3
Algoritmos Evolutivos com Inspiração Quântica e Representação Real – AEIQ–R

Conforme mencionado nos capítulos 1 e 2, os problemas de otimização numérica são um importante assunto de pesquisa e oferecem desafios nas mais diversas áreas como, por exemplo, otimização de plantas industriais, controle, síntese de filtros digitais, treinamento de redes neurais e diversas outras aplicações. O uso de algoritmos evolutivos com inspiração quântica nesta classe de problemas de otimização pode, potencialmente, oferecer inúmeras vantagens. Mas, como acontece com outros tipos de algoritmos evolutivos (sendo os algoritmos genéticos o exemplo mais comum), a representação binária não é, necessariamente, a mais adequada para este tipo de problema, por apresentar algumas particularidades que restringem a capacidade de otimização do algoritmo. Com o objetivo de contornar este problema, este trabalho apresenta um modelo com inspiração quântica que usa uma representação baseada em números reais, sendo, portanto, mais adequada para os problemas de otimização numérica. Além de oferecer uma representação mais adequada, o modelo apresentado neste trabalho, denominado Algoritmo Evolutivo com Inspiração Quântica e Representação Real – AEIQ–$\mathbb{R}$, tem as seguintes vantagens, em relação aos algoritmos genéticos tradicionais e ao AEIQ–$\mathcal{B}$:

- Menor tempo para convergência;
- O conhecimento sobre o problema que está sendo otimizado é armazenado diretamente nos cromossomos quânticos;
- Populações com poucos indivíduos; capacidade de convergir de forma eficiente, mesmo com populações com poucos indivíduos.

3.1
O Modelo de Algoritmo Evolutivo com Inspiração Quântica e Representação Real

O interesse principal deste trabalho é desenvolver um modelo completo, que permita utilizar uma representação numérica em um algoritmo de otimização com inspiração na física quântica. Ao contrário do AEIQ–$\mathcal{B}$, descrito na seção 2.5, deseja-se que este algoritmo permita representar uma superposição de estados

contínuos. Por este motivo, a abordagem usando funções de onda é usada como inspiração para o desenvolvimento do algoritmo.

É importante frisar que, no caso de algoritmos com inspiração quântica, a superposição dos estados é apenas simulada, ou seja, o algoritmo é executado em um computador convencional (daí a denominação inspiração quântica).

As próximas seções deste capítulo apresentam o algoritmo completo do modelo proposto neste trabalho, bem como uma descrição detalhada de cada um dos passos do algoritmo, da representação dos indivíduos com inspiração quântica, dos operadores evolutivos e outros detalhes importantes referentes ao processo de otimização.

3.1.1
O Algoritmo Evolutivo com Inspiração Quântica com Representação Real

A figura 3.1 mostra a listagem completa do algoritmo evolutivo com inspiração quântica com representação real (AEIQ–ℝ).


```
iniciar
1.   t ← 1
2.   Gerar população quântica Q(t) com N indivíduos com G genes
3.   enquanto (t <= T)
4.       E(t) ← gerar indivíduos clássicos observando indivíduos quânticos
5.       se (t=1) então
6.           C(t) ← E(t)
7.       senão
8.           E(t) ← recombinação entre E(t) e C(t)
9.           avaliar E(t)
10.          C(t) ← K melhores indivíduos de [E(t) ∪ C(t)]
11.      fim se
12.      Q(t + 1) ← Atualiza Q(t) usando os N melhores indivíduos de C(t)
13.      t ← t + 1
14.  fim enquanto
fim
```

Figura 3.1: Listagem completa do algoritmo evolutivo com inspiração quântica usando representação real.

Este algoritmo será explicado detalhadamente nas subseções a seguir.

3.1.2
A População Quântica

Da mesma forma que o algoritmo genético com inspiração quântica usando representação binária (AEIQ–$\mathcal{B}$) necessita de uma população de indivíduos que representem a superposição dos possíveis estados que o indivíduo clássico pode assumir ao ser observado, o AEIQ–ℝ também necessita de uma população com a

mesma funcionalidade. No entanto, esta representação deve respeitar a condição de que o conjunto de estados observáveis deve ser contínuo, e não discreto como no caso do AEIQ–$\mathcal{B}$.

Para se conseguir esse objetivo, a população quântica $Q(t)$, em um instante t qualquer do processo evolutivo, é formada por um conjunto de N indivíduos quânticos q_i ($i = 1, 2, 3, .., N$). Cada indivíduo quântico q_i desta população é formado por G genes g_{ij} ($j = 1, 2, 3, ..., G$) que, por sua vez, são formados por funções que representam uma densidade de probabilidade. Os indivíduos quânticos podem ser representados pela equação 3-1.

$$q_i = [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), ..., g_{iG} = p_{iG}(x)] \tag{3-1}$$

Onde $i = 1, 2, 3, ..., N$, $j = 1, 2, 3, ..., G$ e as funções p_{ij} representam as funções densidade de probabilidade. Esta função densidade de probabilidade é usada pelo AEIQ–ℝ para gerar os valores para os genes dos indivíduos clássicos. Em outras palavras, a função $p_{ij}(x)$ representa a densidade de probabilidade de se observar um determinado valor para o *gene quântico* quando a superposição do mesmo for colapsada. De fato, a função $p_{ij}(x)$ pode ser definida como mostra a equação 3-2.

$$p_{ij}(x) = \psi_{ij}^{*}(x)\psi_{ij}(x) \tag{3-2}$$

Onde $\psi_{ij}(x)$ representa a função de onda associada ao gene quântico j do indivíduo i da população quântica e $\psi_{ij}^{*}(x)$ representa o conjugado complexo desta mesma função de onda. A função densidade de probabilidade deve respeitar a propriedade de normalização conforme mostrado na equação 3-3.

$$\int \psi_{ij}(x)^{*}\psi_{ij}(x)dx = \int p_{ij}(x)dx = 1 \tag{3-3}$$

É importante ressaltar que a função densidade de probabilidade deve ser integrável na região do domínio dentro do qual as variáveis que se quer otimizar podem assumir valores. Isto permite calcular a distribuição cumulativa de probabilidade e assim usar a distribuição de probabilidade para se gerar valores para os genes clássicos.

Uma das funções mais simples que se pode usar como função densidade de probabilidade é o pulso quadrado. Esta função pode ser definida pela equação 3-4.

$$p_{ij}(x) = \begin{cases} \frac{1}{U_{ij} - L_{ij}} & \text{se } L_{ij} \leq x \leq U_{ij} \\ 0 & \text{caso contrário} \end{cases} \tag{3-4}$$

Onde L_{ij} é o limite inferior e U_{ij} o limite superior do intervalo no qual o gene j do i–ésimo indivíduo quântico pode assumir valores (colapsar) quando observado. Esta equação respeita a propriedade de normalização mencionada na equação 3-3. Além disso, corresponde a uma distribuição de probabilidade que pode

ser facilmente utilizada em algoritmos computacionais, já que a mesma corresponde a uma distribuição uniforme no intervalo $[L_{ij}, U_{ij}]$.

Um exemplo de um gene quântico formado por um pulso quadrado é mostrado na figura 3.2. Neste exemplo, os limites L_{ij} e U_{ij} da função $p_{ij}(x)$ estão definidos como -1 e 1 respectivamente.

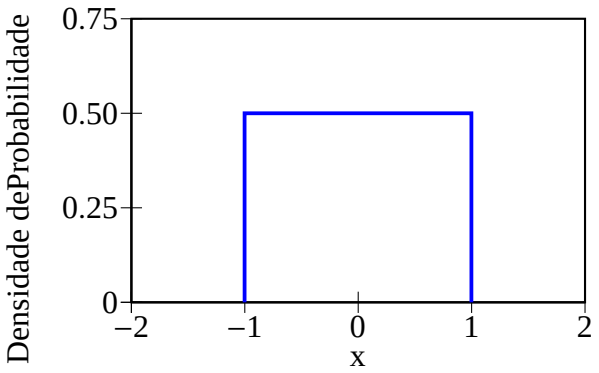

Figura 3.2: Exemplo de um gene quântico do AEIQ–ℝ .

