UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Programação Genética, Redes Neurais
e o Jogo da Minoria

Fabiano Lemes Ribeiro

*Dissertação apresentada
ao Instituto de Física da
Universidade de São Paulo
para a obtenção do título de
Mestre em Ciências*

Orientador : Prof. Dr. Nestor Felipe Caticha Alfonso

Banca Examinadora:

Prof. Dr. Alessandro Paulo S. Moura (IF-USP)

Prof. Dr. Marcus Vinícius Baldo (ICB-USP)

Prof. Dr. Nestor Felipe Caticha Alfonso (IF-USP)

SÃO PAULO
2005

Aos meus amores,
Isabela e Luciane

AGRADECIMENTOS

À Deus, por ser meu guia.

Ao CNPq pelo apoio financeiro, que tornou possível a realização deste trabalho.

Ao Prof. Nestor Caticha pela sua orientação paciente e amiga.

Ao Juan pelos tão motivantes ensinamentos, não apenas relacionados à Programação Genética, mas também à própria vida.

Ao Renato Vicente pelos debates sobre Econofísica.

À Neide pela imensa ajuda com a nossa tão difícil língua portuguesa.

Ao pessoal do apoio computacional, sempre tão prestativos.

À Profa. Tânia Tomé pelas discussões sobre o modelo predador-presa.

Ao Prof. Salinas pelo grande apoio no início de meu mestrado.

Agradeço também aos meus amigos de sala e do Dept. de Física Geral pela amizade sincera. Em especial ao Carlos e ao Evaldo, que se portaram como verdadeiros irmãos em tempos difíceis.

Às secretárias Ivone, Dirce e Silvana, que sempre me ajudaram com grande competência e satisfação.

Aos amigos de Alfenas, que mesmo pela distância, não deixam de estar presentes em meus pensamentos.

À minha mãe e meu pai por terem me ensinado a arte do amor e do respeito. Como vocês me ajudaram!

Aos meus irmãos Kátia, Talisson, Jeferson e Anderson e todos os meus familiares por acreditarem em mim.

Por último quero enfatizar a minha felicidade de ter junto a mim, duas companheiras. Luciane meu amor e Isabela minha filha, não consigo imaginar minha vida sem vocês!

RESUMO

O objetivo desta dissertação foi a implementação de uma plataforma de otimização por Programação Genética (PG) com o intuito de estudar e caracterizar as propriedades estatísticas de uma grande classe de problemas. Esta implementação foi feita através de programas escritos em LISP, executados num *cluster* de computadores com o sistema operacional Linux.

A plataforma foi usada para estudar uma versão do *Jogo da Minoria* (JM) onde seus jogadores utilizam redes neurais para a realização de suas escolhas. Os jogadores foram divididos em dois grupos distintos. O primeiro formado por jogadores que apresentam estratégias estáticas e portanto não adquirem aprendizado. O segundo grupo é formado por jogadores que utilizam um algoritmo de aprendizado para alterar suas estratégias de identificação da minoria. Mostramos que, em determinadas condições, estes jogadores adaptativos conseguem identificar padrões nas escolhas dos jogadores não-adaptativos e assim optam pela decisão da minoria. Porém a eficiência nesta identificação depende do algoritmo de aprendizado utilizado. O algoritmo de aprendizado gerado pela PG se apresentou mais eficiente que outros algoritmos analisados, como, por exemplo, o algoritmo hebbiano. Esta eficiência é caracterizada por uma emergência espontânea de coordenação entre estes jogadores e que lhes proporcionam um melhor desempenho médio por jogador.

ABSTRACT

The aim of this work is to describe an optimization platform through Genetic Programming in order to study and characterize the statistical property of a wide class of problems. This implementation was written in LISP and executed in a cluster of computers running the Linux operational system.

The platform was used to study a version of the Minority Game where the players used neural networks to make their choices. The players were divided into two distinct groups. The first group was made up of players that had quenched strategies and therefore could not learn. The second group had players that used learning algorithms to change their strategies for minority identification. We showed that, under some conditions, the adaptatives players are able to identify patterns in the choices of the non-adaptatives players and can thus benefit by choosing the minority decision. The efficiency in this identification depends on the learning algorithm. The algorithm generated by Genetic Programming is more efficiency than the others algorithms analysed such as hebbian perceptron learning. This efficiency is characterized by a spontaneous emergence of coordination between this players, which permits earning higher scores than average.

RELAÇÃO DE PARÂMETROS

AG : Algoritmos Genéticos;
ES : Expressão Simbólica;
ESF : Expressão Simbólica Fiel;
FMOD : Função de Modulação;
JEST's : Jogadores Estáticos;
JGEN's : Jogadores Genéticos;
JGENr's : Jogadores Genéticos reais;
JGENv's : Jogadores Genéticos virtuais;
JHEBB's : Jogadores Hebbianos;
JINT's : Jogadores Inteligentes;
JM : Jogo da Minoria;
JMP : Jogo da Minoria Padrão;
ME : Macânica Estatística;
PG : Programação Genética;

d : memória do vetor-história e dimensão dos vetores;
 e : índice que representa JINT's;
 $e_{pe}(t)$: erro médio de predição dos JINT's em uma dada rodada t ;
 $E_i(t)$: fitness padronizado do i -ésimo jogador em uma rodada t ;
 E_i : fitness padronizado do i -ésimo jogador após P rodadas.
 $E_{ei}(t)$: fitness padronizado do i -ésimo JINT em uma dada rodada t ;
 $E_e(t)$: fitness padronizado médio dos JINT's em uma dada rodada t ;
 E_e : fitness padronizado médio dos JEST's após P rodadas;
 $\mathcal{F}$: função de transferência;
 $\mathcal{F}$: Conjunto de operadores;
 g : geração corrente;

- g_{max} : número máximo de gerações;
 h : campo local no perceptron;
 $\mathbf{J}_i(t)$: vetor sináptico do i -ésimo perceptron numa dada rodada t ;
 $\mathbf{J}_{ei}(t)$: vetor sináptico do i -ésimo JINT numa dada rodada t .
 $\mathbf{J}_e$: vetor centro de massa dos JINT's;
 $\mathbf{J}_e(g)$: vetor centro de massa dos JINT's no final de uma competição promovida na geração g ;
 $\mathbf{J}_{ei}(g)$: vetor sináptico do i -ésimo JINT's no final de uma competição promovida na geração g ;
 $\mathbf{J}_o$: vetor centro de massa dos JEST's;
 $m(t)$: flutuação das escolhas dos JEST's em uma dada rodada t ;
 M : número de indivíduos que serão copiados para a próxima geração;
 n_i : fitness normalizado do i -ésimo JGEN;
 N : número total de jogadores participantes do JM ;
 N_o : número de JEST's;
 $N_{o+}(t)$: número de JEST's que optaram por +1 em uma dada rodada t ;
 $N_{o-}(t)$: número de JEST's que optaram por -1 em uma dada rodada t ;
 N_e : número de JINT's;
 N_s : número de experimentos;
 N_{pg} : número total de ESF's na população e também o número total de JGEN's;
 o : índice que representa JEST's;
 P : número total de rodadas da competição;
 P_{cros} : probabilidade de acontecer um crossover para se formar dois novos indivíduos;
 P_r : probabilidade de acontecer uma cópia simples para se formar um novo indivíduo;
 P_m : probabilidade de acontecer uma mutação para se formar um novo indivíduo;
 $P(m)$: probabilidade que uma população de perceptrons, com vetores sinápticos homogeneamente distribuídos, adquira um particular valor m , dado um vetor $\mathbf{S}$ aleatório;
 P_a : probabilidade que a resposta da minoria seja afetada pelos JINT's;
 P_n : probabilidade que a resposta da minoria não seja afetada pelos JINT's;
 s : número de tabelas de decisões portadas por cada jogador no JMP .
 $S_e(g)$: entropia fenotípica dos JGENr's
 $\mathbf{S}(t)$: vetor-história numa dada rodada t ;

- $\mathbf{S}$: vetor de entrada do perceptron;
 t : rodada do JM;
 $\mathcal{V}$: conjunto de variáveis;
 β : parâmetro de controle do fitness normalizado;
 $\Delta \mathbf{J}_{ei}(t)$: atualização do vetor sináptico do i -ésimo JINT.
 Δ^*m : desvio padrão de m ;
 η_i : FMOD do i -ésimo JGEN;
 $\phi_i(t)$: ganho do i -ésimo jogador numa dada rodada t .
 $\Phi_i(t)$: fitness do i -ésimo jogador em uma dada rodada t ;
 $\Phi_e(t)$: fitness médio dos JINT em uma dada rodada t ;
 Φ_e : fitness médio dos JINT após P rodadas;
 $\Phi_{ei}(t)$: fitness do i -ésimo JINT em uma dada rodada t ;
 Φ_{ei} : fitness do i -ésimo JINT após P rodadas.
 $\Phi_o(t)$: fitness médio dos JEST's em uma dada rodada t ;
 Φ_o : fitness médio dos JEST's após P rodadas;
 $\Phi_{oi}(t)$: fitness do i -ésimo JEST em uma dada rodada t ;
 Φ_{oi} : fitness do i -ésimo JEST após P rodadas;
 λ : parâmetro de proporção;
 $\rho(\mathbf{w}_a, \mathbf{w}_b)$: parâmetro de correlação entre os vetores $\mathbf{w}_a$ e $\mathbf{w}_b$;
 $\sigma_m(t)$: resposta da minoria numa rodada t ;
 $\sigma_i(t)$: resposta do i -ésimo jogador numa dada rodada t ;
 $\sigma_{oi}(t)$: resposta do i -ésimo JEST numa dada rodada t ;
 $\sigma_{ei}(t)$: resposta do i -ésimo JINT numa dada rodada t ;
 σ^2 : variância;
 Ψ_e : correlação média entre os vetores sinápticos dos JINT's;
 $\Psi_e(g)$: correlação média entre os vetores sinápticos dos JGENr's no final da competição promovida na geração g ;
 Ψ_o : correlação média entre os vetores sinápticos dos JEST's.

Sumário

1	Introdução	1
1.1	Estrutura da Dissertação	4
2	A Programação Genética	5
2.1	Estudo da Linguagem LISP	6
2.2	Os Operadores Genéticos	8
2.3	Algoritmo de Implementação	11
3	O Jogo da Minoria	12
3.1	Definição Matemática do Jogo da Minoria	14
3.2	Jogo da Minoria Padrão	16
4	O Perceptron no Jogo da Minoria	19
4.1	O Perceptron Simples	20
4.2	Aprendizagem de um Perceptron	22
5	Jogo da Minoria com dois Grupos de Jogadores	26
5.1	Definição das Estruturas dos Jogadores	28
5.2	Medidas de desempenho	29
5.3	Perceptrons Hebbianos e Perceptrons Estáticos	31
5.4	Resultados	35
5.5	Discussão	46
6	Resultados da Programação Genética	48
6.1	O Modelo para o Jogo da Minoria	48
6.2	Técnicas de Caracterização da Dinâmica de Evolução	51
6.3	Análise Comparativa entre os Algoritmos de Aprendizado	55
6.4	Análise da Evolução dos Jogadores Genéticos	56

6.5	Discussão	72
7	Conclusões e Perspectivas Futuras	76
	Apêndice A: Uma breve análise de um sistema de escolhas aleatórias	81
	Apêndice B: O cenário Professor-Estudante	85
	Bibliografia	89

Lista de Figuras

2.1	Expressão algébrica $x+x^2$ representada na forma de árvore gramatical e de uma ESF.	8
2.2	Indivíduo, em sua representação de árvore gramatical e de uma expressão algébrica, antes e depois de uma mutação. Um átomo é aleatoriamente selecionado (representado pelo átomo hachurado) e é substituído por um outro átomo de mesmo tipo, escolhido aleatoriamente. Neste exemplo, um operador de multiplicação (*) é substituído por um operador de subtração (-).	9
2.3	Dois indivíduos pais selecionados da população para se realizar o crossover. Os pontos assinalados são escolhidos aleatoriamente; logo depois, os “ramos” provenientes de cada um dos pontos são intercambiados, originando duas novas ESF’s descendentes.	10
2.4	Indivíduos descendentes dos indivíduos da figura (2.3), gerados por crossover.	10
2.5	As três operações de PG utilizadas para a formação de uma geração $g + 1$ através de indivíduos da geração g	10
3.1	Desenho esquemático do JMP numa dada rodada. Neste caso, os agentes tomam suas decisões através dos últimos três resultados da minoria ($d = 3$). A linha mais espessa da figura mostra como a informação flui no sistema: os N agentes consideram o vetor-história $(-1 + 1 + 1)$ e emitem uma resposta. Note que um agente, utilizando a tabela de decisões da tabela (3.1), emitiria a resposta $+1$. As N respostas são transformadas num novo resultado da minoria (neste caso: -1) através da regra da minoria. Esta informação é então apresentada aos jogadores para a atualização dos fitness de suas tabelas de decisões. Finalmente, este resultado da minoria torna-se o próximo número na sequência armazenada pelo vetor-história.	17

4.1	Arquitetura do perceptron simples.	21
4.2	Desenho esquemático do JM entre perceptrons, numa dada rodada t e com uma memória $d = 3$. A linha mais espessa da figura mostra como a informação externa flui no sistema: os N perceptrons consideram o vetor-história $(-1+1+1)$ e emitem uma resposta. As N respostas são transformadas num novo resultado da minoria (neste caso: -1). Esta informação, juntamente com o vetor-história, são então apresentados aos jogadores para a atualização dos fitness e dos vetores sinápticos, sendo estes últimos atualizados através de algum algoritmo de aprendizado. Finalmente, este resultado da minoria torna-se o próximo número na sequência armazenada pelo vetor-história.	23
5.1	Figura esquemática das escolhas dos jogadores numa rodada típica, com todos os JHEBB's realizando a mesma decisão. Nesta figura apresentam-se três situações prováveis: a) $N_e < m$ e os JHEBB's posicionando-se com a minoria dos JEST's; b) $N_e > m$ e os JHEBB's posicionando-se com a maioria dos JEST's; c) $N_e > m$ e os JHEBB's posicionando-se com a minoria dos JEST's. Na primeira situação, a população de JHEBB's é insuficiente para afetar a resposta da minoria; já nas duas últimas situações, a resposta da minoria é sempre contrária à resposta dos JHEBB's.	33
5.2	Distribuição gaussiana de $P(m)$. A área hachurada representa a probabilidade P_a de que os JHEBB's afetem o sinal da minoria.	34
5.3	Gráfico de P_n em função do parâmetro de proporção λ	34
5.4	Gráfico de $\langle \Phi_e \rangle$ em função do parâmetro de proporção λ para diferentes dimensões. Os parâmetros utilizados foram: $P = 400$, $N_o = 1000$ e $N_s = 1000$	35
5.5	Gráfico a): fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e, \text{melhor}}(t)$) e do pior JHEBB ($\Phi_{e, \text{pior}}(t)$) da competição. Gráfico b): erro de predição dos JHEBB's ao longo da competição. Estes resultados são para $\lambda = 0.3$	38
5.6	Gráfico de Ψ_e , $\rho(\mathbf{J}_o, \mathbf{S})$ e $\rho(\mathbf{J}_o, \mathbf{J}_e)$ em função das rodadas da competição. O gráfico menor é uma ampliação do gráfico de $\rho(\mathbf{J}_o, \mathbf{S})$. Estes resultados são para $\lambda = 0.3$	38

- 5.7 Módulo da flutuação m ao longo das rodadas da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (bem menor que 1), a maior parte dos pontos se encontra acima da linha tracejada. Estes resultados são para $\lambda = 0.3$ 39
- 5.8 Gráfico da dinâmica de $\frac{N_{e+}}{N_o}$ e $\frac{N_{e-}}{N_e}$, mostrando a característica periódica das respostas dos JEST's. Pode-se perceber também, os JHEBB's "fugindo" da resposta da maioria dos JEST's. Estes resultados são para $\lambda = 0.3$ 40
- 5.9 Gráfico a): fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e,melhor}(t)$) e do pior JHEBB ($\Phi_{e,pior}(t)$) da competição. Gráfico b): erro de predição dos JHEBB's ao longo da competição. Estes resultados são para $\lambda = 0.6$ 42
- 5.10 Gráfico de Ψ_e , $\rho(\mathbf{J}_o, \mathbf{S})$ e $\rho(\mathbf{J}_o, \mathbf{J}_e)$ em função de t . Estes resultados são para $\lambda = 0.6$ 42
- 5.11 Módulo da flutuação m ao longo das rodadas da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (menor que 1), a maior parte dos pontos ainda se encontra acima da linha tracejada. Estes resultados são para $\lambda = 0.6$ 43
- 5.12 Dinâmica das escolhas dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.6$ 43
- 5.13 gráficos de $\frac{N_{e+}}{N_e}$ em função de $\frac{N_{o+}}{N_o}$. A maior frequência de pontos nos quadrantes 2 e 4 deste gráfico sugere que, na maior parte das rodadas, os JHEBB's realizam uma resposta igual à da minoria dos JEST's. Estes resultados são para $\lambda = 0.6$ 44
- 5.14 Fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e,melhor}(t)$) e do pior JHEBB ($\Phi_{e,pior}(t)$) da competição. Estes resultados são para $\lambda = 1.1$ 45
- 5.15 Gráfico de Ψ_e , $\rho(\mathbf{J}_o, \mathbf{S})$ e $\rho(\mathbf{J}_o, \mathbf{J}_e)$ em função de t . Estes resultados são para $\lambda = 1.1$ 45
- 5.16 Módulo da flutuação m ao longo da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (maior que 1), a maior parte dos pontos se encontra abaixo da linha tracejada. Estes resultados são para $\lambda = 1.1$ 46

6.1	Algoritmo do modelo utilizado.	52
6.2	$\langle \Phi_e \rangle$ em função de λ , para JINT's utilizando diferentes algoritmos de aprendizado. Para o caso dos JINT's utilizando o aprendizado com a PG, foi utilizado $g_{max} = 100$ e a média realizada sobre $N_s = 30$ experimentos diferentes, para cada valor de λ . Para os JINT's utilizando de qualquer um dos demais algoritmos, o fitness médio foi obtido a partir de $N_s = 1000$ para cada valor de λ . Para todos os casos foram utilizados $d = 11$, $P = 400$ e $N_o = 1000$	56
6.3	Gráfico da evolução do fitness médio dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.3$	59
6.4	Gráficos da distribuição do fitness e da entropia fenotípica dos JGENr's em função da geração. No gráfico da distribuição, cada ponto representa o fitness $\Phi_{ei}(g)$ do JGENr i . Uma linha vertical fictícia deste gráfico apresenta a pontuação de cada um dos JGENr's que atuaram na competição promovida na geração g . Estes resultados são para $\lambda = 0.3$	59
6.5	Gráfico a): Fitness médios dos dois grupos de jogadores referente a competição promovida na geração $g = 500$. Gráfico b): Gráfico de $\rho(\mathbf{J}_o, \mathbf{J}_e)$, $\rho(\mathbf{J}_o, \mathbf{S}(t))$ e $\Psi_e(t)$ para esta mesma competição. Estes resultados são para $\lambda = 0.3$	60
6.6	Dinâmica das escolhas dos dois grupos de jogadores referente à competição promovida na geração $g = 500$. Estes resultados são para $\lambda = 0.3$	61
6.7	Gráfico acima: evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Gráfico abaixo: plano de fase entre as quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para cada geração g . Estes resultados são para $\lambda = 0.3$	61
6.8	Gráfico da evolução do fitness médio dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.9$	63
6.9	Gráfico da distribuição do fitness e da entropia fenotípica dos JGENr's em função da geração. Estes resultados são para $\lambda = 0.9$	63
6.10	Gráfico acima: fitness médios dos dois grupos de jogadores referente à competição promovida na geração $g = 500$. Gráfico abaixo: erro médio de predição dos JGENr's referente a essa mesma competição. Estes resultados são para $\lambda = 0.9$	64

