMAD 200 é composto por uma base móvel que possui três motores, três rodas sincronamente guiadas em um sistema não holonômico com um raio de giro nulo, ou seja, pode girar em torno do seu centro. As três rodas são direcionadas simultaneamente por um único motor e giram sob o controle de um segundo motor. Um terceiro motor controla a posição angular da torre. O robô só pode navegar para frente e para trás na direção de alinhamento indicado pelas três rodas (o que caracteriza os sistemas não holonômicos, como acontece com um carro). Sua velocidade máxima é de 24"/s (60,96cm/s) e uma velocidade de rotação máxima de 60 graus por segundo.

O sistema de sonar é composto por 16 canais, fornecendo informações de proximidade de até 255" (6,477m). Está baseado no tempo de deslocamento do sinal acústico desde o seu envio até o retorno. Cada transdutor possui um raio de ação de 25 graus. A circunferência do robô pode ser coberta pelos 16 sensores e estão

¹Procurou-se modelar o sistema para ser executado exatamente com o hardware disponível no LRI/UFRGS

²Caso estivesse disponível, o laser seria a melhor alternativa, por garantir informações mais precisas do ambiente e ainda, ter capacidade de leituras em longa e curta distâncias.

dispostos conforme a Figura 5.4 , que também apresenta, dentre estes sensores, o conjunto utilizado nos experimentos desta tese.

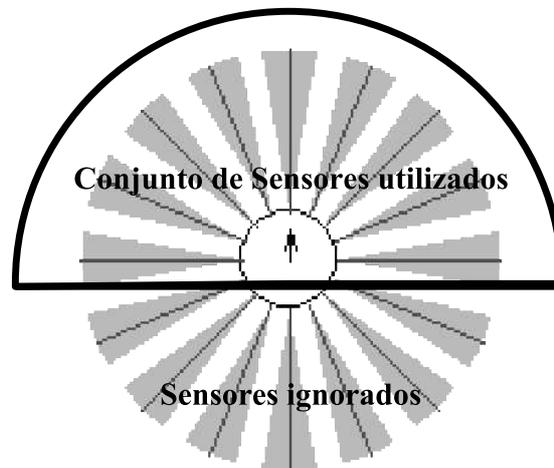


Figura 5.4: Sensores do Robô NOMAD 200

Os sensores de infravermelho formam um sistema composto por 16 canais, com alcance de até 24" (60,96cm), dependendo das condições do ambiente. Cada um dos sensores é composto por dois diodos emissores de luz (LED) e um fotodiodo detector. Pode ser configurado para efetuar uma cobertura total do perímetro do robô (360 graus) utilizando todos os 16 sensores. A distância entre o robô e o objeto é determinado pela intensidade da luz emitida que retorna ao refletir no objeto. Estes sensores estão dispostos da mesma forma que os sonares.

5.3 Identificação e Configuração do Parâmetros

O conjunto de parâmetros do sistema configurado conforme é mostrado na Tabela 5.1 e descritos na seqüência do texto são:

Tabela 5.1: Configuração dos Parâmetros do Sistema

População inicial	10 ou 20 indivíduos
Idade inicial	20 unidades
Operador de Cruzamento	Médio ou Uniforme
Pares para Reprodução	5 pares
Taxa de Mutação	10 %
Similaridade Mínima	60%
Iterações por Geração	500
Limite de Gerações	100
Limite de Leituras Nulas	15
Velocidade do Robô	5"/s

- População Inicial: foram realizados alguns testes para acompanhar e identificar a convergência do AE com diversas configurações de população inicial e, desta

forma, foi definido que para este trabalho o número de 20 indivíduos pode ser satisfatório. Também serão realizados testes com 10 indivíduos na população inicial, para verificar desempenho e diferença neste sistema.

- **Idade Inicial:** considerou-se que 20 gerações é tempo suficiente para que uma determinada regra seja usada ao menos uma vez, para verificar sua relevância.
- **Estratégia de Cruzamento:** Para esta tarefa, será investigada qual a mais adequada, de duas alternativas distintas: a primeira, chamada cruzamento médio, calcula a média aritmética dos valores das situações e das ações dos dois progenitores, mas de forma separada para o segmento antecedente e para o conseqüente. Desta forma, é gerado somente um novo indivíduo, representando uma situação e uma ação intermediárias dos progenitores no espaço. A outra alternativa é o cruzamento uniforme, que realiza uma recombinação do genótipo, gene a gene, desde o antecedente até o conseqüente, gerando dois descendentes distintos.
- **Número de pares para reprodução:** configurado fixo em 5 pares para todos os processos reprodutivos, independente da estratégia de cruzamento utilizada.
- **Taxa de Mutação:** foi definida em 10%.
- **Similaridade Mínima:** sempre que for realizada uma leitura dos sensores de navegação e não existir ao menos um indivíduo com 60% de similaridade na sua parte antecedente, será então criada uma regra a partir desta leitura dos sensores.
- **Número de passos (iterações de sensoriamento-ação)** para cada período de execução foi definido em 500.
- **O número máximo de períodos de execução (gerações)** por experimento foi limitado a 100.
- **Velocidade do Robô:** tanto a velocidade de translação e rotação da base foi determinada em 5"/s.
- **Limites para os intervalos de sensoriamento:** foi identificado que aproximações com distâncias menores ou iguais a 15" (40 cm) do centro do robô - o que significa 6" (15 cm) entre sua extremidade e o obstáculo, ocorriam colisões com alta frequência. Dessa forma, foi considerado o valor de 20" como limite máximo de aproximação, porque nas execuções realizadas desta forma, o robô conseguia abortar um movimento em direção a um obstáculo sem colidir. Os intervalos de sensoriamento, os quais são apresentados na Tabela 5.2 representam a distância, em polegadas do ponto central do robô ao obstáculo. Optou-se por esta forma de representação, ao invés da distância da extremidade, visto que o NOMAD 200 é um robô de formato circular.

5.3.1 O Simulador do NOMAD 200

Assim como o robô, o simulador do NOMAD 200 está disponível. É um software de simulação que mimetiza seu funcionamento de forma satisfatória. Este mesmo ambiente de simulação é utilizado para a comunicação entre o computador

Tabela 5.2: Limites para os intervalos de leitura dos sensores

Lim Inf	Lim Sup	Navegação	Segurança	Interpretação
61	∞	0	-	Muito Distante / Sem sinal de obstáculo
41	60	1	0	Distante
31	40	2	1	Perto
21	30	3	2	Muito perto
0	20	-	3	Perigosamente perto

que executa o sistema e o robô. A Figura 5.5 mostra esta interface. Na janela superior esquerda está o módulo para o usuário criar cenários, dispondo os obstáculos. Abaixo desta janela, está uma janela para a apresentação de mensagens de saída, em modo texto. Ao lado, encontra-se a janela de execução, onde se observa a execução realizada pelo robô, bem como o mapeamento feito pelos sensores. As duas pequenas janelas à direita da imagem, apresentam a leitura dos sensores de infravermelho e sonar.

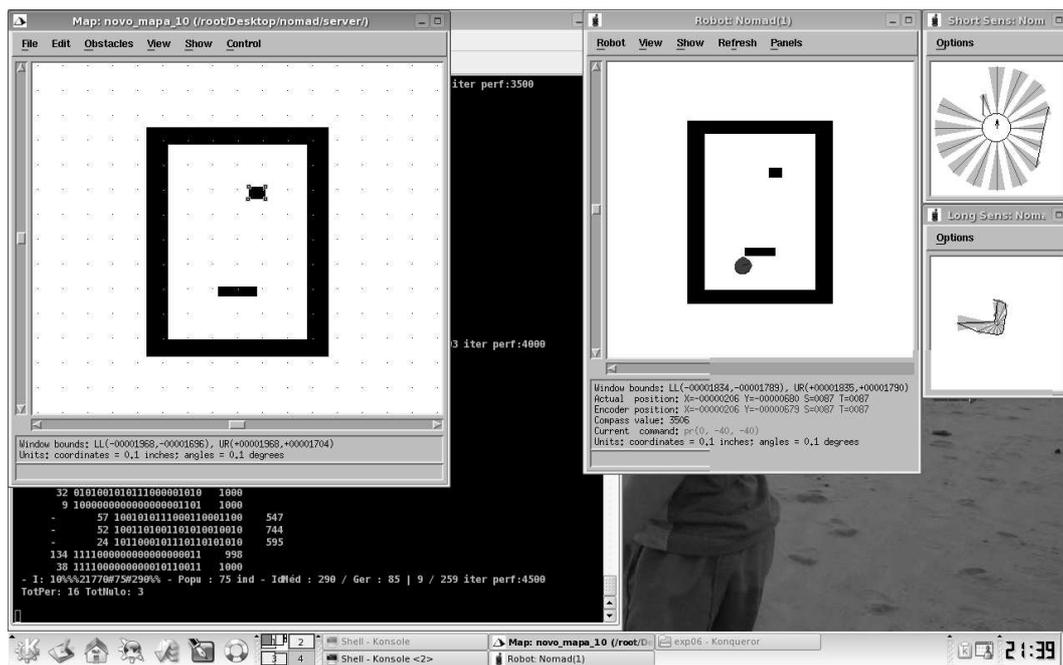


Figura 5.5: Simulador do NOMAD 200

O funcionamento deste software acontece por comunicação entre os programas. Um aplicativo desenvolvido em linguagem C/C++, utilizando bibliotecas de comunicação e controle do robô, é executado e comunica-se com o simulador que executa o respectivo código com um robô simulado, ou então, é responsável pela interface de comunicação com o robô real, tal como mostra a Figura 5.6.

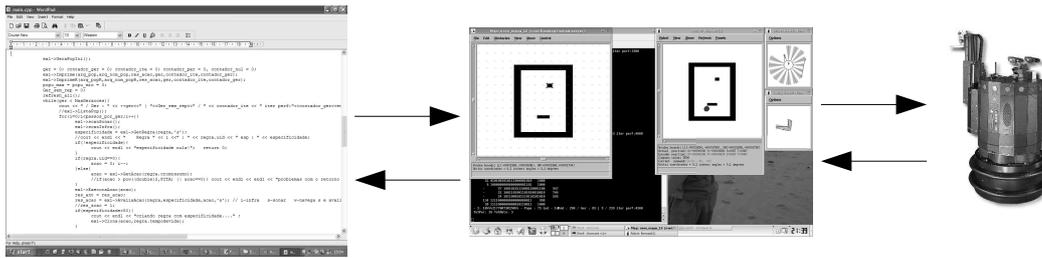


Figura 5.6: Interface de Comunicação com o NOMAD 200

5.4 Experimentos Simulados

Com os parâmetros adequados para o funcionamento do robô já identificados e, conseqüentemente, configurados no sistema, foram iniciados os testes para a avaliação do processo de aprendizagem.

Como se trata de um sistema de execução contínua - em que durante toda a existência do robô, ele estará aprendendo - foram definidos arbitrariamente critérios para determinar o sucesso ou não de cada um dos testes e, desta forma, a condição de parada. O objetivo geral do sistema é fazer com que o robô consiga deslocar-se no cenário sem apresentar situações de perigo ou de movimentação sem referência de alguma parede lateral durante um determinado tempo. Sendo o cenário composto por um ambiente fechado, determinou-se que o objetivo é alcançado somente quando o robô executou um número mínimo de passos de forma satisfatória.

Considerando que nestes experimentos o robô necessita executar pouco menos de 10.000 passos para efetuar uma volta completa no perímetro do cenário, ficou determinado que o objetivo é atingido quando:

- **Resultado 1** : forem executados 10.000 passos satisfatórios, utilizando exclusivamente regras consolidadas;
- **Resultado 2** : forem executados 15.000 passos satisfatórios, utilizando quaisquer regras - consolidadas ou não.

5.4.1 Experimentos Simulados - Ambiente sem Obstáculos

Os primeiros testes foram efetuados utilizando um cenário que mimetiza o espaço disponível para os experimentos com o robô real, que possui dimensões de, aproximadamente, 4.0m x 6.0m. Avaliar o sistema em um ambiente simples como este, executado sobre o cenário apresentado na Figura 5.1(a), permite analisar a regularidade dos resultados. Como a população inicial em todas as execuções é composta pelo mesmo conjunto de indivíduos (variando somente a ação correspondente, que é atribuída aleatoriamente), e a posição inicial do robô também é sempre a mesma, isso faz com que menos fatores aleatórios interfiram no resultado e, conseqüentemente, o perfil das execuções tende a ser muito similar.

Nestes primeiros testes é observado o desempenho a partir da variação do tamanho da população inicial e da metodologia de cruzamento no processo reprodutivo. Também são analisados quais os conjuntos de regras gerados e que situações de leitura dos sensores são contempladas.

5.4.1.1 Experimento 1.1

Os objetivos deste experimento é observar o desempenho do algoritmo para identificar se ele alcança o resultado a que se propõe o sistema, analisar o comportamento da população de regras durante todo o processo e verificar qual o perfil do conjunto de regras consolidadas gerado ao final da execução.

Os parâmetros testados neste experimento são:

- População inicial: 20 indivíduos
- Técnica de cruzamento: Médio

Foram realizados vários conjuntos de testes com esta configuração, os quais, em sua totalidade obtiveram êxito no limite de gerações estabelecido. A Tabela 5.3 apresenta os resultados obtidos em um conjunto de 10 execuções, relacionando os seguintes valores:

- Exec: número da execução;
- Ger: total de gerações necessárias para alcançar a estabilidade;
- Per: total de situações de perigo ocorridas³;
- Nul: total de situações com leitura nula dos sensores ocorridas;
- No Regras: conjunto de regras consolidadas ao final da execução;
- Pop Final: conjunto total de regras - População ao final da execução;
- Id Méd: idade média da população;
- Maior Pop: tamanho da maior população durante a execução;
- Resultado: classificação do resultado da execução.

Tabela 5.3: Resultados do Experimento 1.1

Exec	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop	Resultado
01	27	3	1	3	26	235	42	1
02	39	1	0	4	12	431	24	1
03	29	4	1	3	30	271	45	1
04	41	4	1	7	30	389	47	1
05	38	6	1	6	40	392	58	1
06	31	1	0	6	15	518	25	1
07	31	3	2	5	36	273	45	2
08	27	1	0	4	15	408	24	1
09	29	4	1	3	36	228	44	1
10	33	4	0	6	27	412	42	1

Os resultados mostram que o número de gerações necessário varia pouco entre as execuções e que o conjunto de regras consolidadas também mantém certa regularidade. Na maioria deles, houve somente um evento onde o agente perdeu a referência

³Situações cuja ocorrência ativa o procedimento de segurança descrito em 4.4.4.6

de acompanhamento da parede (leitura nula) e, normalmente, mais de um evento onde ocorreu situação de perigo.

A quantidade de situações de perigo e de leitura nula nos sensores, justamente as que provocam a execução do processo reprodutivo demonstram a variação no tamanho da população do sistema - quanto mais situações indesejáveis, mais processos reprodutivos são disparados e, conseqüentemente, aumenta o número de indivíduos na população. Desta forma, enquanto o sistema não atinge um estado de equilíbrio no comportamento do agente, o tamanho da população cresce; até que ocorra esta estabilidade, quando, por conseqüência do desuso das regras irrelevantes, estas vão sendo eliminadas e, por isso, a população vai diminuindo. Isto é facilmente identificado observando que os valores máximos atingidos pela população são sempre maiores do que a população existente ao final da execução. Em 30% dos casos, inclusive, a população final foi até menor do que a população inicial.

A Tabela 5.4 mostra as médias, valores mínimos e máximos, bem como o desvio padrão calculados a partir destes resultados.

Tabela 5.4: Totalização do Experimento 1.1

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	32,50	3,10	0,70	4,70	26,70	355,70	39,60
Mínimo	27,00	1,00	0,00	3,00	12,00	228,00	24,00
Máximo	41,00	6,00	2,00	7,00	40,00	518,00	58,00
Desvio Padrão	5,10	1,66	0,67	1,49	9,79	97,32	11,46

A diferença existente no número de gerações entre as execuções se deve ao critério de parada - que interfere diretamente em todos os outros valores apresentados. Se o robô continuasse indefinidamente a execução, chegaria ao ponto de possuir uma população formada somente pelas regras consolidadas - o que caracterizaria a super-especialização do conhecimento para atender exclusivamente ao ambiente treinado.

O gráfico da Figura 5.7 apresenta a variação da população de cada uma das 10 execuções do experimento. pode-se identificar um comportamento padrão, onde o número de indivíduos cresce continuamente, efetuando, inclusive, alguns saltos (que são os momentos mais críticos, onde sucessivas situações insatisfatórias ocorrem), para depois atingir a estabilidade dos movimentos. Neste ponto, a população passa a decrescer lentamente, porque as regras que estão em uso são aquelas já consolidadas - não terão acréscimo no tempo de vida - e as outras, que não são utilizadas, tem seu tempo de vida decrementado a cada geração. Também pode ser visto o momento em que a população sofre um declínio acentuado, que representa a eliminação dos indivíduos da população inicial que não foram úteis para atingir os objetivos.

Durante cada execução, a idade média da população também sofre variações. A idade média representa o grau de especialização desta população ao problema, ou seja, se a idade média for elevada, isso quer dizer que a maioria das regras está com seu tempo de vida elevado, o que, conseqüentemente, indica que estas regras estão em uso. Se a maioria das regras está em uso, a população está especializada ao ambiente em questão. Na Figura 5.8 pode-se observar que a idade média cresce na medida que processos reprodutivos são disparados, em conseqüência dos eventos insatisfatórios, e também quando ocorrem eliminações de indivíduos da população. Assim, este aumento da idade média coincide com os movimentos de variação do

tamanho da população. pode-se observar que quando a população permanece estável, acontece uma suave queda na idade média, conseqüência do decremento da idade dos indivíduos não utilizados e que estão caminhando para sua eliminação.

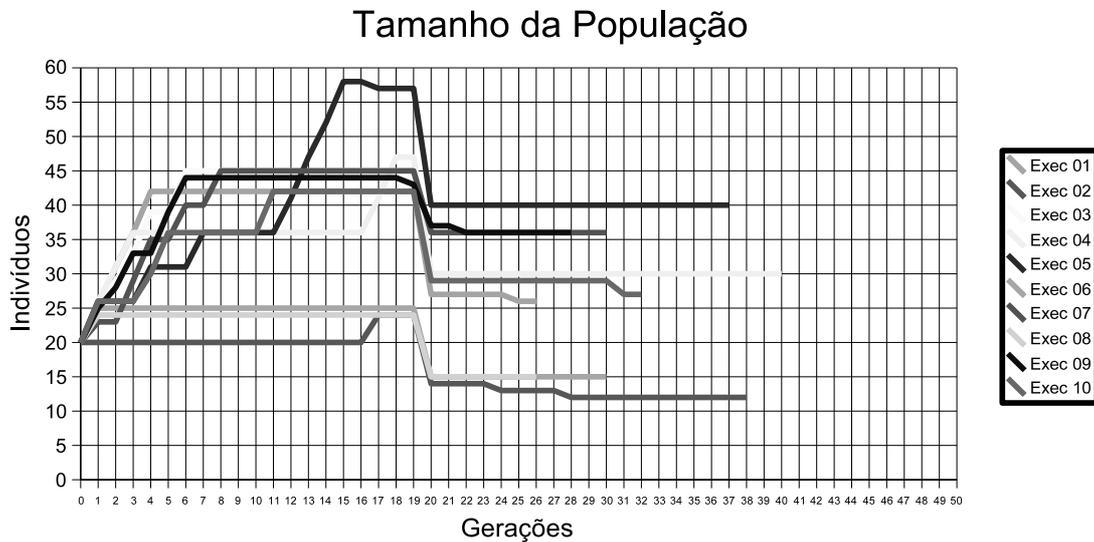


Figura 5.7: Variação da População - Experimento 1.1

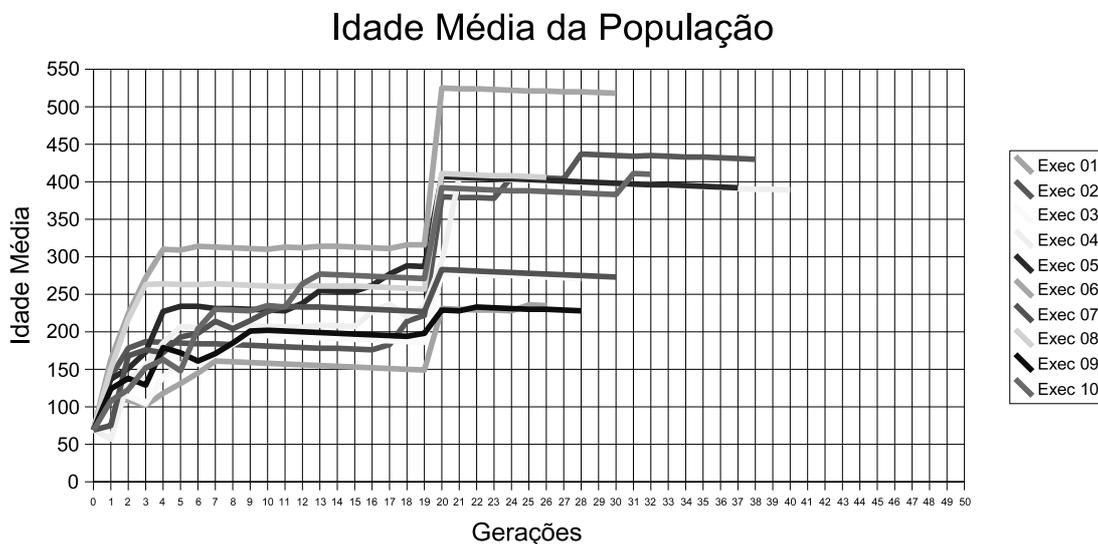


Figura 5.8: Variação da Idade Média da População - Experimento 1.1

A Figura 5.9 apresenta o comportamento de movimentação do robô durante uma execução. Este pode ser considerado o comportamento padrão nestes experimentos: ele inicia os movimentos e passa a confrontar com situações de perigo e leitura nula dos sensores, até que, pelos processos reprodutivos, gere as regras satisfatórias e, conseqüentemente, realize o acompanhamento o número de passos necessário, finalizando com sucesso a execução.