O passo 2 do algoritmo da figura 3.1 consiste, portanto, em gerar um conjunto de N indivíduos quânticos formados por genes que, por sua vez, são formados por funções $p_{ij}(x)$. Para o caso em que $p_{ij}(x)$ é um pulso quadrado, pode-se representar o gene quântico armazenando-se os valores dos limites inferior e superior para cada gene, ou armazenando a posição do ponto central do pulso quadrado e a largura do mesmo. Por exemplo, considerando um indivíduo quântico q_i formado por dois pulsos quadrados $g_{i1}(x)$ e $g_{i2}(x)$, e supondo que estes dois pulsos quadrados têm uma largura igual a 2 e estão posicionados de tal modo que o seu centro está localizado nas posições -0.5 e 0.5 respectivamente, o cromossomo quântico pode ser representado, caso se esteja usando largura e centro como valores para os genes, por $q_i = (\mu_{i1} = -0.5, \mu_{i2} = 0.5, \sigma_{i1} = 2, \sigma_{i2} = 2)$, onde μ_{i1} e μ_{i2} indicam o centro e σ_{i1} e σ_{i2} representam a largura dos dois pulsos quadrados respectivamente. Obviamente a altura desses pulsos deve ser calculada de modo que a propriedade de normalização da função densidade de probabilidade formada pela soma desses pulsos quadrados seja respeitada. Assim, a altura dos pulsos quadrados deve ser tal que a área total dos mesmos seja igual a 1, e pode ser calculada usando-se a equação 3-4. No exemplo dado, cada pulso terá, portanto, uma altura igual a 0.5.

Ao usar o pulso quadrado, pode-se usar, pelo menos, duas estratégias diferentes de inicialização dos mesmos: na primeira, criam-se pulsos quadrados com uma largura igual a $(U_{ij} - L_{ij})/N$ (onde N é o número de indivíduos quânticos) e com seus centros distribuídos uniformemente ao longo de todo o domínio das variáveis; na segunda, cria-se todos os pulsos com o limite inferior e superior igual ao limite inferior e superior do domínio, respectivamente, sob o qual se deseja otimizar a função objetivo. Como exemplo para a primeira estratégia, pode-se considerar um

problema com duas variáveis x e y, ambas no intervalo $[-100, 100]$. Considerando também que a função densidade de probabilidade é representada por um pulso quadrado e que a população quântica inicial $Q(0)$ tem 5 indivíduos quânticos, pode-se representar estes indivíduos quânticos que formam a população inicial da seguinte maneira:

$$Q(0) = \begin{bmatrix} q_1 = [(\mu = -80, \sigma = 40); (\mu = -80, \sigma = 40)] \\ q_2 = [(\mu = -40, \sigma = 40); (\mu = -40, \sigma = 40)] \\ q_3 = [(\mu = 0, \sigma = 40); (\mu = 0, \sigma = 40)] \\ q_4 = [(\mu = 40, \sigma = 40); (\mu = 40, \sigma = 40)] \\ q_5 = [(\mu = 80, \sigma = 40); (\mu = 80, \sigma = 40)] \end{bmatrix}$$

Neste caso, a posição central de cada pulso foi determinada, de modo que os mesmos ficassem uniformemente distribuídos pelo domínio de cada variável. Além disso, a altura de cada pulso deve ser tal que a área total de cada um deles seja igual a 1. Assim sendo, a altura de cada pulso neste exemplo deve ser igual a $1/40 = 0.025$.

Já no caso da segunda estratégia, os indivíduos quânticos seriam representados como:

$$Q(0) = \begin{bmatrix} q_1 = [(\mu = 0, \sigma = 200); (\mu = 0, \sigma = 200)] \\ q_2 = [(\mu = 0, \sigma = 200); (\mu = 0, \sigma = 200)] \\ q_3 = [(\mu = 0, \sigma = 200); (\mu = 0, \sigma = 200)] \\ q_4 = [(\mu = 0, \sigma = 200); (\mu = 0, \sigma = 200)] \\ q_5 = [(\mu = 0, \sigma = 200); (\mu = 0, \sigma = 200)] \end{bmatrix}$$

Neste caso, a altura de cada pulso será igual a $1/200 = 0.005$.

3.1.3 Observação dos Indivíduos Quânticos

Após a inicialização da população quântica no passo 2, mostrado anteriormente, o algoritmo entra no laço principal do processo evolutivo. Este laço será executado por um determinado número T de gerações e é formado por várias tarefas.

O passo 4 é um dos mais importantes do algoritmo. É neste momento que se executa o processo de observação do estado quântico para se gerar indivíduos clássicos, ou seja, indivíduos cujos genes são números reais dentro do intervalo válido do domínio. Para se executar esta observação, usam-se as funções densidade de probabilidade $p_{ij}(x)$ e um gerador uniforme de números aleatórios no intervalo $[0, 1]$. Para cada gene é realizada uma observação através do seguinte procedimento:

1. Gera-se um número aleatório r no intervalo $[0, 1]$;

2. Identifica-se o ponto x tal que, dado que $P_{ij}(x) = \int_{-\infty}^{\infty} p_{ij}(t)dt$, $x = P_{ij}^{-1}(r)$;

3. x é o valor observado para o gene j do indivíduo clássico i.

Este processo, a princípio, implica que o número de indivíduos clássicos gerados é igual ao número de indivíduos quânticos. No entanto, isto não é, necessariamente, verdade. O número de indivíduos clássicos pode ser igual ou maior ao número de indivíduos quânticos, bastando que, ao se gerar os indivíduos clássicos, não se dê preferência a certos indivíduos quânticos no processo de observação, afinal, os indivíduos quânticos não são, a princípio, avaliados como os indivíduos clássicos e, portanto, não podem ser considerados melhores ou piores uns que os outros. Em outras palavras, deve-se garantir que nenhum indivíduo quântico seja preterido ou privilegiado em relação aos outros no que diz respeito ao número de vezes que o mesmo será usado para o processo de observação (geração dos indivíduos clássicos). Uma opção, usada neste trabalho, consiste em garantir que o número de indivíduos clássicos seja um múltiplo inteiro do número de indivíduos quânticos. Assim, dado que o número de indivíduos quânticos é igual a N e supondo-se que o número de indivíduos clássicos seja igual a N_c, define-se $N_c = kN$, onde $k \in \mathbb{N}^+$ indica quantos indivíduos clássicos serão gerados por cada indivíduo quântico.

É importante ressaltar que, o número de indivíduos quânticos não deve ser maior do que o número de indivíduos clássicos ($N \leq N_c$), de modo a garantir um processo de amostragem adequado. Esta restrição se deve ao fato de que os indivíduos clássicos serão usados para atualizar os indivíduos quânticos posteriormente, o que será explicado em detalhes adiante, na seção 3.1.4.

Como exemplo do processo de geração de indivíduos clássicos, pode-se considerar uma população quântica formada por dois indivíduos, cada um composto por 2 genes que usam pulsos quadrados como função densidade de probabilidade. A configuração destes indivíduos é dada na tabela 3.1 e a representação gráfica dos mesmos é mostrada na figura 3.3.

Indivíduo	**Genes**
q_1	$g_{11} = (-\mu_{11} = 5, \sigma_{11} = 20)$, $g_{12} = (\mu_{12} = 0, \sigma_{12} = 20)$
q_2	$g_{21} = (\mu_{21} = 5, \sigma_{21} = 20)$, $g_{22} = (\mu_{22} = 5, \sigma_{22} = 20)$

Tabela 3.1: Exemplo de indivíduos que formam uma população quântica $Q(t)$ em uma geração t qualquer.