- 6.11 Gráfico de $\rho(\mathbf{J}_o, \mathbf{J}_e)$, $\rho(\mathbf{J}_o, \mathbf{S}(t))$ e $\Psi_e(t)$ referente à competição promovida na geração $g = 500$. Estes resultados são para $\lambda = 0.9$ 64
- 6.12 Plano de fase de $\frac{N_{e+}}{N_e}$ e $\frac{N_{o+}}{N_o}$ referente a competição promovida na geração $g = 500$. A maior densidades de pontos nos quadrantes pares sugere que a maioria dos JGENr's se posiciona junto à minoria dos JEST's. Estes resultados são para $\lambda = 0.9$ 65
- 6.13 Gráfico acima: evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Gráfico abaixo: plano de fase entre as quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para cada geração g . Estes resultados são para $\lambda = 0.9$ 65
- 6.14 Gráficos da evolução do fitness médio dos dois grupos de jogadores. Apresenta-se também, a evolução dos fitness dos JGENr's mais e menos pontuados da geração, representados por $\Phi_{e, melhor}$ e $\Phi_{e, pior}$, respectivamente. Estes resultados se referem a $\lambda = 1.5$ 68
- 6.15 Gráfico Acima: Distribuição do fitness entre os JGENr's ao longo das gerações. Neste gráfico, cada ponto representa o fitness obtido por um particular JGENr na geração g . Note uma explícita alteração nesta distribuição a partir de g_c . Gráfico abaixo: evolução da entropia fenotípica entre os JGENr's. Como esta entropia está associada a distribuição do fitness dos JGENr's, também acontece uma alteração significativa em sua dinâmica a partir de g_c . Esta alteração da entropia fenotípica indica uma redução da diversidade de fitness entre os JGENr's. Estes resultados se referem a $\lambda = 1.5$ 69
- 6.16 Gráficos de $\langle ||\mathbf{J}_e|| \rangle$ em função das gerações. O gráfico inferior é igual ao superior, exceto pela região das ordenadas. Estas figuras mostram que no início das gerações (quando $\Phi_e(g)$ é bastante baixo), $\langle ||\mathbf{J}_e|| \rangle$ assume valores altos; porém, a partir de g_c , quando $\Phi_e(g)$ entra no regime de maturidade, a frequência de valores pequenos, assumidos por $\langle ||\mathbf{J}_e|| \rangle$, torna-se evidentemente alta. Estes resultados se referem a $\lambda = 1.5$ 70
- 6.17 Gráfico da evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Estes resultados se referem a $\lambda = 1.5$ 71
- 6.18 Plano de fase referentes às quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para duas regiões de evolução de $\Phi_e(g)$. Estes resultados se referem a $\lambda = 1.5$. 72

6.19	Figuras esquemáticas da disposição dos vetores, num espaço bi-dimensional, referente a uma rodada típica. A figura a) apresenta uma situação em que todos os JGENr's se posicionam junto à minoria dos JEST's, pois $\Psi_e \approx 1$ e $\mathbf{J}_e \approx -\mathbf{J}_o$. A figura b) apresenta uma situação em que $\Psi_e \neq 1$	73
A.1	Gráfico do desvio padrão das escolhas dos perceptrons em função de N_o . Este gráfico apresenta o resultado de simulações sobre uma média de 1000 vetores de entrada independentes para cada valor N_o	83
B.1	Gráfico de e_g em função de t e de α , para diversos níveis de ruído.	87

Lista de Tabelas

3.1	Um exemplo de tabela de decisões para $d = 3$	16
6.2	Parâmetros que foram mantidos fixos em nossos experimentos com a PG.	53

1

INTRODUÇÃO

“Meu pai e o pai de meu pai, antes de mim, abriram aqui as suas tendas... Há doze séculos que os verdadeiros crentes - Deus seja louvado, só eles possuem a verdadeira sabedoria - se estabeleceram neste país, e nenhum deles ouviu jamais falar em palácios subterrâneos, nem tampouco os que os procederam. Então, eis que aparece um franco de um país afastado muitos dias de viagem, vai diretamente ao lugar, pega dum bastão e traça uma linha para cá, uma linha para lá : Aqui, diz ele, está o palácio; e lá está o portão; e mostra-nos o que durante toda a nossa vida esteve sob nossos pés, sem que nada soubéssemos. Maravilhoso! Maravilhoso! Aprendeste isso pelos livros, por artes mágicas ou pelos vossos profetas? Di-lo, ó Bei! Diz-me o segredo da sabedoria!”

Discurso do xequê Abd-er-Rahman ao
arqueólogo inglês Layard

O estudo da Mecânica Estatística (ME) de sistemas desordenados têm despertado, nas últimas décadas, o interesse por métodos de otimização em geral. Isso favoreceu o desenvolvimento de novas técnicas por meio de métodos da ME. Neste trabalho, foi estudada uma técnica computacional de otimização, inspirada em evolução *darwiniana*: a Programação Genética (PG). A estrutura de um problema típico atacado pela PG é a seguinte. Seja dado um problema, um conjunto de representações de soluções candidatas e uma medida de desempenho (*fitness*¹) a estas soluções. Os problemas a serem atacados são não triviais no sentido que pode ser difícil encontrar

¹Todos os termos técnicos apresentados nesta e na próxima seção serão cuidadosamente definidos nos próximos capítulos.

soluções dentro de uma tolerância pré-estabelecida. Porém, uma vez dada a solução candidata, a verificação de seu sucesso é rápida. Começa-se com uma população de soluções candidatas, obtidas aleatoriamente, e, a partir dela, é gerada uma nova população (a nova geração) através dos *operadores genéticos*. Os operadores genéticos são os mecanismos da PG que proporcionam a criação de novas soluções, em uma população, a partir das melhores soluções da população precedente. Exemplos de operadores genéticos são: o *crossover*, que realiza uma mistura das melhores soluções; e a *mutação*, que realiza alterações aleatórias nas melhores soluções. A medida do sucesso de uma solução determina o número de soluções dela derivadas que estarão presentes na próxima geração. Características adequadas tendem a ser mantidas na população e um certo grau de aleatoriedade, presente nos operadores genéticos, possibilita encontrar novas candidatas à solução do problema. Este processo de geração de uma nova população é feito de forma iterativa até que um dado *critério de terminação* seja alcançado.

Há uma liberdade muito grande em escolher a representação das soluções. Entre elas, podemos citar o *Algoritmo Genético* (AG), que usa uma representação simples, não hierárquica, por meio de um vetor de parâmetros de dimensão fixa. Essa técnica, apesar de ser bastante versátil na procura por soluções em uma dada classe de problemas, tem em sua simplicidade um empecilho a uma atuação em problemas mais complexos. A PG, por outro lado, utiliza uma representação mais geral, com soluções representadas como programas. Cada programa implementa um algoritmo que possibilita resolver, com um certo sucesso, o problema dado.

O objetivo desta dissertação foi a implementação de uma plataforma de otimização por PG com o intuito de estudar e caracterizar as propriedades estatísticas de uma grande classe de problemas. Esta implementação foi feita através de programas escritos em LISP, executados num *cluster* de computadores com o sistema operacional Linux. Finalmente, com a implementação desta técnica, foi escolhido o *Jogo da Minoridade* (JM) como o problema a ser atacado. O JM é um exemplo de *sistema de agentes interagentes*, que tem seu ambiente moldado pelas decisões dos jogadores participantes. Os jogadores recebem informações deste ambiente e escolhem uma entre duas alternativas. Essas escolhas são baseadas em algum tipo de estratégia, visando a opção da minoria dos jogadores. A PG será utilizada neste problema para gerar uma estratégia que proporcione, a um dado jogador, condições para realizar a opção da minoria mediante as informações do ambiente. Em nosso modelo, vamos considerar que todos os jogadores utilizam o *perceptron booleano*, de redes

neurais, para a realização de suas escolhas, conforme introduzido em [6], porém, de forma ainda inédita na literatura, divididos em dois grupos distintos. O primeiro é constituído por jogadores que possuem estratégias fixas, não apresentando qualquer adaptação ao ambiente. Esta propriedade destes jogadores impõe certas regularidades previsíveis ao ambiente, que podem ser aproveitadas pelo segundo grupo de jogadores, formado por jogadores adaptativos. Estes jogadores “inteligentes” utilizam algum tipo de *algoritmo de aprendizado* para adaptar suas estratégias ao ambiente, de modo a emitir a resposta da minoria.

A eficiência na identificação de regularidades do ambiente, por parte destes jogadores adaptativos, depende de vários parâmetros, como a quantidade de jogadores adaptativos e do algoritmo utilizado por estes jogadores para realizar as adaptações. Neste sentido, vamos realizar extensas comparações entre desempenhos destes jogadores portando vários algoritmos de aprendizado comuns na literatura e um algoritmo de aprendizado gerado pela PG. Verificaremos que a eficiência de jogadores adaptativos que utilizam o aprendizado com a PG é nítida se comparada a outros algoritmos. Se, por um lado, paga-se pelo custo computacional ao utilizar um algoritmo de aprendizado gerado por esta técnica, ganha-se pelos seus melhores resultados. Mostraremos que esta eficiência é caracterizada por uma emergência espontânea de coordenação entre estes jogadores. Esta coordenação surge apesar dos jogadores agirem de forma egoísta, visando a maximização de ganhos próprios, sem uma escolha consciente de agir pelo bem do grupo.

São descritas também, neste trabalho, algumas características observadas neste modelo com dois grupos de jogadores no JM. Por exemplo, devido à existência de jogadores não-adaptativos, podem ser identificadas as condições de existência de estratégias estáveis entre os jogadores adaptativos. Num JM composto unicamente por jogadores adaptativos, a presença de estratégias estáveis torna-se impossível devido a regra da minoria presente no jogo. Verificaremos que a probabilidade de existência de estratégias estáveis acontece quando a diferença entre o número de jogadores não-adaptativos que optam por uma alternativa e o número de jogadores deste mesmo grupo que optam pela outra for maior que a população dos jogadores adaptativos. Surge aí uma espécie de exploração dos jogadores não-adaptativos pelos jogadores inteligentes, a partir da previsibilidade das escolhas destes primeiros jogadores. Esta diferença citada, que chamaremos de flutuações das escolhas dos jogadores não-adaptativos, será bastante explorada neste trabalho, sendo uma variável relevante para a determinação da probabilidade de identificação da regra da

minoria pelos jogadores inteligentes. Esta probabilidade, como veremos, diminui suavemente e sem *transições de fase*, com o aumento da população dos jogadores adaptativos.

1.1 Estrutura da Dissertação

Esta dissertação está organizada em 7 capítulos.

No capítulo 2, apresenta-se o paradigma da PG. Serão descritos os processos de seleção, os operadores genéticos, o algoritmo de implementação e a linguagem de programação utilizada, o LISP.

No capítulo 3, será introduzido o Jogo da Minoria, bem como a sua formulação matemática. Descrevem-se também algumas quantidades que serão utilizadas em todo o decorrer da dissertação. No final do capítulo, apresenta-se uma pequena revisão dos resultados comuns na literatura a respeito do chamado *Jogo da Minoria Padrão*.

No capítulo 4, introduz-se o perceptron no Jogo da Minoria. Será feito uma breve apresentação da arquitetura do perceptron e de seus mecanismos de aprendizagem. Serão também apresentados alguns algoritmos de aprendizado comuns na literatura e que serão analisados neste trabalho.

No capítulo 5, define-se o modelo do Jogo da Minoria com dois grupos de jogadores. Será apresentada a estrutura dos perceptrons dos jogadores, e descrita a dinâmica proporcionada por este modelo. Finalmente serão apresentados resultados dos experimentos, em que os jogadores inteligentes utilizam o perceptron com aprendizado hebbiano para modificar suas estratégias.

No capítulo 6, apresentam-se os resultados obtidos com a PG no JM com dois grupos de jogadores. Será verificada a eficiência dos jogadores que utilizam esta técnica para realizar suas respostas através de comparações com outros algoritmos. As razões desta eficiência serão detalhadamente discutidas. Uma delas é a emergência espontânea de uma coordenação entre as respostas dos jogadores que utilizam esta técnica.

Finalmente, no capítulo 7, apresentam-se as conclusões e as perspectivas futuras para este trabalho.

2

A PROGRAMAÇÃO GENÉTICA

“Pode-se indagar como é que as variedades, que eu denomino espécies nascentes, acabaram por se transformar em espécies verdadeiras e distintas, as quais, na maioria dos casos, diferem efetivamente muito mais entre si que as variedades de uma mesma espécie; como se formam estes grupos de espécies que constituem o que se denomina gêneros distintos, e que diferem mais uns aos outros que as espécies do mesmo gênero? Todos estes efeitos derivam de uma causa: a luta pela sobrevivência. Devido a esta luta, as variações, por mais fracas que sejam e seja qual for a causa de onde provenham, tendem a preservar os indivíduos de uma espécie e transmitem-se à descendência logo que sejam úteis a esses indivíduos nas suas relações por demais complexas com os outros seres organizados e com as condições físicas da vida. Os descendentes terão, por si mesmos, em virtude disso, maior probabilidade de sobrevivência; porque, dos indivíduos de uma espécie nascidos periodicamente, um pequeno número poderá sobreviver. Denominei este princípio, pelo qual cada variação diminuta, se útil, é preservada, com o termo Seleção Natural, com a finalidade de salientar sua relação com o poder humano de seleção.”

Darwin, A Origem das Espécies, cap III

Inspirada nos processos de evolução por seleção natural, a PG, da mesma forma que o AG [1], emprega métodos genéticos na procura de uma solução ótima para um dado problema. Um estudo completo a respeito da PG pode ser encontrado em [2].

O AG convencional trabalha através da manipulação de seqüências de caracteres que representam candidatos à solução para um dado problema. Porém, para muitos problemas, programas computacionais hierárquicos são a mais natural representação para a solução. O programa candidato à solução deve ter, então, a potencialidade de

alterar seu tamanho e forma, de modo a buscar uma configuração que melhor solucione dado problema, uma vez que o tamanho e a forma do programa que representa a solução são desconhecidos.

A PG trabalha com este paradigma, agindo sobre estruturas hierárquicas (i.e. programas) que não possuem forma ou tamanho determinados *a priori*. A simulação inicia-se com a construção aleatória de N_{pg} programas - uma *população* - com cada um destes programas - um *indivíduo* - diferente de qualquer outro. Todos estes programas são construídos a partir de um conjunto pré-determinado de operadores e variáveis. Este processo de construção deve respeitar algumas regras, de modo a evitar a criação de um programa que retorne uma mensagem de erro quando avaliado.

Logo em seguida, cada indivíduo é julgado com respeito à solução do problema que implementa, recebendo uma pontuação. Essa pontuação, por medir a adaptabilidade ou eficiência de um programa em resolver o problema proposto, será interpretada como o fitness do indivíduo. Um novo passo acontece com a construção de uma nova população de programas (i.e. a nova geração) a partir de seleção *darwiniana* dos melhores indivíduos e pela atuação dos operadores genéticos. O ciclo então recomeça, com estes programas da nova geração sendo julgados, selecionados e os operadores genéticos aplicados novamente. Esse ciclo se desenvolve de forma iterativa até ser alcançado o critério de terminação, dado por um número máximo de gerações, definido por g_{max} , ou por um valor de fitness desejado.

Este capítulo está organizado da seguinte maneira: na seção (2.1) apresenta-se a linguagem de programação utilizada para a implementação da PG, o LISP; na seção (2.2) descreve-se de forma detalhada os operadores genéticos utilizados pela PG; e finalmente, na seção (2.3) descreve-se de forma ordenada o algoritmo com o paradigma da PG.

2.1 Estudo da Linguagem LISP

Embora haja exemplos de implementação da PG na linguagem *C*, a linguagem de programação mais natural para esta técnica é o LISP, e será o caso neste trabalho. A preferência por esta linguagem deve-se ao fato de que no LISP, dados e programas possuem a mesma estrutura, a de uma árvore gramatical. Nossos programas serão, então, *expressões simbólicas* (ES) do LISP, e desta forma, vamos nos referir aos

indivíduos da população como programas ou ES, indistintamente.

Uma ES é dita *fiel* (ESF) se quando avaliada retorna um resultado e não uma mensagem de erro. As ESF's são escritas no LISP como um conjunto ordenado de itens dentro de um par de parênteses. Esses itens podem ser *átomos* ou outras ESF, onde átomo é qualquer operador funcional, uma variável ou um número. Por exemplo, as expressões matemáticas podem ser escritas no LISP, como ESF's que obedecem a forma :

$$\left(< operador > \quad < operando \cdot 1 > \quad < operando \cdot 2 > \quad \dots \quad < operando \cdot n > \right).$$

Um $< operando >$ pode ser uma número, uma variável ou uma outra expressão entre parênteses, i.e. uma outra ESF. Um $< operador >$, é um operador funcional que atua sobre n $<operando>$'s. Operadores que atuam sobre 2 operandos são: a soma, a subtração, a divisão, a multiplicação, etc.; e operadores que atuam sobre 1 operando são: a função seno, a função cosseno, a função logaritmo, dentre outros. Por exemplo, as expressões $(+ \ x \ 2)$ e $(* \ 3 \ (\ln \ x))$ são ESF's válidas para representar as expressões algébricas $x + 2$ e $3 \ln x$, respectivamente. A mais simples ESF é uma variável, por exemplo x . A seguinte ESF mais complexa é um operador seguido do número adequado de variáveis, i.e. $(+ \ x \ y)$ ou $(\text{abs } x)$. Exemplos de *expressões simbólicas infíeis*, i.e. programas que quando avaliadas retornam uma mensagem de erro, são: $(x \ x)$, $(x \ x \ /)$, $(/ \ * \ -)$, $(/ \ x)$, $(x \ * \ x)$, etc..

O primeiro passo para atacar um problema usando a PG é a definição dos conjuntos $\mathcal{V}$ e $\mathcal{F}$, de variáveis e operadores, respectivamente, que formarão as ESF's. A escolha dos componentes destes conjuntos é feita de maneira arbitrária e depende da natureza do problema a ser atacado. Por exemplo, se a solução de um dado problema pode ser representada por um polinômio, uma sensata escolha para estes conjuntos seria: $\mathcal{F} = \{+, -, /, *\}$ e $\mathcal{V} = \{x\}$. Assim, $(+ \ (* \ x \ x) \ x)$ é um exemplo de ESF construída a partir dos elementos dos conjuntos citados, sendo uma forma não única de representar a expressão algébrica $x + x^2$. Na figura (2.1) temos a representação desta expressão algébrica em forma de ESF e de árvore gramatical. Note que cada átomo da ESF é representado por uma “folha” da árvore gramatical.

Figura 2.1: Expressão algébrica $x + x^2$ representada na forma de árvore gramatical e de uma ESF.

Podemos, então, representar por uma ESF cada indivíduo do espaço de programas possíveis para a resolução de um dado problema. A versatilidade desta representação é evidenciada quando se utiliza os operadores genéticos, os quais serão tratados com maiores detalhes na próxima seção.

2.2 Os Operadores Genéticos

Os operadores genéticos são mecanismos utilizados pela PG para modificar a estrutura dos indivíduos constituintes da população, de modo a formar novas estruturas. Essas modificações são realizadas com o intuito de “procurar”, num espaço de programas possíveis, por ESF’s mais eficazes na solução de um dado problema. Embora existam diversos tipos de operadores genéticos, que podem ser vistos em [2], em nosso trabalho utilizaremos apenas a *cópia simples*, a *mutação* e o *crossover*. Esses operadores genéticos são definidos da seguinte forma:

- **Cópia simples:** utilizada para introduzir os melhores indivíduos na próxima geração sem qualquer alteração em suas estruturas. Esse operador garante a preservação de estruturas bem sucedidas. Tal processo está descrito na figura (2.5).
- **Mutação:** realiza mudanças aleatórias na estrutura dos indivíduos. Para acontecer a mutação, um indivíduo da população é sorteado por algum método de seleção baseado no fitness. Um átomo do indivíduo é escolhido aleatoriamente e substituído por um outro átomo do mesmo tipo. Por exemplo, um número pode ser substituído por outro número; um operador que age sobre um

determinado número de operandos deve ser substituído apenas por um outro operador que age sobre a mesma quantidade de operandos, e assim por diante. Na figura (2.2) e (2.5) temos um exemplo de mutação. A estrutura modificada é então introduzida na nova geração. De modo a acelerar a dinâmica, diferentes taxas de mutações podem ser usadas para diferentes tipos de átomos.