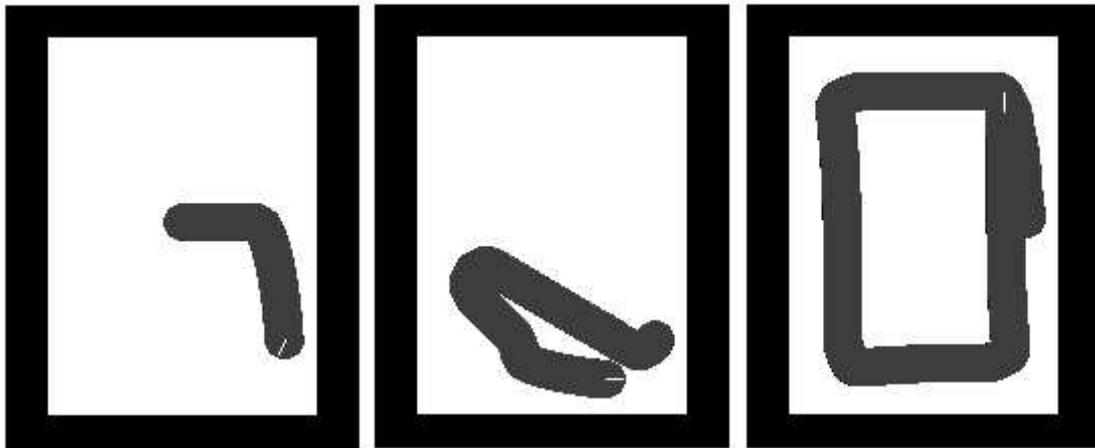


Figura 5.9: Comportamento Padrão do robô - Experimento 1.1

O conjunto de regras geradas ao final de cada uma das execuções, chamado de "Regras Consolidadas", é também muito semelhante, com o seguinte perfil: a "regra zero", acompanhada por mais duas regras, que se destinam ao acompanhamento da parede lateral e as restantes, mais específicas, servem para a execução da curva. A Figura 5.10 mostra um destes conjuntos de regras consolidadas, onde, para cada uma, é apresentado o valor da leitura dos nove sensores e a respectiva ação a ser tomada ao aplicar-se a mesma. Observa-se que este conjunto realiza movimentos de acompanhamento em sentido horário.

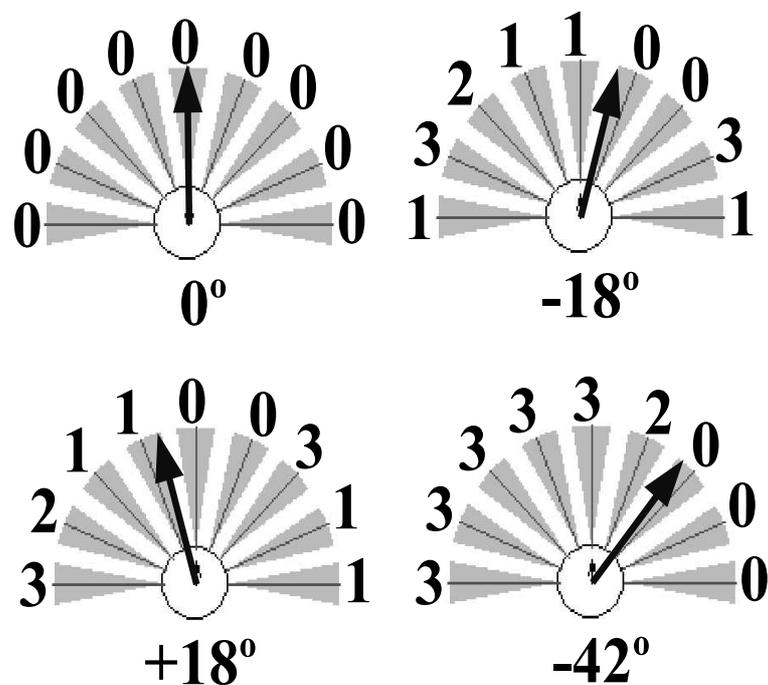


Figura 5.10: Conjunto de Regras Consolidadas - Experimento 1.1

O aspecto inédito deste trabalho - a estratégia de tempo de vida restante para controlar o tamanho da população - pode ser observada na Figura 5.11, que mostra o valor deste atributo para uma parcela dos indivíduos da população (somente aqueles que, em algum momento da execução alcançaram o valor de 300 para o tempo de vida restante), durante cada uma das gerações de uma execução. O comportamento apresentado foi exatamente como esperado: pode-se identificar que uma pequena parcela da população teve seu tempo de vida sendo incrementado, até atingir o estado de consolidação; enquanto que algumas regras foram sumariamente eliminadas e, o restante, teve seu tempo de vida gradativamente decrementado, em direção à sua eliminação da população.

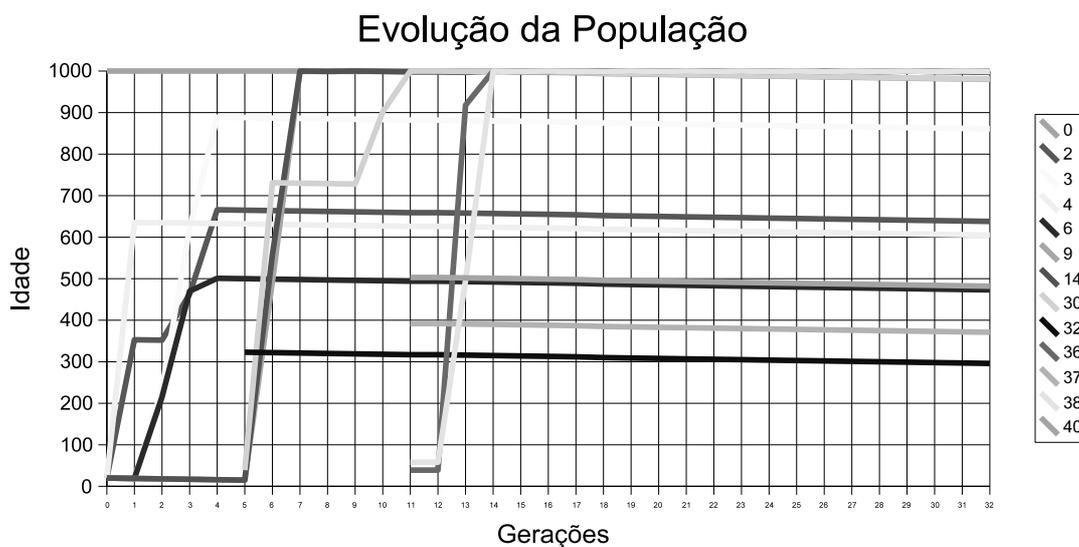


Figura 5.11: Evolução da População - Experimento 1.1

pode-se identificar que a maioria dos indivíduos criados na população inicial acabaram não sendo consolidados e, nem ao menos apresentados neste gráfico, por não alcançarem o patamar de 300 para o tempo de vida restante. Eles foram utilizados nos movimentos iniciais e contribuíram para gerar as novas regras que acabaram por formar o conjunto final. Essas regras que caíram em desuso, foram tendo seu tempo de vida decrementado a cada geração até virem a ser eliminadas.

Por outro lado, as regras que apresentaram utilidade e resultado satisfatório na execução dos movimentos, foram recebendo o incremento até atingir o estágio de consolidadas. Também é visível a eficiência da ação de inserir arbitrariamente uma regra na população quando ocorre um evento insatisfatório. Por exemplo, as regras de número 36 e 38 foram inseridas, quando, a partir de uma determinada leitura (que passou a ser seu segmento antecedente), o robô enfrentou uma situação perigosa, ou então, movimentou-se para o centro do cenário, perdendo a referência da parede a acompanhar. Estas regras foram inseridas na população e, pelo seu uso, foram consolidadas, permanecendo até o final da execução.

5.4.1.2 Experimento 1.2

O objetivo deste experimento é analisar a diferença no desempenho do sistema, ao ser modificada a estratégia de cruzamento no processo reprodutivo. Utilizando o

cruzamento uniforme, cada um dos 5 pares de reprodutores selecionados gera dois novos indivíduos a serem inseridos na população. Os parâmetros modificados neste experimento são:

- População inicial: 20 indivíduos
- Técnica de cruzamento: Uniforme

Com o dobro de indivíduos sendo gerados a cada processo reprodutivo, é evidente que o tamanho da população nestas execuções seja mais elevado do que nas anteriores. No entanto, o critério de parada - número de passos realizados com sucesso - também exerce influência no tamanho da população ao final do experimento, porque, se a execução se prolongasse e mantivesse a estabilidade (sem eventos insatisfatórios), a população convergiria para um pequeno conjunto de regras consolidadas (super-especialização da população).

Também estes experimentos alcançaram 100% de sucesso. Um conjunto de 10 execuções do sistema resultou nos valores apresentados na Tabela 5.5.

Tabela 5.5: Resultados do Experimento 1.2

Exp	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop	Resultado
01	33	3	2	4	38	206	66	1
02	24	2	0	4	23	408	40	1
03	25	1	1	4	23	305	37	1
04	28	2	2	3	44	188	60	1
05	29	2	1	6	32	325	50	1
06	30	4	3	3	50	155	81	1
07	32	3	2	5	51	283	69	1
08	30	0	0	4	7	589	20	2
09	29	1	0	4	17	312	30	1
10	29	3	1	4	42	276	59	1

Os dados nos mostram que o comportamento do sistema é bastante similar ao apresentado anteriormente, onde em quase todos (90%) os testes, o sistema atingiu a condição de parada executando somente movimentos utilizando regras consolidadas.

Uma exceção que ocorreu neste conjunto de testes foi a execução 08 iniciar o processo com regras satisfatórias na população inicial, capazes de alcançar o objetivo do sistema. Desta forma, não aconteceram eventos insatisfatórios e, conseqüentemente, não foram disparados processos reprodutivos. Assim, nenhuma nova regra foi adicionada à população e as regras não utilizadas foram caindo em desuso e, como pode ser visto, a população ao final do experimento se restringe ao conjunto das consolidadas, mais somente três regras. Também a idade média da população é a mais alta entre todas as execuções, devido ao grau de especialização alcançado.

A totalização dos resultados pode ser observada na Tabela 5.6

Como pode ser identificado, a modificação da metodologia de cruzamento para o cenário esparsa não interfere na eficácia do sistema. Todas as execuções tiveram êxito. No entanto, é interessante observar que o comportamento das populações é mais uniforme - não sofrendo tanta variação quanto o experimento utilizando cruzamento médio. Outro ponto que pode-se constatar é que o conjunto com as regras

Tabela 5.6: Totalização do Experimento 1.2

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	28,90	2,10	1,20	4,10	32,70	304,70	51,20
Mínimo	24,00	0,00	0,00	3,00	7,00	155,00	20,00
Máximo	33,00	4,00	3,00	6,00	51,00	589,00	81,00
Desvio Padrão	2,77	1,20	1,03	0,88	14,80	124,29	19,19

consolidadas nesta segunda alternativa sofre menos variação de um experimento para outro, como identifica o desvio padrão.⁴

No entanto, a idade média alcançada neste experimento é inferior à do experimento anterior, sinalizando que a capacidade de especialização do sistema que utiliza o cruzamento médio é superior ao do cruzamento uniforme. Mas, novamente observando o conjunto de regras consolidadas, que é menor, observa-se que o cruzamento uniforme gerou regras mais generalistas.

A Figura 5.12 apresenta a variação no tamanho da população para cada execução do experimento, onde pode ser observado o comportamento similar ao do teste anterior, sem obstáculos: no início a população cresce, até formar um conjunto razoável de regras, para depois decrescer gradativamente. O ponto de queda drástica é a geração em que as regras da população inicial não utilizadas são eliminadas. A Figura 5.13, mostra a variação na idade média da população.

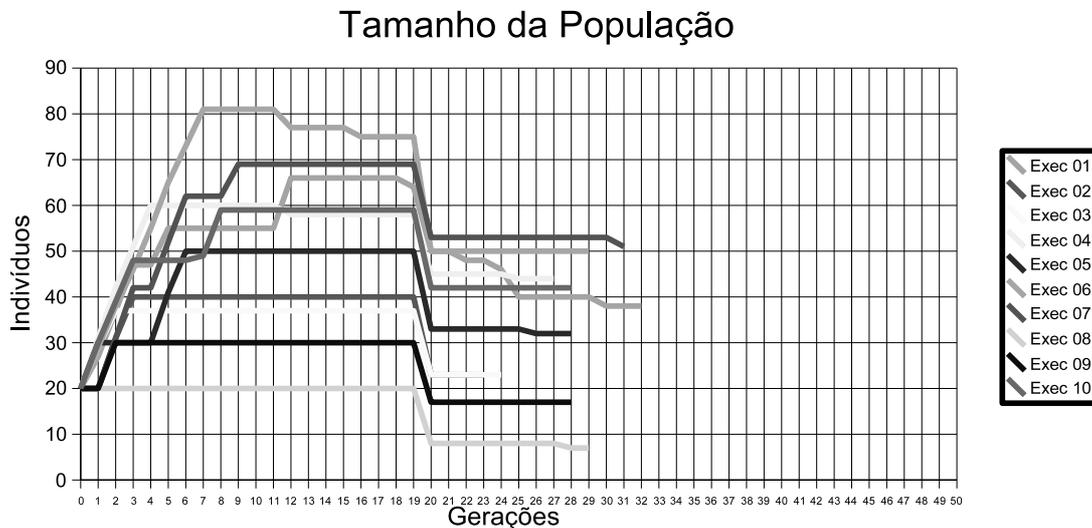


Figura 5.12: Variação da População - Experimento 1.2

5.4.1.3 Experimento 1.3

Para finalizar os testes simulados sem obstáculos, realizou-se um experimento para analisar a profundidade da influência da quantidade de indivíduos na população

⁴Deve se levar em conta que o resultado da execução 08 incrementa o desvio padrão das ocorrências de eventos de perigo e leitura nula, bem como o perfil da população. No entanto, mesmo assim, essa variação é menor neste experimento do que no anterior.

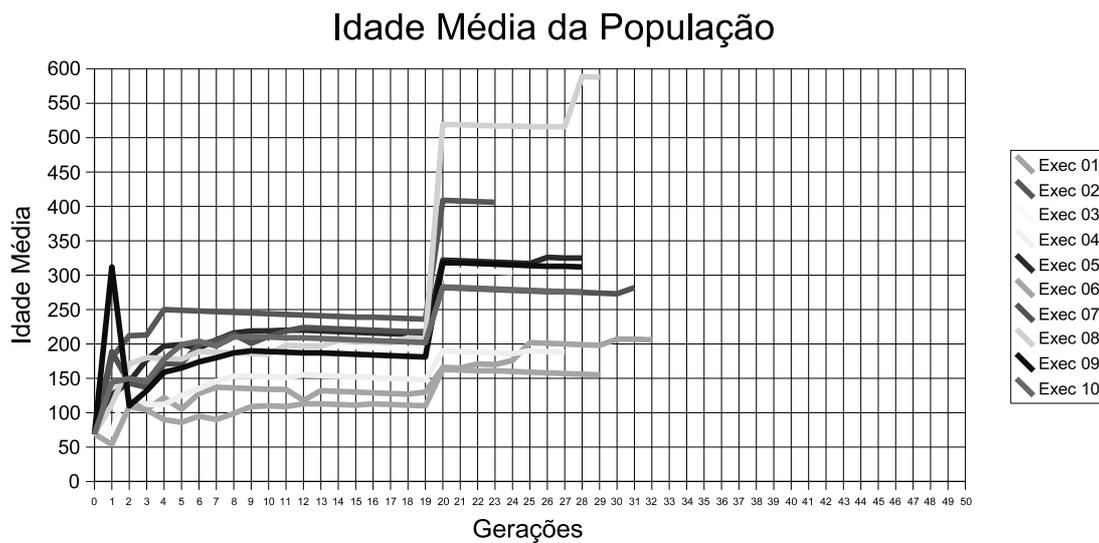


Figura 5.13: Variação da Idade Média da População - Experimento 1.2

inicial. O objetivo é verificar se o sistema tem capacidade de convergir, mesmo iniciando com baixa representatividade no espaço de busca. Isto é importante, visto que nos trabalhos relatados na literatura, o tamanho da população sempre se apresentou como fator crítico, necessitando ser configurado adequadamente. Assim como todos os experimentos, os demais parâmetros são mantidos, variando somente:

- População inicial: 10 indivíduos
- Técnica de cruzamento: Uniforme

A totalização dos resultados pode ser vista na Tabela 5.7.

Tabela 5.7: Totalização do Experimento 1.3

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	26,70	1,60	1,30	3,50	23,00	298,70	36,70
Mínimo	25,00	1,00	1,00	3,00	11,00	193,00	24,00
Máximo	31,00	4,00	2,00	5,00	55,00	415,00	66,00
Desvio Padrão	1,70	0,97	0,48	0,85	12,92	86,56	11,71

Os resultados mostram que para este cenário muito simples, a redução no tamanho da população não só atingiu os resultados esperados, como também colaborou para um ganho razoável nas médias e uniformização do desempenho do sistema, que pode ser constatado pela diminuição dos valores médios, bem como, do desvio padrão entre as execuções.

5.4.2 Experimentos Simulados - Ambiente com Obstáculos

Neste experimento foi utilizado o mesmo cenário do experimento anterior, acrescentando dois obstáculos, resultando no ambiente ilustrado na Figura 5.1(b). O

primeiro obstáculo, posicionado como um pilar em um canto, para que o robô tenha que realizar uma curva em uma situação de sensoriamento frontal e em ambos os lados; e o segundo obstáculo está colocado paralelamente à uma das laterais formando um pequeno corredor, mas com as extremidades razoavelmente distante das laterais perpendiculares. As distâncias entre os obstáculos e as laterais são diferentes, a fim de proporcionar diferentes situações e graus de complexidade. O pilar encontra-se aproximadamente a 1,00 metro de distância de ambos as laterais de seu canto. Já o corredor está a 1,30 metros da lateral paralela a ele e, as suas extremidades estão a 1,5 metros das laterais perpendiculares. Estas diferentes situações de sensoriamento podem exigir, ou um conjunto maior de regras para atender as situações, ou então um conjunto de regras mais generalista, onde cada regra consiga atender a várias situações distintas.

5.4.2.1 Experimento 2.1

O objetivo deste experimento é avaliar o desempenho do sistema, a partir das mesmas configurações utilizadas em ambientes esparsos, mas agora confrontando um cenário com obstáculos. As configurações variáveis utilizadas são:

- População inicial: 20 indivíduos
- Técnica de cruzamento: Médio

Nestas configurações - exatamente iguais às utilizadas no experimento 1.1, o agente alcançou êxito somente em 90% das execuções, o que já demonstra o incremento da complexidade do problema ao serem inseridos obstáculos no cenário e, por conseqüência, limitação para alcançar o objetivo, por parte deste conjunto de parâmetros. A Tabela 5.8 apresenta a totalização dos resultados obtidos em um conjunto de execuções.

Tabela 5.8: Totalização do Experimento 2.1

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	54,44	7,11	2,56	6,78	45,89	357,78	63,44
Mínimo	23,00	0,00	0,00	3,00	7,00	203,00	25,00
Máximo	83,00	15,00	9,00	10,00	90,00	465,00	106,00
Desvio Padrão	19,89	4,68	2,79	2,59	24,55	106,04	26,73

Inicialmente, identifica-se que o número médio de gerações necessário para atingir a estabilidade do sistema aumentou consideravelmente, bem como o número de eventos insatisfatórios enfrentados pelo robô durante cada execução. Isso pode ser explicado pelo fato de que, como o robô deve realizar a volta pelo perímetro do cenário, em algumas execuções acontecem repetidas situações de perigo em um mesmo ponto deste, que ocorrem em voltas subsequentes, até que ele organize seu conjunto de regras e estabilize o comportamento. Novamente houve uma execução que atingiu um tamanho da população menor que a população inicial. Foi também esta população que atingiu o maior valor para a Idade Média, demonstrando seu grau de especialização para o respectivo ambiente.

A variação no tamanho da população, apresentada na Figura 5.14 indica claramente o aumento no desvio médio entre as execuções. O comportamento observado

novamente se repete, exceto pela ausência do ponto de declínio acentuado existente nos experimentos anteriores em algumas das execuções. Essa diferença acontece porque, com a existência de obstáculos e o grau de dificuldade maior, o sistema necessitou selecionar, usar e, conseqüentemente, avaliar um conjunto maior de regras. Dessa forma, aquelas regras avaliadas negativamente já foram sendo eliminadas antes.

A Figura 5.15 apresenta a variação na idade média da população que, mesmo identificando diferenças consideráveis no valor médio para cada execução, o comportamento continua sendo similar, com variações mais acentuadas no início do processo, para depois seguir em crescimento gradativo.

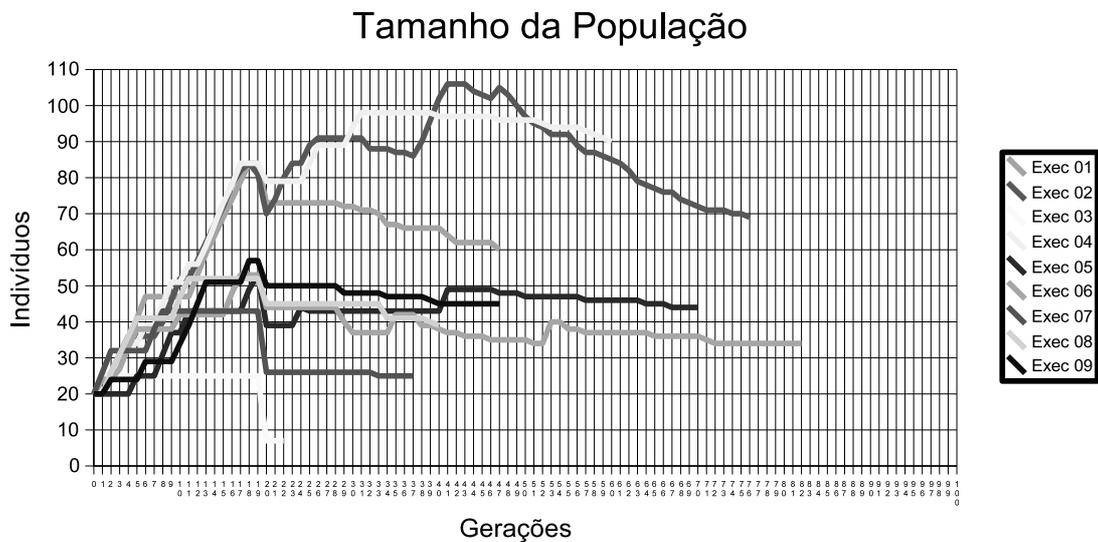


Figura 5.14: Variação da População - Experimento 2.1

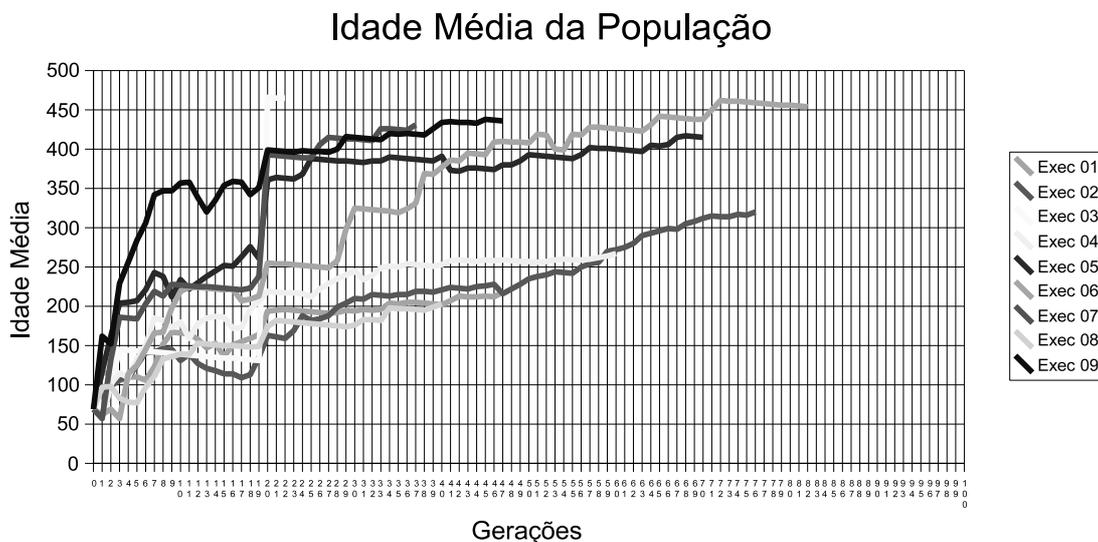


Figura 5.15: Variação da Idade Média da População - Experimento 2.1

A Figura 5.16 apresenta algumas situações onde pode ser observado o comportamento do robô durante as execuções. Neste cenário, o processo acontece da mesma forma que no ambiente esparsos. Ocorrem eventos de leitura nula e o robô se desloca pelo centro do cenário, assim como ocorrem situações de perigo. A diferença mais significativa é que, principalmente para aprender a passar pelos obstáculos, acontecem mais eventos de perigo que nos experimentos anteriores. Como ele está realizando o deslocamento pelo perímetro do cenário, acontecem casos onde o robô necessita efetuar várias voltas até que tenha realmente o conhecimento necessário para não mais ocasionar eventos insatisfatórios.