A função $P_{ij}(x)$ (função cumulativa de probabilidade), associada a cada um dos genes quânticos mostrados anteriormente, pode ser facilmente representada por equações de retas, já que as funções $p_{ij}(x)$ são constantes dentro dos intervalos onde o valor das mesmas é diferente de 0. A figura 3.4 mostra as funções $P_{ij}(x)$

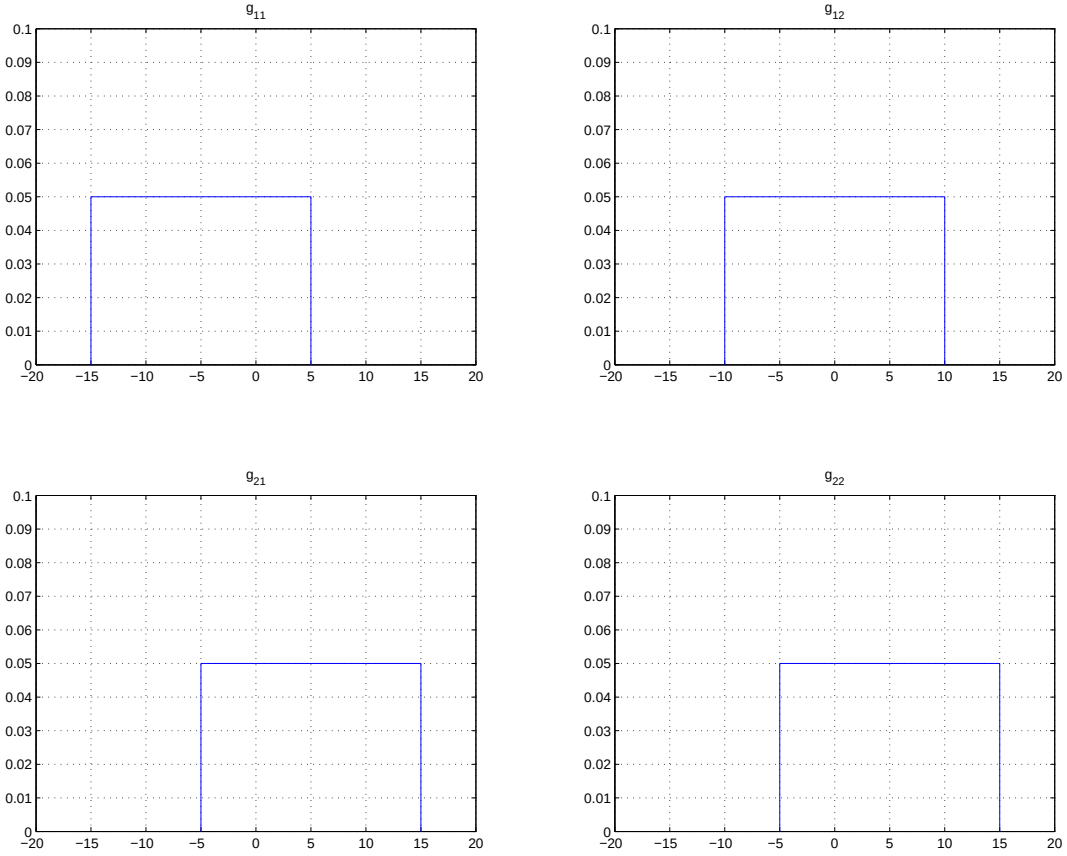

Figura 3.3: Genes de uma população quântica usando pulsos quadrados como função densidade de probabilidade.

associadas às funções densidade de probabilidade $p_{ij}(x)$ mostradas, anteriormente, na figura 3.3.

De posse das funções $P_{ij}(x)$ é possível efetuar a geração dos indivíduos clássicos propriamente dita (passo 4 do algoritmo). Esta observação é feita através de um processo aleatório, descrito anteriormente. Este processo aleatório é determinado pela função de onda de cada gene do indivíduo quântico. Assim, para cada gene j de cada indivíduo quântico i, deve-se gerar um número aleatório r_{ij} no intervalo $[0, 1]$. Com este número pode-se, então, gerar o valor observado para o gene j de cada indivíduo clássico. De fato, o cálculo dos indivíduos clássicos é extremamente simples quando se usa um pulso quadrado para representar os genes que formam o indivíduo quântico, já que a função cumulativa de probabilidade pode ser representada pela equação de uma reta, quando se usa pulsos quadrados (quando não se usa os pulsos quadrados, a função cumulativa de probabilidade pode ser, em geral, facilmente encontrada também). Dado o número r_{ij} gerado aleatoriamente, e usando-se a equação da reta $y(x) = ax + b$, pode-se calcular o valor do gene clássico usando-se a equação 3-5.

$$x_{mj} = r_{mj} * \sigma_{ij} + \left(\mu_{ij} - \frac{\sigma_{ij}}{2}\right) \tag{3-5}$$

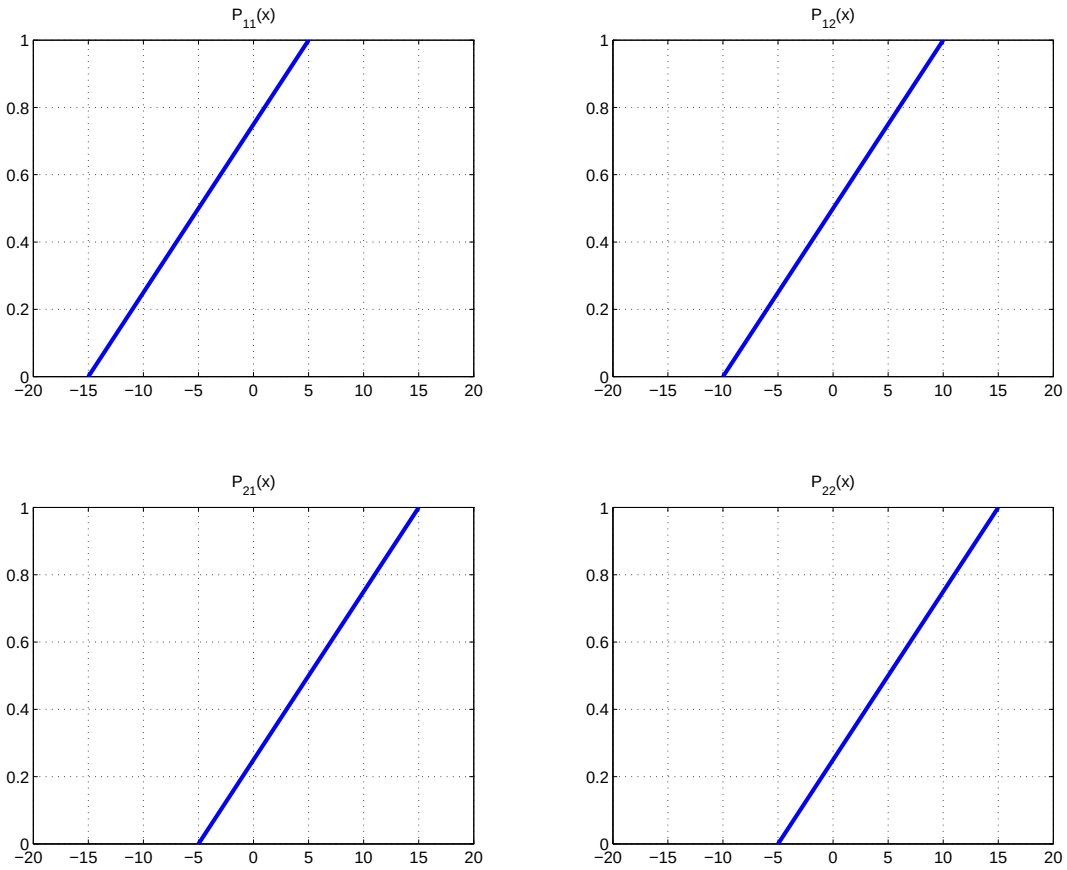

Figura 3.4: Funções cumulativas de probabilidade associadas aos genes de uma população quântica.