Figura 2.2: Indivíduo, em sua representação de árvore gramatical e de uma expressão algébrica, antes e depois de uma mutação. Um átomo é aleatoriamente selecionado (representado pelo átomo hachurado) e é substituído por um outro átomo de mesmo tipo, escolhido aleatoriamente. Neste exemplo, um operador de multiplicação (*) é substituído por um operador de subtração (-).

- **Crossover:** cria novas estruturas através de partes das estruturas de indivíduos pais. A operação de crossover inicia-se com duas ESF's pais e produz duas novas ESF's filhos. Para realizar o crossover, dois indivíduos pais são selecionados, como na operação de mutação, por um dado critério baseado no fitness. Usando uma distribuição de probabilidade uniforme, é sorteado um ponto em cada pai, sendo, estes, os pontos de cruzamento entre eles. Das árvores gramaticais que representam os indivíduos, os “ramos” provenientes de cada ponto sorteado são intercambiados, originando duas novas ESF's descendentes, como descrito nas figuras (2.3), (2.4) e (2.5). As ESF's descendentes são então introduzidas na nova geração. Apesar das ESF's descendentes partilharem a informação genética das ESF's pais, as suas estruturas são completamente diferente das existentes anteriormente.

Figura 2.3: Dois indivíduos pais selecionados da população para se realizar o crossover. Os pontos assinalados são escolhidos aleatoriamente; logo depois, os “ramos” provenientes de cada um dos pontos são intercambiados, originando duas novas ESF’s descendentes.

Figura 2.4: Indivíduos descendentes dos indivíduos da figura (2.3), gerados por crossover.

Figura 2.5: As três operações de PG utilizadas para a formação de uma geração $g + 1$ através de indivíduos da geração g .

Portanto, estes três operadores genéticos, que são responsáveis pelas diferenças dos indivíduos de uma geração e a seguinte, junto ao processo de seleção, serão as

ferramentas da PG, na busca por um programa que melhor solucione o problema atacado.

2.3 Algoritmo de Implementação

O algoritmo implementado com o paradigma da PG será descrito agora. Em ordem de operação podemos listá-lo da seguinte forma:

1. Aleatoriamente, cria-se uma população inicial de N_{pg} indivíduos;
2. Iterativamente, realiza-se os seguintes sub-passos, até que o critério de terminação seja alcançado:
 - (a) Calcula-se o fitness de cada indivíduo da população;
 - (b) Realiza-se uma cópia simples dos M ($M < N_{pg}$) indivíduos possuidores dos melhores fitness. Estes indivíduos são introduzidos na nova geração.
 - (c) Para se construir os restantes ($M - N$) indivíduos da próxima geração, utiliza-se os operadores genéticos, que são escolhidos mediante as seguintes probabilidades:

P_r : Probabilidade de acontecer uma cópia simples para se formar um novo indivíduo. Neste caso, é sorteado *um* indivíduo da população por algum critério baseado no fitness. Realiza-se, então, uma cópia simples deste indivíduo, que é introduzido na nova geração;

P_m : Probabilidade de acontecer uma mutação para se formar um novo indivíduo. Neste caso, é sorteado *um* indivíduo da população de forma análoga ao anterior. Realiza-se, então, uma mutação neste indivíduo e introduz-se o indivíduo gerado na nova geração;

P_{cros} : Probabilidade de acontecer um crossover para se formar dois novos indivíduos. Neste caso, sorteia-se *dois* indivíduos da população, por algum critério baseado no fitness. Realiza-se, então, o crossover nestes indivíduos e seus descendentes são introduzidos na nova geração;
3. O indivíduo com melhor fitness é designado como o resultado final. Este resultado representa a solução (ou uma aproximação da solução) para o problema.

3

O JOGO DA MINORIA

“A verdade sempre repousa com a minoria e a minoria é sempre mais forte que a maioria porque a minoria geralmente é formada por quem realmente possui uma opinião, enquanto que a força da maioria é ilusória, formada por grupos que não possuem opinião - e que no próximo instante (quando for evidente que a minoria é mais forte) mudarão de opinião ... enquanto a verdade, novamente reverte a uma nova minoria.”

Soren Kierkegaard

O Jogo da Minoria (JM), amplamente estudado pela comunidade científica nos últimos anos ([3] a [14]), é um jogo iterado em que um número fixo de jogadores deve escolher uma entre duas alternativas (A ou B), a cada rodada. Todos os jogadores que estiverem na *minority* vencem. Esta regra da minoria, que torna o jogo um sistema de agentes interagentes, mostra fascinantes propriedades e coloca novos desafios à Mecânica Estatística, com seus métodos e idéias. De fato, grande parte da pesquisa relacionada ao JM foi realizada por físicos, embora tenha se originado no contexto da economia no chamado *Problema do Bar El Farol* [3].

El Farol, um bar da cidade de Santa Fe que oferece música irlandesa nas noites de quinta-feira, possui espaço limitado e as noites só serão agradáveis aos freqüentadores se este não estiver tão cheio. O problema do bar *El Farol* é proposto da seguinte maneira: não havendo forma de prever o número de pessoas que estarão presentes, cada freqüentador deve decidir ir ao bar, caso acredite que o mesmo não estará lotado ou, caso contrário, ficará em casa. Estas escolhas devem ser feitas sem comunicação *a priori* entre as pessoas envolvidas e a única informação disponível é o número de freqüentadores presentes nas semanas anteriores. Se um grupo pequeno de pessoas

resolverem ir ao bar, elas terão diversão garantida, pois não havendo superlotação o espaço estará agradável. Nesta situação, o grande grupo de pessoas que preferiram ficar em casa saíram perdendo, pois estão ausentes em uma grande noite no bar *El Farol*. Por outro lado, se um grande grupo de pessoas decidir ir ao bar, superlotando-o, estas terão uma noite bastante desconfortável. Enquanto isso, o pequeno grupo que decidiu ficar em casa usufruirá do bom conforto de seus lares!

Acredita-se que um jogo com os mecanismos do JM engloba características de sistemas onde agentes competem por recursos limitados, como o mercado financeiro. Isso explica o grande interesse neste jogo pela comunidade científica. Podemos exemplificar a aplicação do JM no mercado financeiro observando a situação que expomos a seguir. Muitos compradores e poucos vendedores de um certo produto implica numa alta dos preços. Nestas circunstâncias, o grupo da minoria (vendedores) é o favorecido. Por outro lado, muitos vendedores para poucos compradores deste mesmo produto, causa uma baixa nos preços. Mais uma vez, o grupo que se encontra na minoria (agora os compradores) é o beneficiado.

Existem vários outros contextos, além do mercado financeiro, de aplicações do JM. Por exemplo, um contexto biológico, em que os indivíduos de uma população devem escolher explorar uma dentre duas regiões (região A e região B) de iguais e limitadas quantidades de recursos. O grupo de indivíduos que optar pela decisão da minoria usufruirá de uma maior abundância de recursos. Além desse contexto, temos também: o tráfico de veículos que esperam transitar por uma estrada congestionada [10]; o tráfico de dados na rede, com cada dado fluindo mais rapidamente através de linhas menos carregadas, etc. As duas alternativas de escolhas (*A* e *B*) podem representar, então, diversos tipos de situações ou cenários, como “ir ao bar *El Farol*” ou “ficar em casa”, “comprar” ou “vender”, “migrar” ou “não migrar”, “sim” ou “não”, etc..

O JM pode ser exemplificado, também, através de uma brincadeira de criança chamada “dois-ou-um”: se um grupo de três crianças quer eleger um líder para seus jogos, elas dizem juntas “dois-ou-um” e no final do “um” cada uma delas mostra um ou dois dedos. A criança que estiver na minoria é a vencedora.

Este capítulo está organizado da seguinte forma: na seção (3.1) será definido matematicamente o JM. Será introduzido, nessa seção, algumas quantidades que serão utilizadas em todo decorrer da dissertação; finalmente, na seção (3.2), apresenta-se uma pequena revisão dos resultados comuns na literatura a respeito do Jogo da Minoria Padrão.

3.1 Definição Matemática do Jogo da Minoria

Vamos então introduzir, neste ítem, uma definição matemática para o JM, de forma a equacionarmos a opção da minoria, bem como as pontuações dos jogadores.

O JM considera uma população de N jogadores com interação indireta, isto é, os jogadores não podem identificar a ação individual de outros jogadores, mas reagem a um resultado global. A cada rodada t , $t = 1, \dots, P$, onde P é o número máximo de rodadas, o i -ésimo jogador deve tomar uma decisão entre comprar ou vender, migrar para a região A ou para B , baseada em uma informação global (história do mercado em tempos anteriores, história do número de freqüentadores, circunstâncias políticas, condições de tempo, etc.). Vamos denominar um período completo de P rodadas, uma *competição* do JM.

Representando as duas opções de escolhas por ± 1 , podemos escrever o sinal da minoria, numa particular rodada t , com a seguinte equação:

$$\sigma_m(t) = -\text{sign}\left[\sum_{i=1}^N \sigma_i(t)\right], \quad (3.1)$$

onde sign representa a *função sinal*, definida como:

$$\text{sign}(x) = \begin{cases} +1 & \text{se } x > 0 \\ -1 & \text{se } x \leq 0 \end{cases} \quad (3.2)$$

e $\sigma_i(t) \in \{\pm 1\}$ é a resposta do i -ésimo jogador mediante a informação global. Esta informação global será definida como a seqüência dos últimos d resultados da minoria, armazenados no vetor $\mathbf{S}(t) \in \mathbb{R}^d$, que denominaremos por *vetor-história*. A partir de um vetor inicial aleatório, $\mathbf{S}(t=0) = \{\pm 1\}^d$, são construídos P vetores-história em uma competição, obedecendo a regra :

$$\mathbf{S}(t=0) = (S_\nu, S_{\nu+1}, \dots, S_{\nu+d-1}), \quad (3.3)$$

$$\mathbf{S}(t=1) = (S_{\nu+1}, S_{\nu+2}, \dots, S_{\nu+d-1}, \sigma_m(0)), \quad (3.4)$$

$$\mathbf{S}(t=2) = (S_{\nu+2}, S_{\nu+3}, \dots, \sigma_m(0), \sigma_m(1)), \quad (3.5)$$

e numa rodada $t \geq d$, teremos a configuração:

$$\mathbf{S}(t) = \left(\sigma_m(t-d), \dots, \sigma_m(t-2), \sigma_m(t-1) \right). \quad (3.6)$$

Vamos então atribuir uma pontuação a cada um dos jogadores, de acordo com suas respostas. Consideraremos que todos os jogadores pertencentes à minoria sejam contemplados com um ponto e todos os jogadores pertencentes à maioria sejam penalizados com um ponto. O ganho $\phi_i(t)$ representará a pontuação adquirida ou perdida pelo jogador i em uma dada rodada, sendo expresso pela relação

$$\phi_i(t) = \sigma_i(t) \sigma_m(t). \quad (3.7)$$

Através do ganho deste jogador, podemos introduzir o seu fitness $\Phi_i(t)$ durante uma competição. O fitness deste jogador é dado pelo seu ganho acumulado em t rodadas, expresso pela relação

$$\Phi_i(t) = \sum_{t'=1}^t \phi_i(t'). \quad (3.8)$$

Outra quantidade conveniente é o número total de erros cometido pelo jogador i durante a competição, i.e. o número de vezes que este jogador não escolheu a opção da minoria, até uma rodada t . Vamos denominar esta quantidade por *fitness padronizado*¹ e a representaremos por $E_i(t)$. A conveniência em introduzir o fitness padronizado deve-se à analogia aos mecanismos da Mecânica Estatística de minimização da *energia livre*, uma vez que cada jogador deve utilizar estratégias de modo a minimizar o seu erro de previsão. O fitness padronizado se relaciona com o fitness pela relação

$$E_i(t) = \frac{1}{2} \left(t - \Phi_i(t) \right). \quad (3.9)$$

Ao final de uma competição, este jogador portará, então, o fitness

$$\Phi_i \equiv \Phi_i(t = P), \quad (3.10)$$

e o fitness padronizado

$$E_i \equiv E_i(t = P). \quad (3.11)$$

¹Como veremos na seção (6.2), na PG defini-se por fitness padronizado a variável que representa o grau de adaptabilidade de um indivíduo, de forma idêntica ao fitness; porém, de forma que seu menor valor esteja associado ao melhor indivíduo.

Agora que equacionamos as variáveis utilizadas no JM, podemos introduzir uma dinâmica a seus jogadores, através de um modelo que ficou conhecido como *Jogo da Minoria Padrão* e que será discutido brevemente no próximo item.

3.2 Jogo da Minoria Padrão

Em uma série de publicações de Challet e Zhang ([11] e [12]) e Challet, Marsili e Zecchina ([13] e [14]), o seguinte modelo, conhecido como *Jogo da Minoria Padrão* (JMP), foi introduzido e estudado numérica e analiticamente: cada jogador é equipado com $s \geq 2$ tabelas de decisões, com cada uma prescrevendo uma resposta para cada configuração possível do vetor-história. Cada tabela portará 2^d respostas, com cada uma destas respostas geradas aleatoriamente no início de uma competição.

Um exemplo de uma tabela de decisões em um ambiente com uma memória (ou história) $d = 3$ é apresentada na tabela (3.1). Para estas circunstâncias teremos $2^3 = 8$ configurações possíveis para o vetor-história.

vetor-história	Resposta
+1 + 1 + 1	-1
+1 + 1 - 1	-1
+1 - 1 + 1	+1
+1 - 1 - 1	-1
-1 + 1 + 1	+1
-1 + 1 - 1	+1
-1 - 1 + 1	+1
-1 - 1 - 1	-1

Tabela 3.1: Um exemplo de tabela de decisões para $d = 3$.

Cada jogador, em uma dada rodada, utiliza apenas uma de suas tabelas de decisão, porém pontualiza “virtualmente” cada uma delas através das relações (3.7)-(3.8). A tabela com maior fitness em t será utilizada para a prescrição de uma resposta em $t + 1$. O mecanismo de funcionamento do JMP é esquematizado na figura (3.1).

O processo de atualização da tabela de decisões atuante no JMP faz deste modelo um sistema adaptativo, onde os jogadores buscam uma melhor estratégia para prever o resultado da minoria. No entanto, a dinâmica das respostas dos jogadores pode

Figura 3.1: Desenho esquemático do JMP numa dada rodada. Neste caso, os agentes tomam suas decisões através dos últimos três resultados da minoria ($d = 3$). A linha mais espessa da figura mostra como a informação flui no sistema: os N agentes consideram o vetor-história $(-1 + 1 + 1)$ e emitem uma resposta. Note que um agente, utilizando a tabela de decisões da tabela (3.1), emitiria a resposta $+1$. As N respostas são transformadas num novo resultado da minoria (neste caso: -1) através da regra da minoria. Esta informação é então apresentada aos jogadores para a atualização dos fitness de suas tabelas de decisões. Finalmente, este resultado da minoria torna-se o próximo número na sequência armazenada pelo vetor-história.

promover o surgimento de dois casos extremos em uma dada rodada. O primeiro se dá quando somente um jogador está do lado da minoria. O segundo, acontece quando $\frac{N-1}{2}$ jogadores estão na minoria. De um ponto de vista social, o segundo extremo é preferível, pois o aproveitamento global dos recursos, em uma dada rodada, é máximo; enquanto que para o primeiro temos um péssimo aproveitamento global, pois a quantidade de jogadores no lado da minoria é mínimo. Em geral, espera-se que a população de jogadores comporte-se entre estes dois extremos, sendo mais eficiente quanto menores as *flutuações das escolhas* em torno do valor médio. Por flutuação das escolhas entendemos como a diferença entre o número de jogadores que optaram por uma alternativa e os que optaram pela outra.

A eficiência global pode ser, então, caracterizada pela variância

$$\sigma^2(t) = \left\langle \left(\sum_i \sigma_i(t) \right)^2 \right\rangle, \quad (3.12)$$

onde a média é sobre os exemplos possíveis. Pequenos valores da variância correspondem a um estado eficiente, isto é, o aproveitamento dos recursos, neste caso, é grande.

Respostas aleatórias de todos os jogadores, proporcionam $\sigma^2 = N$. Desta forma, um valor $\sigma^2/N < 1$ indica uma boa coordenação entre os jogadores. Por outro lado, para $\sigma^2/N > 1$ temos um mau aproveitamento dos recursos globais, causado por uma espécie de “efeito manada” : um grande número de jogadores adapta-se às tabelas de decisões bem sucedidas, provocando uma inversão no resultado esperado da minoria e aumentando a flutuação.

Diversas análises e simulações presentes na literatura ([4], [5] e [9]) mostram que a variância depende unicamente da função $z \equiv 2^d/N$, dada pela fração entre o número de configurações possíveis do vetor-história e o número de jogadores. Tais simulações, além de demonstrar que resultados qualitativos independem de s , apresentam também que :

- para $z > z_c \approx 0.33740$, existe um bom aproveitamento dos recursos globais ($\sigma^2/N < 1$), provocado por uma boa coordenação entre os jogadores;
- para $z = z_c$, σ^2/N atinge um valor mínimo e menor que 1, indicando ser esta a configuração que melhor aproveita os recursos globais;
- para $z < z_c$ temos $\sigma^2/N > 1$, mostrando que nesta configuração o sistema apresenta um aproveitamento ruim dos recursos globais.

O JMP foi resolvido analiticamente através do tratamento da escolha entre as duas tabelas de decisões como uma variável de spin, e aplicando o modelo de réplicas para encontrar o estado fundamental do sistema. Estes resultados podem ser encontrados em [11] [12].

4

O PERCEPTRON NO JOGO DA MINORIA

“Um dos grandes desafios intelectuais das próximas décadas é a questão da organização do cérebro. Qual mecanismo básico para o armazenamento da memória? Quais são os processos que servem como interface entre os processos químicos básicos e as específicas operações não estatísticas do cérebro? Sobretudo, como a formação conceitual é arquivada no cérebro humano? Eu pergunto a mim mesmo se o espírito dos físicos que irão estar envolvidos nestes estudos não será parecido com o que moveu os fundadores do fundamento racional da termodinâmica.”

C. N. Yang - (1972)

No capítulo anterior, introduzimos o JM e a forma como os jogadores fazem suas escolhas no chamado Jogo da Minoria Padrão. Na ocasião, vimos que as escolhas eram realizadas a partir de tabelas que prescreviam uma resposta a cada configuração possível do ambiente. Vimos também que o processo de aprendizado estava relacionado com a escolha da mais bem pontuada de suas tabelas, para a utilização em uma próxima rodada. Nos capítulos posteriores analisaremos um cenário alternativo ao descrito pelo JMP, onde os jogadores utilizam *perceptrons* para suas decisões, alterando suas estratégias de acordo com a decisão da minoria. Esta versão do JM foi inicialmente introduzida e estudada em [6]. É necessário, então, uma discussão prévia do *Perceptron Simples* e de seus mecanismos de aprendizagem, o que faremos neste capítulo.

Em 1962, Frank Rosenblatt e outros [15] publicaram um tipo específico de rede neural, chamado por eles de *Perceptron*. Este nome foi escolhido devido ao fato de o considerarem um modelo simplificado do mecanismo biológico do processamento de informações sensoriais (*percepção*). Este trabalho veio a trazer uma nova luz para

os chamados estudos de *Redes Neurais Artificiais*, promovidos por uma publicação de Warren McCulloch e Walter Pitts [16] em 1943, que propunha uma teoria geral do processamento de informação baseada em redes de elementos de decisão binária, análogos aos neurônios biológicos (em sua forma bastante simplificada) [17]. Nesse contexto, o aprendizado por uma rede de neurônios consiste em encontrar configurações de sinapses que gerem saídas desejadas. McCulloch e Pitts mostraram que tais redes poderiam realizar quaisquer tarefas imagináveis. Porém, a rede neural proposta ainda possuía um problema não resolvido: como escolher as sinapses da rede de forma que tarefas específicas sejam realizadas por uma máquina?

Em 1961, Caianiello [18] construiu uma teoria estatística, usando idéias da Mecânica Estatística, e incorporou o aprendizado baseado nas propostas feitas por Hebb [19] sobre o processamento de informação no cérebro¹; surgindo no ano seguinte, o perceptron, apresentado pelo grupo de Rosenblatt, o qual demonstrou que este simples e rico modelo podia generalizar tarefas!