Ao efetuar a passagem junto a um obstáculo, muitas vezes pode ser observado um comportamento que gera um início de curva em volta deste obstáculo (Semelhante às três imagens superiores da Figura 5.16), como se o sistema estivesse ignorando a lateral do cenário e somente acompanhando o obstáculo. No entanto, ao finalizar esta passagem, o robô retorna a utilizar a lateral como referência para sua movimentação.

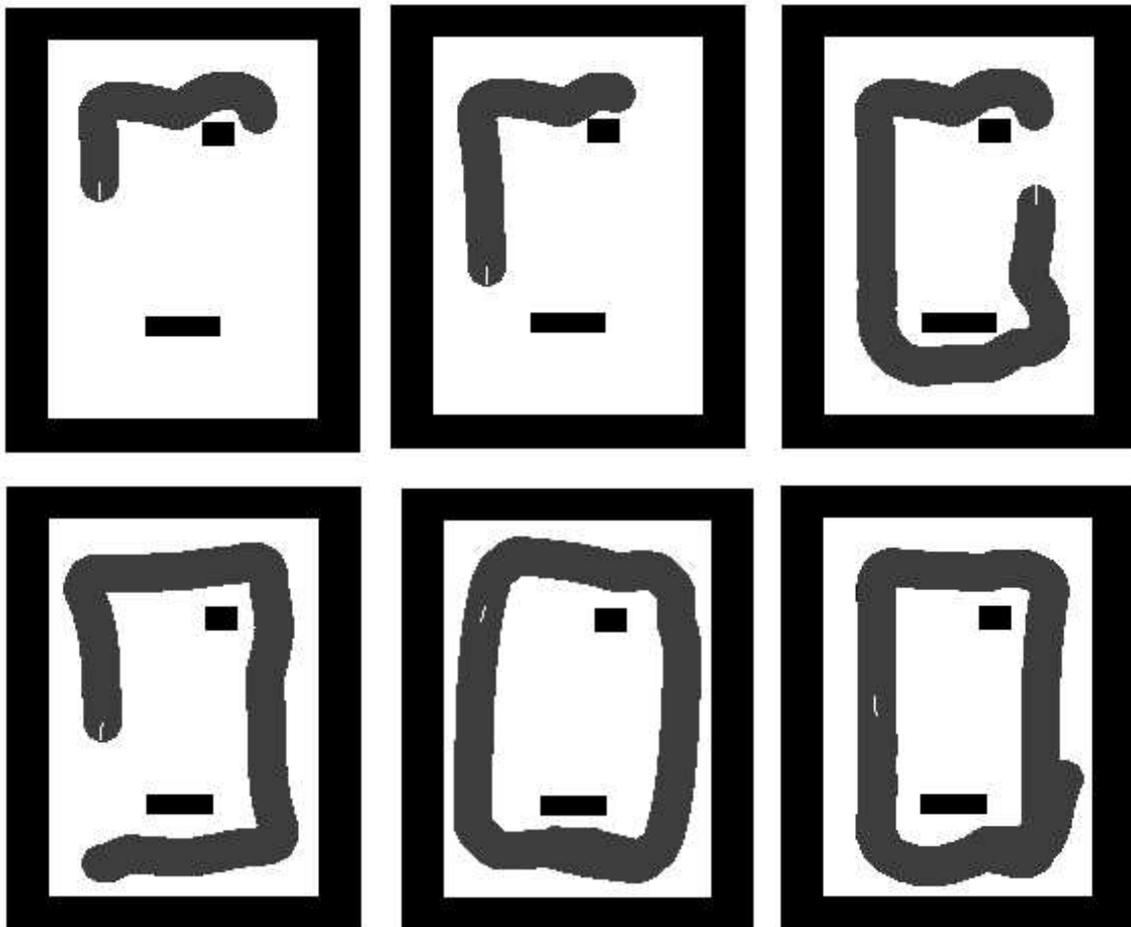


Figura 5.16: Comportamento Padrão do robô - Experimento 2.1

Assim como nos testes anteriores, os conjuntos de regras geradas em cada uma das execuções continuam sendo também similares. Neste caso, em torno de seis regras fazem parte de um bom conjunto. Aqueles que apresentam mais regras consolidadas possuem, entre elas, regras que estão em processo de desuso, mas, como ainda possuíam idade restante acima de 900, continuam na categoria de consoli-

dadas. A Figura 5.17 mostra um destes conjuntos de regras geradas durante uma execução. Observa-se se que este conjunto realiza movimentos de acompanhamento em sentido anti-horário.

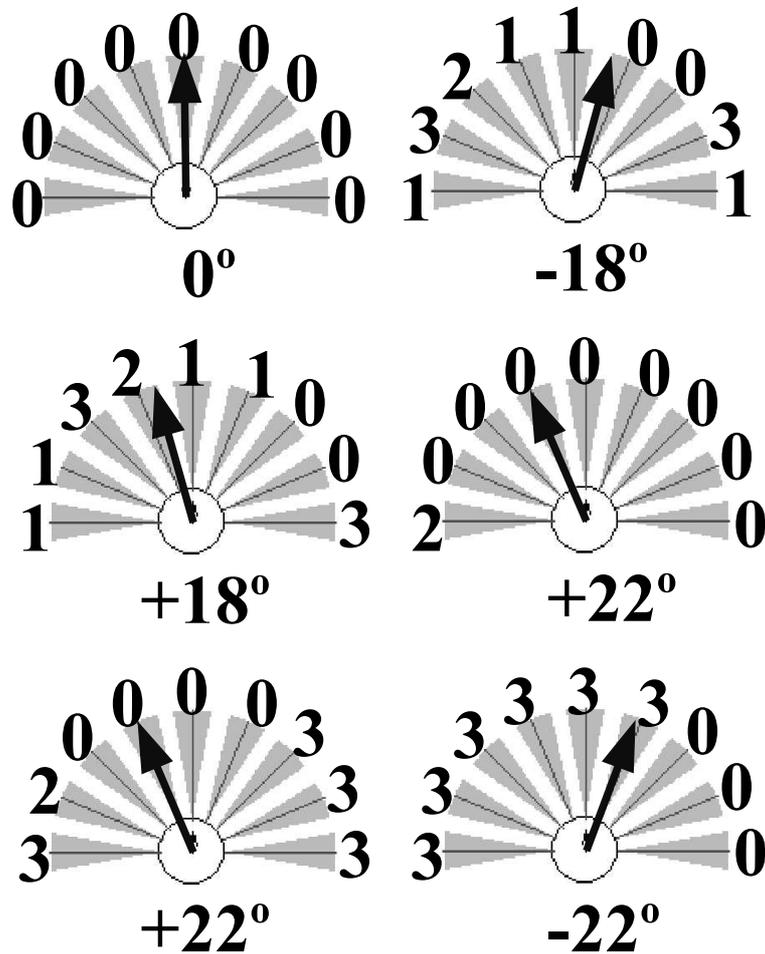


Figura 5.17: Conjunto de Regras Consolidadas - Experimento 2.1

5.4.2.2 Experimento 2.2

Neste experimento, assim como no 1.2, o objetivo é analisar a diferença do desempenho entre o cruzamento médio e o cruzamento uniforme. Assim, a população e o cruzamento neste teste são:

- População inicial: 20 indivíduos
- Técnica de cruzamento: Uniforme

A primeira característica observada neste experimento, antes mesmo dos valores resultantes, é que para esta configuração, o sistema apresenta êxito em 100% dos testes, o que, a priori, já demonstra maior eficiência. A totalização dos resultados é apresentada na Tabela 5.9.

Tabela 5.9: Totalização do Experimento 2.2

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	56,60	5,60	2,30	7,30	52,00	314,30	80,80
Mínimo	34,00	1,00	1,00	5,00	21,00	224,00	50,00
Máximo	73,00	9,00	6,00	10,00	96,00	517,00	118,00
Desvio Padrão	13,71	2,32	1,70	2,00	25,45	81,93	23,04

pode-se observar que, além de garantir sucesso em todas as execuções, novamente, o uso do cruzamento uniforme - assim como nos experimentos sem obstáculos - passou a apresentar uma pequena melhora no desempenho do sistema, principalmente garantindo maior uniformidade nas execuções.

Durante o acompanhamento dos experimentos pôde ser identificado que determinadas situações exigiam muito mais iterações até alcançar um aprendizado satisfatório. Um exemplo disso é quando o robô está acompanhando uma lateral com a distância máxima possível e, à sua frente está um obstáculo paralelo a esta lateral, exigindo que ele se desloque mais próximo da mesma - para apresentar distância segura tanto para esta quanto para o obstáculo. O que acontece é que o robô atinge um estado de perigo repetidas vezes, quando passa pelo mesmo ponto. Isto acontece porque as últimas regras utilizadas e também as situações sensoriadas até o momento do perigo não contemplam explicitamente situações desta natureza. Nesta situação, novamente a inserção do novo indivíduo, contemplando a situação de perigo colabora com o sistema. Certamente, a colaboração não é direta, gerando a regra que faltava para a estabilização do sistema, mas sim, uma regra com código genético que contempla uma situação de maior proximidade dos obstáculos. Desta forma, ela poderá ser de grande utilidade nos próximos processos reprodutivos.

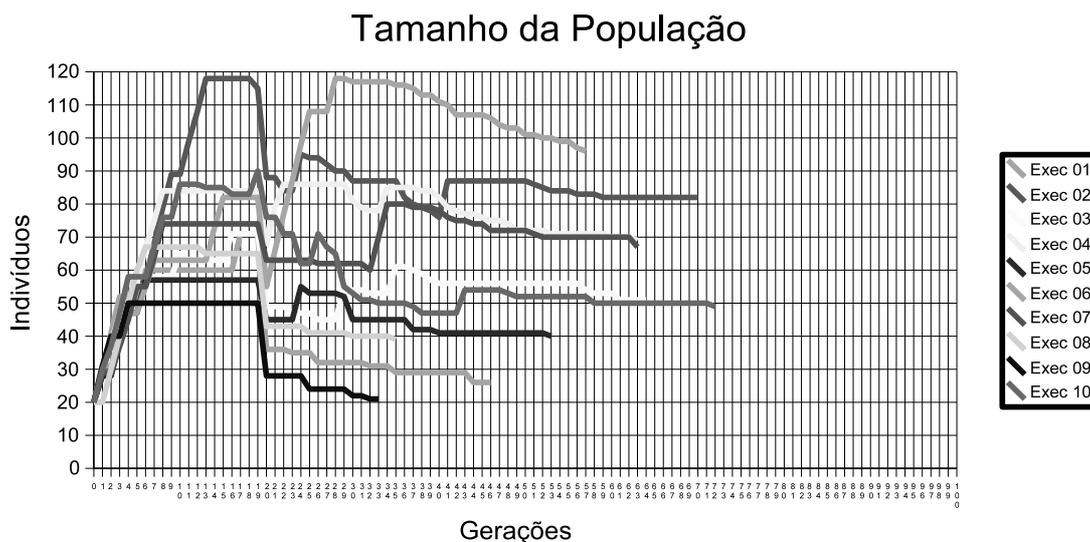


Figura 5.18: Variação da População - Experimento 2.2

A Figura 5.18 apresenta a variação no tamanho da população para cada execução do experimento, onde se identifica mais indivíduos na população - decorrente do uso

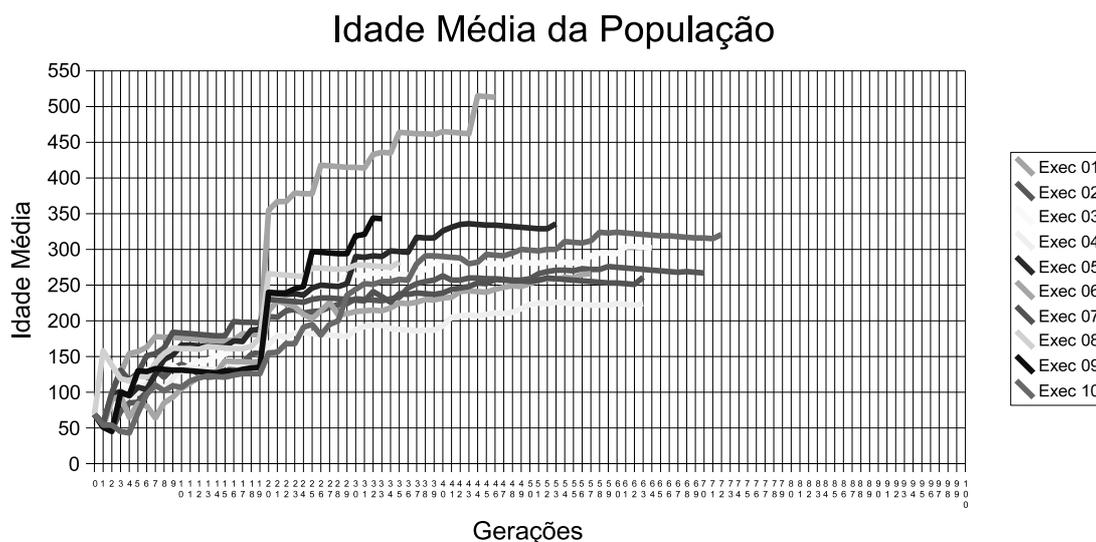


Figura 5.19: Variação da Idade Média da População - Experimento 2.2

do cruzamento uniforme - e a permanência do aspecto de diminuição acentuada no número de indivíduos na geração 20. Isto acontece porque, mesmo testando e usando mais indivíduos nas primeiras iterações do processo, mais regras satisfatórias novas são geradas e, conseqüentemente, algumas daquelas regras da população inicial acabaram não sendo nem ao menos testadas, devido à baixa similaridade em relação às outras. A Figura 5.19 mostra a variação na idade média da população onde, exceto uma execução, as outras apresentaram relativa uniformidade.

5.4.2.3 Experimento 2.3

Por fim, o objetivo deste último experimento simulado é novamente verificar a possibilidade do uso da população inicial reduzida e se este parâmetro proporciona ganho para o sistema também em ambientes com obstáculos. Assim, foi utilizado:

- População inicial: 10 indivíduos
- Técnica de cruzamento: Uniforme

Nestas configurações, somente em 80% das execuções, o robô conseguiu atingir o resultado desejado dentro do limite de gerações estipulado. Provavelmente as outras execuções viriam a convergir, no entanto, em um número muito elevado de gerações, o que demonstra a ineficiência causada pelo reduzido número de indivíduos na população inicial. A Tabela 5.10 mostra a totalização dos resultados obtidos neste experimento.

Estes resultados mostram que, quando confrontado com situações mais complexas, o sistema apresenta resultados menos satisfatórios, quando a população inicial for reduzida. Também corre maior risco de se encontrar em um máximo local, fazendo com que o sistema não convirja para a solução desejada. Isso pode ser constatado pelos 20% de não-convergência, assim como também pela alta variação existente nos resultados que alcançaram êxito.

Tabela 5.10: Totalização do Experimento 2.3

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	52,29	6,57	2,00	8,14	76,14	264,14	88,57
Mínimo	35,00	2,00	1,00	4,00	30,00	224,00	35,00
Máximo	83,00	20,00	3,00	21,00	218,00	339,00	226,00
Desvio Padrão	15,70	6,13	0,82	5,90	66,74	47,65	64,04

5.5 Experimentos com o Robô NOMAD 200

Para comprovar realmente a eficácia do sistema no mundo real, foram realizados testes utilizando o robô NOMAD 200. Como nos experimentos simulados, o uso de uma população inicial reduzida não apresentou desempenho satisfatório para ambientes com obstáculos, todos os testes realizados com o robô real tem este parâmetro do sistema configurado para 20 (vinte) indivíduos.

5.5.1 Experimentos Reais - Ambiente sem Obstáculos

Os primeiros testes foram realizados com o mesmo objetivo da simulação: verificar e comprovar o funcionamento do sistema e também a variação do tipo de cruzamento, mas agora com as características de execução no mundo real. Como foi dito anteriormente, um ambiente simulado é controlado, diferentemente do mundo real, onde a interferência de variáveis aleatórias é grande.

O ambiente real construído para estes testes caracteriza-se como um retângulo, que possui dimensões de aproximadamente 4.0m x 6.0m. Duas laterais são formadas pelas próprias paredes da sala, o que implica em irregularidades, tais como canaletas do cabeamento de rede e de energia elétrica, bem como as caixas para as tomadas elétricas. As outras duas laterais são formadas por placas de madeira. Tanto as placas quanto as paredes possuem cores claras, o que produz um bom resultado para o infravermelho, quando da leitura dos sensores de segurança.

Uma diferença substancial entre os experimentos simulados e reais diz respeito ao número de passos necessários para que o robô realize uma volta completa pelo cenário. Os 10.000 passos do simulador são agora somente 2.500 passos. Através da observação de variação do número de passos em diferentes experimentos, tanto simulados quanto reais, chegou-se a conclusão que isso é diretamente dependente do número de instruções computacionais que cada iteração possui. Quando o sistema é executado plotando resultados no monitor, de forma iterativa, já é possível observar-se alguma variação, mesmo em simulação. E, no caso do robô real, conclui-se que a diferença é maior devido ao processo de comunicação entre o computador base - onde está sendo executado o sistema - e o próprio robô.

No entanto, essa diferença não interfere no desempenho do sistema - somente exige uma quantidade menor de passos e, conseqüentemente, de gerações necessárias para atingir o objetivo. Dessa forma, qualquer análise comparativa entre a simulação e as execuções com o robô real deve ser realizada observando esta proporcionalidade.

Nestes experimentos, como o robô necessita executar pouco mais de 2.000 passos para efetuar uma volta completa no perímetro do cenário, ficou determinado que o objetivo é atingido quando o sistema efetuar:

- **Resultado 1** : 2.500 passos satisfatórios, utilizando exclusivamente regras con-

solidadas;

- **Resultado 2** : 4.000 passos satisfatórios, utilizando quaisquer regras - consolidadas ou não.

5.5.1.1 Experimento 3.1

Os objetivos deste experimento são comprovar o funcionamento do sistema no mundo real, utilizando a mesma configuração do ambiente simulado, com a técnica de **cruzamento médio**, e realizar uma análise comparativa dos resultados, a fim de identificar as principais diferenças entre execuções simuladas e reais.

O sistema alcançou o resultado esperado em 100% das execuções deste experimento, comprovando que, assim como para os ambientes simulados, tem condições de funcionar no mundo real. A Tabela 5.11 apresenta uma bateria de testes, composta por 5 execuções, com os respectivos resultados de cada uma delas. Pode ser observado em todas as execuções que o sistema apresentou comportamento similar - o que comprovou a regularidade do sistema.

Tabela 5.11: Resultados do Experimento 3.1

Exp	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop	Resultado
01	11	2	1	3	38	114	38	2
02	16	4	2	3	53	144	53	1
03	11	2	1	3	35	116	35	2
04	20	3	7	3	66	99	66	2
05	10	4	2	3	31	151	31	2

O número reduzido de gerações necessárias para a estabilização do sistema, em comparação aos testes simulados se deve à diferença no número de passos necessários para a volta completa no perímetro. Um ponto a ser observado neste experimento é que o tamanho da população ao final da execução é sempre igual ao maior tamanho da população. Isso acontece porque o sistema estabilizou antes da vigésima geração - quando os indivíduos da população inicial que não foram usados vão sendo descartados. Outro valor que sofre influência dessa variação é o tempo de vida restante do indivíduo. Como o robô executa menos passos, também é um número menor de vezes que o processo de avaliação das regras acontece. Por isso, a idade média apresentada aqui também é menor que nas simulações.

Para se fazer uma análise mais completa, a Tabela 5.12 apresenta a totalização dos valores do experimento.

Tabela 5.12: Totalização do Experimento 3.1

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	13,60	3,00	2,60	3,00	44,60	124,80	44,60
Mínimo	10,00	2,00	1,00	3,00	31,00	99,00	31,00
Máximo	20,00	4,00	7,00	3,00	66,00	151,00	66,00
Desvio Padrão	4,28	1,00	2,51	0,00	14,57	21,88	14,57

As diferenças identificadas entre o experimento simulado e o real, descartando as provenientes da diferença do número de passos, foram:

- a variação no número de gerações no mundo real é maior, mostrando que as execuções são menos uniformes que na simulação;
- o robô real enfrenta mais situações onde perde a referência do objeto a acompanhar (evento de leituras nulas);
- a grande maioria das execuções reais é finalizada com resultado igual a 2, o que significa atingir o objetivo utilizando não somente regras consolidadas, mas também aquelas em avaliação;
- curiosamente, o número de regras consolidado é igual em todas as execuções;

O gráfico da Figura 5.20 apresenta a variação da população de cada uma das 5 execuções do experimento. Pode se identificar que, assim como na simulação, o comportamento padrão, o número de indivíduos cresce, para depois atingir a estabilidade dos movimentos. Aqui não há a fase de descréscimo da população, porque não houve gerações suficiente para isso.

Na Figura 5.21 pode se observar que a idade média também cresce na medida que processos reprodutivos são disparados, para quando o sistema atingir a estabilidade, acontecer uma suave queda, em consequência do decremento da idade dos indivíduos não utilizados.