Onde x_{mj} é o j-ésimo gene do m-ésimo indivíduo clássico que se está gerando, r_{mj} é o número aleatório sorteado para o gene do indivíduo que está sendo observado, e μ_{ij} e σ_{ij} são a posição do centro e a largura do pulso quadrado que está sendo usado para observar o gene clássico.

Para tornar o processo mais claro, considere o exemplo de uma população quântica como a mostrada na figura 3.3. Considere também que, para este exemplo, deseja-se gerar uma população clássica com 4 indivíduos. Deste modo, o número de indivíduos $N_c = kN \rightarrow k = N_c/N$ é tal que $k = 2$ e, portanto, cada indivíduo quântico será usado para gerar dois indivíduos clássicos. Supondo-se que o gerador de números aleatórios forneceu 2 números diferentes para gerar os dois genes do primeiro indivíduo clássico – $\{0.1, 0.8\}$ – e, usando-se a equação 3-5 ($x_{11} = 0.1*20+(-5+10)$ e $x_{12} = 0.8*20+(0+10)$), o indivíduo clássico observado é, portanto, $\{7, 26\}$.

Neste ponto existe a possibilidade de se realizar uma operação de recombinação entre os indivíduos da população nova (geração atual) e da população antiga (geração anterior), equivalente ao passo 8 do algoritmo da figura 3.1. O uso desta função de recombinação é interessante para permitir o aproveitamento do material genético já presente na população clássica, melhorando a capacidade do algoritmo

explorar regiões próximas dentro do espaço de busca. Neste trabalho, é usada uma operação de recombinação similar à que é usada no algoritmo de evolução diferencial, descrito em (Storn95).

Em seguida, os indivíduos da população nova são avaliados (passo 9 do algoritmo da figura 3.1) pela função de avaliação que se deseja otimizar. Apesar de ser possível, não é, provavelmente, conveniente usar operações de mutação nos indivíduos novos ou antigos, já que os indivíduos quânticos, por si só, já introduzem um efeito aleatório (*exploration*) nas populações sendo evoluídas.

Com a nova população clássica gerada, deve-se tomar uma decisão sobre como substituir os indivíduos de uma eventual população pré-existente (passo 10 do algoritmo da figura 3.1). De fato, a estratégia de substituição dos indivíduos da população da geração anterior pelos indivíduos criados na nova geração deve ser usada a todo momento, com exceção da primeira geração, onde não existe uma população anterior. Existem várias abordagens possíveis para esta etapa do algoritmo. Elas podem ser:

1. Substituir todos os indivíduos da população antiga pelos indivíduos da população nova (equivalente a ausência de elitismo e *steady–state* dos algoritmos genéticos convencionais);

2. Substituir todos os indivíduos da população antiga pelos indivíduos da população nova, mas preservando o melhor indivíduo da população antiga (equivalente ao elitismo dos algoritmos genéticos tradicionais);

3. Substituir os *n* piores indivíduos da população antiga pelos *n* melhores indivíduos da população nova (equivalente ao *steady-state* dos algoritmos genéticos tradicionais;

4. Substituir os *n* piores indivíduos da população antiga por *n* indivíduos quaisquer da população nova.

Cada uma dessas abordagens apresenta vantagens e desvantagens. A opção 1 diminui o tempo de processamento pois elimina a necessidade de ordenação dos indivíduos da população antiga e da população nova. No entanto, esta abordagem tende a introduzir muito ruído, já que os indivíduos da população antiga são completamente substituídos e a avaliação do melhor indivíduo pode piorar significativamente, dependendo do problema que se deseja otimizar. A opção 2 resolve este problema preservando o indivíduo mais apto da população clássica na população nova. A opcão 3, apesar de exigir a ordenação dos elementos da população antiga e da população nova, apresenta a alternativa mais conservadora de busca, sem provocar grandes alterações na população que está sendo evoluída. Finalmente, a opção 4

combina o conservadorismo da opção 3, mantendo os indivíduos mais bem avaliados na população em evolução, mas diminuindo o tempo de processamento por não necessitar avaliar todos os N_c indivíduos da população nova.

3.1.4 Atualização da População Quântica

Finalmente, após a geração dos indivíduos clássicos, é necessário atualizar os indivíduos da população quântica (passo 12 do algoritmo da figura 3.1). Este procedimento depende, até certo ponto, do tipo de função densidade de probabilidade que se definiu para usar nos genes quânticos. Em geral, este processo deve:

- Reduzir o espaço de busca da função que se quer otimizar. No AEIQ–ℝ isto é feito reduzindo-se o tamanho da região onde a função densidade de probabilidade (genes quânticos) tem probabilidade diferente de 0;
- Mapear as regiões mais promissoras do espaço de busca. Isto deve ser feito aumentando-se a probabilidade de se observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem-sucedidos da população clássica.

Fica claro que, na primeira geração, por não se ter encontrado ainda as regiões mais promissoras do espaço de busca, o tamanho do domínio para a função que se quer otimizar deve ser o maior possível para que todas as regiões do espaço de busca sejam mapeadas. Em outras palavras, supondo-se que se deseja otimizar uma função $f(x)$ no intervalo $x \in [-100, 100]$, os genes de todos os indivíduos quânticos devem, preferencialmente, estar definidos de tal forma que todos os valores possíveis dentro do intervalo $[-100, 100]$ tenham chance de serem observados. Para o caso em que a função densidade de probabilidade é um pulso quadrado, estes valores devem ter, idealmente, a mesma chance de serem observados. Obviamente que, no caso de se ter informações *a priori* sobre regiões mais promissoras, pode-se redefinir as funções densidade de probabilidade de tal modo que os indivíduos quânticos ofereçam maiores chances de que genes na região mais promissora do espaço de busca sejam observados com mais freqüência do que em outras regiões.

Supondo-se, então, que uma função densidade de probabilidade com a forma de pulso quadrado foi usada para os genes quânticos e, para garantir que as duas condições citadas acima sejam satisfeitas, pode-se considerar o seguinte processo de atualização dos genes quânticos:

1. Modificar a largura dos pulsos de modo que o espaço de buscas seja reduzido;
2. Modificar a posição dos pulsos de modo que o ponto central dos mesmos coincida com o valor dos genes de um conjunto de indivíduos da população clássica.

Mais uma vez é possível adotar diversas estratégias para realizar cada uma das duas tarefas descritas anteriormente. Para o passo 1, pode-se usar um decaimento exponencial da largura dos pulsos quadrados, um decaimento linear ou, ainda, um processo similar ao usado em algoritmos de estratégia evolutiva chamado *"regra do* $1/5$*"* (Michalewicz94). A regra do $1/5$ faz com que a modificação de largura dos pulsos seja executada de forma homogênea para todos os genes quânticos e para todos os indivíduos quânticos. A heurística usada para determinar se a largura do gene será aumentada ou diminuída é a seguinte (Michalewicz94): se menos de 20% da população clássica criada na geração atual tiver uma avaliação melhor do que na geração anterior, a largura do gene é reduzida; se esta taxa for maior do que 20%, a largura do gene é aumentada; caso a taxa seja exatamente igual a 20%, nenhuma alteração é feita. Matematicamente, isto pode ser representado pela equação 3-6.

$$\sigma_{ij} = \begin{cases} \sigma_{ij} \cdot \delta & \varphi < 1/5 \\ \sigma_{ij} / \delta & \varphi > 1/5 \\ \sigma_{ij} & \varphi = 1/5 \end{cases} \tag{3-6}$$

Onde σ_{ij} é a largura do j-ésimo gene do i-ésimo indivíduo quântico em $Q(t)$, δ é um valor arbitrário, em geral no intervalo $]0, 1[$ e φ é a taxa que indica quantos indivíduos da nova população clássica foram melhores do que os indivíduos da população clássica na geração anterior.