Em nosso trabalho vamos considerar que cada jogador do JM, sabendo desta propriedade do perceptron, a de realizar uma dada tarefa, utiliza-o para realizar suas escolhas. A questão com que estes jogadores devem se deparar será: *como escolher os valores para os pesos sinápticos para obter uma resposta desejada e assim vencer o JM?* Tentaremos, em nosso trabalho, resolver esta questão para um cenário específico, analisando a performance de um conjunto de algoritmos de aprendizado.

O presente capítulo está organizado da seguinte forma: no item (4.1) faremos uma breve apresentação do perceptron e de sua arquitetura, bem como a forma de realização de sua resposta; no item (4.2) introduziremos os mecanismos de aprendizagem de um perceptron e os algoritmos de aprendizado que serão utilizados neste trabalho.

4.1 O Perceptron Simples

Numa rede neural biológica, um típico neurônio coleta sinais vindos de outros

¹Donald O. Hebb, em 1949, sugeriu que o aprendizado se dá através de modificações nas sinapses em consequência das atividades dos neurônios pré e pós-sinápticos sob a influência de estímulos externos. Segundo Hebb: “*Quando um axônio da célula A está próximo o suficiente para excitar uma célula B e repetidamente ou persistentemente influi no seu disparo, algum processo de crescimento ou mudança metabólica ocorre em uma ou ambas as células tal que a influência de A no disparo de B seja aumentada*”.

neurônios, por meio de árvores de estruturas finas chamadas *dentritos* ou até mesmo pelo próprio corpo celular. Tal neurônio envia impulsos elétricos por um longo tubo conhecido como *axônio*, que se divide em centenas de ramificações. No fim de cada ramificação, uma estrutura chamada *sinapse* converte os impulsos vindos do axônio em atividades de inibição ou excitação nos neurônios conectados. Quando um neurônio recebe entradas excitantes que sejam suficientemente grandes, comparadas com as inibitórias, e com um certo limiar de disparo, ele envia um impulso de resposta por seu axônio [17].

Semelhante a uma rede neural biológica, o perceptron simples consta de um corpo central, dentritos, axônio e as sinapses. Um conjunto $\{S^k\}$ de impulsos externos chegam ao perceptron pelos dentritos, com cada impulso k ponderado por uma relevância variável J^k , num processo análogo à sinapse biológica. O perceptron simples converte este padrão de impulsos em um único impulso de saída, chamado de resposta do perceptron. A arquitetura do perceptron simples, com um número de entradas (dentritos) d , é apresentada na figura (4.1).

Figura 4.1: Arquitetura do perceptron simples.

Representando os impulsos de entrada pelo vetor $\mathbf{S} \in \mathbb{R}^d$ e os pesos sinápticos pelo vetor $\mathbf{J}$, podemos introduzir o campo local h no perceptron, definido por

$$h \equiv \frac{\mathbf{J} \cdot \mathbf{S}}{\|\mathbf{J}\|}, \quad (4.1)$$

onde $\mathbf{J} \cdot \mathbf{S} = \sum_{k=1}^d J^k S^k$ é o produto escalar usual. O vetor $\mathbf{J}$ é também chamado de *vetor sináptico*, e o vetor $\mathbf{S}$ de *vetor de entrada*².

²Veremos na próxima seção, quando introduzirmos o perceptron no JM, que este vetor de entrada será justamente o vetor-história, justificando a mesma representação, $\mathbf{S}$.

Finalmente, para um certo campo local, a resposta do perceptron é dada por

$$\sigma = \mathcal{F}(h) \quad (4.2)$$

onde $\mathcal{F}$ é a chamada *função de transferência*.

Conforme mencionado anteriormente, consideraremos que cada jogador do JM utiliza de um perceptron simples para realizar suas escolhas. Porém, como definimos o JM com duas opções: $+1$ ou -1 , faz-se necessário que os jogadores utilizem perceptrons portadores de uma função de transferência dada por $\mathcal{F}(x) = \text{sign}(x)$. Esta função de transferência define o chamado *perceptron booleano*³. Através desta função de transferência, a resposta do perceptron i , em uma dada rodada t , é dada por

$$\sigma_i(t) = \text{sign}\left(h_i(t)\right) = \text{sign}\left(\frac{\mathbf{J}_i(t) \cdot \mathbf{S}(t)}{\|\mathbf{J}_i\|}\right). \quad (4.3)$$

A resposta de um perceptron depende, então, tanto da função de transferência como também da distribuição dos pesos de suas sinápses. Em analogia aos neurônios biológicos, podemos interpretar σ como a resposta de um neurônio (o perceptron) mediante os estímulos $\{S^k\}$, provenientes do “mundo externo”, ponderados pelos pesos sinápticos $\{J^k\}$.

4.2 Aprendizagem de um Perceptron

Definida a estrutura do perceptron e a forma com que este realiza uma resposta, podemos agora introduzir os mecanismos de aprendizagem do perceptron para sua utilização no JM. Por *aprendizagem* entendemos a modificação, introduzida por dados externos, de parâmetros internos, de modo que o perceptron tenha uma maior probabilidade de reproduzir a resposta desejada. Então, ao final de cada rodada t do JM, é apresentado ao jogador i o par de informação externa

$$\left(\sigma_m(t), \mathbf{S}(t)\right),$$

³O perceptron booleano pode ser visto como um spin de Ising à temperatura zero e campo externo H nulo, em que o campo local é devido a d spins $\{S^k\}$ fixos com constantes de troca $\{J^k\}$. Na literatura de redes neurais, este elemento também é conhecido como neurônio de McCulloch-Pitts.

juntamente com sua informação interna

$$\left(\sigma_i(t), h_i(t), \mathbf{J}_i(t) \right).$$

Neste momento, esse jogador induz uma modificação nos pesos sinápticos do seu perceptron correspondente, de modo a tentar reproduzir a resposta da minoria na rodada seguinte, i.e. realizar a opção desejada $\sigma_i(t+1) = \sigma_m(t+1)$. Um desenho esquemático deste modelo para o JM, bem como o fluxo da informação externa, é apresentado na figura (4.2).

Figura 4.2: Desenho esquemático do JM entre perceptrons, numa dada rodada t e com uma memória $d = 3$. A linha mais espessa da figura mostra como a informação externa flui no sistema: os N perceptrons consideram o vetor-história $(-1 + 1 + 1)$ e emitem uma resposta. As N respostas são transformadas num novo resultado da minoria (neste caso: -1). Esta informação, juntamente com o vetor-história, são então apresentados aos jogadores para a atualização dos fitness e dos vetores sinápticos, sendo estes últimos atualizados através de algum algoritmo de aprendizado. Finalmente, este resultado da minoria torna-se o próximo número na sequência armazenada pelo vetor-história.

Os algoritmos que realizam esta aprendizagem, chamados de *algoritmos de aprendizado*, possuem diversas variações. Porém, a maioria reduz-se ao mesmo princípio

iterativo: a atualização do vetor sináptico através de uma combinação linear do vetor sináptico prévio $\mathbf{J}_i(t)$ e o vetor $\mathbf{S}(t)$. O coeficiente da combinação linear pode depender das informações externas e internas. Esta atualização, realizada mediante uma informação imediatamente anterior, é chamada de aprendizado *on-line*[20].

Considera-se, neste trabalho, quatro algoritmos de aprendizado on-line bem conhecidos:

- O *Aprendizado Hebbiano*, definido pela atualização:

$$\mathbf{J}_i(t+1) = \mathbf{J}_i(t) + \frac{1}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t). \quad (4.4)$$

Um exemplo bastante prático e eficiente de aprendizado de um perceptron através deste algoritmo pode ser conferido no apêndice (B). Neste apêndice tratamos o conhecido problema de redes neurais denominado por *cenário professor-estudante com ruído multiplicativo*;

- O *Aprendizado do Perceptron*, que realiza uma atualização hebbiana no vetor sináptico apenas se a resposta emitida for diferente da desejada. Ou seja,

$$\mathbf{J}_i(t+1) = \begin{cases} \mathbf{J}_i(t) + \frac{1}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t) & \text{se } \sigma_i(t) \neq \sigma_m(t) \\ \mathbf{J}_i(t) & \text{se } \sigma_i(t) = \sigma_m(t) \end{cases}. \quad (4.5)$$

Este algoritmo pode ser então condensado na forma :

$$\mathbf{J}_i(t+1) = \mathbf{J}_i(t) + \frac{(1 - \sigma_i(t)\sigma_m(t))}{2} \frac{1}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t); \quad (4.6)$$

- O *Aprendizado com Normalização*, que realiza uma atualização hebbiana e depois normaliza o vetor sináptico resultante, impondo $\mathbf{J}_i(t+1) \cdot \mathbf{J}_i(t+1) = 1$. Este algoritmo pode ser expresso pela forma :

$$\mathbf{J}_i(t+1) = \frac{\mathbf{J}_i(t) + \frac{1}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t)}{\|\mathbf{J}_i(t) + \frac{1}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t)\|}; \quad (4.7)$$

- O *Aprendizado com a Programação Genética*, que utiliza a PG para modular a atualização hebbiana. A atualização pode ser expressa na forma:

$$\mathbf{J}_i(t+1) = \mathbf{J}_i(t) + \frac{\eta_i(t)}{\sqrt{d}}\sigma_m(t)\mathbf{S}(t), \quad (4.8)$$

onde $\eta_i(t)$ é uma função que fornece o tamanho da correção de aprendizagem. Esta quantidade, que depende de variáveis externas e interna, é chamada de *função de modulação* (FMOD) e será o objeto da PG em nosso trabalho. Para o caso $\eta_i(t) = 1$, recupera-se o aprendizado hebbiano em (4.4).

Nos próximos capítulo, analisaremos o desempenho destes algoritmos na previsão da resposta da minoria diante de um modelo para o JM.

5

JOGO DA MINORIA COM DOIS GRUPOS DE JOGADORES

“Se os tubarões fossem homens, eles seriam mais gentis com os peixes pequenos. Se os tubarões fossem homens, eles fariam construir resistentes caixas do mar, para os peixes pequenos, com todos os tipos de alimentos dentro, tanto vegetais, quanto animais. (...) Para que os peixinhos não ficassem tristonhos, eles dariam cá e lá uma festa aquática, pois os peixes alegres tem gosto melhor que os tristonhos. Naturalmente também haveria escolas nas grandes caixas, nessas aulas os peixinhos aprenderiam como nadar para a goela dos tubarões. (...) Eles seriam ensinados de que o ato mais grandioso e mais belo é o sacrifício alegre de um peixinho, e que todos eles deveriam acreditar nos tubarões, sobretudo quando esses dizem que velam pelo belo futuro dos peixinhos. (...) Também haveria uma religião ali. Se os tubarões fossem homens, eles ensinariam essa religião. E só na barriga dos tubarões é que começaria verdadeiramente a vida. (...) Curto e grosso, só então haveria civilização no mar, se os tubarões fossem homens.”

Bertold Brecht

Embora o texto de Brecht nos inspire um sentimento de indignação, por sugerir a opressão de uma classe de seres ignorantes por outra, de seres mais inteligentes, nele estão presentes os ingredientes básicos da natureza e das relações sociais. Seja a natureza, com os processos de seleção natural, sejam as relações sociais, como as interações num mercado financeiro, em ambos os casos encontram-se indivíduos que buscam, primordialmente, seu benefício próprio. Enquanto um indivíduo biológico “tenta” deixar um maior número de descendentes, um agente financeiro busca a obtenção de lucros. Além disto, em ambos os contextos, oportunidades de ganho e de propagação de descendentes não são desperdiçadas. Desta forma é necessário

interpretarmos que estes indivíduos são inteligentes, i.e. dispõem de algum tipo de estratégia em busca de suas intenções egoístas.

No JM, onde os jogadores também visam o lucro próprio, a comunidade científica tem encontrado características de coordenação e cooperação entre os jogadores, como mostrado na seção (3.2). Porém, não pretendemos encontrar mais um cenário de melhores proveitos dos recursos globais. Pretendemos, sim, encontrar condições necessárias para que uma parcela da população (os tubarões) tirem proveito dos demais (os peixinhos) para uma ascensão no JM. Para isso, vamos analisar o JM disputado, simultaneamente, por jogadores pertencentes a dois grupos distintos. O primeiro grupo será constituído por jogadores que utilizam estratégias fixas para realizar suas decisões. Já o segundo será constituído por jogadores munidos de estratégias que se alteram de acordo com algum algoritmo de aprendizado. Devido a esta presença de aprendizado nas estratégias dos jogadores deste grupo, vamos chamá-los de *Jogadores Inteligentes* (JINT's).

O que se intenta responder é: haverá alguma condição para que os JINT's consigam tirar proveito dos jogadores não adaptativos para vencer o JM? E se há, existirão algoritmos melhores que outros, para esta prática? Verificaremos que as respostas a estas perguntas são afirmativas. Os JINT's conseguem tirar proveito dos demais se a proporção entre o número de jogadores de cada grupo for menor que um certo valor. Verificaremos também, por simulações computacionais, que o desempenho dos JINT's varia de acordo com o algoritmo de aprendizado utilizado.

Os JINT's utilizarão perceptrons com vetores sinápticos atualizados, a cada rodada, por um dos algoritmos de aprendizado apresentados na seção (4.2), de modo a aprender a regra da minoria. Já os demais jogadores utilizarão perceptrons com vetores sinápticos fixos, ao longo das rodadas, não adquirindo aprendizado. Este carácter fixo dos vetores sinápticos representa bem uma estratégia também fixa.

Este capítulo está organizado da seguinte maneira. Na seção (5.1) introduziremos a estrutura dos perceptrons destes dois grupos de jogadores, bem como uma discussão geral a respeito da dinâmica proporcionada por este cenário. Na seção (5.3) descreveremos nossas simulações para o caso de todos os JINT's utilizando o algoritmo de aprendizado hebbiano. Porém, a análise comparativa entre os desempenhos de diferentes algoritmos de aprendizado, foi guardada para o próximo capítulo.

5.1 Definição das Estruturas dos Jogadores

Vamos então definir a estrutura dos perceptrons destes dois grupos de jogadores. No grupo dos JINT's, o i -ésimo jogador é equipado com um perceptron de vetor sináptico $\mathbf{J}_{ei}(t) \in \mathbb{R}^d$, que será a sua estratégia de escolha em uma rodada t . De agora em diante, vamos adotar o índice (e) para representar quantidades associadas a jogadores pertencentes a este grupo.

O vetor sináptico do JINT i é atualizado, a cada rodada, pela relação

$$\mathbf{J}_{ei}(t+1) = \mathbf{J}_{ei}(t) + \Delta\mathbf{J}_{ei}(t), \quad (5.1)$$

onde $\Delta\mathbf{J}_{ei}(t)$ representa a atualização realizada pelo algoritmo de aprendizado utilizado. Esta atualização é feita a partir do vetor inicial $\mathbf{J}_{ei}(t=0)$, com $\|\mathbf{J}_{ei}(t=0)\| = 1$, gerado aleatoriamente. A decisão $\sigma_{ei}(t)$ deste jogador, é gerada pela regra (4.3).

No grupo de jogadores com estratégias fixas, o j -ésimo jogador utiliza um perceptron com vetor sináptico $\mathbf{J}_{oj} \in \mathbb{R}^d$, imutável ao longo das rodadas, como estratégia de escolha. De agora em diante, vamos adotar o índice (o) para representar quantidades associadas aos jogadores deste grupo. O vetor sináptico do jogador j é gerado aleatoriamente no início da competição, obedecendo à normalização $\|\mathbf{J}_{oj}\| = 1$. Por estas características, vamos denominar os jogadores pertencentes a este grupo por *Jogadores Estáticos* (JEST's). A decisão $\sigma_{oj}(t)$ do JEST j , na rodada t , será tomada de forma análoga ao grupo anterior, mediante a regra (4.3).

Considerando N_e o número de JINT's e N_o o número de JEST's, podemos escrever o resultado da minoria, numa dada rodada t , como

$$\sigma_m(t) = -\text{sign} \left[\sum_{j=1}^{N_o} \sigma_{oj}(t) + \sum_{i=1}^{N_e} \sigma_{ei}(t) \right]. \quad (5.2)$$

Com esta informação, pode-se atualizar a pontuação (fitness) do i -ésimo JINT e do j -ésimo JEST pela equação (3.7).

Como os vetores sinápticos dos JEST's são gerados de forma aleatória e com módulos unitários, para N_o grande teremos uma distribuição homogênea destes vetores num espaço de d dimensões. Sendo assim, é interessante olharmos para as flutuações das escolhas dos JEST's. A flutuação $m(t)$ entre as escolhas dos JEST's para um dado vetor $\mathbf{S}(t)$, será definido pela diferença

$$m(t) \equiv N_{o+}(t) - N_{o-}(t), \quad (5.3)$$

onde $N_{o+}(t)$ é o número de JEST's que optaram por $+1$ e $N_{o-}(t)$ o número de JEST's que optaram por -1 na rodada t . No apêndice A, demonstra-se que esta grandeza possui o valor médio $\langle m \rangle = 0$ e o desvio padrão

$$\Delta^* m = \sqrt{N_o}. \quad (5.4)$$

Este último resultado nos permite observar que o *valor esperado* de $\sigma_m(t)$ não depende das opções dos JINT's para $N_e < \sqrt{N_o}$. Nestas circunstâncias, mesmo com todos os N_e JINT's realizando a mesma opção, o sinal da minoria, dito pelos JEST's, não é afetado. Desta forma haverá a possibilidade de vitória de todos os JINT's em uma mesma rodada, mesmo com todos eles utilizando uma mesma estratégia.

A partir destas observações, pode-se definir N_e em termos do desvio padrão das escolhas dos JEST's :

$$N_e \equiv \lambda \sqrt{N_o}, \quad (5.5)$$

onde λ é um parâmetro de proporção e o objeto de nosso estudo. Mais adiante, vamos demonstrar que o grau de sucesso dos JINT's está intimamente relacionado ao valor deste parâmetro.

5.2 Medidas de desempenho

Antes da descrição de nossos resultados, devemos apresentar as quantidades usadas para caracterizar o processo de aprendizagem dos JINT's. Além das quantidades introduzidas na seção (3.1) (como o fitness e o fitness padronizado), utilizaremos as seguintes :

- **fitness médio**, que representa a média do fitness entre os indivíduos do mesmo grupo, é dado por

$$\Phi_\kappa(t) = \frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} \Phi_{\kappa i}(t), \quad (5.6)$$

onde $\kappa = \{o, e\}$ designa o grupo que se refere e $\Phi_{\kappa,i}(t)$ representa o fitness do jogador i pertencente ao grupo κ . Em muitas situações, neste trabalho, ser-nos-á útil o valor do fitness médio em $t = P$, e por isto definiremos

$$\Phi_{\kappa} \equiv \Phi_{\kappa}(t = P) \quad (5.7)$$

o fitness médio do grupo κ no final da competição;

- **fitness padronizado médio**, que representa a média de erros dos jogadores do mesmo grupo. Esta quantidade é dada de forma análoga à anterior, pela forma

$$E_{\kappa}(t) = \frac{1}{N_{\kappa}} \sum_{i=1}^{N_{\kappa}} E_{\kappa i}(t); \quad (5.8)$$

- **Erro médio de predição**, definido por

$$e_{p\kappa}(t) = \frac{1}{t} E_{\kappa}(t), \quad (5.9)$$

que nos fornece a probabilidade que um dado jogador, pertencente ao grupo κ , erre o sinal da minoria em uma rodada t ;

- **vetor centro de massa**, que é definido em termos da soma de todos os vetores sinápticos dos jogadores de cada um dos grupos. Esta quantidade é equacionada pela forma

$$\mathbf{J}_{\kappa} \equiv \frac{1}{N_{\kappa}} \sum_{i=1}^{N_{\kappa}} \mathbf{J}_{\kappa i}. \quad (5.10)$$

É importante observarmos que o vetor centro de massa dos JINT's ($\mathbf{J}_e$) varia ao longo das rodadas, enquanto que o vetor centro de massa dos JEST's ($\mathbf{J}_o$) é constante;

- **Parâmetro de correlação**, definido por

$$\rho(\mathbf{w}_a, \mathbf{w}_b) \equiv \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{\|\mathbf{w}_a\| \|\mathbf{w}_b\|} = \cos \phi, \quad (5.11)$$

que nos dá a correlação entre dois vetores, onde ϕ é o ângulo entre estes vetores num hiperespaço de d dimensões;

- **Correlação média entre jogadores**, que nos dá a correlação média entre os vetores sinápticos de perceptrons pertencentes a jogadores do mesmo grupo. Esta quantidade é definida por

$$\Psi_{\kappa} \equiv \frac{1}{N_{\text{pares}}} \sum_{ij} \rho(\mathbf{J}_{\kappa i}, \mathbf{J}_{\kappa j}), \quad (5.12)$$

representando a média empírica sobre todos os pares (i, j) possíveis de vetores sinápticos pertencentes ao grupo κ . Para o caso em que todos os vetores sinápticos do grupo κ apontam para a mesma direção, num espaço de d -dimensões, esta quantidade têm seu valor máximo, 1; no caso oposto, quando todos os vetores sinápticos estão homogeneamente distribuídos, como é o caso dos JEST's, esta quantidade se anula.