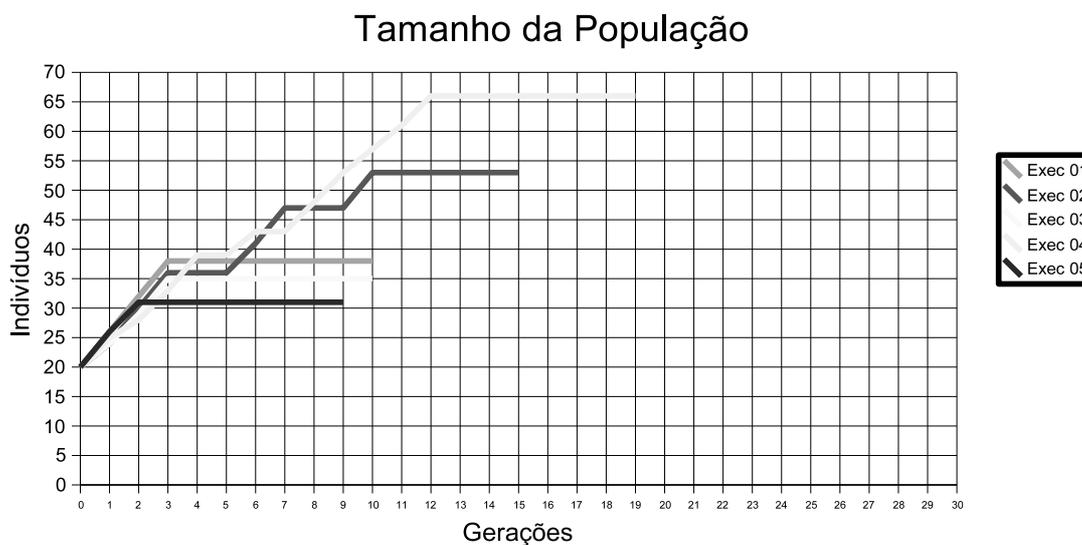


Figura 5.20: Variação da População - Experimento 3.1

A Figura 5.22 apresenta o comportamento de movimentação do robô durante as execuções. Também este pode ser considerado o comportamento padrão nestes experimentos: tal como na simulação, inicia confrontando com situações de perigo e leitura nula dos sensores, até que realiza o acompanhamento de modo satisfatório

Uma diferença significativa entre a simulação e o experimento real é o perfil do conjunto de regras consolidadas ao final de cada teste real ser muito semelhante para todas as execuções. Foi realizada uma análise da composição da população em cada uma das execuções. O que foi identificado é que aquelas regras que foram criadas e inseridas na ocorrência dos eventos insatisfatórios é que vieram a ser responsáveis por

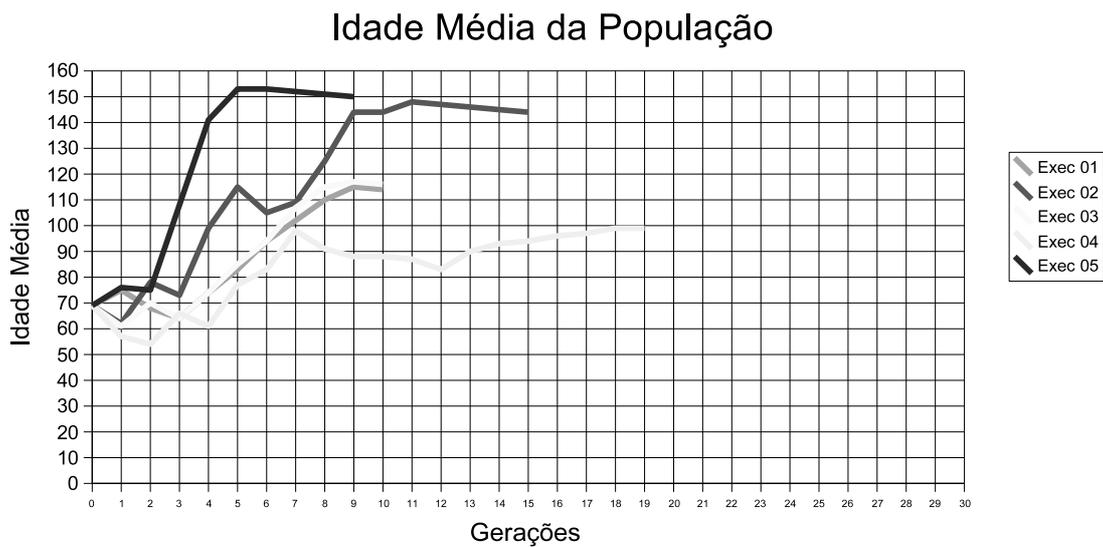


Figura 5.21: Variação da Idade Média da População - Experimento 3.1

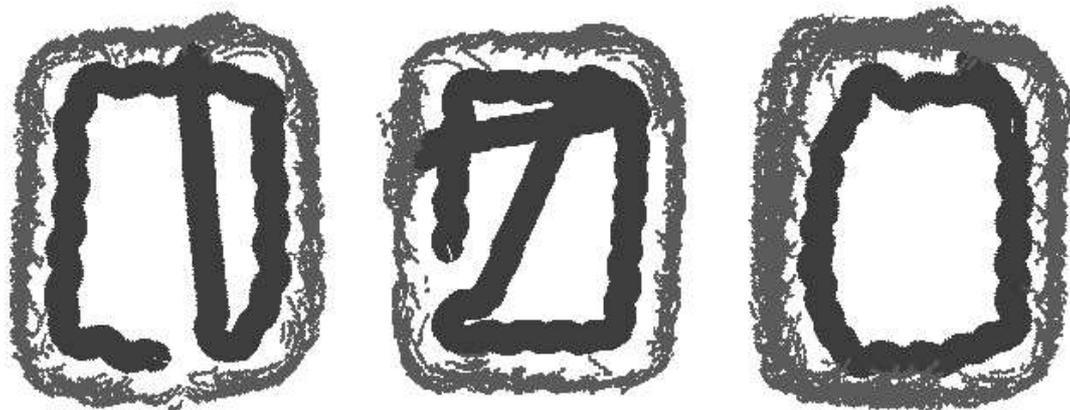


Figura 5.22: Comportamento Padrão do robô - Experimento 3.1

levar o sistema à estabilidade. A Figura 5.23 apresenta graficamente a composição do conjunto de regras que alcançaram a categoria de consolidadas ao final de uma das execuções.

No entanto, não se pode afirmar que o processo reprodutivo foi completamente desnecessário. O fato de que as execuções tiveram seu resultado final classificado como 2 (4.000 passos satisfatórios utilizando quaisquer regras da população), indica que outras regras, que não somente aquelas já consolidadas, também estiveram sendo usadas durante o período de estabilização do sistema.

5.5.1.2 Experimento 3.2

Os objetivos deste experimento são analisar o desempenho do sistema real ao ser utilizada a técnica do **cruzamento uniforme** e, também comparar com os resultados do experimento 1.2.

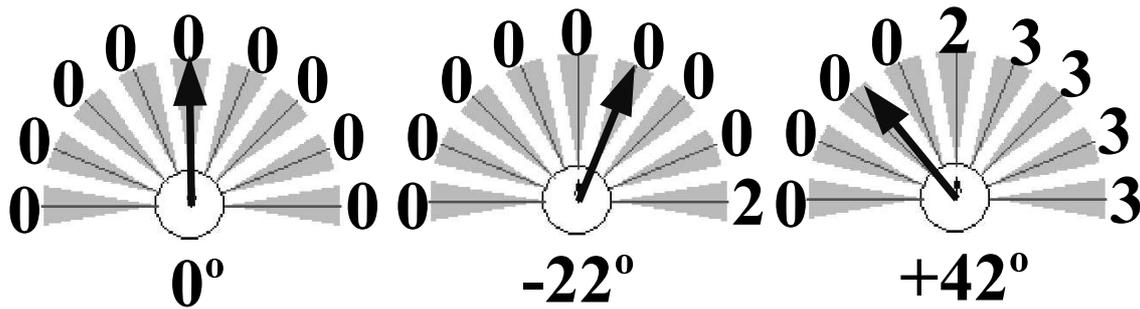


Figura 5.23: Conjunto de Regras Consolidadas - Experimento 3.1

Neste teste, o sistema alcançou resultado satisfatório em 100% das execuções. Os resultados de um conjunto de 5 execuções é apresentado na Tabela 5.13.

Tabela 5.13: Resultados do Experimento 3.2

Exp	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop	Resultado
01	12	3	1	3	59	88	59	2
02	10	1	1	3	37	107	37	2
03	13	4	2	4	61	95	61	2
04	22	8	3	4	90	104	118	1
05	23	8	2	4	88	117	110	2

Somente duas das cinco execuções ultrapassaram a marca de 20 gerações. Mas isso já permite observar que o número de indivíduos na população ao final da execução é menor do que o tamanho máximo atingido pela população durante todo o processo. Assim como o experimento anterior, a maioria das execuções atingiu o sucesso com resultado igual a 2. E, comprovando uma característica das execuções com o robô real, caracterizado pela dificuldade de leitura dos sensores, pode se identificar que em todas as execuções houve perda da referência da lateral (leitura nula), diferentemente do experimento simulado, no qual em que algumas execuções isso não ocorria.

A Tabela 5.14 apresenta a totalização dos resultados do experimento.

Tabela 5.14: Totalização do Experimento 3.2

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	16,00	4,80	1,80	3,60	67,00	102,20	77,00
Mínimo	10,00	1,00	1,00	3,00	37,00	88,00	37,00
Máximo	23,00	8,00	3,00	4,00	90,00	117,00	118,00
Desvio Padrão	6,04	3,11	0,84	0,55	22,19	11,17	35,18

Aparentemente, os experimentos realizados com o cruzamento médio apresentaram melhores resultados - não fosse a particularidade dos conjuntos de regras consolidados gerados.

A Figura 5.24 apresenta a variação no tamanho da população para cada execução do experimento e a Figura 5.25, a variação na idade média da população.

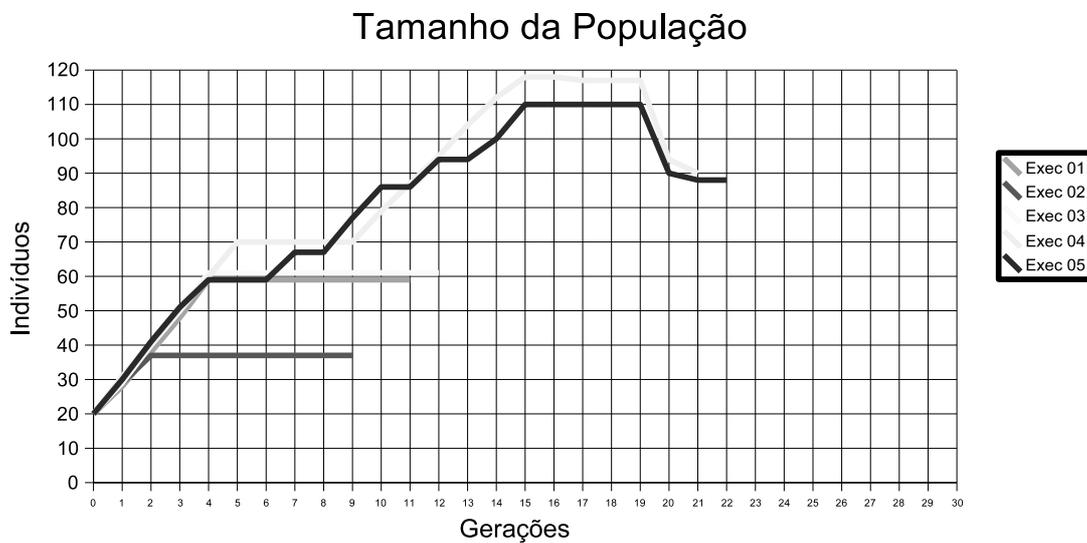


Figura 5.24: Variação da População - Experimento 3.2

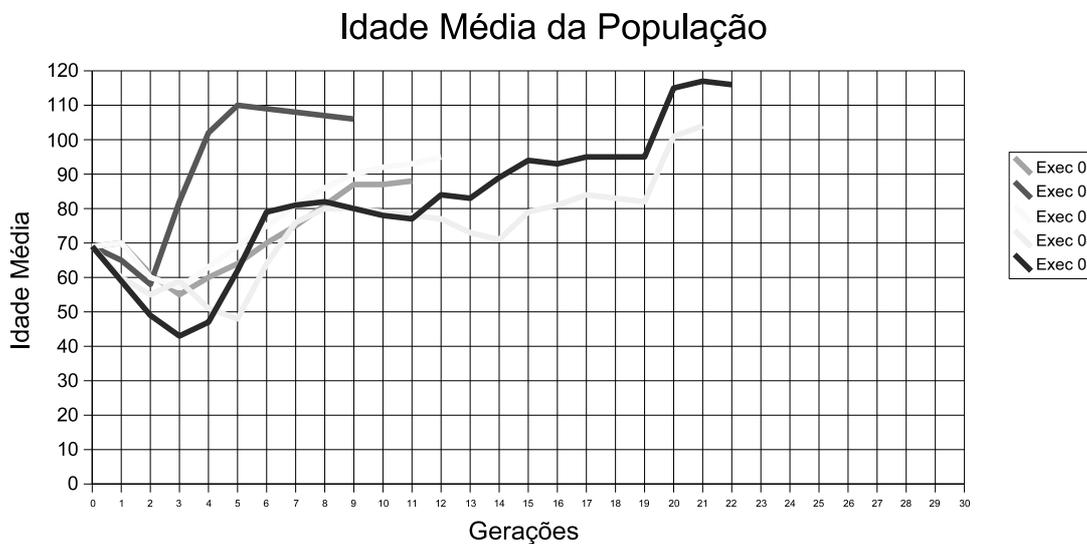


Figura 5.25: Variação da Idade Média da População - Experimento 3.2

Neste experimento, com o cruzamento uniforme também ocorreu o mesmo fato em uma parcela das execuções. No entanto, ao ser efetuada uma análise mais criteriosa em cada conjunto de regras existentes ao final do processo, sejam consolidadas ou não, verificou-se que o processo reprodutivo atuou de forma satisfatória, gerando regras mais adaptadas a partir daquelas inseridas arbitrariamente na ocorrência dos eventos de perigo e leitura nula. Isso foi constatado a partir das características encontradas nas regras consolidadas, as quais apresentavam semelhança à estas regras

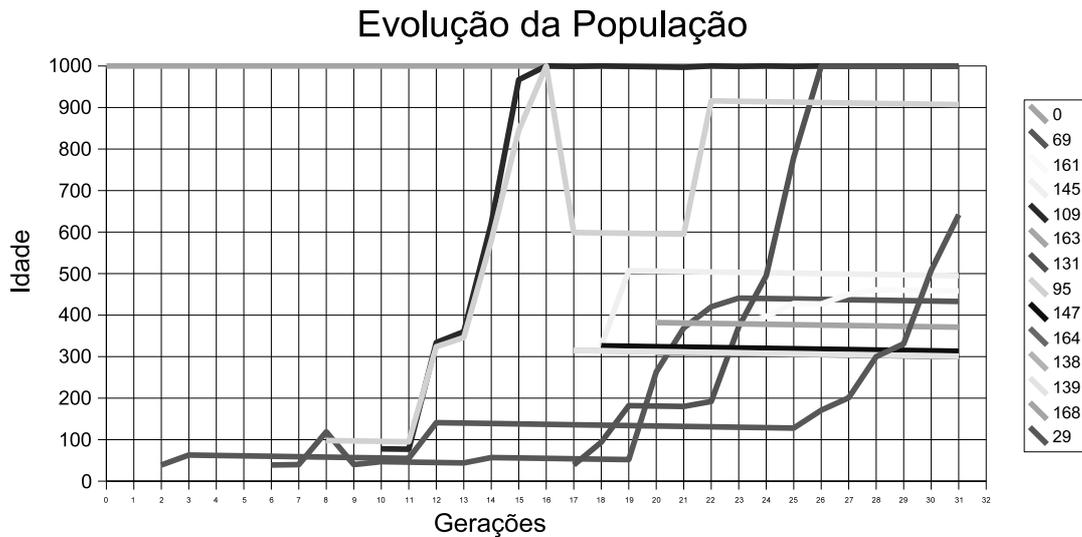


Figura 5.27: Evolução da População - Experimento 3.2

5.5.2 Experimentos Reais - Ambiente com Obstáculos

Os experimentos com o NOMAD 200 em cenário esparsos apresentaram resultados satisfatórios, comprovando o funcionamento do sistema para ambientes de baixa complexidade. Assim como nos testes em simulação, a técnica desenvolvida neste trabalho foi confrontada com ambientes mais complexos. No mesmo cenário utilizado no experimento anterior, foram inseridos dois obstáculos (coluna e corredor), nas posições correspondentes conforme a descrição apresentada no Experimento 2.1.

5.5.2.1 Experimento 4.1

O primeiro teste real com obstáculos utilizou a técnica de **cruzamento médio** e, para o limite estipulado em 100 gerações, somente 85% das execuções obtiveram êxito e alcançaram a estabilização. O problema observado foi que algumas vezes ocorriam repetidos eventos de perigo para o robô em um determinado ponto do cenário. Todas as vezes que ele tornava a passar por aquele ponto, um novo evento de perigo acontecia, demonstrando dificuldade em convergir para a solução. Assim, o limite de gerações era ultrapassado e as execuções onde isso ocorria eram classificadas como insucesso.

A Tabela 5.15 apresenta a totalização dos resultados do experimento.

Tabela 5.15: Totalização do Experimento 4.1

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	18,80	5,80	2,40	3,40	61,80	148,80	62,60
Mínimo	12,00	3,00	1,00	3,00	40,00	126,00	40,00
Máximo	31,00	11,00	4,00	5,00	93,00	182,00	93,00
Desvio Padrão	7,40	3,11	1,14	0,89	19,70	21,49	19,58

A Figura 5.28 apresenta a variação no tamanho da população para cada execução dos experimentos e a Figura 5.29, a variação na idade média da população.

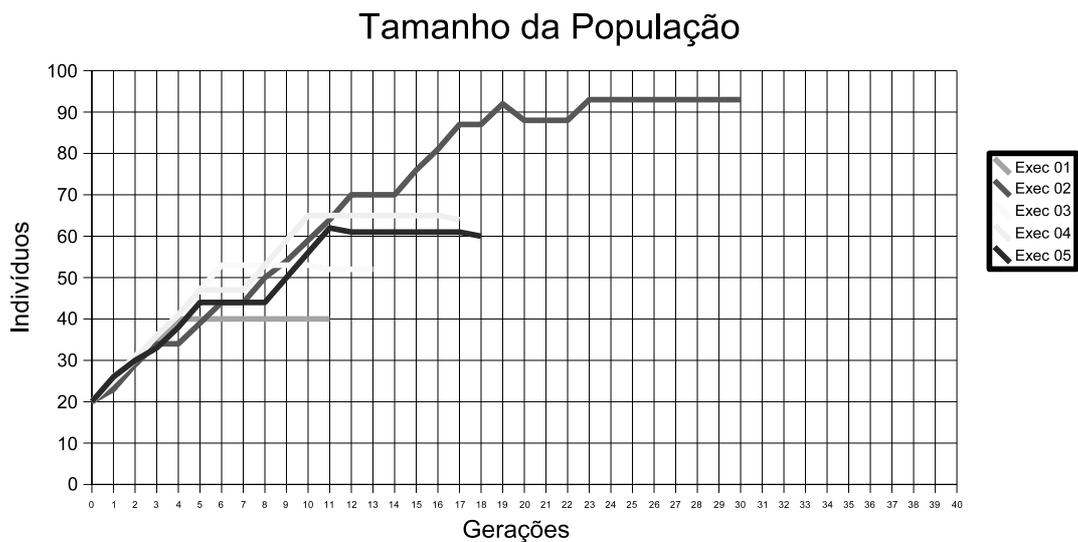


Figura 5.28: Variação da População - Experimento 4.1

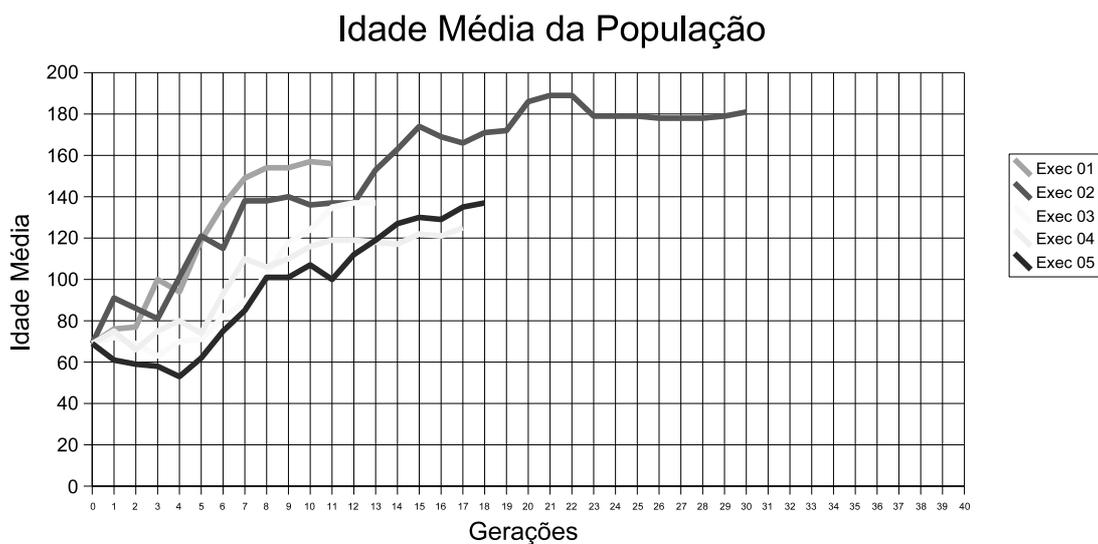


Figura 5.29: Variação da Idade Média da População - Experimento 4.1

A Figura 5.30 apresenta o comportamento de movimentação do robô durante as execuções com obstáculos. Também este pode ser considerado o comportamento padrão nestes experimentos: tal como na simulação, inicia confrontando com situações de perigo e leitura nula dos sensores, até que realiza o acompanhamento de modo satisfatório

5.5.2.2 Experimento 4.2

Novamente modifica-se a técnica de cruzamento, para cruzamento uniforme. A Tabela 5.16 apresenta a totalização dos resultados do experimento.

A Figura 5.31 apresenta a variação no tamanho da população para cada execução do experimentos e a Figura 5.32, a variação na idade média da população.