Esta operação de redimensionamento da largura dos genes da população quântica pode ser feita em cada geração ou pode usar um intervalo de tempo maior (por exemplo, a cada 5 gerações). Intervalos menores para o redimensionamento aceleram o processo de convergência, mas podem levar o algoritmo a ficar preso em mínimos locais. Por outro lado, intervalos grandes podem evitar que o aprisionamento aconteça, mas podem retardar o processo de otimização.

O argumento para o uso desta regra se baseia na seguinte heurística: se menos de 20% dos indivíduos da população clássica gerada tiverem melhorado de uma geração para outra, é provável que o algoritmo esteja fazendo a busca em uma região muito grande e, neste caso, pode ser interessante reduzir o espaço de buscas; se mais de 20% dos indivíduos tiverem melhorado de uma geração para outra, é provável que o algoritmo esteja fazendo a busca em uma região muito pequena (e portanto achando, facilmente, indivíduos com avaliações melhores) e o mesmo deve tentar ampliar a região de busca; a taxa de 20% é considerada ideal e, portanto, nenhuma alteração é feita caso essa taxa seja medida.

Já no passo 2 da atualização dos genes quânticos, o processo pode ser definido em termos de quais indivíduos da população clássica serão usados para atualizar os pulsos da população quântica. Pode-se escolher os indivíduos mais aptos, os indivíduos menos aptos, um conjunto aleatório de indivíduos ou usar um processo de roleta para selecionar quais os indivíduos que farão parte do *pool* de indivíduos

clássicos que serão usados para atualizar os indivíduos quânticos. Após escolher quais indivíduos serão usados (o número de indivíduos clássicos escolhidos deve ser igual ao número de indivíduos quânticos e, portanto, o número de indivíduos clássicos não pode ser menor do que o número de indivíduos quânticos), o centro dos pulsos de cada indivíduo quântico deve ser modificado em relação ao valor de cada gene quântico. Em geral, pode-se usar um cálculo que desloque o centro dos pulsos na direção do ponto indicado pelo gene do indivíduo clássico. Por exemplo, supondo-se que o centro do gene quântico na geração t seja dado por $\mu_{ij}(t)$ e o valor do gene clássico seja dado por x_{ij}, então, pode-se calcular a nova posição do gene quântico na geração $t + 1$, através da combinação convexa dada pela equação $\mu_{ij}(t + 1) = \mu_{ij}(t) + \lambda(x_{ij} - \mu_{ij})$, onde $\lambda \in [0, 1]$ indica o percentual que se quer deslocar o centro do gene quântico na direção do gene clássico.

Todos esses passos do algoritmo devem ser repetidos por um número T de gerações, de acordo com o que foi mostrado na figura 3.1.

Além do algoritmo definido na figura 3.1, é possível definir um diagrama que mostra como as partes que constituem o modelo se relacionam entre si. Este diagrama pode ser visto na figura 3.5.

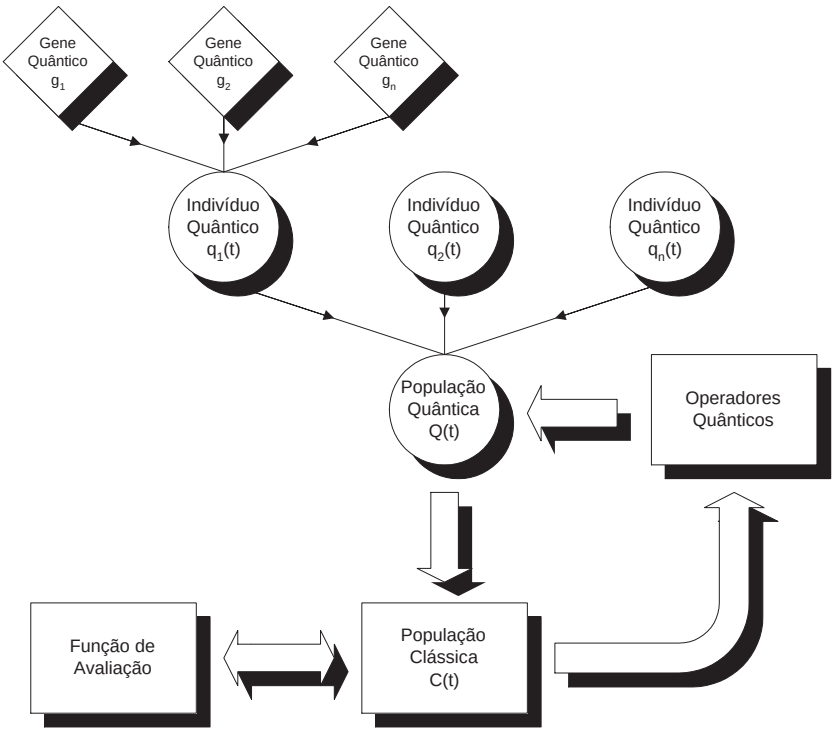

Figura 3.5: Diagrama completo do Sistema Evolutivo com Inspiração Quântica

Este diagrama mostra que a população quântica $Q(t)$ é composta de indivíduos quânticos que, por sua vez, são formados por genes quânticos. Esta população quântica é usada para gerar uma população clássica que é avaliada e usada para modificar os indivíduos da população quântica em um processo iterativo.

3.2
Usando o Modelo da Partícula na Caixa com o AEIQ–R

Conforme mostrado no capítulo 2, é possível idealizar um modelo de uma partícula confinada em uma caixa com barreiras de potencial infinito e calcular a função de onda para este sistema físico. Nesta seção, pretende-se mostrar como é possível usar este modelo hipotético para simular um gene quântico do AEIQ–ℝ.

Para isto, deve-se supor que, para cada gene quântico que se deseja representar, deve-se confinar uma partícula (por exemplo, um elétron) em um poço com barreiras de potencial infinito. Se estas barreiras de potencial infinito forem posicionadas em relação a um referencial qualquer, que permita relacionar a posição desta barreira com os limites do domínio da função que se quer otimizar, pode-se, então, usar a função de onda para a posição do elétron como o gene quântico.

Como exemplo, pode-se imaginar que se deseja otimizar uma função qualquer de duas variáveis x e y e que estas variáveis podem assumir valores no intervalo $[-100, 100]$. O AEIQ–ℝ que se pretende usar para otimizar esta função tem dois indivíduos quânticos, cada um com dois genes quânticos. Neste caso, pode-se criar um conjunto de 4 poços de potencial com um elétron confinado (um para cada gene quântico, conforme mencionado anteriormente) e com um nível energético tal que o valor de n (através do qual é possível definir o número de picos da função de onda, conforme explicado no capítulo 2) seja igual a 1. As barreiras de potencial infinito devem ser, caso se deseje que as funções densidade de probabilidade associadas a cada elétron se estendam por todo o domínio do problema na geração inicial, posicionadas de modo que, em relação a algum referencial pré-definido, as mesmas estejam na posição -100 e 100. Um exemplo de um elétron preso entre barreiras de potencial, sendo utilizado como gene quântico, pode ser visto na figura 3.6.