5.3 Perceptrons Hebbianos e Perceptrons Estáticos

Descreve-se, nesta seção, os experimentos para o caso em que cada JINT utiliza o perceptron com o aprendizado hebbiano (equação (4.4)) para “treinar” seus vetores sinápticos com a regra da minoria. Vamos, por esta razão, denominá-los *Jogadores Hebbianos* (JHEBB's).

A atualização realizada pelo aprendizado hebbiano na equação (5.1), é dada por

$$\Delta \mathbf{J}_{ei}(t) = \frac{\sigma_m(t) \mathbf{S}(t)}{\sqrt{d}}. \quad (5.13)$$

Esta atualização, por depender apenas de variáveis externas $(\sigma_m(t), \mathbf{S}(t))$, é igual para todos os JHEBB's (independe de i). Sendo assim, é bastante plausível que os vetores sinápticos dos JHEBB's “esqueçam” rapidamente as suas configurações iniciais e apontem todos a uma mesma direção. Nestas circunstâncias, todos os JHEBB's tomarão sempre uma mesma decisão.

A figura (5.1) apresenta, de forma esquemática, o histograma das escolhas dos jogadores para três situações típicas. Nesta ilustração considera-se que todos os N_e JHEBB's realizam a mesma decisão, e que os JEST's proporcionam uma flutuação m entre suas escolhas. A análise da figura nos permite concluir que:

- Para $N_e < m$ (figura (5.1-a)), os JHEBB's podem posicionar-se no lado oposto à maioria dos JEST's, sem que sua população implique em uma alteração no

resultado da minoria;

- Para $N_e > m$ (figura (5.1-b,c)) os JHEBB's estarão sempre do lado da maioria, independente da opção comum realizada por eles. Esta situação mostra a principal característica do JM: se existe um número significativo de jogadores utilizando uma mesma estratégia, estes afetam o resultado da minoria, implicando em um resultado diferente do previsto.

De um ponto de vista mais geral, podemos extrair as conclusões que seguem. Nas rodadas em que a flutuação dos JEST's assume $|m| > N_e$, basta que os JHEBB's identifiquem a tendência dos JEST's através da informação global¹; com a tendência identificada, os JHEBB's podem se posicionar junto à minoria dos JEST's, sem que sua população afete σ_m . No entanto, em rodadas que a flutuação dos JEST's assume $|m| < N_e$, e com todos os JHEBB's realizando uma resposta comum, estes últimos estarão sempre no lado da maioria. Para este último caso, a população de JHEBB's afeta drasticamente o resultado da minoria, implicando que σ_m seja diferente do previsto.

Podemos, então, mesmo que de forma aproximada, estimar a probabilidade P_a de que $|m|$ seja menor que N_e numa dada rodada. O cálculo desta quantidade é importante, pois esta representa a probabilidade de que o sinal da minoria seja afetado pelas decisões dos JHEBB's. No apêndice (A) discute-se a probabilidade de que uma população de perceptrons, com vetores sinápticos homogeneamente distribuídos, adquira um particular valor m , dado um vetor $\mathbf{S}$ aleatório. Esta probabilidade é a gaussiana:

$$P(m) = \frac{1}{\sqrt{2\pi N_o}} e^{-\frac{m^2}{2N_o}}. \quad (5.14)$$

Pode-se então determinar P_a , interpretando-a como a área hachurada sob a curva $P(m)$ da figura (5.2).

Através da equação (5.14), temos

$$P_a(N_e) = \int_{|m| < N_e} P(m) dm = \operatorname{erf}\left(\frac{\sqrt{2}}{2} \frac{N_e}{\sqrt{N_o}}\right), \quad (5.15)$$

onde $\operatorname{erf}(x)$ é a *função erro*, definida por

¹Veremos logo em seguida que esta identificação da tendência dos JEST's é uma tarefa passível de ser realizada.

Figura 5.1: Figura esquemática das escolhas dos jogadores numa rodada típica, com todos os JHEBB's realizando a mesma decisão. Nesta figura apresentam-se três situações prováveis: a) $N_e < m$ e os JHEBB's posicionando-se com a minoria dos JEST's; b) $N_e > m$ e os JHEBB's posicionando-se com a maioria dos JEST's; c) $N_e > m$ e os JHEBB's posicionando-se com a minoria dos JEST's. Na primeira situação, a população de JHEBB's é insuficiente para afetar a resposta da minoria; já nas duas últimas situações, a resposta da minoria é sempre contrária à resposta dos JHEBB's.

$$\text{erf}(y) \equiv \frac{2}{\sqrt{\pi}} \int_0^y e^{-x^2} dx. \quad (5.16)$$

Pela definição (5.5), podemos re-escrever P_a em termos apenas do fator λ :

$$P_a(\lambda) = \text{erf}\left(\frac{\lambda}{\sqrt{2}}\right). \quad (5.17)$$

Já a probabilidade P_n de que os JHEBB's não afetem σ_m , é simplesmente a probabilidade complementar à P_a , isto é, $P_n = 1 - P_a$. Sendo assim, o resultado (5.17) nos dá

Figura 5.2: Distribuição gaussiana de $P(m)$. A área hachurada representa a probabilidade P_a de que os JHEBB's afetem o sinal da minoria.

$$P_n(\lambda) = \text{erfc}\left(\frac{\lambda}{\sqrt{2}}\right), \quad (5.18)$$

onde $\text{erfc}(x)$ é a *função erro complementar*, definida por $\text{erfc}(x) = 1 - \text{erf}(x)$. O gráfico deste resultado está apresentado na figura (5.3), onde percebemos um decréscimo suave de P_n com λ .

Figura 5.3: Gráfico de P_n em função do parâmetro de proporção λ .

É claro que o resultado (5.18) não condiz exatamente com a realidade, uma vez que para obtê-lo consideramos $\{\mathbf{S}\}$ gerados de forma aleatória e independente; lembrando que pela definição do JM (a situação real) cada vetor-história é gerado a partir do vetor-história anterior. Porém, tal resultado nos dá uma idéia qualitativa a respeito de um decréscimo suave do fitness Φ_e com o aumento de λ , em analogia ao comportamento encontrado para $P_n(\lambda)$. Este decréscimo suave pode ser verificado experimentalmente, e está descrito no item seguinte.

5.4 Resultados

Para analisar o desempenho dos JHEBB's neste modelo proposto, foi realizado um grande número de experimentos independentes. Cada experimento significa uma simulação com uma nova “semente” no gerador de números aleatórios, de modo a obtermos diferentes situações. Em uma dada simulação j , o valor do fitness médio entre os JHEBB's, na rodada P , é dada por $\Phi_e^{(j)}$, determinado pelas equações (5.6) e (5.7). Para um total de N_s simulações diferentes, podemos determinar a média empírica desta quantidade por

$$\langle \Phi_e \rangle = \frac{1}{N_s} \sum_{j=1}^{N_s} \Phi_e^{(j)}. \quad (5.19)$$

A figura (5.4) apresenta os resultados obtidos para $\langle \Phi_e \rangle$ em função do fator de proporção λ . Note que $\langle \Phi_e \rangle$ obedece um decaimento suave e sem transição de fase, à medida que se aumenta o parâmetro λ . Este resultado está de alguma forma relacionado ao resultado (5.18), onde vimos que $P_n(\lambda)$ também decresce de forma suave. Além do mais, estes resultados mostram que os JHEBB's conseguem bons resultados em uma competição ($\langle \Phi_e \rangle > 0$) para $\lambda < 1$.

Figura 5.4: Gráfico de $\langle \Phi_e \rangle$ em função do parâmetro de proporção λ para diferentes dimensões. Os parâmetros utilizados foram: $P = 400$, $N_o = 1000$ e $N_s = 1000$.

Para encontrar os mecanismos utilizados pelos JHEBB's para conseguir Φ_e positivo, vamos fixar um certo valor de λ e analisar a dinâmica de uma competição. Analisaremos três casos particulares, dois deles com $\lambda < 1$ e um terceiro com $\lambda > 1$.

Resultados para $\lambda = 0.3$

Nas figuras (5.5) a (5.8), apresentam-se os resultados obtidos para uma particular competição, onde utilizou-se $\lambda = 0.3$, $d = 11$, $N_o = 1000$ e $P = 400$. A figura (5.5) apresenta os fitness médios dos dois grupos. Nessa figura tem-se também, o gráfico de $\Phi_{e, \text{melhor}}(t)$ e $\Phi_{e, \text{pior}}(t)$, que representam a evolução do fitness do melhor e do pior JHEBB da competição, respectivamente.

Podemos observar ganhos positivos para os JHEBB's. Além disso, há uma região de aumento constante na pontuação dos JHEBB's a partir de $t = 220$. Este aumento constante está relacionado ao surgimento de uma periodicidade no vetor-história, como pode ser conferido pela análise da figura (5.6). Esta figura apresenta o gráfico de $\rho(\mathbf{J}_o, \mathbf{S}(t))$ em função das rodadas da competição. Como $\mathbf{J}_o$ é constante, a periodicidade observada deve-se ao vetor $\mathbf{S}(t)$.

Antes de realizarmos uma interpretação desta periodicidade no vetor-história devemos fazer a seguinte análise. Como todos os JHEBB's realizam a mesma correção (5.13), é de se esperar, como já foi mencionado, uma convergência de seus vetores sinápticos para uma direção comum. A figura (5.6) evidencia este fato através do gráfico de $\Psi_e(t)$. A convergência $\Psi_e \rightarrow 1$, para t suficientemente grande, evidencia que os vetores sinápticos dos JHEBB's tendem a apontar na mesma direção. Desta forma, todos eles irão tomar a mesma decisão, independente do vetor-história. Com os JHEBB's realizando uma resposta comum, evidentemente estes só conseguirão um ganho positivo em rodadas que $|m| > N_e$. Mas, como $\lambda = 0.3$ é um valor bem menor que 1, a inequação citada é obedecida na maior parte das rodadas. O gráfico (5.7) comprova este fato, apresentando que a densidade de pontos acima da reta tracejada, que representa N_e , é maior que a densidade de pontos abaixo. Neste mesmo gráfico, podemos perceber uma região de periodicidade nas escolhas dos JEST's, gerada pela periodicidade em $\mathbf{S}$. Como esta periodicidade proporciona valores de $|m|$ bem maiores que N_e , os JHEBB's podem se posicionar tranquilamente junto à minoria dos JEST's, sem qualquer interferência no sinal da minoria.

A periodicidade do vetor-história pode, então, ser interpretada da seguinte forma.

Como os JEST's respondem da mesma maneira a vetores-história iguais, a partir do momento em que este vetor repetir um valor, já assumido, repetirá toda uma sequência de valores, periodicamente. Isto, é claro, sob a condição de que todos os σ_m , gerados por esta sequência, não sejam afetados pela população dos JHEBB's. Na simulação descrita, esta última condição é perfeitamente obedecida a partir de $t \approx 210$, o que proporcionou a periodicidade observada.

Outra característica interessante desta dinâmica está relacionada à correlação entre os vetores centro-de-massa dos dois grupos. Esta correlação é apresentada na figura (5.6), que apresenta o gráfico de $\rho(\mathbf{J}_e, \mathbf{J}_o)$ ao longo das rodadas da competição. Antes da periodicidade do vetor-história, temos uma situação em que $\mathbf{J}_e$ tende a se direcionar contrariamente ao vetor $\mathbf{J}_o$, i.e. $\rho(\mathbf{J}_e, \mathbf{J}_o) \rightarrow -1$. Este resultado é perfeitamente justificável, pois é de se esperar que a resposta da maioria dos JEST's concorde com a resposta de um perceptron fictício que possui o vetor $\mathbf{J}_o$ como vetor sináptico. O aprendizado dos JHEBB's consiste, então, em encontrar uma configuração para seus vetores sinápticos, de modo a se direcionar contrariamente a $\mathbf{J}_o$. Porém, com o surgimento da periodicidade de $\mathbf{S}(t)$, a informação sobre $\mathbf{J}_o$, contida no vetor $\Delta \mathbf{J}_{ei}(t) \propto \sigma_m(t) \mathbf{S}(t)$, fica restrita a apenas um grupo de valores. No entanto, esta informação incompleta já é suficiente para que $\mathbf{J}_o \cdot \mathbf{J}_e(t) < 0$, implicando que a resposta comum entre os JHEBB's seja diferente da resposta da maioria dos JEST's e sem que sua população influencie o valor de $\sigma_m(t)$.

A figura (5.8) mostra os JHEBB's “fugindo” da opção da maioria dos JEST's. Quando $\frac{N_{o+}}{N_o} > 0.5$ (maioria dos JEST's optam por +1), então $\frac{N_{e+}}{N_e} < 0.5$ (JEST's optam por -1). Quando $\frac{N_{o+}}{N_o} < 0.5$ (maioria dos JEST's optam por -1), então $\frac{N_{e+}}{N_e} > 0.5$ (JEST's optam por +1). Esta dinâmica garante a vitória dos JHEBB's na competição, mesmo com todos os seus jogadores realizando uma mesma resposta!

Figura 5.5: Gráfico a): fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e,melhor}(t)$) e do pior JHEBB ($\Phi_{e,pior}(t)$) da competição. Gráfico b): erro de predição dos JHEBB's ao longo da competição. Estes resultados são para $\lambda = 0.3$.

Figura 5.6: Gráfico de Ψ_e , $\rho(J_o, S)$ e $\rho(J_o, J_e)$ em função das rodadas da competição. O gráfico menor é uma ampliação do gráfico de $\rho(J_o, S)$. Estes resultados são para $\lambda = 0.3$.

Figura 5.7: Módulo da flutuação m ao longo das rodadas da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (bem menor que 1), a maior parte dos pontos se encontra acima da linha tracejada. Estes resultados são para $\lambda = 0.3$.

Figura 5.8: Gráfico da dinâmica de $\frac{N_{o+}}{N_o}$ e $\frac{N_{e+}}{N_e}$, mostrando a característica periódica das respostas dos JEST's. Pode-se perceber também, os JHEBB's "fugindo" da resposta da maioria dos JEST's. Estes resultados são para $\lambda = 0.3$.

Resultados para $\lambda = 0.6$

Nas figuras (5.9) a (5.13) apresentam-se os resultados obtidos para $\lambda = 0.6$, $d = 11$, $N_o = 1000$ e $P = 400$. Pela análise da figura (5.9), percebemos que os JHEBB's só iniciam uma pontuação crescente a partir de $t > 100$. Com este valor de λ (maior que o analisado anteriormente) temos um cenário mais difícil para os JHEBB's. Agora a probabilidade que $|m|$ seja inferior a N_e é maior, porém não tão grande a ponto de impedir um aprendizado por parte dos JHEBB's. Pela figura (5.11), percebemos que a densidade de pontos acima da linha pontilhada

ainda é maior que a densidade de pontos abaixo. Esta característica possibilita que na maior parte das rodadas da competição, σ_m não seja afetado pelos JHEBB's. Com os JHEBB's se posicionando junto à minoria dos JEST's, estes terão uma quantidade maior de ganhos positivos do que negativos, implicando em uma vitória na competição.

O gráfico da figura (5.10) comprova que todos os JHEBB's tendem a direcionar seus vetores sinápticos para uma direção comum ($\Psi_e \rightarrow 1$) e opostamente ao vetor $\mathbf{J}_o$ ($\rho(\mathbf{J}_e, \mathbf{J}_o) \approx -1$) para t suficientemente grande.

Na figura (5.12), temos a dinâmica das escolhas dos dois grupos. Como no obtido para $\lambda = 0.3$, os JHEBB's tentam “fugir” da opção da maioria dos JEST's.

Figura 5.9: Gráfico a): fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e,melhor}(t)$) e do pior JHEBB ($\Phi_{e,pior}(t)$) da competição. Gráfico b): erro de predição dos JHEBB's ao longo da competição. Estes resultados são para $\lambda = 0.6$.

Figura 5.10: Gráfico de Ψ_e , $\rho(J_o, S)$ e $\rho(J_o, J_e)$ em função de t . Estes resultados são para $\lambda = 0.6$.

Figura 5.11: Módulo da flutuação m ao longo das rodadas da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (menor que 1), a maior parte dos pontos ainda se encontra acima da linha tracejada. Estes resultados são para $\lambda = 0.6$.

Figura 5.12: Dinâmica das escolhas dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.6$.

Figura 5.13: gráficos de $\frac{N_{e+}}{N_e}$ em função de $\frac{N_{o+}}{N_o}$. A maior frequência de pontos nos quadrantes 2 e 4 deste gráfico sugere que, na maior parte das rodadas, os JHEBB's realizam uma resposta igual à da minoria dos JEST's. Estes resultados são para $\lambda = 0.6$.

Resultados para $\lambda = 1.1$

Finalmente, para $\lambda = 1.1$ (> 1) o grupo de JHEBB's afeta drasticamente a decisão da minoria. Os resultados das simulações estão apresentados nas figuras (5.14) a (5.16), que apresentam um desempenho dos JHEBB's pior que o dos JEST's. Nesta simulação foram utilizados os parâmetros : $d = 11$, $N_o = 1000$ e $P = 400$.

A figura (5.16) mostra que a densidade de pontos abaixo da linha pontilhada é bem superior à densidade de pontos acima. Isto implica que $|m|$ é quase sempre inferior a N_e . Nestas condições, σ_m é quase sempre afetado pelas decisões dos JHEBB's.

Figura 5.14: Fitness médio dos dois grupos de jogadores ao longo das rodadas da competição, juntamente com a evolução do fitness do melhor JHEBB ($\Phi_{e, \text{melhor}}(t)$) e do pior JHEBB ($\Phi_{e, \text{pior}}(t)$) da competição. Estes resultados são para $\lambda = 1.1$.

Figura 5.15: Gráfico de Ψ_e , $\rho(\mathbf{J}_o, \mathbf{S})$ e $\rho(\mathbf{J}_o, \mathbf{J}_e)$ em função de t . Estes resultados são para $\lambda = 1.1$.

Figura 5.16: Módulo da flutuação m ao longo da competição. A linha tracejada representa o valor de N_e . Note que, para este valor de λ (maior que 1), a maior parte dos pontos se encontra abaixo da linha tracejada. Estes resultados são para $\lambda = 1.1$.

5.5 Discussão

Não é difícil perceber a analogia entre este nosso modelo proposto (para o caso de λ suficientemente pequeno) e o cenário *Professor-Estudante* de redes neurais. Enquanto que o vetor sináptico correspondente a um particular JHEBB tenta se paralelizar ao vetor $-\mathbf{J}_o$ (através de seu algoritmo de aprendizado), de forma análoga o vetor sináptico de um perceptron estudante busca o vetor sináptico do perceptron professor [21] [22] [23]. É claro que existem diferenças significativas entre estes dois ambientes, como, por exemplo, a geração dos vetores de entrada $\mathbf{S}(t)$. Em nosso modelo, os vetores de entradas (os vetores-história) estão intimamente correlacionados, enquanto que no cenário professor-estudante estes são gerados de forma totalmente independente. Porém, parece que esta diferença não interfere significativamente em nossa comparação. Além disto, para uma analogia mais precisa, devemos estender nossa comparação ao caso do cenário professor-estudante com ruído multiplicativo. Neste caso, esse ruído do cenário professor-estudante seria o análogo à probabilidade $P_a(\lambda)$ de σ_m ser afetado pelos JHEBB's, em nosso modelo. Maiores detalhes a respeito do cenário professor-estudante com ruído multiplicativo são en-

contrados no apêndice (B).