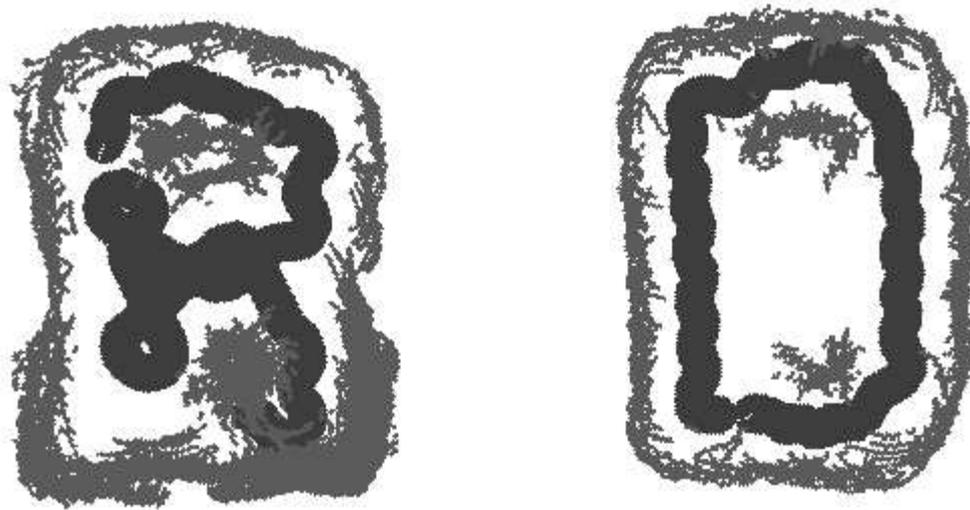


Figura 5.30: Comportamento Padrão do robô - Experimento 4.1

Tabela 5.16: Totalização do Experimento 4.2

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	17,80	5,40	2,20	3,60	85,20	119,20	88,80
Mínimo	10,00	2,00	0,00	3,00	42,00	84,00	42,00
Máximo	28,00	10,00	5,00	5,00	145,00	148,00	154,00
Desvio Padrão	7,16	3,21	1,79	0,89	38,21	24,24	42,04

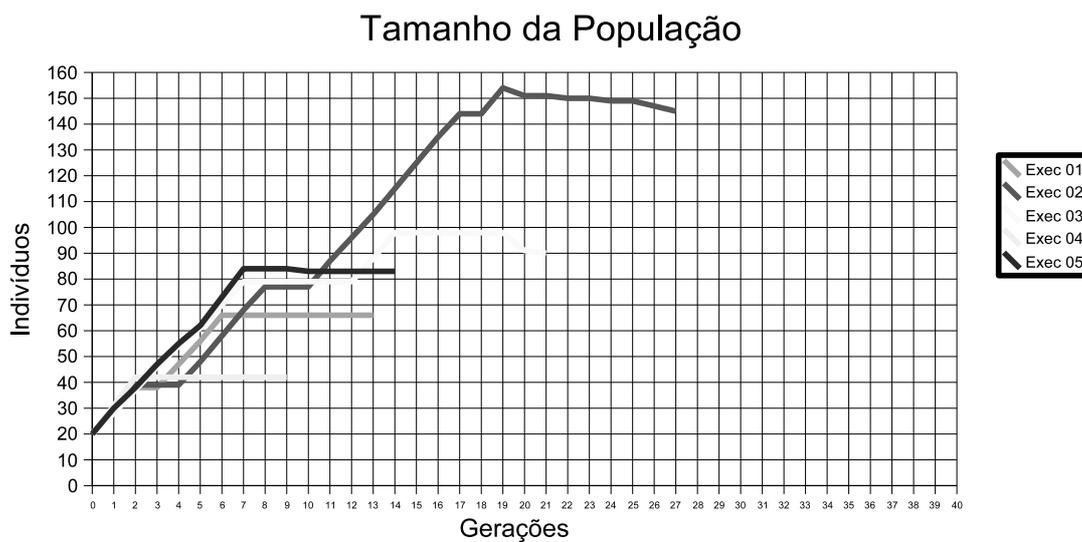


Figura 5.31: Variação da População - Experimento 4.2

Os primeiros experimentos realizados com o robô real em cenários com obstáculos mostraram que o sistema consegue apresentar relativa semelhança com o desem-

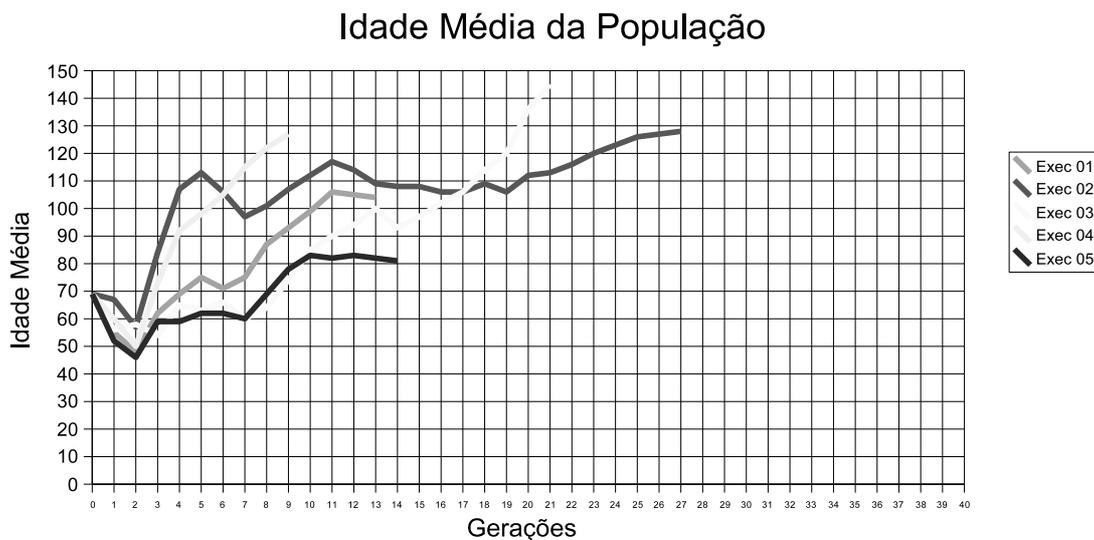


Figura 5.32: Variação da Idade Média da População - Experimento 4.2

penho obtido em testes simulados com um ambiente similar. Novamente, a composição do conjunto de regras consolidadas difere, devido ao aspecto já mencionado anteriormente, que, como as leituras dos sensores não possuem regularidade, a generalização das regras é acentuada, fazendo com que menos regras contemplem um número maior de situações.

5.5.3 Experimentos Reais - Ambiente com Obstáculos e Beco

Para a realização destes experimentos, além da modificação do cenário pela inserção do ponto onde o robô deve retornar por não existir saída, foi necessário alterar os critérios de parada. Como o perímetro do cenário nestes testes é maior, o robô necessita executar pouco mais de 3.600 passos para efetuar uma volta completa. Desta forma, ficou determinado que o objetivo é atingido quando ocorrerem:

- **Resultado 1** : 4.000 passos satisfatórios, utilizando exclusivamente regras consolidadas;
- **Resultado 2** : 6.000 passos satisfatórios, utilizando quaisquer regras - consolidadas ou não.

5.5.3.1 Experimento 5.1

O objetivo deste experimento é confrontar o sistema com um ambiente mais complexo, onde o processo de convergência até a estabilização necessite um número maior de regras e de gerações. Desta forma, será possível avaliar melhor a estratégia de controle populacional. Novamente foi utilizada a técnica de **cruzamento médio** que, assim como no experimento 4.1, alcançou a convergência do sistema em somente 85% das execuções.

A Tabela 5.17 apresenta a totalização dos resultados do experimento.

Como pode ser observado, o número de gerações necessárias nesta configuração do cenário é maior e, principalmente a quantidade de eventos de perigo ocorridos.

Tabela 5.17: Totalização do Experimento 5.1

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	53,00	17,40	3,60	5,80	105,20	158,40	118,20
Mínimo	27,00	6,00	2,00	3,00	54,00	134,00	63,00
Máximo	86,00	38,00	6,00	8,00	215,00	206,00	228,00
Desvio Padrão	22,86	13,56	1,67	2,28	67,14	28,06	68,69

A Figura 5.33 apresenta a variação no tamanho da população para cada execução do experimento e a Figura 5.34, a variação na idade média da população.

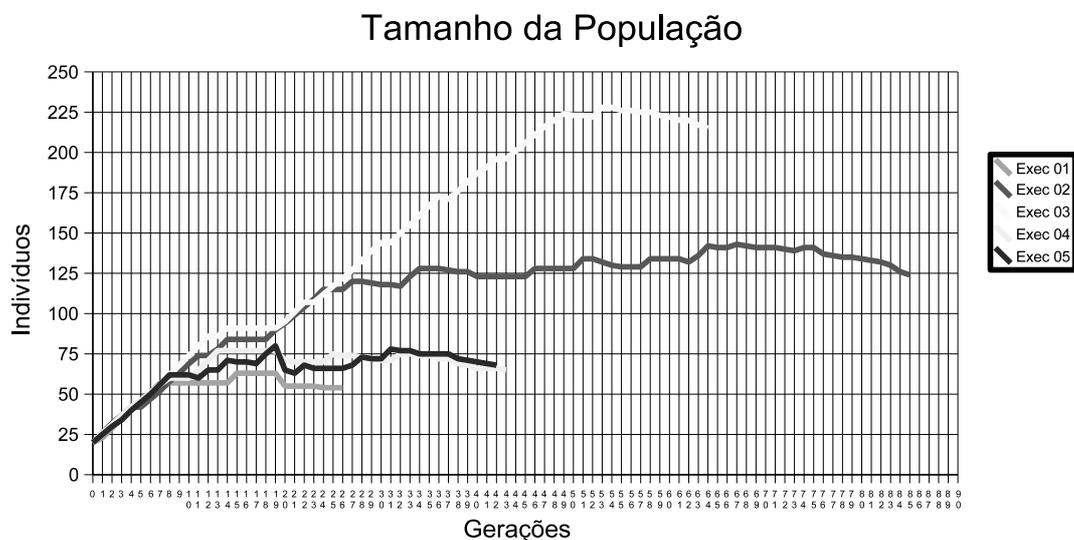


Figura 5.33: Variação da População - Experimento 5.1

A Figura 5.35 apresenta o comportamento de movimentação do robô durante as execuções com obstáculos e beco. A maior diferença nesta situação é que podem existir no cenário, algumas áreas onde o robô consegue realizar pequenas voltas, aparentando ser uma movimentação satisfatória, mas que acabam por levá-lo novamente a eventos de perigo. Isso pode ser observado na movimentação contínua e em círculos realizada entre o obstáculo que representa a coluna e o canto convexo da entrada do beco.

5.5.3.2 Experimento 5.2

Seguindo na mesma metodologia, agora foi realizado um experimento utilizando o mesmo cenário do anterior, mas fazendo uso da técnica de **cruzamento uniforme**, a fim de analisar comparativamente o desempenho das duas técnicas de cruzamento. A Tabela 5.18 apresenta a totalização dos resultados do experimento.

Novamente o sistema, em todas as execuções, consegue alcançar o objetivo dentro do limite de gerações estabelecido. O que realmente comprova que a forma de cruzamento que permite geração de indivíduos com maior diversidade apresenta melhores resultados para os objetivos deste trabalho.

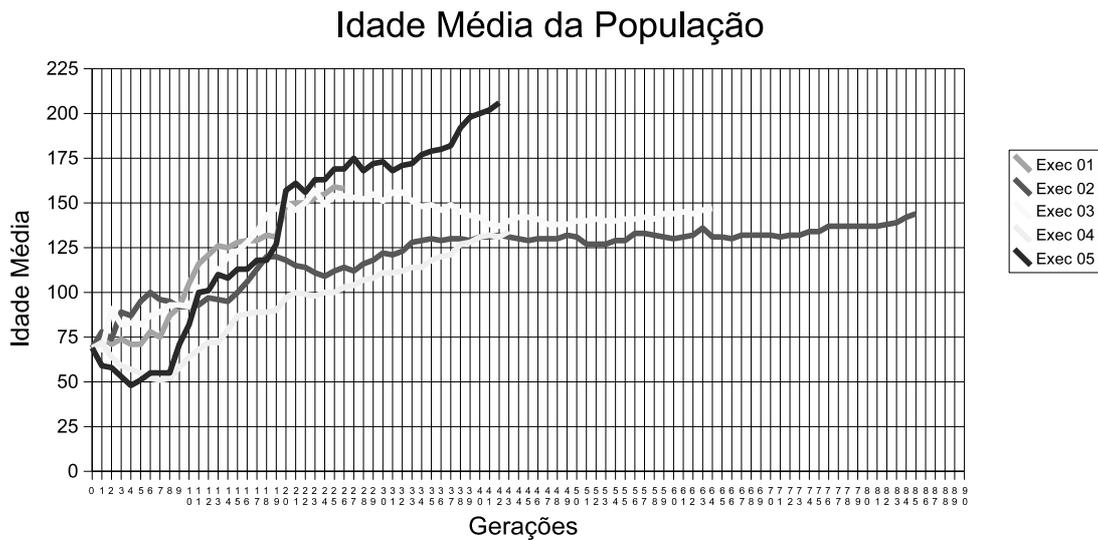


Figura 5.34: Variação da Idade Média da População - Experimento 5.1

Tabela 5.18: Totalização do Experimento 5.2

	Ger	Per	Nul	No Regras	Pop Final	Id Méd	Maior Pop
Média	28,60	8,00	2,00	4,20	91,60	161,80	102,60
Mínimo	23,00	3,00	1,00	3,00	51,00	104,00	56,00
Máximo	44,00	13,00	3,00	5,00	121,00	201,00	135,00
Desvio Padrão	8,73	3,74	1,00	0,84	27,26	37,80	29,85

A Figura 5.36 apresenta a variação no tamanho da população para cada execução do experimento e a Figura 5.37, a variação na idade média da população.

5.5.4 Experimentos Reais - Execução Contínua

Os testes anteriores demonstraram a capacidade do sistema de construir um conjunto satisfatório de regras para movimentação livre de colisão e acompanhamento de paredes. No entanto, uma característica que pôde ser observada durante o acompanhamento dos experimentos, é que em algumas vezes o sistema se apresenta relativamente estável e ocorre um evento insatisfatório - causando um prolongamento de algumas gerações até que atinja novamente o critério de estabilização e parada.

5.5.4.1 Experimento 6.1

O objetivo deste experimento é observar o comportamento do sistema em execuções mais prolongadas, onde o sistema já possui um conjunto de regras consolidadas e continua a execução, para além da condição de parada antes estabelecida. Estes testes foram realizados no cenário composto pelos obstáculos e o beco; sendo executado somente utilizando a técnica de cruzamento uniforme, visto que foi identificado que o cruzamento médio apresenta resultados satisfatórios somente para cenários pouco complexos.

A Figura 5.38 mostra o acompanhamento de uma execução (6.1a), relacionando

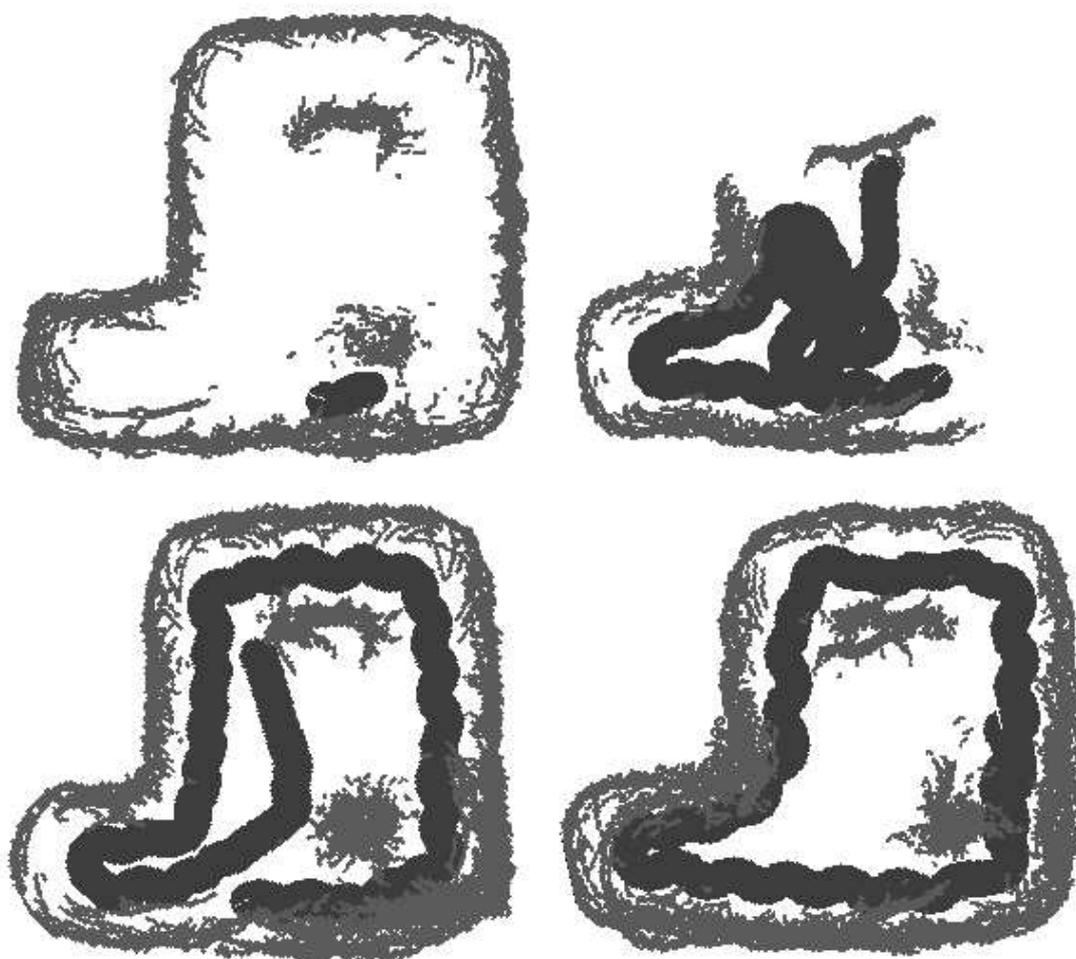


Figura 5.35: Comportamento Padrão do robô - Experimento 5.1

os valores correspondentes ao tamanho da população, que, assim como nos demais testes, apresenta um incremento acentuado no início do processo, passando a decrescer assim que o sistema atinge a estabilização. Também mostra o valor da idade média no decorrer das gerações. Mostrando ainda a distância mínima que o robô está dos objetos (laterais ou obstáculos) ao final de cada geração.

Na linha que mostra a variação da população também estão expostas as gerações que, pela ocorrência de eventos insatisfatórios, dispararam processos reprodutivos. O crescimento da população coincide justamente com estes eventos.

Esta execução apresentou primeiramente eventos de perigo e, logo após, de leituras nulas, até que, a partir da geração 13 passou a apresentar comportamento satisfatório. Pelos critérios de parada e avaliação, esta execução seria encerrada na geração 25, quando alcançou os 6.000 passos satisfatórios. Pode ser observado que, com o sistema já estabilizado, a idade média continua crescendo e, conseqüentemente, o tamanho da população diminui. Isto acontece porque as regras que não mais apresentam utilidade vão tendo sua idade decrementada, até que, finalmente, são eliminadas da população.

Se esta fosse uma execução em simulação, o comportamento do robô continuaria o mesmo indefinidamente. No entanto, no mundo real, pelas características de ruídos

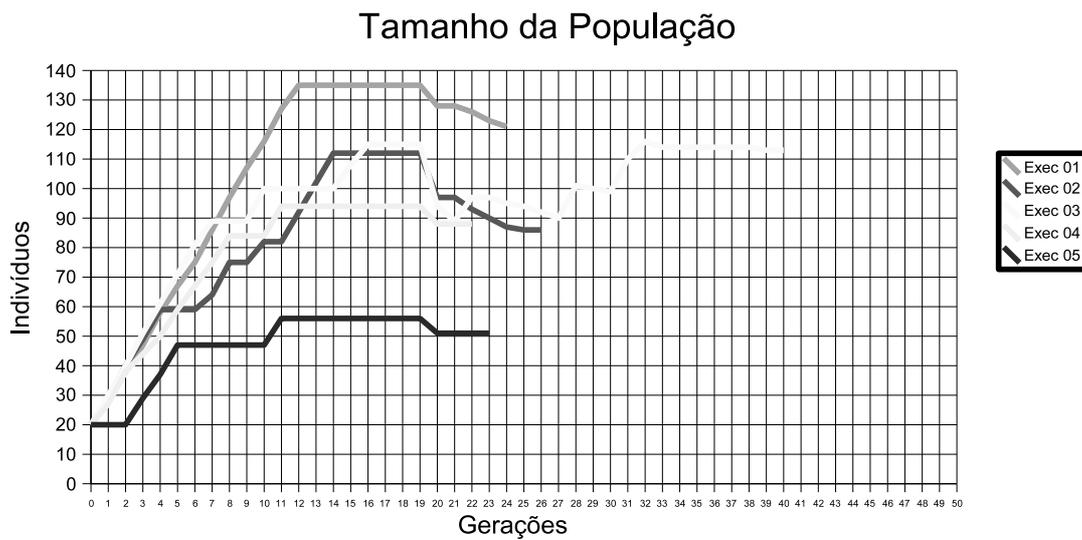


Figura 5.36: Variação da População - Experimento 5.2

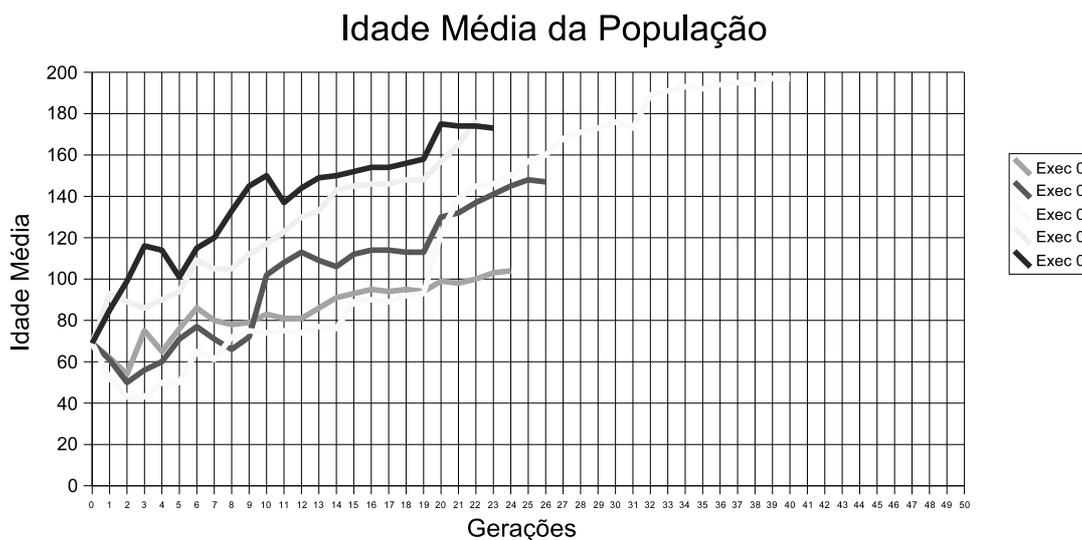


Figura 5.37: Variação da Idade Média da População - Experimento 5.2

nas leituras de sensores e outras ocorrências externas, tais como atrito, inércia, etc, acontecem situações em que o robô executa um movimento insatisfatório e, conseqüentemente, dispara novo processo reprodutivo, para ajustar seu conjunto de regras. Estas ocorrências estão registradas no gráfico, entre as gerações 56 e 77. A cada nova reprodução, a população é incrementada, indicado pelos picos mostrados no gráfico, mas o controle populacional realizado a partir da idade de cada indivíduo segue eliminando aqueles irrelevantes e, portanto, o tamanho da população continua decrescendo.