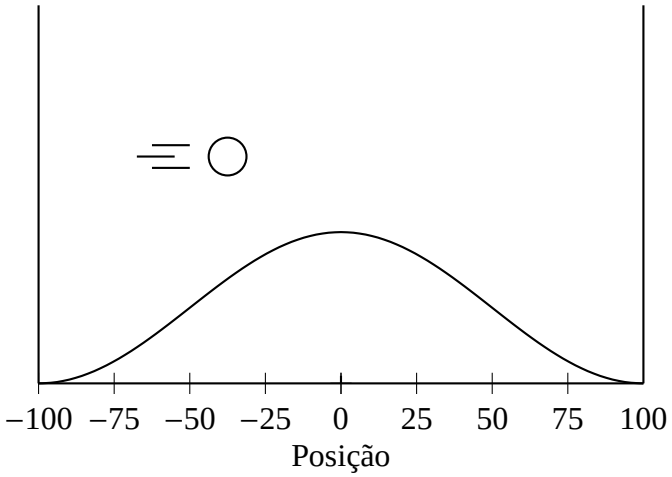

Figura 3.6: Diagrama de um modelo de partícula na caixa para uso no AEIQ–ℝ.

Nesta figura pode-se ver, além das barreiras de potencial nas posições -100 e 100, a função densidade de probabilidade associada à função de onda desta partí-

cula. O fato desta função de onda ser uma senóide faz com que a probabilidade de que as posições próximas às barreiras de potencial tenham menos chance de serem observadas ao se medir a posição do elétron. Isto, no entanto, não oferece problemas. É sempre possível definir as barreiras de potencial em posições além das fronteiras do domínio, de modo a permitir que as chances das posições mais próximas às bordas tenham maiores chances de serem observadas, descartando-se os genes clássicos que venham a ser, eventualmente, observados fora do domínio. Uma outra opção consiste em criar vários indivíduos quânticos diferentes, posicionando-se as barreiras de potencial ao longo do domínio (por exemplo, poderia-se usar três indivíduos quânticos com as barreiras de potencial nos intervalos $[-200, 0]$, $[-100, 100]$ e $[0, 200]$).

Para se gerar os indivíduos clássicos, basta observar a posição do elétron dentro de cada um dos poços de potencial. Em outras palavras, a observação da posição do elétron dentro da caixa é equivalente ao processo de observação da população quântica no AEIQ–ℝ descrito na seção anterior. Com os valores obtidos para a posição do elétron nessas observações, é possível ter indivíduos que possam ser avaliados da maneira tradicional. Ou seja, é possível, de fato, substituir a parte com inspiração quântica do algoritmo, por um sistema físico que possa fornecer o efeito quântico desejado.

Após a geração dos indivíduos clássicos e a avaliação dos mesmos, a atualização dos indivíduos quânticos é realizada. Esta atualização consiste, basicamente, em alterar o posicionamento e a distância entre as barreiras de potencial. Por exemplo, pode-se imaginar que, após as avaliações dos indivíduos clássicos, identifique-se que o indivíduo mais bem avaliado tem o gene relativo à variável x igual a 25. Além disso, define-se que, a cada geração, a largura do gene quântico deva ser igual a 60% da largura na geração anterior (neste caso, portanto, a nova largura será igual a 120). Assim, a nova configuração do gene quântico será aquela na qual as barreiras de potencial estejam localizadas na posição -35 e 85, como mostrado na figura 3.7.

Portanto, através deste exemplo, verifica-se que é possível usar as propriedades e características de um modelo quântico hipotético como um método de implementar os genes quânticos do AEIQ–ℝ . Deseja-se assim tornar mais clara a relação entre o algoritmo implementado e os elementos da física quântica nos quais o mesmo algoritmo se inspira.

3.3 Analogia entre o AEIQ–R e os Algoritmos Culturais

Uma importante característica do AEIQ–ℝ faz com que o mesmo tenha um comportamento semelhante ao apresentado pelos algoritmos culturais: as funções densidade de probabilidade usadas para representar os genes dos indivíduos quânti-

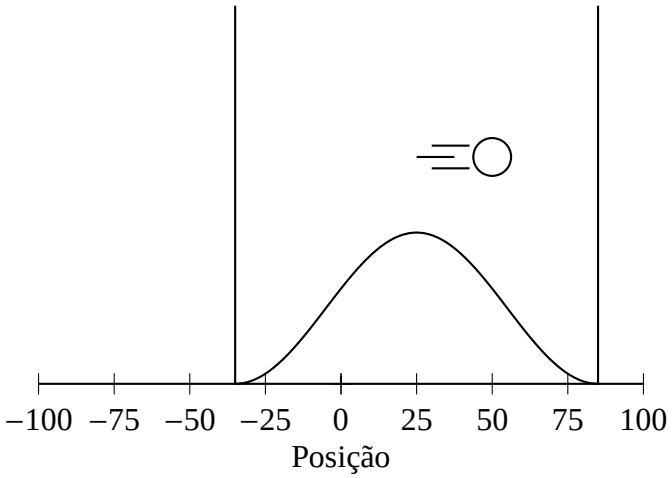

Figura 3.7: Diagrama do modelo de partícula na caixa para uso no AEIQ–ℝ após a atualização do gene quântico.

cos em $Q(t)$ representam uma forma de conhecimento normativo (esta forma de conhecimento é explicada no capítulo 2). Em outras palavras, este tipo de informação guarda semelhanças funcionais com o modelo de espaço de crenças dos algoritmos culturais. As principais diferenças entre os dois algoritmos estão no fato de que, nos algoritmos culturais, o espaço de crenças é usado para influenciar a avaliação dos indivíduos da população e influenciar também, indiretamente a geração de novos indivíduos. Além disso, os indivíduos da população, por sua vez, atuam no espaço de crenças modificando a sua forma. Já no caso do AEIQ–ℝ, a população quântica (que nesta comparação teria uma funcionalidade equivalente ao espaço de crenças) gera novos indivíduos para a população clássica, influenciando diretamente a criação dos novos indivíduos. De maneira análoga, os indivíduos da população clássica atuam na população quântica, alterando a forma e posição dos genes quânticos.

Além de guiar melhor o processo de otimização, da mesma maneira que nos algoritmos culturais, esta característica sugere uma importante possibilidade do uso do AEIQ–ℝ: como o conhecimento normativo fica armazenado no indivíduo quântico, este indivíduo pode ser utilizado para tarefas de otimização onde seja importante alimentar a população inicial com algum tipo de conhecimento. Um exemplo deste tipo de tarefa de otimização é o aprendizado em tempo real. Mais especificamente, pode-se imaginar o seguinte cenário: supondo que se deseja otimizar a fila de embarque de produtos em um porto através do planejamento do uso dos equipamentos e dos píeres que compõem o porto. Uma primeira otimização é feita e uma solução é produzida pelo algoritmo. Em seguida, pequenas alterações no cenário do porto (por exemplo, a quebra de um equipamento) fazem com que uma nova otimização seja necessária. Obviamente, o fato da mudança no cenário não ter sido drástica, faz com que a solução encontrada na otimização anterior, apesar de não ser mais a melhor possível, seja, provavelmente, uma solução acima da média

em termos de avaliação. Em um algoritmo genético convencional, pode-se usar o melhor (ou os melhores) indivíduo encontrado para alimentar a população inicial da nova tarefa de otimização. Com o AEIQ–ℝ , pode-se usar, não só os indivíduos da população clássica final, como também os indivíduos quânticos finais (sendo que os pulsos estão, provavelmente, com a largura próxima de zero e, portanto, devem ser aumentados antes de reiniciar o processo). Desta forma, a realimentação do processo evolutivo deixará de ser pontual e passará a englobar toda uma região promissora com respeito à solução procurada, de maneira similar aos espaços de crença nos algoritmos culturais. A principal diferença entre a população quântica e o modelo de espaço de crenças é que o conhecimento normativo é tratado de maneira probabilística no primeiro modelo e de maneira uniforme no segundo.