6

RESULTADOS DA PROGRAMAÇÃO GENÉTICA

“Todos os espíritos repetem, desde Bacon, que somente são reais os conhecimentos que repousam sobre fatos observados.”

A. Comte

Vamos descrever, neste capítulo, os resultados obtidos para o JM entre dois grupos de jogadores (JINT's e JEST's), onde os JINT's utilizam perceptrons com o algoritmo de aprendizado gerado pela PG para treinar seus vetores sinápticos. Por esta razão, vamos denominar estes jogadores de *jogadores genéticos* (JGEN's).

Este capítulo está organizado da seguinte forma: na seção (6.1) introduziremos um modelo para uma utilização eficiente da PG entre os JINT's. Este modelo faz com que apenas as melhores funções de modulação (FMOD's), de uma população, sejam efetivamente utilizada pelos JINT's. O critério de geração, avaliação e pontuação destas FMOD's serão discutidos em detalhes; na seção (6.2) introduziremos algumas quantidades que serão relevantes para a análise dos experimentos; na seção (6.3) apresentaremos um gráfico comparativo entre os desempenhos de JINT's utilizando diversos algoritmos de aprendizado. Verificaremos a grande eficácia do aprendizado com a PG em comparação aos demais, para qualquer valor de λ ; finalmente na seção (6.4), analisaremos os mecanismos presentes nos JGEN's que os tornam mais eficazes que os JINT's com outros algoritmos de aprendizado.

6.1 O Modelo para o Jogo da Minoria

Como foi exposto na relação (5.13), os JHEBB's realizavam atualizações comuns a seus vetores sinápticos, utilizando unicamente informações externas. No

aprendizado com a PG, ao contrário, cada JGEN atualiza seus respectivos vetores sinápticos de uma forma particular, utilizando informações externas e internas.

A atualização realizada pelo aprendizado com a PG na equação (5.1) é dada por

$$\Delta \mathbf{J}_{ei}(t) = \eta_i(t) \frac{\sigma_m(t) \mathbf{S}(t)}{\sqrt{d}}, \quad (6.1)$$

onde η_i , que depende de variáveis externas e internas, é a FMOD que implementa o algoritmo de aprendizado associado ao JGEN i . Em nossas simulações da PG no JM, considera-se unicamente a evolução da população $\{\eta_i\}$; desta forma, vamos representar as $\{\eta_i\}$ como ESF's (programas) e construí-las a partir do conjunto de operadores

$$\mathcal{F} = \{+, -, \%, *, \text{srt}, \text{rlog}, \text{rexp}, \text{abs}, p., vv+, vv-, cv*\}, \quad (6.2)$$

e do conjunto de variáveis

$$\mathcal{V} = \{\sigma_m, \mathbf{S}, \sigma, E, h, \mathbf{J}\}. \quad (6.3)$$

Devemos ressaltar que a escolha dos elementos destes conjuntos é feita de forma bastante arbitrária, mas baseada em conhecimento prévio sobre PG e aprendizagem em perceptrons.

Dentro de $\mathcal{F}$ encontramos a soma, a subtração, a divisão protegida, a multiplicação, a raiz quadrada protegida, o logaritmo protegido, a função exponencial protegida, o valor absoluto, o produto escalar, a soma de vetores, a subtração de vetores e o produto de um escalar por um vetor. As oito primeiras têm argumentos escalares (um ou dois) e o resultado é um escalar. O produto escalar aceita como argumento dois vetores, e o resultado é um escalar. A soma e a subtração vetorial aceitam dois vetores como argumento e o resultado é um vetor. O produto de um escalar por um vetor aceita como argumento um escalar e um vetor, e o resultado é um vetor. As funções protegidas são funções cujos domínios de definição foram estendidos, de modo a aceitar um maior número de argumentos. A raiz quadrada protegida é a raiz quadrada do valor absoluto do argumento. Desta forma, estendemos seu domínio aos números negativos. A função exponencial é bem definida no domínio dos números reais, porém, para evitarmos *overflows*, impomos um limite superior para o seu argumento. O logaritmo protegido de um argumento é o logaritmo do valor absoluto do argumento, de modo a estendermos o seu domínio para os números reais negativos. Impomos também que o argumento do logaritmo

possui um limite inferior. Finalmente, a divisão protegida evita divisões por zero. Se o valor absoluto do denominador é menor que um certo valor muito pequeno ($\sim 10^{-13}$), a divisão protegida retorna um número grande ($\sim 10^{13}$).

Dentro de $\mathcal{V}$ temos o sinal da minoria e o vetor-história, que são as variáveis externas. Temos também a resposta do jogador, o total de erros cometidos na previsão do sinal da minoria, o campo local no perceptron e o vetor sináptico, que são as variáveis internas.

Uma das FMOD's passíveis de serem construídas a partir dos elementos dos conjuntos $\mathcal{F}$ e $\mathcal{V}$ é a expressão (escrita em termos de ESF do LISP):

$$(+ (* \sigma_m E) (p. (vv+ S J) (cv* (abs h) S))),$$

que, em forma algébrica, proporciona a seguinte expressão ao JGEN i que a utiliza:

$$\eta_i(t) = |h_i| \sigma_m(t) E_i(t) (\mathbf{S}(t) + \mathbf{J}_i(t)) \cdot \mathbf{S}(t). \quad (6.4)$$

Em nosso experimento, a PG envolve duas escalas de tempo. Uma escala de tempo rápida e uma escala de tempo longa. A escala de tempo rápida é medida em rodadas do JM, possuindo um intervalo de tempo completo, dado pelo período P de uma competição. Já a escala de tempo longa será medida em gerações, sendo onde acontece a evolução dos programas pela PG. A cada geração, ou para sermos mais exatos, ao final de cada competição, o programa η_i é avaliado de acordo com o fitness Φ_{ei} do JGEN i que implementa. Através dos mecanismos da PG (operadores genéticos e seleção), cria-se a nova geração de funções de modulação que, por sua vez, serão utilizadas pelos JGEN's em uma nova competição.

Embora sejam os operadores genéticos que permitem a busca por programas eficientes, é de se esperar, devido ao caráter estocástico destes operadores, a criação de programas que proporcionem desempenhos ruins aos JGEN's que implementam. Para resolver este inconveniente, adota-se o modelo que segue. Considere a população $\{\eta_i\}$ (composta por N_{pg} programas) maior que o número de JGEN's que irão participar de uma competição (N_e jogadores). Pode-se então associar a estes N_e jogadores, que denominaremos *jogadores genéticos reais* (JGENr's) apenas os melhores programas gerados. Paralelamente, considere que os demais ($N_{pg} - N_e$) programas estejam associados a *jogadores genéticos virtuais* (JGENv's); os JGENv's contabilizam suas pontuações pelas equações (3.7) e (3.8) e atualizam seus vetores sinápticos, da mesma forma que os JGENr's, porém suas respostas não influenciam

o resultado da minoria; desta forma, diz-se que estes jogadores participam de forma “virtual” em uma competição, justificando a sua denominação. Temos, então, duas categorias de JGEN’s, os JGENr’s e os JGENv’s. Numa dada competição, todos os N_{pg} JGEN’s (virtuais ou reais) são pontuados e atualizados, porém apenas os N_e JGENr’s estão *realmente* competindo.

No final da geração (término da competição), todos os programas $\{\eta_i\}$, $i = 1, \dots, N_{pg}$, são ordenados de acordo com o fitness de seus JGEN’s associados. Os N_e melhores programas são *copiados* e associados aos JGENr’s para serem utilizados novamente na próxima geração (competição). São então gerados ($N_{pg} - N_e$) novos programas (através dos operadores genéticos), que serão associados aos JGENv’s. Um programa recém criado poderá estar associado a um JGENr em uma competição futura, se proporcionar uma boa pontuação ao seu JGENv correspondente. Este modelo é esquematizado na figura (6.1).

Com o início de uma nova competição (geração), todos os jogadores (JEST’s, JGENr’s e JGENv’s) têm seus fitness zerados e o vetor sináptico inicial novamente gerado (de forma aleatória e com módulo unitário). Já a informação global, contida no vetor-história, deve ser transmitida ao longo das gerações. Esta característica é obtida através da imposição:

$$\mathbf{S}(t = P, g) = \mathbf{S}(t = 0, g + 1), \quad (6.5)$$

onde g é a geração corrente. Note que o vetor-história é, agora, uma função de t e g , i.e. $\mathbf{S} = \mathbf{S}(t, g)$, com $\mathbf{S}(t = 0, g = 0)$ gerado aleatoriamente.

Em todos os experimentos que se descrevem a partir de agora, foram mantidos fixos alguns parâmetros. Estes parâmetros e seus respectivos valores, estão apresentados na tabela (6.2).

6.2 Técnicas de Caracterização da Dinâmica de Evolução

Para monitorar a dinâmica do processo evolutivo na população de programas, utilizam-se diversas quantidades. Descrevem-se então, neste item, algumas quantidades mais relevantes para o nosso estudo.

Na natureza, o fitness é a força que direciona a seleção darwiniana, da mesma forma como no AG e PG. Na PG, a medida do fitness pode estar representada de diferentes formas, embora estas formas estejam associadas entre si, e sempre

Figura 6.1: Algoritmo do modelo utilizado.

Parâmetro	valor
N_{pg}	600
M	$\max(N_e, \frac{N_{pg}}{10})$
P_m	0.001
P_{cros}	0.999
P_r	0.0
P	400
d	11
N_o	1000

Tabela 6.2: Parâmetros que foram mantidos fixos em nossos experimentos com a PG.

representando o grau de adaptabilidade de uma determinado indivíduo. Três formas diferentes de se representar o fitness, conforme sugeridos em [2], são definidas a seguir:

- o *fitness absoluto* - medida do fitness que está ligado à terminologia natural do problema. Por exemplo, no jogo da minoria, o ganho acumulado de um dado jogador em uma competição (Φ_{ei}), pode ser interpretado como o fitness absoluto. Neste caso particular, quanto maior o fitness absoluto, mais adaptado é o jogador. Costuma-se chamar o fitness absoluto apenas por fitness¹, salvo em situações onde possa ser confundido com as demais formas de fitness;
- o *fitness padronizado* - é definido de forma que seu menor valor esteja sempre associado ao indivíduo mais adaptado. No caso do JM, o fitness absoluto não obedece a este critério. Porém, se considerarmos o número de erros acumulado pelo jogador i em uma competição (E_{ei}), teremos a obediência a esta definição²;
- o *fitness normalizado* - é definido de modo que a sua soma sobre todos os indivíduos da população (no caso dos JGEN's, os N_{pg} programas) seja igual a 1. Uma forma de definirmos o fitness normalizado n_i do indivíduo i é inspirada no fator de Boltzmann:

$$n_i = \frac{e^{-\beta E_i}}{\sum_i e^{-\beta E_i}}, \quad (6.6)$$

¹Que é o caso neste trabalho.

²Foi justamente por esta razão que na seção (3.1) denominamos E_i (equação (3.9)) por fitness padronizado.

onde β é um parâmetro de controle e E_i o fitness padronizado deste indivíduo. Se considerarmos os JGEN's, n_i representa a probabilidade de que o programa que implementa o jogador i seja selecionado pelos operadores genéticos para compor os programas da próxima geração.

A partir do fitness normalizado, pode-se definir a *entropia fenotípica* [24] [25] do grupo κ , pela expressão

$$S_\kappa(g) = - \sum_j^{N_\kappa} n_j \ln(n_j). \quad (6.7)$$

Esta entropia é dita “fenotípica” por ser uma função das propriedades expressadas pelo grupo κ em uma geração g ; essa quantidade é máxima quando todos os N_κ jogadores têm o mesmo fitness, e a emergência de um indivíduo diferente é marcada por um decréscimo desta entropia.

Diversas quantidades já introduzidas na seção (5.2) ganham agora, com a introdução da escala de tempo longa, uma dependência em g . Por exemplo, o fitness médio dos JGENr's em uma competição ($\Phi_e \equiv \Phi_e(t = P)$), deve ser expresso como uma função das gerações, $\Phi_e(g)$. De forma análoga, temos as seguintes funções de g :

- $\Phi_o(g)$, que representará o fitness médio dos JEST's, obtido na competição promovida na geração g ;
- $\mathbf{J}_o(g)$, que representará o vetor centro de massa dos JEST's³ na geração g ;
- $\Phi_{ei}(g)$, que representará o fitness do i -ésimo JGENr ao final de cada competição ($t = P$);

•

$$\mathbf{J}_{ei}(g) \equiv \mathbf{J}_{ei}(t = P, g), \quad (6.8)$$

que representará o vetor sináptico do JGENr i no final de cada competição;

•

$$\mathbf{J}_e(g) \equiv \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{J}_{ei}(g), \quad (6.9)$$

que representará o vetor centro de massa dos JGENr's no final de cada competição;

³Lembrando que a cada nova competição (geração), é criada uma nova população de JEST's, o que implica novos vetores sinápticos e, portanto, um novo $\mathbf{J}_o$.

•

$$\Psi_e(g) \equiv \left\langle \rho(\mathbf{J}_{ei}(g), \mathbf{J}_{ej}(g)) \right\rangle, \quad (6.10)$$

que representará a correlação média entre os vetores sinápticos dos JGENr's no final de cada competição. De forma análoga a equação (5.12), $\langle \dots \rangle$ representa a média sobre os pares (i, j) possíveis;

•

$$\langle \|\mathbf{J}_{ei}(g)\| \rangle = \frac{1}{N_e} \sum_{i=1}^{N_e} \|\mathbf{J}_{ei}(g)\|, \quad (6.11)$$

que representará o valor médio do módulo $\|\mathbf{J}_{ei}(g)\|$ no final de cada competição.

6.3 Análise Comparativa entre os Algoritmos de Aprendizado

Agora que foi introduzido o modelo para a utilização do aprendizado com a PG e as técnicas de caracterização da dinâmica de evolução, podemos realizar uma comparação entre este e os demais algoritmos de aprendizado apresentados na seção (4.2). Para isto, foi realizado um grande número de experimentos independentes. Com estes experimentos foi possível obter valores esperados para o fitness destes jogadores mediante o algoritmo de aprendizado utilizado, de forma similar ao descrito na seção (5.4), onde, através de N_s simulações independentes, determinamos o valor esperado do fitness ($\langle \Phi_e \rangle$) pela relação (5.19). Para o caso dos experimentos com JGEN's, o cálculo do valor deste fitness médio é feito sobre os resultados da última competição, promovida na geração g_{max} .

A figura (6.2) apresenta os valores de $\langle \Phi_e \rangle$ para cada um dos algoritmos de aprendizado em função do fator de proporção λ . Podemos perceber uma nítida superioridade dos JGENr's em relação aos demais algoritmos. Esta superioridade não acontece somente para regiões de λ 's pequenos, mas principalmente nas vizinhanças de $\lambda = 1$. Nesta região, os JGEN's ainda conseguem ganhos positivos numa competição frente a JEST's, o que não é possível para os demais algoritmos.

Os demais algoritmos apresentam desempenhos bastante próximos para regiões de $\lambda < 1$. Para $\lambda > 1$ o aprendizado com a normalização merece destaque. Nesta

Figura 6.2: $\langle \Phi_e \rangle$ em função de λ , para JINT's utilizando diferentes algoritmos de aprendizado. Para o caso dos JINT's utilizando o aprendizado com a PG, foi utilizado $g_{max} = 100$ e a média realizada sobre $N_s = 30$ experimentos diferentes, para cada valor de λ . Para os JINT's utilizando de qualquer um dos demais algoritmos, o fitness médio foi obtido a partir de $N_s = 1000$ para cada valor de λ . Para todos os casos foram utilizados $d = 11$, $P = 400$ e $N_o = 1000$.

região este algoritmo se mostra levemente superior ao aprendizado hebbiano e ao aprendizado do perceptron, apesar de não conseguir pontuação positiva.

6.4 Análise da Evolução dos Jogadores Genéticos

Uma vez observada a eficiência dos JGENr's frente aos JEST's, cabe agora efetuar uma descrição mais apurada dos mecanismos presentes nestes jogadores. Nesta particular seção, vamos apresentar os resultados de nossos experimentos onde, para um dado valor fixo de λ , analisou-se a evolução dos parâmetros internos

destes jogadores; parâmetros tais como fitness, estrutura de suas FMOD's e vetores sinápticos.

A PG, apesar de sua natureza estocástica, proporcionando resultados diferentes para cada experimento realizado, apresenta visíveis padrões em sua dinâmica. Através de um grande número de experimentos, foi verificada a existência de dois regimes de evolução para o fitness médio $\Phi_e(g)$. O primeiro acontece para $\lambda < 1$ e é caracterizado por $\Phi_e(g)$ sempre positivo e sem qualquer alteração significativa ao longo das gerações. Já o segundo, que acontece para $\lambda > 1$, $\Phi_e(g)$ apresenta valores bem menores que zero nas primeiras gerações, porém aumenta gradativamente até convergir para um dado valor. Este valor de convergência depende diretamente de λ , sendo menor quanto maior é este parâmetro de proporção. Como sugerem os resultados apresentados na figura (6.2), $\Phi_e(g)$ converge para um valor positivo, caso λ seja próximo de 1, porém negativo para λ suficientemente grande. Nos experimentos que se apresentam a seguir, utilizamos valores de λ que abrangem perfeitamente estes dois regimes de evolução.

Resultados para $\lambda = 0.3$

Os resultados de um experimento com $\lambda = 0.3$ aparecem nas figuras (6.3) a (6.7). A figura (6.3) apresenta que $\Phi_e(g)$ não possui qualquer alteração significativa ao longo das gerações. No gráfico da figura (6.4), podemos verificar que a distribuição do fitness e a entropia fenotípica dos JGENr's - que são quantidades relacionadas - também não sofrem grandes alterações em sua evolução.

Pode-se verificar que a dinâmica apresentada pelos JGENr's nas competições deste experimento apresentam características bastante parecidas com as dos JHEBB's. Por exemplo, a tentativa, por parte dos JGENr's, em opor seus vetores sinápticos ao vetor centro de massa dos JEST's, como pode ser observado nos gráficos da figura (6.7). Nesta figura, podemos verificar $\rho(\mathbf{J}_e(g), \mathbf{J}_o(g))$ sempre negativo e $\Psi_e(g) \approx +1$. Esta característica de perseguição a $-\mathbf{J}_o$, de forma análoga aos experimentos com JHEBB's, implica que os JGENr's sempre tendem a optar por uma resposta contrária à maioria dos JEST's. Um fato interessante é que, como este valor de λ proporciona periodicidades freqüentes no vetor-história⁴, evidentemente a informação sobre $\mathbf{J}_o(g)$ contida no par $(\mathbf{S}(t, g), \sigma_m(t, g))$ fica bastante restrita.

⁴Por razões discutidas na seção (5.4).

Este fato impede aos JGENr's oporem seus vetores sinápticos à direção de $\mathbf{J}_o(g)$ em uma dada competição, justificando $\rho(\mathbf{J}_e(g), \mathbf{J}_o(g))$ diferente de -1 , o que não impede, no entanto, os JGENr's de se posicionarem adequadamente à minoria.

Em uma particular geração do experimento descrito em (6.3), analisamos a dinâmica completa da competição correspondente. Os gráficos das figuras (6.5) a (6.6) apresentam os resultados obtidos. O gráfico da figura (6.6) apresenta a evolução de $\frac{N_{e+}}{N_e}(t)$ e $\frac{N_{o+}}{N_o}(t)$ nesta competição. Pode-se observar uma clara periodicidade nas escolhas dos jogadores dos dois grupos. Este fato acontece devido à periodicidade no vetor-história, como pode ser confirmado através de $\rho(\mathbf{J}_o, \mathbf{S}(t))$ na figura (6.5).

Figura 6.3: Gráfico da evolução do fitness médio dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.3$.

Figura 6.4: Gráficos da distribuição do fitness e da entropia fenotípica dos JGENr's em função da geração. No gráfico da distribuição, cada ponto representa o fitness $\Phi_{ei}(g)$ do JGENr i . Uma linha vertical fictícia deste gráfico apresenta a pontuação de cada um dos JGENr's que atuaram na competição promovida na geração g . Estes resultados são para $\lambda = 0.3$.