A Figura 5.39 mostra o acompanhamento de outra execução, apresentando os valores dos mesmos parâmetros do gráfico anterior.

Acompanhar as execuções do sistema em modo contínuo permite constatar a

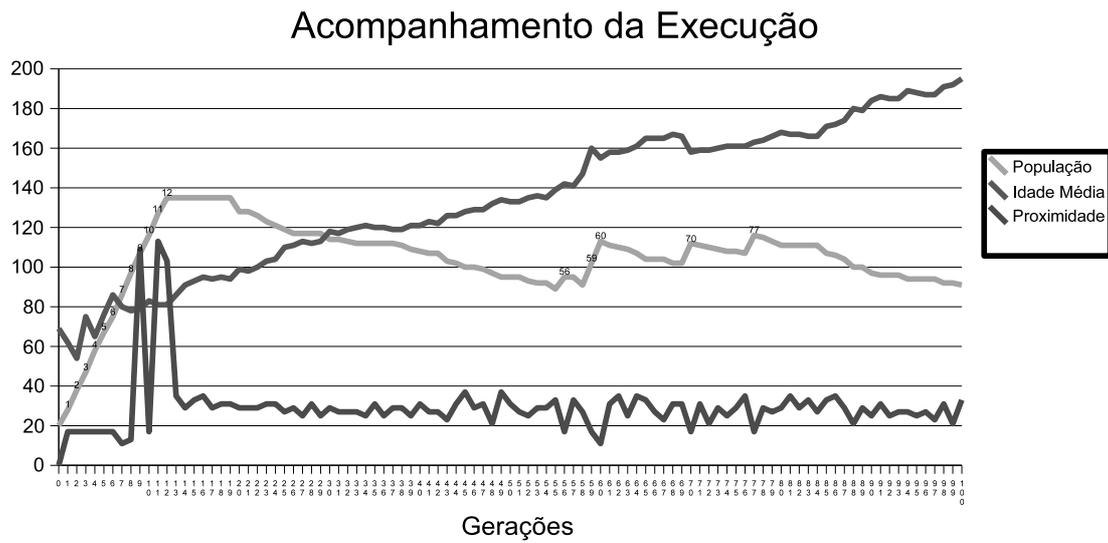


Figura 5.38: Execução Contínua - Experimento 6.1a

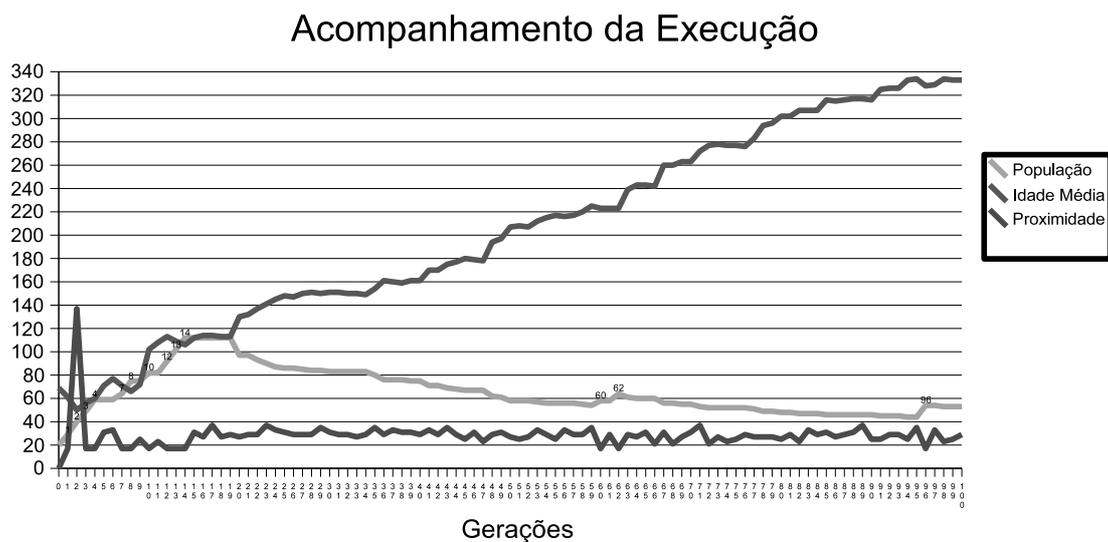


Figura 5.39: Execução Contínua - Experimento 6.1b

validade das estratégias utilizadas neste sistema, que habilitam o robô a continuar executando sua tarefa e, se caso se depare com alguma situação indesejada, retoma o processo evolutivo, a fim de ajustar sua base de conhecimento.

5.5.4.2 Experimento 6.2

O objetivo deste experimento é analisar o sistema em execuções contínuas, onde, durante o processo, o ambiente sofre modificações. Desta forma pode ser analisado o desempenho do sistema para:

- ambientes instáveis;
- aprendizado incremental: processo onde já existem regras prévias consolidadas

e o sistema deve realizar o aprendizado a partir deste conjunto existente;

A primeira execução foi realizada com o cenário sendo modificado do ambiente esparso para o ambiente com obstáculos - configurando uma transposição entre os experimentos 3.2 e 4.2. O robô inicia o processo sem conhecimento prévio, assim como todos os outros experimentos, e, quando o sistema estiver estabilizado, o cenário sofre modificações. A Figura 5.40 mostra o acompanhamento da execução, apresentando os valores do tamanho da população, com a indicação das gerações em que houveram processos reprodutivos; sua idade média e também a proximidade do robô com os objetos do cenário.

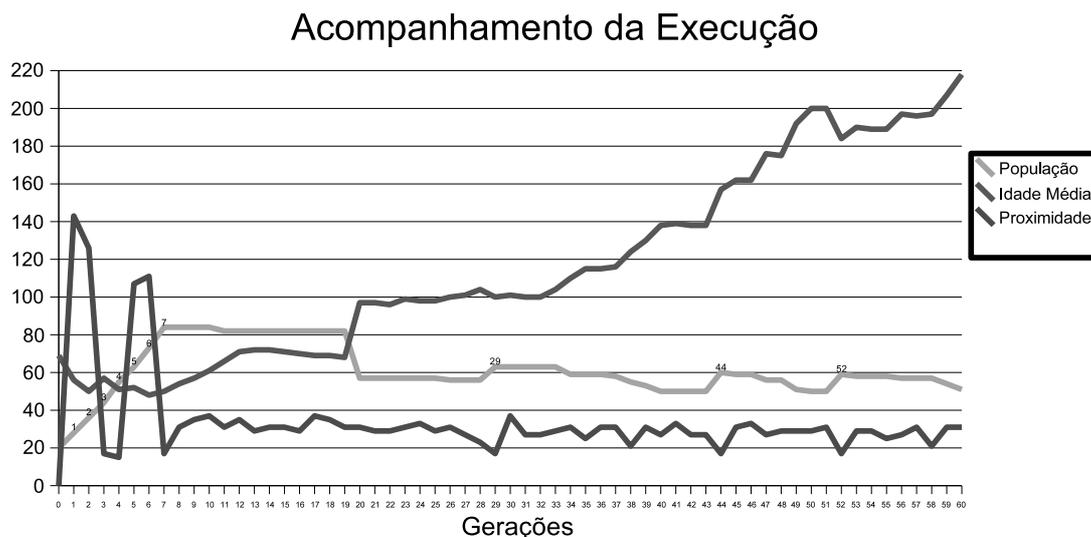


Figura 5.40: Execução com Modificação do Cenário - Experimento 6.2a

O gráfico da figura 5.41 mostra a evolução de uma parcela do conjunto de regras nesta execução. São apresentadas somente as regras que, em algum momento do processo, tiveram sua idade igual ou superior a 300 unidades, configurando-se no mínimo como regras "em crescimento".

Nas primeiras gerações ocorrem os eventos indesejados, que disparam as reproduções e, conseqüentemente, incrementam o tamanho da população, bem como a idade média. A estabilidade do sistema é alcançada na geração 15, totalizando 3 ocorrências de perigo e 4 de leitura nula, constituindo 3 regras consolidadas.

O que pode ser observado no gráfico é que, entre a geração 8 e 15, o número de indivíduos já permanece estável, enquanto a idade média sofre um incremento, até a geração 13. Esse incremento é resultado da avaliação positiva das regras que vieram a compor o grupo das consolidadas.

Na geração 22 foram inseridos os obstáculos no cenário, os quais, na primeira volta, foram transpostos pelo robô sem nenhum evento insatisfatório. O que ocorreu é que na população já haviam algumas regras razoavelmente adequadas à situação, que foram utilizadas. Pode ser observado também o pequeno aumento na idade média da população quando isso ocorre.

No entanto, na segunda volta que o robô realiza no cenário com obstáculos, ocorre um evento de perigo, registrado na geração 29. Este evento dispara uma reprodução, mas não gera mais nenhuma regra satisfatória.

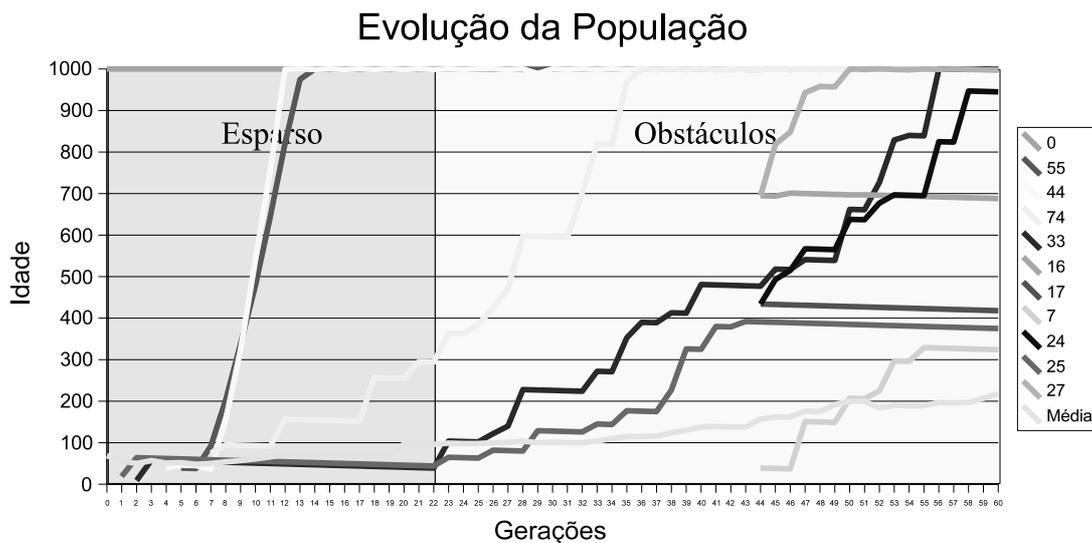


Figura 5.41: Evolução da População - Experimento 6.2a

Na geração 35, o sistema já consolidou mais uma regra, daquelas já existentes antes da modificação do cenário e, na geração 37, o sistema já teria atingido novamente o critério de parada, com a realização de 4.000 passos satisfatórios.

Neste tempo, o sistema continua utilizando as regras 33 e 25, duas regras geradas nos primeiros processos reprodutivos, mas que se mostraram mais adequadas ao cenário com obstáculos.

A partir da geração 44, quando o robô já atingiu 7.000 passos satisfatórios ocorre mais um evento de perigo, surgindo novas regras, ainda melhores, o que pode ser identificado pelo ponto inicial delas apresentar idade elevada⁵. A regra 25 não mais se torna necessária, o que pode ser constatado pelo declínio da sua idade, deste ponto até o final do experimento.

Novamente, na geração 52 houve outro evento de perigo, mas que não gerou nenhuma nova regra relevante ao sistema. Este foi mais um evento insatisfatório em decorrência das leituras ruidosas dos sensores. Por fim, na geração 60, atingindo novamente os 4.000 passos satisfatórios, a execução é interrompida.

A execução a seguir foi efetuada com a ocorrência de duas modificações no ambiente. A primeira, passando de espurso para com obstáculos e, quando o sistema apresentou estabilidade neste, o ambiente foi novamente modificado, com a inserção do beco.

A Figura 5.42 mostra a variação da população, da idade média e da proximidade de objetos durante todo o experimento.

O início da execução foi bem atípico, com a ocorrência de vários eventos de perda de referência das laterais, logo nas primeiras gerações. Assim que conseguiu ajustar o conjunto de regras necessárias, depois de 5 eventos nulos e somente 1 de perigo, estabilizou e teria atendido o critério de parada na geração 15 - quando alcançou 4.000 passos satisfatórios. Na geração 20, foram inseridos os dois obstáculos (coluna e corredor), o que ocasionou um evento de perigo na geração 22. Nesse

⁵Como está descrito em 4.4.1.4, a idade inicial de uma regra gerada em um cruzamento é a média das idades de seus progenitores.

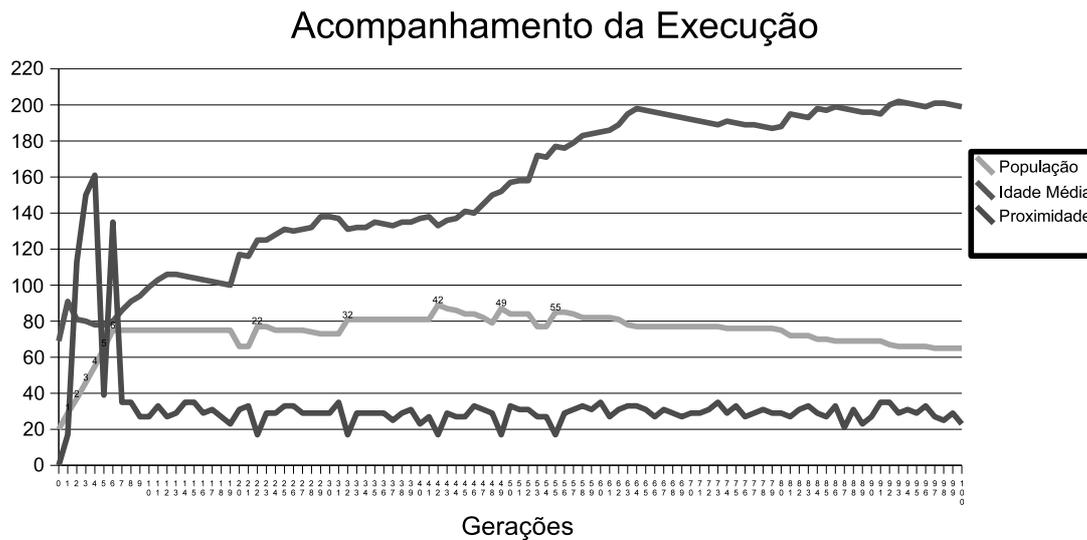


Figura 5.42: Execução com Modificação do Cenário - Experimento 6.2b

momento, aconteceu o único processo reprodutivo necessário para estabilizar novamente o sistema, o que ocorreu na geração 31, quando alcançou novamente 4.000 passos satisfatórios. Na geração 40 foi então inserido o beco, iniciando a última fase do experimento. O que aconteceu é que foram necessárias 3 passagens do robô pelo canto de entrada do beco (curva convexa), com a ocorrência de eventos de perigo, para que conseguisse finalmente ajustar o conjunto de regras. O critério de parada para este cenário, que é de 6.000 passos, seria alcançado na geração 68. No caso específico desta execução, deixou-se o sistema atingir a geração 100 para concluir.

A parcela da população que teve participação significativa neste experimento é apresentada na Figura 5.43. São apresentadas somente aquelas regras cujo tempo restante de vida alcançou, em algum momento do processo, valor igual ou maior a 500 (quinhentas) unidades, ou seja, somente aquelas que, em algum ponto da execução foram identificadas como "candidatas à consolidadas". Para o cenário esparso, foram evoluídas somente 2 regras (25 e 39) que, juntamente com a regra 0, formavam o conjunto consolidado; já para o ambiente com obstáculos, foi necessário adicionar mais duas regras (12 e 34), mas preservando as três primeiras. Por fim, para o final da execução, onde o cenário contava com obstáculos e o beco, o conjunto consolidado era formado por seis regras, as quais eram as três primeiras juntamente com outra três novas (17, 53 e 11). Desta forma, as duas regras geradas para a etapa intermediária foram substituídas. No caso específico deste experimento, a regra 39, gerada ainda no início do processo, para movimentar o robô em ambiente sem obstáculos, servia também para a movimentação entre obstáculos paralelos, porque previa sensores ativos em ambos os lados. O que permitiu seu uso no primeiro estágio da execução é justamente a característica de generalização das regras e sua aplicação por índice de similaridade.

O experimento possibilitou identificar que, ao confrontar um cenário novo, o robô consegue deslocar-se melhor no início do que nas situações de partida do processo sem conhecimento prévio algum. Algumas regras aprendidas anteriormente são preservadas e, também colaboram na geração de novas regras. O que pôde ser

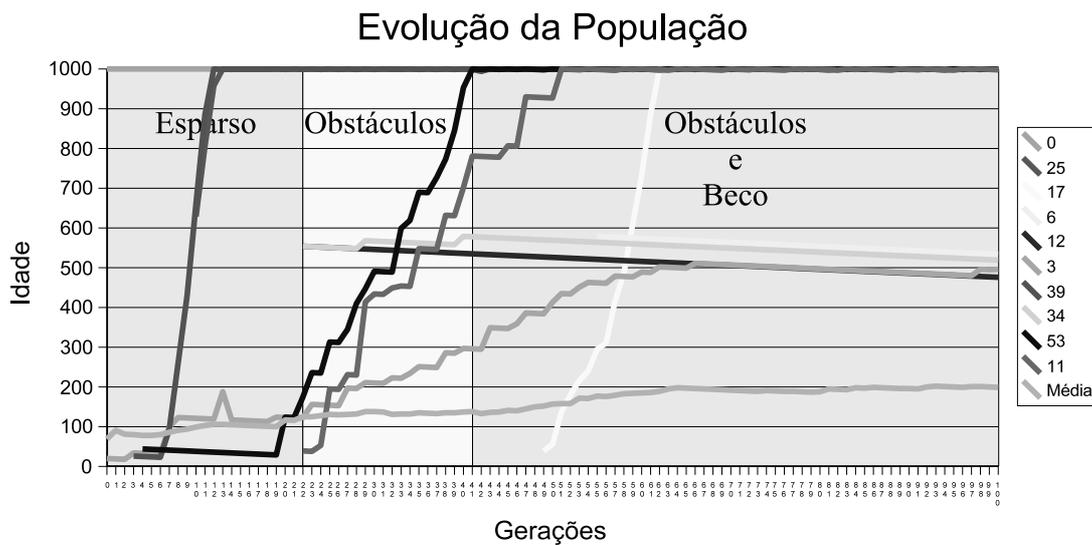


Figura 5.43: Evolução da População - Experimento 6.2b

observado é que a adaptação integral do sistema para uma nova configuração do cenário pode, algumas vezes, custar até mais gerações do que o aprendizado realizado para este mesmo cenário, quando o robô inicia o processo sem conhecimento prévio. Mas, apesar dessa constatação, o desempenho médio pode ser considerado melhor, em virtude da capacidade de movimentação apresentada no início do processo, que possibilita movimentação efetiva do robô em alguns espaços do ambiente desconhecido.

5.6 Discussão dos Resultados

Ao final dos experimentos, podem ser descritos alguns comentários importantes observados nessa etapa do trabalho.

A estratégia proposta, inspirada em AGI, para controlar o tamanho da população e restringi-la a somente o conjunto de regras necessárias para atingir o objetivo proposto, apresentou resultados satisfatórios, como pode ser constatado analisando os gráficos de variação do tamanho da população, bem como da variação da idade média.

Os experimentos simulados proporcionaram uma avaliação preliminar do sistema proposto nesta tese. Os resultados obtidos podem ser considerados satisfatórios, comprovando sua eficácia, tanto para cenários muito simples, sem obstáculos, quanto para ambientes onde o robô é confrontado com obstáculos, em situações mais complexas.

Uma situação observada durante alguns experimentos simulados, foi o surgimento de uma solução como a mostrada na Figura 5.44, que apresenta o comportamento do robô quando o sistema gera uma subpopulação. Este pode ser considerado um comportamento satisfatório pelos padrões de avaliação, visto que ele está acompanhando corretamente uma parede e não colide. Não ocorrem situações de perigo e também é mantida a referência da parede nestes experimentos. Como se trata de um sistema reativo, o robô não tem registro do tamanho total do cenário, para

saber se ainda lhe resta espaço a percorrer. O que pode ser constatado é que nos experimentos que iniciavam com a população composta por 20 indivíduos, o índice de ocorrência dessa situação ficou abaixo de 5%; e nos experimentos com população inicial reduzida, este índice estava próximo de 10%.

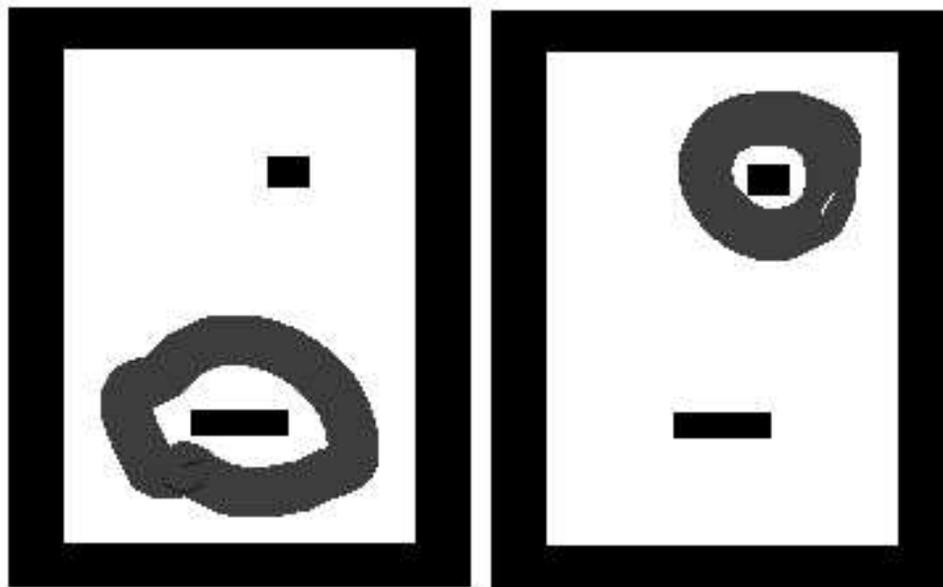


Figura 5.44: Comportamento de Sub-População - Experimento 2.3

Os resultados obtidos nos testes com o NOMAD 200 mostram que a abordagem proposta por este trabalho alcança êxito tanto para ambientes simulados, quanto para o mundo real. As principais diferenças observadas dizem respeito ao problema da acuracidade no sensoriamento. O nível de ruído existente no sinal obtido pelos sensores ocasiona uma queda na credibilidade do que o robô está sentindo, o que implica na necessidade ainda maior de generalização do uso das regras de movimentação.