3.4
Modelo Neuro-Evolutivo – Aprendizado Supervisionado

O AEIQ–ℝ pode ser utilizado para otimizar qualquer tipo de função numérica. Em geral, os resultados obtidos pelo algoritmo são superiores aos obtidos pelos algoritmos genéticos clássicos com relação ao tempo de convergência do processo de otimização (alguns resultados do uso deste modelo para a otimização deste tipo de problemas são mostrados no capítulo 4). Além de apresentar este melhor desempenho em diversas situações, o modelo aqui proposto se destaca também nos processos neuro-evolutivos, devido ao fato de não apresentar problemas de supertreinamento, não necessitar de um conjunto de dados para validação e por ser capaz de realizar o aprendizado de forma *online*, conforme mencionado anteriormente.

O modelo aqui proposto usa o AEIQ–ℝ para treinar uma rede neural recorrente (FRNN – *Fully-Recurrent Neural Network*), otimizando os pesos e a topologia da mesma, de modo a minimizar o erro na saída, aprendendo, assim, a reconhecer os padrões de entrada fornecidos.

A rede neural recorrente é uma rede que tem realimentação nas suas ligações, da saída dos processadores para as entradas dos mesmo, além das ligações das entradas para os processadores e dos processadores para as saídas, como nas redes *multilayer perceptron* (Haykin99). Este tipo de rede apresenta bons resultados em problemas de previsão de séries (Haykin99) e em problemas de controle (Gomez03), pelo fato de que as realimentações introduzem um efeito de memória nestas redes (Gomez03). A figura 3.8 mostra um diagrama de uma rede neural recorrente com duas entradas, três processadores e uma saída (os *biases* não são mostrados para não sobrecarregar o diagrama).

Outra importante característica de uma rede neural recorrente é que, através da manipulação dos pesos, em particular ao se zerar alguns deles, é possível redefinir a topologia da rede neural, e até mesmo reduzir a rede recorrente a uma rede

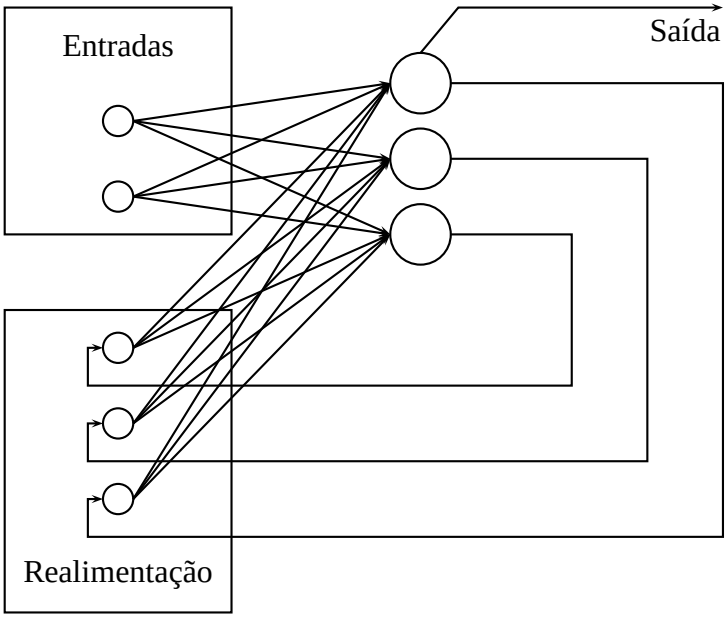

Figura 3.8: Diagrama mostrando uma rede neural recorrente. Este diagrama não mostra as ligações dos *biases*.

multilayer perceptron. Por exemplo, na rede mostrada na figura 3.8, caso todos os pesos das entradas para o primeiro neurônio sejam zero e caso todos os pesos de realimentação sejam iguais a zero, a rede irá se transformar numa rede *multilayer perceptron* com duas entradas, dois processadores na camada escondida e um processador de saída.

O cromossomo clássico usado para fazer o treinamento desta rede contém todos os pesos necessários para todas as conexões da rede neural que se quer treinar. O cálculo do número de genes necessários para o cromossomo que representa os pesos da rede neural pode ser feito através da equação 3-7:

$$num\_genes = N_e \times N_p + N_p^2 + N_p \tag{3-7}$$

Onde N_e indica o número de entradas e N_p indica o número de processadores. O primeiro termo desta equação calcula o número de pesos necessários para realizar as ligações entre as entradas e os processadores. O segundo termo, calcula o número de pesos necessários para realizar as ligações recorrentes entre os processadores e o último termo é o número de *biases* necessários (um para cada processador).

Assim, o indivíduo quântico necessário para gerar os indivíduos clássicos, deve ser inicializado de modo que a função densidade de probabilidade permita que os valores dos genes clássicos observados estejam dentro de uma faixa aceitável como pesos para uma rede neural. Neste trabalho, em todos os estudos de caso mostrados no capítulo 4, o domínio usado para os pesos está no intervalo $[-1, 1]$. Estes valores para os limites inferior e superior do intervalo foram encontrados

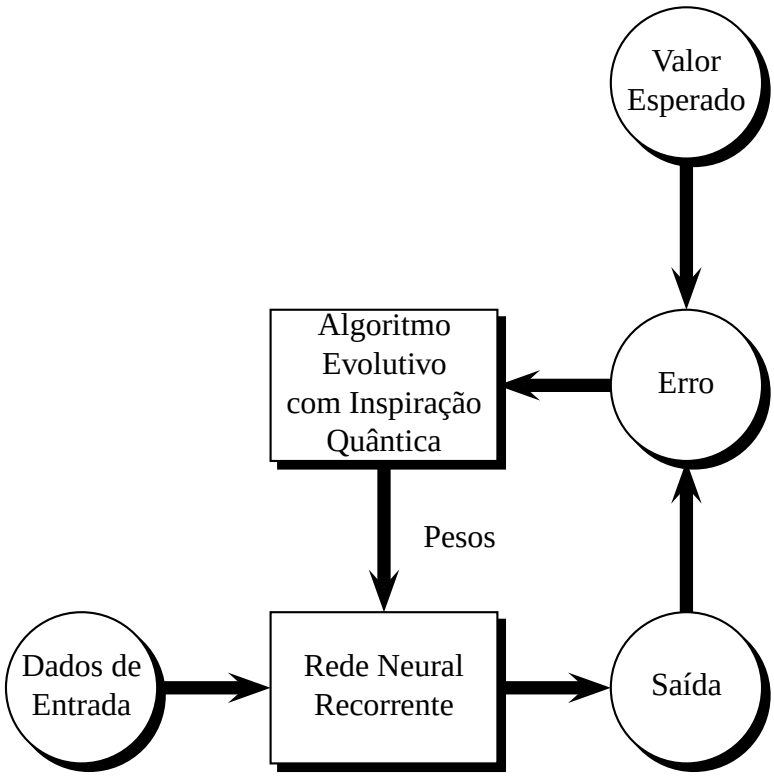

Figura 3.9: Modelo de Aprendizado Supervisionado usando o AEIQ–ℝ.

experimentalmente e podem variar de problema para problema.

Cada cromossomo clássico gerado deve, naturalmente, ser avaliado. Esta avaliação é feita construindo-se a rede neural recorrente com os pesos determinados pelo indivíduo clássico. Os padrões de entrada são então apresentados e a saída obtida pela rede é comparada com as saídas desejadas. Esta comparação permite que se faça o cálculo do erro médio, que pode ser usado como o valor da avaliação do indivíduo clássico em questão.

Na figura 3.9 é mostrado um diagrama de blocos, que ilustra o funcionamento do modelo de aprendizado supervisionado para redes neurais recorrentes usando o AEIQ–ℝ.

O número de neurônios e a topologia de uma rede neural são dois importantes parâmetros na configuração da mesma. Em geral, a topologia deve ser determinada empiricamente ou pela experiência pessoal do usuário. Quando se usam algoritmos como o *backpropagation* (Haykin99) para realizar o treinamento das redes, este parâmetro pode afetar seriamente o desempenho das mesmas. Processadores em excesso podem fazer com que a rede memorize o conjunto de treinamento ao invés de aprender a função que mapeia as entradas nas saídas, resultando em uma má capacidade de generalização por parte da mesma. Por outro lado, o uso de poucos

processadores pode retardar o processo de aprendizado ou até mesmo inviabilizá-lo.