Figura 6.5: Gráfico a): Fitness médios dos dois grupos de jogadores referente a competição promovida na geração $g = 500$. Gráfico b): Gráfico de $\rho(\mathbf{J}_o, \mathbf{J}_e)$, $\rho(\mathbf{J}_o, \mathbf{S}(t))$ e $\Psi_e(t)$ para esta mesma competição. Estes resultados são para $\lambda = 0.3$.

Figura 6.6: Dinâmica das escolhas dos dois grupos de jogadores referente à competição promovida na geração $g = 500$. Estes resultados são para $\lambda = 0.3$.

Figura 6.7: Gráfico acima: evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Gráfico abaixo: plano de fase entre as quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para cada geração g . Estes resultados são para $\lambda = 0.3$.

Resultados para $\lambda = 0.9$

Os resultados obtidos em um experimento com $\lambda = 0.9$ estão apresentados nas figuras (6.8) a (6.13). A figura (6.8) apresenta a evolução de $\Phi_e(g)$. Note que, como no ítem anterior, esta quantidade não apresenta alterações significativas, embora agora $\Phi_e(g)$ oscile em torno de valores - positivos - bem menores que os do ítem anterior. Já a entropia fenotípica (figura (6.9)) apresenta uma leve flutuação nas primeiras gerações seguida de uma estabilização.

Como vimos pelos resultados da figura (6.2), para $\lambda = 0.9$ os JHEBB's não conseguem pontuações positivas, porém os JGENr's conseguem! Vamos então analisar uma particular competição do experimento descrito em (6.8) de modo a verificar a dinâmica dos JGENr's. Os seus resultados estão descritos pelas figuras (6.10) a (6.12). A figura (6.10) apresenta que tanto Φ_e quanto $\Phi_{e,pior}$, que representa o JGENr's menos pontuado, são positivos.

O gráfico da figura (6.11) apresenta a evolução dos vetores nesta competição. Pode-se perceber que o vetor $\mathbf{J}_e(t)$ se opõe quase totalmente a $\mathbf{J}_o$, ($\rho(\mathbf{J}_e(t), \mathbf{J}_o) \approx -1$). Entretanto, os vetores sinápticos dos JGENr's não apresentam uma direção comum ($\Psi_e \neq 1$), o que é bastante sensato. Afinal, se todos os JGEN's apontarem seus vetores sinápticos na mesma direção, isso causaria exatamente o que prejudica os JHEBB's. Como vimos na seção (5.4), os vetores sinápticos dos JHEBB's convergiam praticamente para a mesma direção, o que fazia com que todos eles tomassem a mesma decisão. Este fato proporcionava (para este valor de λ) uma alta probabilidade de influência dos JHEBB's no sinal da minoria, o que tornava σ_m diferente da prevista por estes jogadores.

A dinâmica das escolhas dos jogadores desta competição está descrita na figura (6.12). No gráfico desta figura, cada ponto representa $\frac{N_{e+}}{N_e}$ em função de $\frac{N_{o+}}{N_o}$ para uma dada rodada t . Pode-se perceber a grande quantidade de pontos nos quadrantes pares, sugerindo que a maioria dos JGENr's se posiciona junto à minoria dos JEST's.

Figura 6.8: Gráfico da evolução do fitness médio dos dois grupos de jogadores. Estes resultados são para $\lambda = 0.9$.

Figura 6.9: Gráfico da distribuição do fitness e da entropia fenotípica dos JGENr's em função da geração. Estes resultados são para $\lambda = 0.9$.

Figura 6.10: Gráfico acima: fitness médios dos dois grupos de jogadores referente à competição promovida na geração $g = 500$. Gráfico abaixo: erro médio de predição dos JGENr's referente a essa mesma competição. Estes resultados são para $\lambda = 0.9$.

Figura 6.11: Gráfico de $\rho(J_o, J_e)$, $\rho(J_o, S(t))$ e $\Psi_e(t)$ referente à competição promovida na geração $g = 500$. Estes resultados são para $\lambda = 0.9$.

Figura 6.12: Plano de fase de $\frac{N_{e+}}{N_e}$ e $\frac{N_{ot}}{N_o}$ referente a competição promovida na geração $g = 500$. A maior densidade de pontos nos quadrantes pares sugere que a maioria dos JGENr's se posiciona junto à minoria dos JEST's. Estes resultados são para $\lambda = 0.9$.

Figura 6.13: Gráfico acima: evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Gráfico abaixo: plano de fase entre as quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para cada geração g . Estes resultados são para $\lambda = 0.9$.

Resultados para $\lambda = 1.5$

Deve-se agora analisar um experimento que apresenta o que chamamos de segundo regime de evolução de $\Phi_e(g)$, que foi observado em todos os experimentos com $\lambda > 1$. Descrevem-se nesta seção os resultados de um experimento com $\lambda = 1.5$. A figura (6.14) apresenta a evolução de $\Phi_e(g)$ e $\Phi_o(g)$, que são os fitness médios dos dois grupos, e de $\Phi_{e, \text{melhor}}$ e $\Phi_{e, \text{pior}}$, que são os fitness dos JGENr's mais e menos pontuados da geração, respectivamente. Essa figura mostra uma constante melhora no desempenho médio dos JGENr's ao longo das gerações, com a convergência de $\Phi_e(g)$ para um valor maior que zero. Esse valor é atingido em menos de 100 gerações, porém, apenas a partir da geração $g_c \approx 280$, é que $\Phi_e(g)$ assume, preferencialmente, valores positivos. Pode-se então dividir a evolução de $\Phi_e(g)$ em três regiões distintas: a primeira, acontece entre $g = 0, \dots, 60$ e representa a região de aumentos gradativos de $\Phi_e(g)$; a segunda, que esta compreendida entre $g = 60, \dots, g_c$, representa uma região de instabilidades neste fitness médio; e a terceira, que está em $g > g_c$, representando a região de maturidade dos JGENr's. Note também que o fitness médio dos JEST's, $\Phi_o(g)$, praticamente não se altera ao longo das gerações, se mantendo sempre em uma região um pouco abaixo de zero.

O surgimento desse novo comportamento de $\Phi_e(g)$ em g_c acontece juntamente com transições de dinâmica em outras quantidades, o que sugere um certo vínculo. No gráfico apresentado pela figura (6.15), podemos verificar que a distribuição do fitness entre os JGENr's, juntamente com sua entropia fenotípica, sofre mudanças significativas a partir de g_c . Nessa ocasião, acontece uma brusca redução na diversidade fenotípica entre os JGENr's. Nos gráficos das figuras (6.17) e (6.18), apresentam-se a dinâmica de $\Psi_e(g)$ e $\rho(J_e(g), J_o(g))$. Em particular, na figura (6.18), tem-se gráficos do espaço de fase destas duas quantidades, com cada plano associado a uma das regiões distintas de evolução de $\Phi_e(g)$. Na figura mencionada, percebe-se que no gráfico referente a $g < g_c$ há duas regiões onde se distribuem os pontos. Em cada uma destas regiões, $\rho(J_e(g), J_o(g))$ assume valores negativos e positivos e os valores positivos estão associados a $\Psi_e(g) \approx 0$, ou melhor, a uma distribuição homogênea dos vetores sinápticos dos JGENr's. Evidentemente, para estas configurações, o fitness médio dos JGENr's deve ser parecido com o dos JEST's. Finalmente, na região de maturidade, $\rho(J_e(g), J_o(g))$ assume valores negativos em grande parte das gerações, e $\Psi_e(t = P, g)$ se estabelece em um dado valor próximo de 0.3.

Estes resultados sugerem que os JGENr's serão bem sucedidos em uma competição, se conseguirem encontrar valores ótimos para as duas grandezas, $\rho(\mathbf{J}_o, \mathbf{J}_e(g))$ e $\Psi_e(g)$. Todavia, neste regime de evolução ($\lambda > 1$), ao contrário do regime descrito nos dois itens anteriores, é necessário um significativo número de gerações para que os JGENr's encontrem FMOD's que realizem estas tarefas.

Uma condição necessária que se imagina de imediato, para que estes valores ótimos sejam alcançados pelos JGENr's, é um tamanho reduzido para os módulos dos vetores sinápticos. Obviamente, um JGENr's com $\|\mathbf{J}_{ei}\|$ muito grande (causado por uma FMOD η_i que assume valores altos), tem dificuldades em girar seu vetor sináptico, o que dificulta a procura por valores ótimos. O gráfico apresentado na figura (6.16) descreve a evolução do $\langle \|\mathbf{J}_{ei}\| \rangle$. Note que a evolução desta quantidade acompanha as três regiões de evolução de $\Phi_e(g)$, apresentando 1) uma diminuição gradativa na primeira região de evolução; 2) uma região de instabilidade; e finalmente 3) uma brusca diminuição em g_c . Este resultado demonstra o vínculo entre $\Phi_e(g)$ com $\langle \|\mathbf{J}_{ei}\| \rangle$. Podemos então inferir que o processo de seleção, presente na PG, age de modo a privilegiar funções de modulação que proporcionam vetores sinápticos menores. É claro que tais funções devem exibir outros tipos de qualidades para serem bem sucedidas, mas um baixo valor para $\langle \|\mathbf{J}_{ei}\| \rangle$ parece ser imprescindível, pelo menos no que diz respeito a um melhor ganho global por parte dos JGENr's.

Figura 6.14: Gráficos da evolução do fitness médio dos dois grupos de jogadores. Apresenta-se também, a evolução dos fitness dos JGENr's mais e menos pontuados da geração, representados por $\Phi_{e, \text{melhor}}$ e $\Phi_{e, \text{pior}}$, respectivamente. Estes resultados se referem a $\lambda = 1.5$.

Figura 6.15: Gráfico Acima: Distribuição do fitness entre os JGENr's ao longo das gerações. Neste gráfico, cada ponto representa o fitness obtido por um particular JGENr na geração g . Note uma explícita alteração nesta distribuição a partir de g_c . Gráfico abaixo: evolução da entropia fenotípica entre os JGENr's. Como esta entropia está associada a distribuição do fitness dos JGENr's, também acontece uma alteração significativa em sua dinâmica a partir de g_c . Esta alteração da entropia fenotípica indica uma redução da diversidade de fitness entre os JGENr's. Estes resultados se referem a $\lambda = 1.5$.

Figura 6.16: Gráficos de $\langle ||J_e|| \rangle$ em função das gerações. O gráfico inferior é igual ao superior, exceto pela região das ordenadas. Estas figuras mostram que no início das gerações (quando $\Phi_e(g)$ é bastante baixo), $\langle ||J_e|| \rangle$ assume valores altos; porém, a partir de g_c , quando $\Phi_e(g)$ entra no regime de maturidade, a frequência de valores pequenos, assumidos por $\langle ||J_e|| \rangle$, torna-se evidentemente alta. Estes resultados se referem a $\lambda = 1.5$.

Figura 6.17: Gráfico da evolução de $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$. Estes resultados se referem a $\lambda = 1.5$.

Figura 6.18: Plano de fase referentes às quantidades $\rho(\mathbf{J}_o(g), \mathbf{J}_e(g))$ e $\Psi_e(g)$ para duas regiões de evolução de $\Phi_e(g)$. Estes resultados se referem a $\lambda = 1.5$.

6.5 Discussão

Na figura (6.19) apresenta-se de forma esquemática, a posição relativa entre os vetores em questão, num espaço bi-dimensional. Todos os JGENr's, cujos vetores sinápticos se encontram na área hachurada devem emitir a mesma resposta que um perceptron fictício, possuidor do vetor $\mathbf{J}_o$ como vetor sináptico; ou seja, emitirão a mesma resposta que a maioria dos JEST's. Por outro lado, os JGENr's, cujos vetores sinápticos se encontram fora desta área, respondem de forma idêntica à minoria dos JEST's.

A figura (6.19-a) apresenta uma situação em que todos os JGEN's se posicionam junto a minoria dos JEST's. Para um valor de λ suficientemente pequeno, esta situação, com $\Psi_e \approx 1$ e $\mathbf{J}_e \approx -\mathbf{J}_o$, é a almejada; pois, nessas condições, todos os JGENr's emitirão a resposta da minoria, praticamente para qualquer vetor $\mathbf{S}$. A situação descrita na figura (6.19-a), com valores pequenos de λ , proporciona a

Figura 6.19: Figuras esquemáticas da disposição dos vetores, num espaço bi-dimensional, referente a uma rodada típica. A figura a) apresenta uma situação em que todos os JGENr's se posicionam junto à minoria dos JEST's, pois $\Psi_e \approx 1$ e $\mathbf{J}_e \approx -\mathbf{J}_o$. A figura b) apresenta uma situação em que $\Psi_e \neq 1$.

distribuição apresentada na figura (5.1-a).

Para valores de λ próximos ou maiores que 1, a situação se torna um pouco mais complicada para os JGENr's. Nesse caso, se todos estes jogadores emitirem uma mesma resposta, a probabilidade $P_a(\lambda)$ ⁵ fica próxima de 1, causando σ_m diferente da previsão. Assim, uma configuração de vetores, como a apresentada na figura (6.19-a) não é conveniente. Os JGENr's devem buscar uma melhor configuração de seus vetores sinápticos, i.e. um valor ótimo para Ψ_e . A figura apresentada em (6.19-b), mostra uma situação em que $\Psi_e \neq 1$ e, conseqüentemente, com alguns JGENr's se posicionando na área hachurada, o que lhes proporcionaram uma resposta diferente da comum entre os demais jogadores do seu grupo. Essa espécie de “sacrifício”, se feito por um número adequado de jogadores, possibilita que a maioria dos JGENr's ainda continuem emitindo uma resposta contrária à maioria dos JEST's e, assim, σ_m continua inalterado pelas decisões dos JGENr's. Este número de sacrificados deve ser tanto maior quanto maior for o valor de λ . É claro que se o valor de λ for muito grande, o número de jogadores a serem sacrificados é a maioria dos JGENr's, o que evidentemente, proporcionará ganhos negativos a estes jogadores, tornando impossível, por estas vias, uma vitória deste grupo. Portanto, de um ponto de vista “comunitário”, o sacrifício de jogadores é fundamental para que os JGENr's

⁵Dado pela equação (5.17), e que representa a probabilidade de que σ_m seja afetado pelas decisões de JINT's possuidores de uma resposta comum.

obtenham maiores fitness médios.

No experimento descrito para $\lambda = 1.5$, mais especificamente o resultado apresentado na figura (6.18), verificamos que $\Psi_e = 0$ não é interessante para estes jogadores, pois proporciona desempenhos médios bem semelhantes a JEST's. Sendo assim, deve haver valores de Ψ_e entre 0 e 1, que proporcionam melhores desempenhos aos JGENr's. E foi exatamente isto que observamos nas simulações com $\lambda = 0.9$ e $\lambda = 1.5$, quando verificamos que $\Psi_e(g)$ se estabiliza em um dado valor maior que 0, tanto menor quanto maior o valor de λ .

Podemos concluir então que a maioria dos JGENr's só consegue prever σ_m , porque estes jogadores realizam, simultaneamente, duas tarefas: a primeira consiste na tentativa de opor seus vetores sinápticos ao vetor $\mathbf{J}_o$. Para o caso de λ 's pequenos, a realização desta tarefa já é suficiente; a segunda consiste na alteração de seus vetores, de modo a encontrar um *ângulo médio ótimo* entre seus vetores sinápticos, ou seja, encontrar um valor ótimo para Ψ_e . No caso do primeiro regime de evolução de $\Phi_e(g)$, estas tarefas são realizadas pelos JGENr's já nas primeiras gerações. Porém, no segundo regime ($\lambda > 1$), é necessário um número significativo de gerações para que os JGENr's encontrem FMOD que proporcione a realização destas tarefas.

Como pode ser verificado pela figura (6.15), a geração que se inicia a realização destas tarefas é bastante explícita, sendo marcado por uma brusca diminuição da diversidade fenotípica dos JGENr's, além da alteração de dinâmica de outras quantidades correlacionadas, mostrando a situação de coordenação entre suas respostas.

Deve ser lembrado que estas tarefas são realizadas sem qualquer informação externa além do par $(\mathbf{S}(t), \sigma(t))$. Ou seja, um dado JGENr realiza estas tarefas com uma total ignorância a respeito de particularidades dos demais jogadores, seja outro JGENr, seja um JEST. Outro fato importante é o de que a realização dessas tarefas (que inclui o “sacrifício”), apesar de contribuir para uma melhora do fitness médio dos JGENr's, são feitas, por estes jogadores, devido à tentativa de maximizar o fitness individual e não uma escolha “intencional” de agir pelo bem do grupo.

Outra questão que merece destaque é com relação ao tempo de CPU necessário para a utilização da PG. A utilização desta técnica demanda um custo computacional que cresce significativamente com o número de gerações percorridas. Em nossas análises com $g_{max} = 100$, um experimento típico demanda de uma a duas horas de tempo de CPU. Em contrapartida, um experimento com $g_{max} = 500$, que foi o caso na maior parte dos experimentos descritos, chega a demandar de um a dois dias de

tempo de CPU.

7

CONCLUSÕES E PERSPECTIVAS FUTURAS

“Não se deve invejar o físico teórico. A natureza, ou, mais precisamente, a experiência, é um juiz inexorável e um não muito agradável crítico do seu trabalho. Nunca diz ‘sim’ a uma teoria. Nos casos mais bem sucedidos diz apenas ‘talvez’ e na vasta maioria deles exclama simplesmente ‘não’. Provavelmente toda a teoria será um dia denegada - um não para a maioria das teorias, logo após a sua formulação.”

Albert Einstein - 1922

Nesta dissertação, descrevemos a construção de uma plataforma computacional para a PG. Para testá-la estudamos o JM composto por jogadores que utilizam o perceptron para realizar suas escolhas. Os jogadores foram divididos em duas categorias distintas. A primeira formada por jogadores que alteram constantemente suas estratégias a partir de um algoritmo de aprendizado (os JINT's); e a segunda, composta por jogadores que não alteram suas estratégias (os JEST's).

Através de uma grande quantidade de experimentos, foi possível verificar condições que proporcionam bons desempenhos aos JINT's. Uma análise qualitativa seguida de uma comprovação experimental, permitiram verificar que o desempenho dos JINT's está diretamente relacionado com o tamanho relativo de sua população. Quando sua população é pequena, se comparada às flutuações das escolhas dos JEST's, os JINT's conseguem bons resultados, independente do algoritmo de aprendizado utilizado. Porém o desempenho dos JINT's decai suavemente, e sem transição de fase, com o aumento de sua população, chegando a ser pior que o desempenho dos JEST quando $N_e \approx \sqrt{N_o}$, i.e. $\lambda \approx 1$.

Foi possível observar também, diferenças de desempenhos médios entre os JINT's, de acordo com o algoritmo de aprendizado utilizado. Contudo, em todos os algoritmos de aprendizado, pelo menos no caso de λ 's pequenos, o aprendizado consistiu na identificação do vetor centro de massa dos JEST's através da informação global. O algoritmo de aprendizado de cada JINT age de forma a opor o vetor sináptico, deste jogador, ao vetor centro de massa dos JEST's. Este fato é justificável por duas razões. A primeira é que a maioria dos JEST's emitem respostas que concordam com a resposta de um perceptron fictício, possuidor do vetor centro de massa destes jogadores como vetor sináptico. A segunda é que, para valores pequenos de λ , a probabilidade de que σ_m seja afetado pelos JINT's é pequena, o que dá direito a estes jogadores posicionarem-se impunemente à minoria dos JEST's, mesmo que todos estes jogadores adaptativos realizem uma resposta comum.

Para valores de λ em torno ou maior que 1, as decisões dos JINT's afetam drasticamente o sinal da minoria, se todos estes emitirem uma mesma resposta. Este fato implica σ_m diferente do previsto pelos jogadores adaptativos. Sendo assim, nesta região de λ , fica explícita a diferença de desempenho dos JINT's de acordo com o algoritmo de aprendizado utilizado.

Analizamos detalhadamente dois algoritmos de aprendizado singulares entre si: 1) o algoritmo de aprendizado hebbiano, que, para t suficientemente grande, proporciona estratégias idênticas aos jogadores que o utilizam e, como foi mostrado, é terrivelmente prejudicial a estes jogadores para λ suficientemente grande; 2) o algoritmo de aprendizado gerado pela PG, que se mostrou claramente eficiente neste cenário, se comparado aos outros algoritmos de aprendizado analisados.