Isto porque a ocorrência de diferentes leituras para a mesma situação, faz com que seja ainda mais importante a capacidade de utilizar uma mesma regra para diferentes situações, as quais possuem um índice de similaridade aceitável com a respectiva regra. Isso pode ser constatado ao comparar os experimentos reais, onde as regras inseridas como cópias dos eventos insatisfatórios acabam tendo maior probabilidade de vir a ser consolidadas, diferentemente dos experimentos simulados, nos quais estas mesmas regras têm sua maior utilidade na geração de novas regras em processos reprodutivos.

O que acontece é que, como as leituras no simulador são mais precisas, o sistema conduz o processo evolutivo para gerar regras mais semelhantes às situações sensorizadas - visto que elas se repetem. Já no mundo real, como não existe regularidade das leituras, as primeiras regras que conseguem apresentar algum resultado razoável, acabam por ser utilizadas em todas as situações possíveis (generalização).

Assim como nos experimentos simulados, também foi constatado o surgimento de soluções como a mostrada na Figura 5.45, que apresentam o comportamento do robô em uma solução sub-ótima. Como nos experimentos reais as leituras apresentam muito ruído e, com isso, a cada passagem do robô pelo mesmo local podem

acontecer leituras distintas dos sensores, a ocorrência deste tipo de comportamento ficou abaixo de 3% das execuções. Inclusive, um aspecto diferente da simulação é que com o robô real também ocorria o fato de que, após vários passos movimentando-se em volta do obstáculo, o robô volta a acompanhar a lateral do cenário. Isso acontece novamente devido aos ruídos na leitura dos sensores de navegação, o que não existe na simulação.

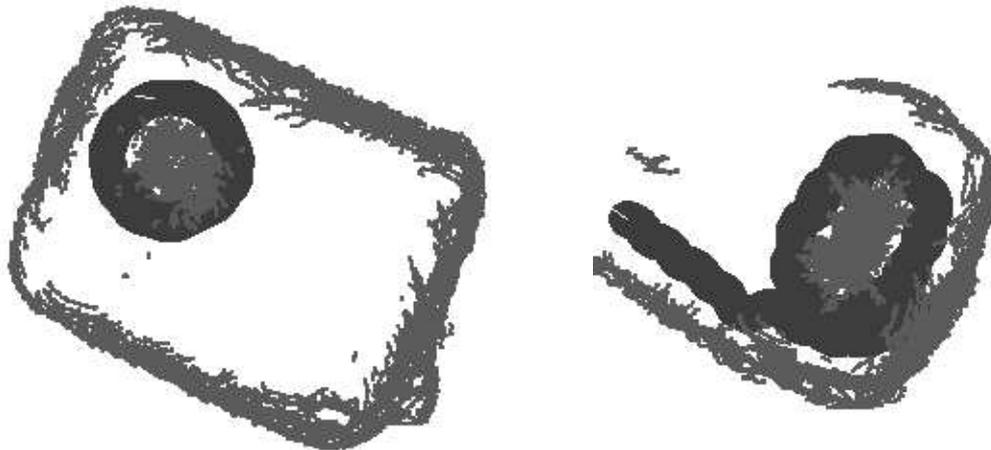


Figura 5.45: Comportamento de Sub-População - Experimento 4.3

O propósito de ser um sistema de aprendizagem contínua, onde o robô está constantemente interagindo com o ambiente e, de acordo com a necessidade, ajusta seu conhecimento através da evolução e processos reprodutivos pôde ser claramente observada nos experimentos, os quais não foram usados os critérios de parada. A observação, tanto do comportamento do robô movimentando-se no cenário, quanto da dinâmica da população de regras indicou que a estratégia possibilita o aprendizado mesmo em situações de dificuldade na leitura dos sensores e, em casos extremos, quando o robô depara-se com modificações em seu ambiente.

O uso de dois diferentes métodos de cruzamento foi apresentado, porque durante a etapa de configuração do sistema, quando foram testadas várias configurações dos diferentes parâmetros, o desempenho alcançado pelo cruzamentos médio e uniforme era similar. Assim, optou-se por testá-los nas diferentes configurações de cenários, o que comprovou a superioridade do cruzamento uniforme para este tipo de aplicação. Uma descrição e análise mais detalhada sobre estes métodos é apresentada a seguir.

5.6.1 Diferenças entre os Métodos de Cruzamento

O método de cruzamento uniforme apresentou resultados ligeiramente melhores que o cruzamento médio. A diferença substancial é que, enquanto no cruzamento médio, uma única regra gerada representa uma situação mediana entre as duas reprodutoras, no cruzamento uniforme, existe uma possibilidade bem maior de configurações para o par de descendentes gerados - que apresenta, desta forma, maior probabilidade de contemplar as situações necessárias para a convergência do sistema.

Para analisar a situação observada e descrita nos experimentos anteriores, que as execuções que utilizam o cruzamento uniforme em seu processo reprodutivo apresentam melhores resultados, foi elaborado um exemplo. As Figura 5.46 e Figura 5.47

apresentam, para o mesmo par de reprodutores, os resultados para cruzamento médio e cruzamento uniforme.

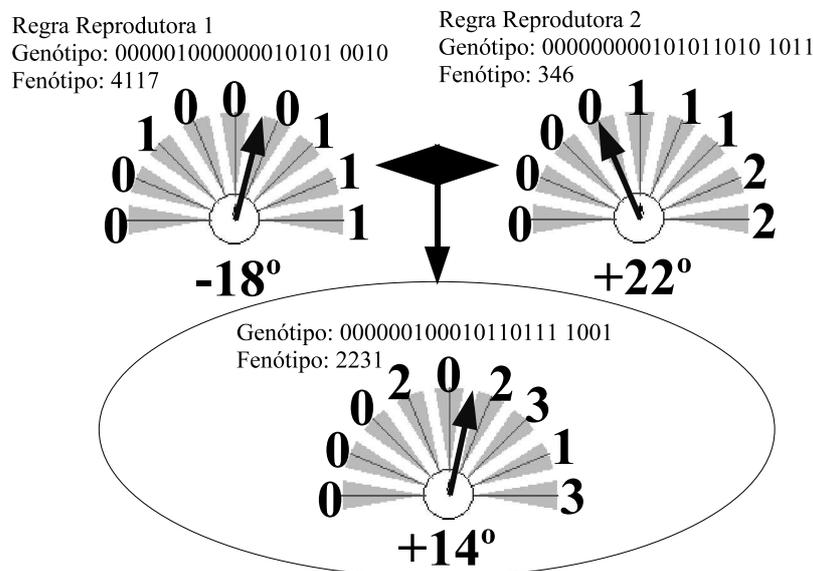


Figura 5.46: Resultado de Reprodução utilizando Cruzamento Médio

No caso de utilizar o método de cruzamento médio, é gerada somente uma nova regra que, apesar de apresentar uma nova configuração de leitura dos sensores, continua sendo somente uma única alternativa de movimentação para esta situação. E, como o sistema funciona a partir da combinação de pares de regras de aproximação/repulsão para realizar o movimento de acompanhamento das paredes, é evidente que, mesmo que esta regra seja válida para o processo de aprendizado, ainda não é o suficiente. Provavelmente esta regra será utilizada e, conseqüentemente, participará de novos processos reprodutivos, o que possibilitará gerar as novas regras que ainda se fazem necessárias. O custo do uso do cruzamento médio acaba sendo que mais iterações se tornam necessárias para que o sistema venha a convergir.

Por outro lado, utilizando o cruzamento uniforme, dentre as várias possibilidades, estão as duas apresentadas na Figura 5.47.

O exemplo mostra que com os dois métodos de cruzamento é possível gerar regras adequadas para ajustar o conhecimento a determinadas situações. A vantagem proporcionada pelo método do cruzamento uniforme para esta aplicação está no fato de gerar dois descendentes distintos a partir do par de reprodutores - gerando até mesmo duas regras com ações em direções contrárias, caso estes reprodutores assim se apresentem.

5.6.2 Comparação com Trabalhos Similares

Em um trabalho como o de Grefenstette e Schultz (GREFENSTETTE; SCHULTZ, 1994), que a modelagem do sistema prevê que cada indivíduo representa o conjunto de regras necessário para a movimentação do robô, é necessário que o projetista deste sistema tenha conhecimento do número aproximado de regras que devem compor este conjunto. Isto faz com que o conhecimento do domínio da aplicação tenha que ser mais profundo, o que aumenta a dificuldade de implementação de um sistema

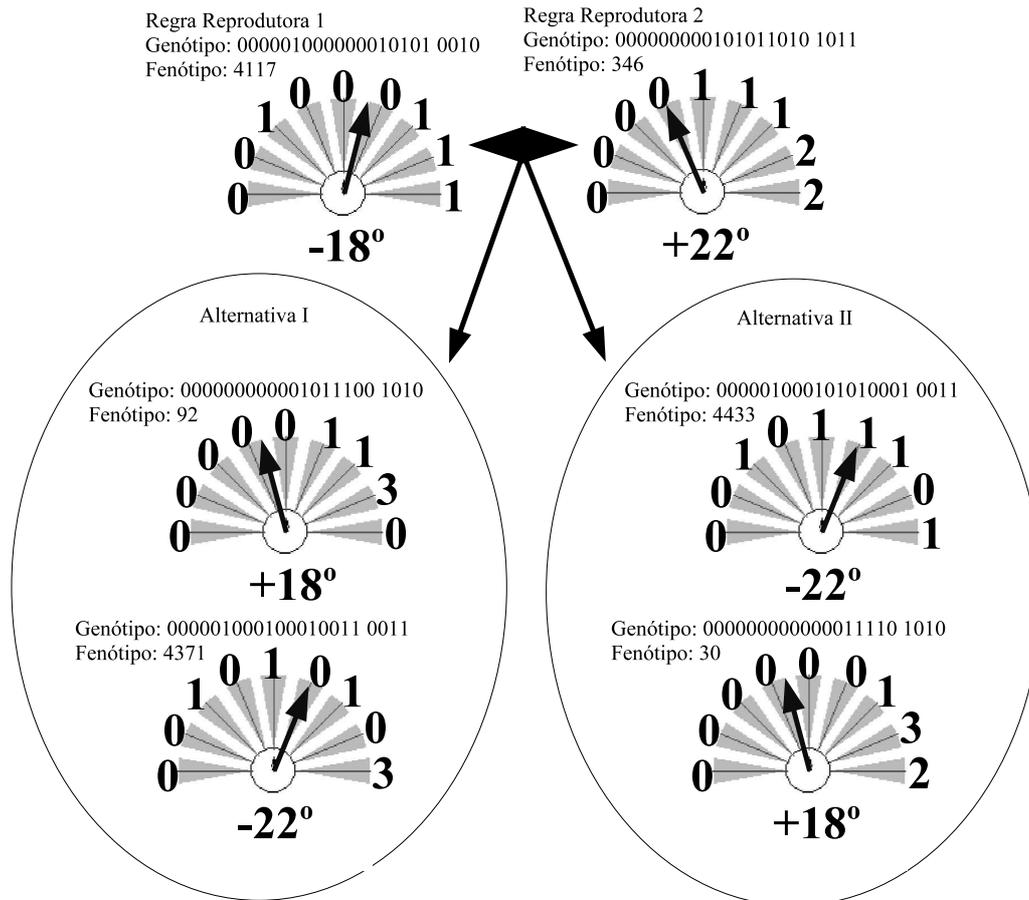


Figura 5.47: Resultado de Reprodução utilizando Cruzamento Uniforme

com esta finalidade. Em contraposição à este trabalho, usar uma representação onde cada indivíduo é composto por somente uma regra, pareceu ser a forma mais simples de modelar um sistema onde a população poderia ter tamanho variável.

Dentre as dificuldades enfrentadas, o custo computacional e também a definição do tamanho da população foram citadas como críticas. Como cada indivíduo era composto por várias regras, isso implica que o espaço de busca para esse AE seja consideravelmente maior do que quando se utiliza a abordagem de somente uma regra por indivíduo. E, considerando que um AE possui a característica de, a cada ciclo do algoritmo, realizar a busca em vários pontos simultaneamente, todas as regras que se apresentam como soluções satisfatórias acabam sendo indicadas pelo sistema com parte integrante da solução. Portanto, o uso da abordagem por idades para controlar o tamanho e a qualidade da população de regras, elimina totalmente essas preocupações, dispensando o projetista do sistema definir qual o tamanho ideal da população, bem como do conjunto de regras a ser gerado.

Como já foi descrito na Seção 3.3, este sistema funcionava com dois módulos distintos, um realizando o aprendizado em simulação e o outro fazendo uso das soluções encontradas executando em um robô real. Na versão posterior do sistema (GREFENSTETTE, 1996), o robô vai monitorando possíveis mudanças no ambiente, alterando o modelo do mundo existente no processo simulado, para conseguir evoluir os conjuntos de regras que serão então usados pelo robô real. Essa

abordagem permite acelerar o aprendizado, pelo fato da simulação ser mais rápida. No entanto, como o próprio autor aponta, existem deficiências nesta abordagem (GREFENSTETTE; SCHULTZ, 1994), pelas substanciais diferenças entre simulação e execução real. Assim, o fato do aprendizado ser realizado totalmente no robô real indica maior probabilidade de sucesso em ambientes desconhecidos.

No trabalho aqui apresentado, o aprendizado também acontece de forma continuada durante a execução, através dos erros e acertos das ações executadas, não necessitando o registro em memória do mapeamento do cenário. Isto apresenta tanto pontos positivos, como negativos. A não necessidade de memória para o mapeamento é um ponto positivo, mas, por outro lado, como o agente é desenvolvido a partir de uma arquitetura reativa e, conseqüentemente, não possui memória das ações passadas, também não tem conhecimento do domínio do ambiente, o que faz com que não saiba se está revisitando um ponto do cenário ou se já tenha se deslocado por toda sua extensão. Desta forma, alguma modalidade de registro do mapeamento do cenário poderia ser incorporada ao sistema, para melhorar sua capacidade exploratória.

Assim como neste trabalho, nos experimentos de Cazangi (CAZANGI, 2004), o processo reprodutivo também acontece somente quando da ocorrência de eventos pré-determinados (colisão, captura ou monotonia). Aqui tem-se somente as situações de perigo e leitura nula, devido ao propósito da aplicação. No entanto, naquele trabalho, somente a partir do momento que o evento ocorre, o sistema realiza a varredura na população para verificar quais são os possíveis candidatos a reprodutores de novas regras, a partir da similaridade destes indivíduos com a situação enfrentada. Portanto, uma diferença significativa é que, na estratégia desta tese, essa busca de candidatos à reprodução acontece concomitantemente à movimentação do robô. A cada movimento realizado, é efetuada a atribuição do índice de utilidade de cada indivíduo. Desta forma, o custo computacional da reprodução acaba por ser diluído durante o processamento de todo um período de execução e não fica concentrado somente para o momento específico em que os cruzamentos devem ser realizados.

Outra diferença é que aqui os novos indivíduos são inseridos sem eliminar seus progenitores, o que causa incremento populacional, mas, por outro lado, não corre-se o risco de eliminar um indivíduo que poderia ser melhor do que seu descendente, o que já é largamente visto na literatura como estratégia de elitismo.

Além disso, para a situação específica em que ocorreu o evento insatisfatório, é criada e inserida arbitrariamente uma nova regra, composta em seu antecedente, pela configuração lida pelos sensores e, em seu conseqüente, pelo inverso da ação da regra utilizada com insucesso. A justificativa para essa atitude é que, com a ocorrência do evento, não se faz necessário um processo evolutivo para atender especificamente esta situação, já que ela é conhecida. O que se acredita ser mais importante, é justamente a criação de regras que não mais conduzam o sistema a uma situação como a que gerou o evento insatisfatório. Para isso, o processo evolutivo desencadeado neste momento seleciona os reprodutores, executando uma seleção por roleta, baseada no índice de utilidade das regras durante a última etapa de execução. Conseqüentemente, esta estratégia tende a gerar regras para ser utilizadas em situações similares às identificadas naqueles movimentos exatamente anteriores à respectiva situação.

A dificuldade recorrente, apontada pelos pesquisadores, nos trabalhos encontra-

dos na literatura, que é a definição do tamanho da população (GREFENSTETTE; SCHULTZ, 1994) (CAZANGI; FIGUEIREDO; ZUBEN, 2003) (CHAMINDRA, 2003), foi eliminada neste trabalho pela introdução do operador de idade. Normalmente o uso de uma população de tamanho fixo implica em não trabalhar com um número reduzido de indivíduos, porque isto representa uma fraca amostragem do espaço de busca. Assim, a estratégia de idades controlando a permanência do indivíduo na população eliminou esta preocupação. E, como pôde ser observado nos resultados apresentados, a população cresce na medida que mais indivíduos são necessários para participar do processo de busca e otimização do AE, mas, em contrapartida, quando o sistema passa a apresentar os resultados esperados, os indivíduos desnecessários vão sendo gradativamente eliminados, até que restam somente aqueles realmente úteis para atingir o objetivo proposto.

6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo propor uma nova estratégia para evoluir conhecimento, de maneira autônoma, em robótica móvel e implementá-la em um sistema de navegação autônoma. Os fundamentos para esta estratégia são os AGs e SCs, técnicas evolucionárias bem conhecidas pela comunidade científica e já utilizadas em sistemas similares e adicionalmente, através da introdução de um novo operador, de controle de idade, inspirado nos AGIs. Um ponto de extrema importância é que o conhecimento evoluído está representado de forma explicitamente declarativa, para permitir que um usuário possa compreender o significado contido em cada regra e, inclusive, se assim o quizer, contribuir com a inserção "ad hoc" de outras regras, caso venha a julgar interessante.

A fim de situar o trabalho no campo da CE, foram apresentadas resumidamente as principais características dos AGs, AGIs e SCs, seus operadores e funcionalidades. Na sequência, foi apresentada a RE, uma sub-área da robótica que constrói seus controladores baseados em AE, descrevendo-se um resumo dos seus principais trabalhos já desenvolvidos. O Capítulo 4 descreve o sistema proposto, sua arquitetura, seus módulos e componentes. O funcionamento de cada módulo é explicado. Por fim, são apresentados os experimentos realizados em ambientes simulados e, também, executados em um robô NOMAD 200.

Normalmente, quando o foco de um trabalho é a "evolução computacional", tem-se um problema onde a solução é encontrada pelo incremento da adaptação de um conjunto de indivíduos ao longo de várias gerações. Se o foco da investigação for o "aprendizado", busca-se incrementar as habilidades de um único indivíduo ao longo de sua existência e, para tanto, tradicionalmente são utilizadas outras técnicas, que não envolvem AE. Aqui tomou-se um rumo diferenciado, buscando implementar aprendizado a partir de um processo de evolução computacional.

Uma diferença importante entre AGs convencionais e a estratégia aqui apresentada é que neste trabalho a solução final para o problema não está representada individualmente em cada indivíduo da população, mas sim, pela configuração de uma pequena parcela do conjunto de indivíduos. Isso gera a necessidade de implementar estratégias diferenciadas para o controle populacional, a fim de que a busca realizada pelo AG não homogenize a população.

Considerando a análise da literatura, que mostra que foram desenvolvidos apenas poucos trabalhos relacionados a aprendizado evolucionário com regras em robótica, os quais citavam a limitação do uso de uma população de tamanho fixo; e analisando os resultados alcançados nos experimentos, acredita-se que este trabalho contribuiu de forma significativa para o desenvolvimento de sistemas em RE, assim como para a própria CE.

Na condução do trabalho, os experimentos preliminares, anteriores aos apresentados nesta tese, embora realizados somente em simulação, geraram resultados interessantes que puderam ser considerados satisfatórios e comprovaram a importância da abordagem desta proposta, visto que foram submetidos e aprovados artigos para eventos (SCHNEIDER; PRESTES; BARONE, 2004a) (SCHNEIDER; PRESTES; BARONE, 2004b), bem como validaram a proposta para esta tese.

Estes primeiros testes não apresentavam regularidade nos resultados, variando muito nos totais do número de gerações, população final e até mesmo no êxito do processo. Portanto, evidenciaram a necessidade de ajustes no sistema, principalmente no que diz respeito ao mapeamento das melhores ações correspondentes às regras, porque, muitas vezes o sistema apresentava comportamento não satisfatório, mesmo possuindo regras com um bom índice de similaridade à leitura dos sensores; ou seja, a base era composta por regras que contemplavam situações confrontadas pelo robô, mas que executavam ações que levam o robô a um estado de risco ou, até mesmo, à uma colisão.

Este problema no conjunto de regras foi verificado porque o processo de aprendizagem é responsável por executar duas buscas (mapeamentos) distintas. A primeira, mais importante e mais complexa, deve descobrir, dentre as quase 300 mil regras possíveis (considerando regras constituídas por nove sensores de dois elementos cada), aquelas que são realmente importantes e representativas (generalização) para o ambiente onde o robô está agindo. A segunda busca, diz respeito ao mapeamento da melhor ação a ser realizada por cada uma das regras. A solução implementada neste caso foi o uso mais intensificado de mutações no segmento responsável pela indicação da ação.

Realizando os devidos ajustes, principalmente na função de avaliação, os resultados obtidos nos experimentos simulados com os diferentes tipos de cenários nos mostra que a abordagem é válida e alcança desempenho razoavelmente satisfatório no objetivo a que se propõe, ou seja, construir e manter uma base de regras, a qual é mantida com relativa diversidade genética, o que evidencia a existência de regras para situações diversas das enfrentadas momentaneamente.

Um aspecto que proporcionou ganho significativo ao sistema foi o procedimento de inserir a regra referente à leitura do evento insatisfatório ocorrido (perigo ou leitura nula). Desta forma, passa a existir na população de regras, ao menos um indivíduo que carrega as características enfrentadas pelo robô e não superadas. Em algumas execuções, principalmente em ambiente esparsos, o sistema acabou utilizando somente estas regras, mostrando que até mesmo um programa baseado em uma heurística simples poderia solucionar este caso em específico.

No entanto, para cenários mais complexos, as regras consolidadas ao final do experimento eram, normalmente, diferentes daquelas inseridas nesses eventos, e representavam ainda melhor a generalização das situações enfrentadas pelo robô. No entanto, é possível constatar que a diferença entre estas regras não é elevada, comprovando que justamente as regras inseridas, copiando a situação dos sensores, colaboraram nos processos evolutivos ocorridos na seqüência da execução.

Mais uma vez fica comprovada a hipótese de que trabalhos com robótica móvel devem, na medida do possível, ser realizados diretamente com os robôs reais. As diferenças existentes em um ambiente controlado de simulação acabam por simplificar ao extremo as implementações que, quando executadas no robô, não conseguem alcançar os mesmos resultados, inviabilizando o trabalho.