Para se contornar este problema, o modelo de treinamento usando o AEIQ–ℝ usa um método de aprendizado descrito em (Gomez03). Neste método, um operador especial chamado *lesion* é utilizado durante o processo de aprendizado. Este operador funciona da seguinte forma: quando a evolução pára de progredir por um certo número de gerações (este número de gerações η é definido pelo usuário), a melhor rede encontrada é reavaliada, removendo cada um de seus neurônios, um por vez. Se a avaliação desta nova rede não piorar mais do que um determinado nível após a remoção do neurônio *i*, pode-se considerar que este neurônio não é importante para a rede e o mesmo é removido. Se nenhum neurônio puder ser removido, isto pode significar que a rede está precisando de mais neurônios para aprender corretamente. Sendo assim, mais um neurônio é inserido na rede, com todos os pesos inicialmente iguais a zero. A inserção de neurônios é feita com peso igual a zero, para que isso não altere a avaliação dos indivíduos imediatamente (zerar os pesos da entrada e da saída de um neurônio equivale a remover o neurônio da rede). Além disso, conforme já foi mencionado anteriormente, dado que a rede neural recorrente pode se transformar em uma rede com múltiplas camadas, conclui-se que, além do número de neurônios, o número de camadas da rede neural também será otimizado automaticamente, de acordo com os pesos que forem encontrados para determinadas ligações entre os processadores.

Através do uso deste operador e da capacidade do processo de treinamento de encontrar a topologia mais adequada automaticamente, espera-se que o AEIQ–ℝ seja capaz de otimizar os pesos da rede neural com um tempo computacional muito menor, já que através deste método de aprendizado, não é necessário realizar o processo de identificação do número ideal de processadores da rede, o que exige um esforço computacional relativamente alto em relação ao processo de aprendizado da rede como um todo.

Finalmente, é importante destacar que, foi observado durante os experimentos que, o fato de se usar o AEIQ–ℝ como método de treinamento, torna desnecessário o uso de um conjunto de dados para validação da rede neural. De acordo com (Bishop95), devido ao fato de as redes com treinamento bayesiano usarem distribuições de probabilidade para a determinação dos pesos, a otimização dos valores ideais para os coeficientes de regularização (os coeficientes de regularização são usados para minimizar a ocorrência de super-treinamento) da rede pode ser feita sem a necessidade do uso de conjuntos de validação. Como o AEIQ–ℝ também usa distribuições de probabilidade para encontrar estes pesos, pode-se concluir que este método de aprendizado também não necessita do uso destes conjuntos de validação para o aprendizado.

3.5
Modelo Neuro-Evolutivo – Aprendizado por Reforço

Nesta seção, pretende-se apresentar um modelo de aprendizado por reforço, onde se usa redes neurais para aprender a realizar tarefas de controle e para as quais o reforço só é fornecido quando o agente alcança o estado final ou o estado de falha. Neste caso, o objetivo é treinar a rede para que a mesma seja capaz de levar um determinado sistema de um estado a outro (por exemplo, equilibrar um pêndulo, regular a temperatura de um ambiente, entre outros casos).

Ao contrário do modelo de aprendizado supervisionado, neste caso não existem dados de saída para comparação e cálculo do erro, mas sim um estado (ou um conjunto de estados) finais que se deseja alcançar. A rede neural então lê o estado inicial deste sistema, e dá uma resposta na sua saída. Essa resposta é usada pelo atuador para mudar o estado do sistema que então será usado como nova entrada para a rede neural. Esse processo é repetido sucessivamente até que a rede neural consiga levar o sistema ao estado final desejado ou que um certo número de passos seja executado. O diagrama da figura 3.10 resume o funcionamento deste modelo.

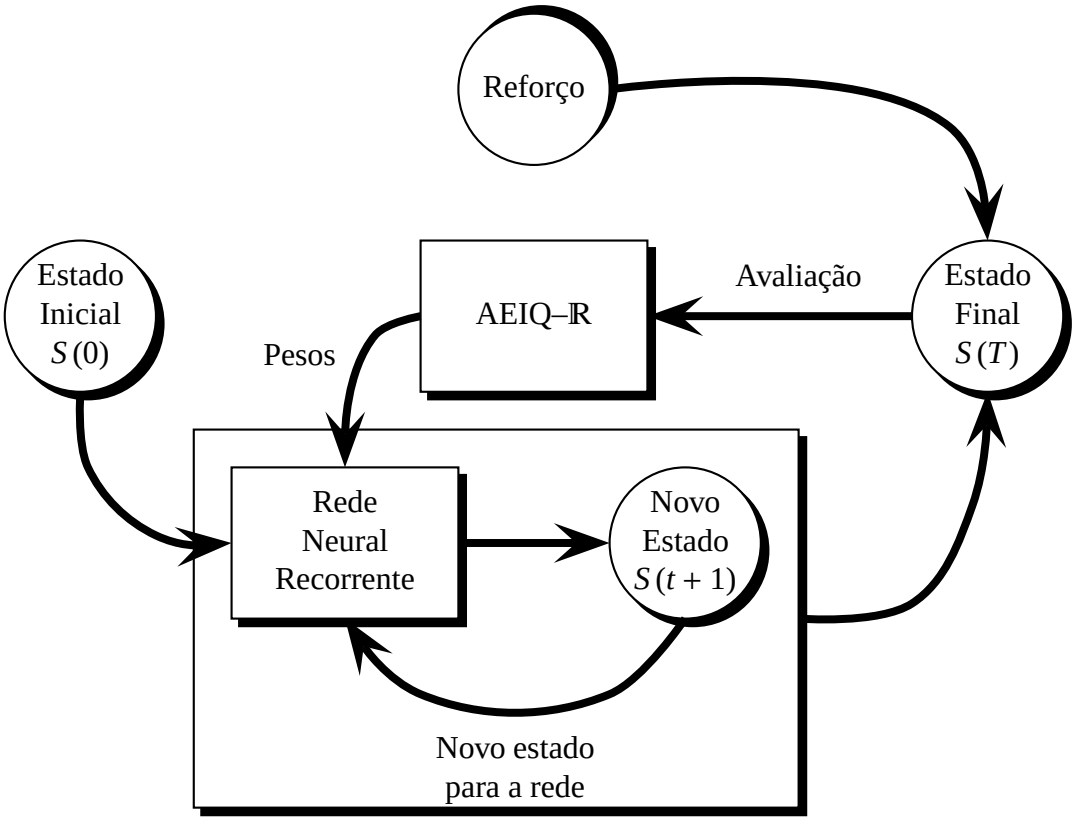

Figura 3.10: Modelo de Aprendizado por Reforço usando o AEIQ–ℝ.

Neste diagrama, pode-se observar que, a partir de um estado inicial, a rede neural é usada para gerar uma saída que indica qual ação o sistema deve tomar. Esta ação muda o estado corrente do sistema e este novo estado é usado como entrada

para a mesma rede neural. Uma nova saída é gerada e desta forma o processo é repetido continuamente até que um certo número limite de passos tenha sido executado ou até que o sistema atinja o estado final desejado. O sistema é então avaliado pelo AEIQ–ℝ (em geral, usando-se o número de passos necessários para se atingir o objetivo) que, por sua vez, através do processo de evolução, gera um novo conjunto de pesos para a rede neural. Estes novos pesos são então usados para repetir o processo de controle novamente.

Este modelo de aprendizado usa o mesmo método de treinamento apresentado na seção anterior. Assim, o algoritmo de aprendizado também é capaz de definir a topologia e o número de neurônios necessários para a rede neural.