Os resultados dos experimentos com o algoritmo de aprendizado gerado pela PG foram apresentados no capítulo 6. Neste capítulo, foi possível concluir que o fitness médio dos JGENr's é maior que zero, mesmo em regiões de $\lambda > 1$. A razão para este bom desempenho global entre os JGENr's deve-se à realização simultânea de duas tarefas: a tentativa de oposição de seus vetores sinápticos ao vetor centro de massa dos JEST's e a descoberta de um *ângulo médio ótimo* entre seus vetores sinápticos, ou melhor, um valor ótimo para Ψ_e , compreendido em $0 < \Psi_e < 1$. Para $\lambda < 1$ foi verificado que, já nas primeiras gerações, os JGENr's encontram FMOD's que proporcionam a realização desta tarefa. Isto caracterizou o que denominamos de primeiro regime de evolução de $\Phi_e(g)$. Porém, para $\lambda > 1$, a descoberta de FMOD's que realizam tais tarefas, exigem um número significativo de gerações. Esse caso foi denominado de segundo regime de evolução de $\Phi_e(g)$. Neste regime, a geração

que se inicia a realização destas tarefas, é bastante explícita, sendo marcada por uma brusca diminuição da diversidade fenotípica dos JGENr's, além da alteração da dinâmica de outras quantidades correlacionadas. O fato intrigante é que estas tarefas são realizadas pelos JGENr's sem qualquer informação além do par externo $(\mathbf{S}(t), \sigma_m(t))$. Ou seja, um dado JGENr realiza estas tarefas com uma total ignorância a respeito de particularidades dos demais jogadores, seja outro JGENr, seja um JEST. Outro fato importante é que a realização destas tarefas, embora proporcione bons fitness médio para a população dos JGENr como um todo, devido à emergência de uma coordenação entre seus jogadores, é encaminhada pelos JGENr's sem uma escolha “consciente” de agir pelo bem do grupo, uma vez que cada JGENr's visa, primordialmente, a maximização de seu fitness individual.

Ainda em relação ao segundo regime de evolução de $\Phi_e(g)$, verificou-se que os mecanismos de seleção, presentes na PG, agem de forma a privilegiar FMOD's que proporcionam certas particularidades. Por exemplo, foi possível observar uma maior sobrevivência de FMOD's os quais proporcionam vetores sinápticos com baixos módulos. Esta característica possibilita uma maior flexibilidade destes vetores na procura por configurações ótimas de Ψ_e e $\rho(\mathbf{J}_o, \mathbf{J}_e)$. Mas, eventualmente, esta não é a única característica que deve estar presente em FMOD's eficientes neste cenário, sendo necessário um trabalho mais apurado na procura de outras peculiaridades. Uma boa sugestão seria a análise do comportamento destas FMOD's diante da *surpresa* [20][24], que pode revelar, por exemplo, aspectos da interpretação da informação por estes jogadores. Outra propriedade das FMOD's a ser analisada são os valores assumidos por estas funções ao longo de uma competição. Resultados preliminares mostram que JINT's que utilizam o algoritmo de aprendizado hebbiano, mas com pequenos *passos de aprendizado*¹, apresentam fitness médios bastante altos². Evidentemente valores pequenos para os passos de aprendizado também proporcionam módulos pequenos aos vetores sinápticos dos jogadores, o que justifica, pelo menos de forma superficial, as características das FMOD's encontradas pela PG.

¹Para o caso do aprendizado hebbiano puro, o passo de aprendizado é 1.

²Este fato é bastante razoável se considerarmos os resultados apresentados em [6]. Nesta referência, foi mostrado que em um JM formado somente por perceptrons adaptativos, portadores do algoritmo de aprendizado hebbiano, porém com um *passo de aprendizado* próximo de zero, tem-se um excelente desempenho global

Reflexões Finais

O JM, além de seus aspectos diretamente relevantes à economia, apesar de ainda estar bem longe do que chamaríamos de *modelo geral para o mercado*, tem promovido a utilização não menos especial às seguintes áreas: sistemas auto-organizáveis, vidros de spin, teoria de jogos, dinâmicas de população, seleção natural, ecologia, teoria de informação e agora, a PG.

A observação de melhores desempenhos de algoritmos gerados pela PG no modelo do JM proposto neste trabalho veio a acrescentar ainda mais o escopo de atuação desta técnica de programação. A PG vem sendo aplicada nas mais diversas áreas, obtendo resultados bastante animadores. Nas áreas da eletrônica, por exemplo, esta técnica tem sido bastante explorada, em especial na automação de circuitos analógicos, em sistemas de controles e *design* de antenas. Também podemos citar as suas aplicações em autômatos celulares e no seqüenciamento de proteínas[26]. Na área de redes neurais, estes algoritmos de aprendizado gerados pela PG têm sido utilizados na generalização de regras, obtendo melhores resultados se comparado com algoritmos de aprendizado tradicionais.

Entretanto, um empecilho à utilização da PG é o seu alto custo computacional. Dada esta característica, a *computação em paralelo* é uma necessidade, embora todo o trabalho descrito na dissertação tenha sido feito de forma convencional. Resultados apresentados em [27] mostram que os experimentos com a PG, feitos de forma convencional, demandam muito mais tempo - em geral, dias de tempo de CPU - que os efetuados por meio da programação em paralelo, a qual exige apenas algumas horas. A implementação de uma plataforma de programas que proporcione a introdução da programação em paralelo parece ser o primeiro passo a ser dado em direção a uma melhor utilização das técnicas apreendidas neste trabalho.

A

UMA BREVE ANÁLISE DE UM SISTEMA DE ESCOLHAS ALEATÓRIAS

Vamos analisar brevemente, neste apêndice, as flutuações das escolhas de perceptrons booleanos mediante um vetor de entrada qualquer. Primeiramente realizaremos uma revisão teórica e depois apresentaremos uma simulação computacional.

Considere uma população de N_o perceptrons, com seus vetores sinápticos estáticos (não adquirem aprendizado) e formando uma distribuição homogênea num espaço d -dimensional. A partir de um vetor de entrada $\mathbf{S}$ aleatório, estes perceptrons dividem-se em duas sub-populações. A primeira, composta por N_{o+} perceptrons que, pela regra (4.3), apresentam $+1$ como resposta. Já a segunda é composta por N_{o-} perceptrons que apresentam -1 como resposta. Devido à distribuição homogênea dos vetores sinápticos destes perceptrons, podemos interpretar, sem qualquer perda de generalidade, que este sistema possui as mesmas propriedades do chamado *Problema do Caminho Aleatório*¹[28][29]. Nesse problema, um indivíduo se desloca sobre uma reta, a partir da origem, dando passos (de comprimento igual) para a direita, com probabilidade p , ou para a esquerda, com probabilidade $1 - p$. O problema consiste em encontrar a probabilidade $P_N(N_1)$ que este indivíduo, num total de N passos, efetue N_1 passos para a direita. Como resultado, temos:

$$P_N(N_1) = \frac{N!}{N_1!(N - N_1)!} p^{N_1} (1 - p)^{(N - N_1)}. \quad (\text{A.1})$$

No caso dos perceptrons booleanos homogeneamente distribuídos, as opções $+1$ e -1 são análogas às opções *esquerda* e *direita*, e possuem probabilidades iguais de serem escolhidas, i.e. $p = \frac{1}{2}$. Desta forma, a partir da equação (A.1), a probabilidade que este sistema de N_o perceptrons assuma um particular valor N_{o+} , dado um vetor

¹Do Inglês *The Random Walk Problem*

de entrada qualquer, será :

$$P(N_{o+}) = \frac{N_o!}{N_{o+}!(N_o - N_{o+})!} \left(\frac{1}{2}\right)^{N_o}. \quad (\text{A.2})$$

Dispondo desta função de probabilidades, podemos calcular o valor médio, ou valor esperado de uma dada função $f(N_{o+})$, pela definição

$$\langle f(N_{o+}) \rangle = \sum_{N_{o+}=0}^{N_o} f(N_{o+}) P(N_{o+}). \quad (\text{A.3})$$

Podemos, portanto, encontrar o valor esperado de N_{o+} e de N_{o+}^2 , que são

$$\langle N_{o+} \rangle = \langle N_{o-} \rangle = \frac{N_o}{2} \quad (\text{A.4})$$

e

$$\langle (N_{o+})^2 \rangle = \langle (N_{o-})^2 \rangle = \frac{N_o^2}{4} + \frac{N_o}{4}, \quad (\text{A.5})$$

respectivamente.

A quantidade m é então definida como a flutuação das escolhas dos perceptrons (conforme seção (5.1)), isto é, a diferença entre estas duas sub-populações:

$$m \equiv N_{o+} - N_{o-} = 2N_{o+} - N_o. \quad (\text{A.6})$$

Considerando os resultados (A.4) e (A.5), o valor médio desta quantidade deve ser nulo e sua variância $\langle (\Delta m)^2 \rangle = N_o$, onde $\Delta m \equiv m - \langle m \rangle$. A partir desta variância, podemos determinar o desvio padrão entre as escolhas destes perceptrons, dado por

$$\Delta^* m \equiv \sqrt{\langle (\Delta m)^2 \rangle} = \sqrt{N_o}. \quad (\text{A.7})$$

Este resultado pode ser verificado experimentalmente através da simulação computacional apresentada pelo gráfico da figura (A.1). Este gráfico apresenta o desvio padrão das escolhas destes perceptrons em função de N_o . Nesta simulação, os perceptrons respondem a vetores de entrada gerados aleatoriamente e de forma totalmente independente. Podemos perceber claramente a dependência do desvio padrão com $\sqrt{N_o}$.

Figura A.1: Gráfico do desvio padrão das escolhas dos perceptrons em função de N_o . Este gráfico apresenta o resultado de simulações sobre uma média de 1000 vetores de entrada independentes para cada valor N_o .

Finalmente, podemos reescrever a probabilidade (A.2) em termos de m

$$P(m) = \frac{N_o!}{\left(\frac{N_o+m}{2}\right)!\left(\frac{N_o-m}{2}\right)!} \left(\frac{1}{2}\right)^{N_o}, \quad (\text{A.8})$$

que para N_o grande, nos permite utilizar a aproximação gaussiana:

$$\lim_{N_o \rightarrow \infty} P(m) = \frac{1}{\sqrt{2\pi N_o}} e^{-\frac{m^2}{2N_o}}. \quad (\text{A.9})$$

B

O CENÁRIO PROFESSOR-ESTUDANTE

Vamos apresentar, neste apêndice, um exemplo de cenário onde um algoritmo de aprendizado pode ser utilizado para se generalizar uma dada regra. Consideraremos aqui um *cenário de aprendizagem* conhecido na literatura de redes neurais por *cenário professor-estudante* [21] [22][23], onde um perceptron (o *estudante*) tenta inferir a *regra* (um mapeamento de entradas e saídas), definida por um outro perceptron (o *professor*).

O perceptron professor é possuidor de um vetor sináptico $\mathbf{B} \in \{\pm 1\}^d$ fixo, com d dimensões, que emite uma resposta binária dado um vetor de entrada $\mathbf{S} \in \{\pm 1\}^d$, através da regra:

$$\sigma_B(\mathbf{S}) = \text{sign}(b(\mathbf{S})), \quad (\text{B.1})$$

onde $b(\mathbf{S})$ é o campo local do professor, definido de forma análoga a (4.1), por

$$b(\mathbf{S}) \equiv \frac{\mathbf{B} \cdot \mathbf{S}}{\sqrt{d}}. \quad (\text{B.2})$$

Desta forma, o professor dita uma regra Booleana $\sigma_B(\mathbf{S}) = \pm 1$.

A tarefa do estudante, que possui um vetor sináptico ajustável $\mathbf{J} \in \mathbf{R}^d$ e arquitetura idêntica à do professor, consiste na busca de uma classificação correta da resposta do professor. Isto equivale a fazer com que o seu vetor sináptico convirja para a regra $\mathbf{B}$. A informação que é apresentada ao estudante para o aprendizado chega na forma de pares entrada-saída $(\mathbf{S}(t), \tilde{\sigma}_B(t))$, que são os *exemplos* da regra. A saída $\tilde{\sigma}_B(t)$ é a versão corrompida da resposta original (σ_B) do professor, cuja corrupção (ou ruído) é introduzida a partir de

$$\tilde{\sigma}_B = \xi \sigma_B, \quad (\text{B.3})$$

onde $\xi = \pm 1$. Se considerarmos χ como o *nível de ruído* presente, a probabilidade de inversão do sinal da resposta do professor será $P(\xi = -1) = \chi$. Este tipo de corrupção é conhecido na literatura por *ruído multiplicativo* [20].

No final de P exemplos, teremos um conjunto de pares ordenados entrada-saída do professor, que chamaremos de *lista de aprendizagem*:

$$\mathcal{L}_P = \left\{ \left(\mathbf{S}(1), \tilde{\sigma}_B(1) \right), \dots, \left(\mathbf{S}(t), \tilde{\sigma}_B(t) \right), \dots, \left(\mathbf{S}(P), \tilde{\sigma}_B(P) \right) \right\}, \quad (\text{B.4})$$

onde cada um destes P vetores de entradas são gerados de forma totalmente aleatória e independente. Os ajustes ao vetor sináptico do estudante será realizado a partir desta lista de aprendizagem e do algoritmo de aprendizado hebbiano que, de acordo com a equação (4.4), resultará na atualização

$$\mathbf{J}(t+1) = \mathbf{J}(t) + \frac{1}{\sqrt{d}} \tilde{\sigma}_B(t) \mathbf{S}(t). \quad (\text{B.5})$$

O estado instantâneo $\mathbf{J}(t)$ representa a *hipótese* que o estudante faz sobre a regra no instante t .

Como medida de desempenho do estudante ao longo de P exemplos, tomar-se-á o erro de generalização (e_g), definido como a probabilidade de que uma nova questão $\mathbf{S}(t+1)$, independente daquelas usadas anteriormente, produza uma resposta, no aluno, diferente do professor. Pode-se mostrar que esta quantidade é dada por

$$e_g(\mathbf{B}, \mathbf{J}(t)) = \frac{1}{\pi} \arccos \left[\frac{\mathbf{B} \cdot \mathbf{J}(t)}{\sqrt{d} \cdot \|\mathbf{J}(t)\|} \right]. \quad (\text{B.6})$$

O gráfico apresentado na figura (B.1) apresenta a evolução de e_g para diversos níveis de ruído. Apresenta-se também, neste gráfico, a evolução desta quantidade em função de $\alpha \equiv \frac{t}{d}$. Pode-se observar uma convergência de e_g para zero, com um comportamento assintótico com $\alpha^{-\frac{1}{2}}$ e independente de χ .

Figura B.1: Gráfico de e_g em função de t e de α , para diversos níveis de ruído.

Uma previsão teórica deste comportamento é apresentada em [30], onde encontrou-se que o e_g deve obedecer, para α suficientemente grande, a expressão

$$e_g = \frac{1}{\sqrt{2\pi}|1-\chi|} \alpha^{-1/2}. \quad (\text{B.7})$$

Esta expressão, assim como nossos resultados experimentais, diz que, não importando o nível de ruído presente na saída do sinal do professor, o algoritmo de aprendizado hebbiano irá sempre apresentar um erro de generalização com um decaimento com $\alpha^{-1/2}$; a menos, obviamente, para $\chi = 1$, onde nenhum aprendizado deve acontecer.

Embora este apêndice tenha tratado unicamente do aprendizado hebbiano, todos os algoritmos de aprendizado citados na seção (4.2) conseguem generalizar uma regra, no cenário professor-estudante.

BIBLIOGRAFIA

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- [3] W. B. Arthur . *Inductive Reasoning and Bounded Rationality (The El Farol Problem)*. Amer. Econ. Review 84 , 406, 1994.
- [4] R. Manuca, Y. Li, R. Riolo and R. Savit. *The structure of adaptive competition in Minority Games*. Physica A 282 (2000) 559-608.
- [5] E. Moro. *The minority game: an introductory guide* .Advances in Condensed Matter and Statistical Physics (2004).
- [6] W. Kinzel, R. Metzler and I. Kanter. *Dynamics of Interacting Neural Networks*. J. Phys. A : Math. Gen. 33 (2000).
- [7] R. Meltzer. Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Bayerischen Julius-Maximilians-Universität Würzburg, (2002).
- [8] M. Sysi-Aho, A. Chakraborti and K. Kaski. *Intelligent Minority Game with genetic-crossover strategies* (2002).
- [9] J. Almeida and J. Menche. *Quantifying a Certain "Advantage Law": Minority Game with above the Rules Agents*. Brazilian Journal of Physics 55 , 4 Dec.2003.
- [10] K. Nagel, S. Rasmussen, C Barrett. *Self-organization of complex structures: from individual to collective dynamics*. Edited by F. Schweitzer (Gordon and Breach, London p. 579) (1997).
- [11] D.Challet and Y. C. Zhang .*Emergence of Cooperation and Organization in a evolutionary game*. Physica A 246, 407 (1997).

-
- [12] D.Challet and Y. C. Zhang. *On the Minority Game: Analytical and numeric studies* . Physica A 256, 514 (1998).
 - [13] D. Challet , M. Marsili and Zecchina. *Statistical mechanics of systems with heterogeneous agentes: Minority games* Phys. Rev. Lett. 84 ,1824 (2000).
 - [14] D. Challet, M. Marsili and R. Zecchina. Phys. Rev. Lett 85, 5008 (2000).
 - [15] F. Rosenblatt. *Principles of neurodynamics, Spartan, New York* (1962).
 - [16] W. S. McCulloch and W. Pitts. *A logical calculus of ideas immanent in nervous activity* , *Bull. Math. Biophys.*, **5**, 115 (1943).
 - [17] E. A. de Oliveira. *Relações entre aprendizagem dentro e fora de equilíbrio termodinâmico*. Dissertação de Mestrado - Universidade de São Paulo. Instituto de Física, 2000.
 - [18] E. R. Caianiello. *Outline of a theory of thought processes and thinking machines*, *J. Theor. Biol.*, **1**, 204 (1961).
 - [19] D. O. Hebb . *The organization of behavior: A neurophysiological theory* , *J. Wiley, New York* (1949).
 - [20] O. Kinouchi. *Aprendizagem ótima em percétrons a partir de exemplos com ruído*. Dissertação de doutorado. Universidade de São Paulo, Instituto de Física, 1996.
 - [21] M. Oppen and W. Kinzel. *Statistical mechanics of generalization*. Physics of Neural Networks III (1994).
 - [22] J.Hertz and R. G. Palmer. *Introduction to the theory of neural computation*. Addison-Wesley, 1991.
 - [23] D. J. Amit. *Modeling Brain Function - The World of Attractor Neural Networks*. Cambridge University Press, 1989.
 - [24] J. P. Neirotti and N. Caticha. *Dynamics of the Evolution of Learning algorithms by Selection*. Physical Review E 67, 041912 (2003).
 - [25] C. Adami, C. Ofria, T. C. Colier. *Evolution of biological complexity*. Proc. Nat. Acad. Sci. USA 97 (2000).

-
- [26] J. R. Koza *Genetic Programming II: Automatic Discovery of Reusable Programs*. Book, the MIT Press (1994).
 - [27] J. P. Neirotti and N. Caticha. *Parallel LISP Genetic Programming in a beowulf cluster: an application to the study of learning algorithms*. Preprint submitted to Elsevier Science (2004).
 - [28] F. Reif. *Fundamental of Statistical and thermal physics* . Macgraw-hill Book Company, 19th printing (1987).
 - [29] S.R.A. Salinas. *Introdução à Física Estatística*. Edusp (1997).
 - [30] M. Copelli, O. Kinouchi and N. Caticha. *Equivalence between learning in noisy perceptrons and tree committee machines*. Physical Review E 53, 6341 (1995).
 - [31] D. S. Touretzky, *Common LISP: A Gentle Introduction to Symbolic Computation*. Carnegie Mellon University. The Benjamin-Cummings Publishing Company, Inc.,1990.
 - [32] R. O. Duda, P. E. Hart, D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, second Edition, 2001.