É claro que deve se relatar que, para um sistema de aprendizado autônomo, onde o robô necessita continuamente realizar leituras dos sensores para mensurar a qualidade dos movimentos que faz, um sistema de sensoricamente composto por sonares não é o mais adequado, devido ao nível de ruído existente no sinal. Robôs mais modernos não mais utilizam esta tecnologia, que foi substituída por sensores de varredura laser, além de sistemas de visão que permitem identificar alvos.

Outra dificuldade enfrentada foi a definição precisa de uma métrica para constatar o sucesso, ou não, do processo de aprendizado do sistema. Como se trata de um sistema que deve interagir continuamente com o ambiente, foi considerado que, quando o comportamento estiver estável por um determinado número de passos, o objetivo foi alcançado, assim foi possível realizar medições de desempenho dos experimentos.

Nos testes iniciais, os processos reprodutivos eram disparados ao final de cada etapa de execução, o que ocasionava que o sistema normalmente não alcançava um estado satisfatório e se mantinha nele estável. Isto porque poderiam ocorrer situações onde o algoritmo tentava testar a eficiência das novas regras que surgiam na população, em detrimento daquelas já consolidadas. A solução encontrada para isto foi realizar processos reprodutivos somente se fizer necessário ajustar o conjunto de regras, em decorrência de eventos indesejados, o que se mostrou uma estratégia eficiente.

Portanto, este trabalho mostra que é possível realizar um processo de evolução do conhecimento (aprendizado) para um único agente robótico, utilizando um AE, onde o aspecto relacionado ao tamanho da população e volume de regras que representa o conhecimento final é controlado internamente pelo sistema, através da estratégia de idades incorporada. E ainda, que este conhecimento, por estar declarado de forma explícita, pode ser interpretado facilmente pelo usuário.

Por fim, esta abordagem, assim como qualquer outro AE, é dependente da função de avaliação, que atribui os conceitos de satisfação/insatisfação à regra, dependendo do resultado da aplicação de sua ação. No entanto, como a estratégia de controle populacional pela idade realiza a tarefa de manter ou não um indivíduo na população, a função de avaliação deixa de ser a única responsável pela indicação dos bons indivíduos existentes, o que permite maior flexibilidade na sua modelagem.

6.1 Contribuições

O desenvolvimento deste trabalho apresenta várias contribuições, dentre as quais podem ser destacadas:

- a principal contribuição, de caráter inédito, é a inserção dos atributos "IDADE" e "USABILIDADE" em um sistema baseado em SC. Uma abordagem com idades, bastante similar, já havia sido utilizada em AGs, mas com o objetivo de eliminar as deficiências decorrentes da pressão seletiva nos processos de seleção de reprodutores. Neste trabalho, a usabilidade substitui o atributo de energia e respectivos cálculos do SC tradicional, no processo de escolha das regras, simplificando a implementação. Além disso, permite ser utilizado como índice de ajuste, para que possam ser usadas técnicas convencionais de seleção, mais uma vez, simplificando a implementação. A idade é responsável por preservar ou eliminar os indivíduos da população, através de estratégias

de penalização e recompensa. Assim é possível manter uma população de regras de tamanho variável, permitindo, ainda, manter a diversidade genética na população e evitar a sua homogenização, bem como isentar o modelador do sistema da definição destes parâmetros;

- o uso da idade permite que o sistema seja projetado sem a necessidade de conhecimento prévio de quantas regras devem formar um conjunto razoável ao final do processo evolutivo, bem como, qual o tamanho da população para que esta realize a busca de forma satisfatória. Isto reduz consideravelmente a necessidade de representação do domínio da aplicação;
- sistemas de aprendizagem em robótica normalmente utilizam redes neurais como técnica principal, o que "encapsula" o conhecimento. Utilizando uma forma de representação declarativa para as regras, é possível analisar mais detalhadamente o conhecimento gerado. Também um usuário pode até inserir, a priori, regras que foram geradas em experiências anteriores, se considerar importantes para o funcionamento futuro do sistema. Isso é interessante, porque, muitas vezes não se consegue modelar um problema ou solução, bem como trabalhar com exemplos, porque o conhecimento que existe é difícil de ser formalizado;
- possuindo o conhecimento armazenado em uma base de regras, é possível realizar processos de "aprendizagem incremental", onde habilidades previamente conhecidas podem ser informadas ao robô antes de iniciar uma nova execução;

6.2 Possibilidades de Trabalhos Futuros

Para a continuidade deste trabalho, alguns tópicos foram identificados, dentre os quais, considerou-se que os mais importantes são:

- utilizar robôs com sensoriamento mais preciso, que possam informar com mais acuracidade a real situação do ambiente. A sugestão inicial é passar a utilizar robôs com sistemas de sensoriamento por laser e, complementarmente, de visão computacional, o que permitiria, além de aprender a navegar, também reconhecer objetivos;
- simplificar a construção da função de avaliação, construindo um framework ou um sub-sistema onde o usuário possa inserir regras heurísticas para penalização e recompensa, ou ainda, poderia ser utilizada uma forma de apresentar situações desejadas e indesejadas, que poderia ser então hibridizada com uma implementação utilizando RNAs ou Lógica Nebulosa;
- utilizar uma população de robôs, onde estaria sendo executado outro algoritmo evolucionário, em um nível superior ao do aprendizado, que guiaria a evolução desta população, caracterizando a evolução embarcada. A característica particular em um sistema destes é que a carga genética trocada ente os robôs seria composta pelas regras já consolidadas por eles. Está se analisando a possibilidade de utilizar um time de futebol de robôs com essa finalidade;

REFERÊNCIAS

- BOOKER, L. B. Improving the Performance of Genetic Algorithms in Classifier Systems. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1., 1985, Mahwah, NJ, USA. **Proceedings...** [S.l.]:Lawrence Erlbaum Associates, 1985. p.80–92.
- BROOKS, R. A. A Robust Layered Control System for a Mobile Robot. In: SHAFER, G.; PEARL, J. (Ed.). **Readings in Uncertain Reasoning**. San Mateo, CA: Kaufmann, 1990. p.204–213.
- BROOKS, R. A. A Robust Layered Control System for a Mobile Robot. In: IYENGAR, S. S.; ELFES, A. (Ed.). **Autonomous Mobile Robots: control, planning, and architecture**. Los Alamitos, CA: IEEE Computer Society, 1991. v.2, p.152–161.
- BROOKS, R. A Robust Layered Control System for a Mobile Robot. **IEEE Journal of Robotics and Automation**, [S.l.], v.2, p.14 – 23, 1986.
- BULL, L. learning classifier systems: a brief introduction. In: BULL, L. (Ed.). **Applications of Learning Classifier Systems**. [S.l.]: Springer, 2004. p.3–14.
- BULLOCK, S. Co-evolutionary design: implications for evolutionary robotics. In: EUROPEAN CONFERENCE ON ARTIFICIAL LIFE, 3., 1995, Granada, Spain. **Proceedings...** [S.l.: s.n.], 1995.
- CAZANGI, R. R. **Uma proposta evolutiva para controle inteligente em navegação autônoma de robôs**. 2004. Dissertação (Mestrado em Engenharia Elétrica) — CPG/FEEC - Unicamp, Campinas (SP), Brasil.
- CAZANGI, R. R.; FIGUEIREDO, M. F. Sistema Autônomo Inteligente Baseado em Computação Evolutiva Aplicado à Navegação de Robôs Móveis. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 5., 2001. **Anais...** [S.l.: s.n.], 2001.
- CAZANGI, R. R.; FIGUEIREDO, M. F.; ZUBEN, F. J. V. Sistema Autônomo Evolutivo em Aplicações Embarcadas para Navegação de Robôs. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 6., 2003, Bauru (SP) Brasil. **Anais...** [S.l.: s.n.], 2003. v.1, p.704–709.
- CAZANGI, R. R.; ZUBEN, F. J. V.; FIGUEIREDO, M. F. A classifier system in real applications for robot navigation. In: CONGRESS ON EVOLUTIONARY COMPUTATION, CEC, 2003, Canberra. **Proceedings...** [S.l.]:IEEE Press, 2003. p.574–580.

CAZANGI, R. R.; ZUBEN, F. J. V.; FIGUEIREDO, M. F. Autonomous navigation system applied to collective robotics with ant-inspired communication. In: CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION, GECCO, 2005, Washington DC, USA. **Proceedings...** New York: ACM Press, 2005. v.1, p.121–128.

CHAMINDRA, R. **Obstacle Avoidance in a Mobile Robot using the Learning Classifier**. Brisbane, Australia, 2003.

CLIFF, D.; HARVEY, I.; HUSBANDS, P. **Incremental evolution of neural network architectures for adaptive behaviour**. Brighton, England, UK: University of Sussex, School of Cognitive and Computing Sciences, 1992. (Cognitive Science Research Paper CSRP256).

CLIFF, D.; HARVEY, I.; HUSBANDS, P. Explorations in Evolutionary Robotics. **Adaptive Behavior**, [S.l.], v.2, n.1, p.71–108, 1993.

DARWIN, C. **The Origin of Species**. New York: Modern Library, 1998.

DAVIS, L. **Handbook of Genetic Algorithms**. New York: Van Nostrand Reinhold, 1991.

DORIGO, M. Alecsys and the AutoMouse: learning to control a real robot by distributed classifier systems. **Machine Learning**, [S.l.], v.19, n.3, p.209–240, 1995.

DORIGO, M.; COLOMBETTI, M. Robot Shaping: developing autonomous agents through learning. **Artificial Intelligence**, [S.l.], v.71, n.2, p.321–370, 1994.

DORIGO, M.; SCHNEPF, U. Genetics-based Machine Learning and Behaviour Based Robotics: a new synthesis. **IEEE Transactions on Systems, Man and Cybernetics**, [S.l.], v.23, n.1, p.141–154, 1993.

FICICI, S.; WATSON, R.; POLLACK, J. Embodied Evolution: a response to challenges in evolutionary robotics. In: EUROPEAN WORKSHOP ON LEARNING ROBOTS, 8., 1999. **Proceedings...** [S.l.: s.n.], 1999. p.14–22.

FLOREANO, D. Evolutionary Mobile Robotics. In: QUAGLIARELLI, D.; PERIAUX, J.; POLONI, C.; WINTER, G. (Ed.). **Genetic Algorithms in Engineering and Computer Science**. [S.l.]: Chichester: John Wiley and Sons, 1997.

FLOREANO, D.; MONDADA, F. Automatic Creation of an Autonomous Agent: genetic evolution of a neural-network driven robot. In: FROM ANIMALS TO ANIMATS: INTERNATIONAL CONFERENCE ON SIMULATION OF ADAPTIVE BEHAVIOR, SAB, 1994. **Proceedings...** Cambridge: MIT Press, 1994.

FLOREANO, D.; MONDADA, F. Evolution of Homing Navigation in a Real Mobile Robot. **IEEE Transactions on Systems, Man and Cybernetics — Part B: Cybernetics**, [S.l.], v.26, n.3, p.396–407, 1996.

FOGEL, D. B. **Evolutionary Computation**: toward a new philosophy of machine intelligence. 2nd ed. NY: IEEE Press, 2000.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA: Addison-Wesley, 1989.

GREFENSTETTE, J. J. Genetic learning for adaptation in autonomous robots. In: INTERNATIONAL SYMPOSIUM ON ROBOTICS AND MANUFACTURING, 6., 1996. **Proceedings...** [S.l.]:ASME Press, 1996. v.6, p.265–270.

GREFENSTETTE, J. J.; RAMSEY, C. L.; SCHULTZ, A. Learning Sequential Decision Rules Using Simulation Models and Competition. **Machine Learning**, [S.l.], v.5, p.355–381, 1990.

GREFENSTETTE, J.; SCHULTZ, A. An Evolutionary Approach to Learning in Robots. In: MACHINE LEARNING WORKSHOP ON ROBOT LEARNING, 1994, New Brunswick, NJ. **Proceedings...** [S.l.: s.n.], 1994.

GU, D.; HU, H.; REYNOLDS, J.; TSANG, E. GA-based Learning in Behaviour Based Robotics. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, 2003, Kobe, Japan. **Proceedings...** [S.l.: s.n.], 2003.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor, MI: University of Michigan Press, 1975.

JENNINGS, N. R.; WOOLDRIDGE, M. J. Applications of Intelligent Agents. In: JENNINGS, N. R.; WOOLDRIDGE, M. J. (Ed.). **Agent Technology: foundations, applications, and markets**. Heidelberg: Springer-Verlag, 1998. p.3–28.

JONG, K. A. D. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems**. 1975. Tese (Doutorado em Ciência da Computação) — University of Michigan.

JONG, K. A. D. Learning with Genetic Algorithms: an overview. **Machine Learning**, [S.l.], v.3, n.2-3, p.121–138, 1988.

JONG, K. A. D. Using genetic algorithms to learn task programs: the pitt approach. **Machine Learning**, [S.l.], v.3, n.2-3, 1988.

KATAGAMI, D.; YAMADA, S. Interactive Classifier System for Real Robot Learning. In: IEEE INTERNATIONAL WORKSHOP ON ROBOT-HUMAN INTERACTION, ROMAN, 2000, Osaka, Japan. **Proceedings...** [S.l.: s.n.], 2000. p.258–263.

KOVACS, T.; LANZI, P. L. **A Learning Classifier Systems Bibliography**. [S.l.]: School of Computer Science, University of Birmingham, 1999. (CSRP-99-19).

KOZA, J. R. Concept formation and decision tree induction using the genetic programming paradigm. In: INTERNACIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE, PPSN, 1., 1991, Dortmund, Germany. **Proceedings...** Berlin: Springer-Verlag, 1991. p.124–128. (Lecture Notes in Computer Science, v.496).

KOZA, J. R. Evolution of subsumption using genetic programming. In: EUROPEAN CONFERENCE ON ARTIFICIAL LIFE. TOWARDS A PRACTICE OF AUTONOMOUS SYSTEMS, 1., 1992, Paris, France. **Proceedings...** [S.l.]:MIT Press, 1992. p.110–119.

KOZA, J. R. Evolution of a subsumption architecture that performs a wall following task for an autonomous mobile robot via genetic programming. In: HANSON, S. J.; PETSCHKE, T.; KEARNS, M.; RIVEST, R. L. (Ed.). **Computational Learning Theory and Natural Learning Systems**. Cambridge, MA, USA: MIT Press, 1994. v.2, p.321–346.

MEYER, J.-A.; HUSBANDS, P.; HARVEY, I. Evolutionary Robotics: a survey of applications and problems. In: EUROPEAN WORKSHOP ON EVOLUTIONARY ROBOTICS, EVOROBOT, 1., 1998. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1468).

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. [S.l.]: New York: Springer Verlag, 1996.

MIGLINO, O.; NAFASI, K.; TAYLOR, C. E. Selection for Wandering Behavior in a Small Robot. **Artificial Life**, [S.l.], v.2, n.1, p.101–116, 1994.

MITCHELL, M. **An introduction to genetic algorithms**. [S.l.]: MIT Press, Cambridge, MA, 1996.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997.

MONDADA, F.; FRANZI, E.; IENNE, P. Mobile Robot Miniaturization: a tool for investigation in control algorithms. In: INTERNATIONAL SYMPOSIUM ON EXPERIMENTAL ROBOTS, EXPERIMENTAL ROBOTICS III, LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES, 3., 1993, Kyoto. **Proceedings...** Berlin: Springer-Verlag, 1993. v.200, p.501–513.

MORIARTY, D. E.; SCHULTZ, A. C.; GREFENSTETTE, J. J. Evolutionary Algorithms for Reinforcement Learning. **Journal of Artificial Intelligence Research**, [S.l.], v.11, p.199–229, 1999.

NOLFI, S. Evolving non-trivial behavior on autonomous robots: adaptation is more powerful than decomposition and integration. In: EVOLUTIONARY ROBOTICS: FROM INTELLIGENT ROBOTICS TO ARTIFICIAL LIFE, ER, 1997, Ottawa, Canada. **Proceedings...** [S.l.]:AAI Books, 1997. p.21–48.

NOLFI, S.; FLOREANO, D. Learning and Evolution. **Autonomous Robots**, [S.l.], v.7, n.1, p.89–113, 1999.

NOLFI, S.; FLOREANO, D. **Evolutionary Robotics**: the biology, intelligence and technology of self-organizing machines. [S.l.]: MIT Press, 2000.

NOLFI, S.; FLOREANO, D.; MIGLINO, O.; MONDADA, F. How to Evolve Autonomous Robots: different approaches in evolutionary robotics. In: INTERNATIONAL WORKSHOP ON ARTIFICIAL LIFE, 4., 1994, Cambridge. **Proceedings...** MA: MIT Press, 1994. p.190–197.

NOLFI, S.; PARISI, D. Evolving non-Trivial Behaviors on Real Robots: an autonomous robot that picks up objects. In: CONGRESS OF THE ITALIAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE, 3., 1995. **Proceedings...** [S.l.: s.n.], 1995. p.243–254.

NORDIN, P.; BANZHAF, W. Genetic Programming Controlling a Miniature Robot. In: AAAI SYMPOSIUM ON GENETIC PROGRAMMING, 1995, MIT, Cambridge, MA. **Proceedings...** Menlo Park: CA: AAAI, 1995. v.1, p.61–67.

NORDIN, P.; BANZHAF, W. An On-Line Method to Evolve Behavior and to Control a Miniature Robot in Real Time with Genetic Programming. **Adaptive Behavior**, [S.l.], v.5, n.2, p.107–140, 1996.

PFEIFER, R.; IIDA, F.; BONGARD, J. **New Robotics**: design principles for intelligent systems. [S.l.]: MIT Press, 2004.

POLLACK, J. B.; LIPSON, H.; FICICI, S.; FUNES, P.; HORNBY, G.; WATSON, R. Evolutionary Techniques in Physical Robotics. In: INTERNATIONAL CONFERENCE ON EVOLVABLE SYSTEMS: FROM BIOLOGY TO HARDWARE, 3., 2000, Edinburgh. **Proceedings...** Berlin: Springer-Verlag, 2000. p.175–186. (Lecture Notes in Computer Science, v.1801).

QUINN, M. Evolving Communication without Dedicated Communication Channels. In: EUROPEAN CONFERENCE ON ARTIFICIAL LIFE, ECAL, 6., 2001, Prague, Czech Republic. **Proceedings...** Berlin: Springer, 2001. p.357–366. (Lectures Notes in Computer Science).

REYNOLDS, C. An evolved, vision-based model of obstacle avoidance behavior. In: REYNOLDS, C. (Ed.). **Artificial Life III**. [S.l.]: Redwood: Addison-Wesley, 1993.

SCHNEIDER, A.; PRESTES, E.; BARONE, D. Proposta de Aprendizagem Evolucionária para Robôs Móveis. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2004, Salvador (BA). **Anais...** Porto Alegre(RS): SBC, 2004.

SCHNEIDER, A.; PRESTES, E.; BARONE, D. Aprendizagem Evolucionária: uma abordagem para robótica móvel. In: SIMPÓSIO BRASILEIRO DE REDES NEURAIS, SBRN, 8., 2004, São Luis (MA), Brasil. **Anais...** [S.l.: s.n.], 2004.

SERRADA, A. P. **Una introducción a la Computación Evolutiva**. Valladolid: Instituto de Tecnologías Avanzadas de la Producción, Universidad de Valladolid, 1996. Technical Report. (PS96).

SIMÕES, E. V. Robótica Evolutiva. In: BARONE, D. A. C. (Ed.). **Sociedades Artificiais**: a nova fronteira da inteligência nas máquinas. Porto Alegre (RS) Brasil: Bookman, 2003. p.251–275.

SIMÕES, E. V.; BARONE, D. A. C. Predation: an approach to improving the evolution of real robots with a distributed evolutionary controller. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, ICRA, 2002, Washington, DC, USA. **Proceedings...** [S.l.]:IEEE, 2002. v.1, p.664–669.

SIMÕES, E. V.; DIMOND, K. R. An Evolutionary Controller for Autonomous Multi-Robot Systems. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 1999, Tokyo, Japan. **Proceedings...** [S.l.: s.n.], 1999. v.6, p.596–601.

SIMÕES, E. V.; DIMOND, K. R. Embedding a Distributed Evolutionary System into a Population of Autonomous Mobile Robots. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2001, Tucson, Arizona (USA). **Proceedings...** [S.l.: s.n.], 2001. v.2, p.1069–1074.

THOMPSON, A. An evolved circuit, intrinsic in silicon, entwined with physics. In: INTERNATIONAL CONFERENCE ON EVOLVABLE SYSTEMS: FROM BIOLOGY TO HARDWARE, ICES, 1996. **Proceedings...** Berlin: Springer-Verlag: 1997, 1996. p.390–405. (LNCS, v.1259).

THOMPSON, A. Artificial Evolution in the Physical World. In: EVOLUTIONARY ROBOTICS: FROM INTELLIGENT ROBOTS TO ARTIFICIAL LIFE, ER, 1997, Ottawa, Canada. **Proceedings...** [S.l.]:AAI Books, 1997. p.101–125.

TUCI, E. **An exploration on the evolution of learning behaviour using robot based models**. 2004. Tese (Doutorado em Ciência da Computação) — University of Sussex, Brighton (UK).

VARGAS, P. A.; LYRA, C.; ZUBEN, F. J. V. Application of Learning Classifier Systems to the On Line Reconfiguration of Electric Power Distribution Networks. In: BULL, L. (Ed.). **Applications of Learning Classifier Systems**. [S.l.]: Springer, 2004. p.260–275.

WANG, L. Computational intelligence in autonomous mobile robotics - A review. In: INTERNATIONAL SYMPOSIUM ON MICROMECHATRONICS AND HUMAN SCIENCE. MHS, 2002, Nayoga. **Proceedings...** [S.l.]:IEEE, 2002. p.227–235.

WATSON, R. A.; FICICI, S. G.; POLLACK, J. B. Embodied Evolution: embodying an evolutionary algorithm in a population of robots. In: CONGRESS ON EVOLUTIONARY COMPUTATION, 1999, Mayflower Hotel, Washington D.C., USA. **Proceedings...** [S.l.]:IEEE Press, 1999. v.1, p.335–342.

WATSON, R. A.; FICICI, S. G.; POLLACK, J. B. Embodied Evolution: distributing an evolutionary algorithm in a population of robots. **Robotics and Autonomous Systems**, [S.l.], v.39, n.1, p.1–18, 2002.

WERGER, B. B.; MATARIC, M. J. **Exploiting Embodiment in Multi-Robot Teams**. Los Angeles: University of Southern California, Institute for Robotics and Intelligent Systems, 1999. Technical Report. (IRIS-99-378).

WHITLEY, D. An overview of evolutionary algorithms: practical issues and common pitfalls. **Information and Software Technology**, [S.l.], v.43, n.14, p.817–831, 2001.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: theory and practice. **The Knowledge Engineering Review**, [S.l.], v.10, n.2, p.115–152, 1995.

YEPES, I. **Algoritmo Genético com Idades para o Controle de Trajetória de um Braço Mecânico**. 1999. Trabalho de Conclusão de Curso — Universidade Regional Integrada do Alto Uruguai e das Missões, Erechim (RS), Brasil.