

2. Estado da Arte

2.1. Computação Quântica

Um computador quântico é um dispositivo capaz de efetuar operações lógicas utilizando propriedades da mecânica quântica, tais como sobreposição e interferência. A principal vantagem destes computadores é a viabilização da execução de alguns problemas que através de computadores clássicos seriam inviáveis em tempo de processamento. Além de dispositivos quânticos, estes computadores necessitam de algoritmos que sejam capazes de tirar proveito destas propriedades quânticas. Podem-se citar alguns algoritmos mais comuns como o *Shor* (em Han e Kin *apud* Shor [4], Abs da Cruz [2], e Moore e Narayanan [24]), e o *Grover* (em Han e Kin *apud* Grover [4]).

Segundo Abs da Cruz [2], pesquisas na área de computação quântica se concentram em duas partes:

- Desenvolvimento de hardwares que viabilizem o uso de computadores quânticos;
- Desenvolvimento de novos algoritmos que sejam mais eficientes nos computadores quânticos, do que os algoritmos equivalentes em computadores clássicos.

No algoritmo Shor, o principal objetivo é a fatoração em números primos de números naturais. O algoritmo deverá encontrar dois números primos, cujo produto seja igual ao número fornecido pelo usuário. Entre outras palavras, seja C um número fornecido pelo usuário, e A e B dois *números primos*, o algoritmo deve encontrar A e B tal que: $A \times B = C$. Desta forma, quanto maior for este valor fornecido para fatoração, maior o tempo de processamento necessário através de um computador clássico para resolver este tipo de problema, crescendo exponencialmente o nível de dificuldade. Já em um computador quântico, segundo Abs da Cruz, esta tarefa se resolveria com pouco esforço computacional.

Em mecânica quântica, é possível que uma partícula esteja em dois ou mais estados ao mesmo tempo. Uma famosa metáfora denominada o gato de Schrödinger expressa esta realidade: imagine que um gato está dentro de uma cai-

xa, com probabilidades a priori de 50% de chances de estar vivo e 50% de chances de estar morto. Para a mecânica quântica, até abrir-se a caixa e verificar como está o gato, ele deve ser considerado vivo e morto ao mesmo tempo. A esta capacidade de estar simultaneamente em vários estados chama-se *superposição*.

Esta propriedade de superposição de estados faz com que dispositivos quânticos sejam exponencialmente mais eficientes que um dispositivo clássico. Desta forma, um computador quântico passar a ser preferível no lugar de um computador clássico, pelo grau de eficiência que este apresenta. Conforme se observa em Abs da Cruz [2] *apud* Rieffel:

“Computação quântica torna-se mais eficiente, pois se utiliza de conceitos de paralelismo, de tal forma que todos os espaços de busca de um problema passam a ser considerados, mesmo que estes não venham a ser utilizados.”

Computadores clássicos interpretam impulsos elétricos, positivos ou negativos, através de representações binárias, 1 ou 0, respectivamente. A cada impulso elétrico, dá-se o nome de bit (*em inglês*, BInary digiT). Um bit é a menor unidade de informação em computadores clássicos. Em computadores quânticos, a menor unidade de informação é chamada de q-bit (ou quantum bit). Um q-bit pode representar um estado 1, um estado 0, ou uma superposição de ambos estados. Um estado Ψ de um q-bit, segundo Han e Kin [4], pode ser representado por:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (2-1)$$

Ou seja, o bit é substituído pelo bit quântico, o q-bit, e os valores 0 e 1 de um bit são substituídos pelos vetores $|0\rangle$ e $|1\rangle$, representador por:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2-2)$$

Essa notação utilizada em mecânica quântica é conhecida por *notação de Dirac*. No formalismo, o estado de um sistema num dado instante de tempo pode ser representado por uma função complexa das posições ou dos momentos de cada partícula que compõe o sistema. Essa representação é chamada função de onda. O termo função de onda segundo a mecânica quântica tem um significado bastante diferente dependendo do contexto, seja na física clássica, seja no eletromagnetismo clássico.

Por causa da relação concreta entre função de onda e localização de uma partícula num espaço de posições, muitos textos sobre mecânica quântica têm um enfoque "ondulatório". Assim, embora o termo "função de onda" seja usado como

sinônimo "coloquial" para vetor de estado, não é recomendável, já que não só existem sistemas que não podem ser representados por funções de onda, mas também que o termo função de onda leva a imaginar erroneamente que há algum meio que ondula no sentido mecânico. O uso moderno do termo função de onda é para qualquer vetor ou função que descreva o estado de um sistema físico pela expansão em termos de outros estados do mesmo sistema, tal como acontece com a função Ψ conforme dado em (2-1).

A interpretação de um q-bit conforme função de onda (2-1) é que ele esta simultaneamente nos estados "0" e "1". Isto faz com que a quantidade de informação armazenada no estado Ψ possa ser infinita. Entretanto, esta quantidade infinita de informação esta no nível quântico. Para torná-la observável no nível clássico, precisa-se fazer uma medida.

Conforme se observa em Motta [25], a mecânica quântica diz que o processo de medida altera o estado de um q-bit, fazendo-o assumir o estado "0", com probabilidade α^2 , ou o estado "1", com probabilidade β^2 . A esta propriedade da mecânica quântica dá-se o nome de amplitude de probabilidade¹. Tem-se então que em um q-bit, os possíveis estados de um sistema serão representados através do conceito de amplitude de probabilidade. Vem-se que α e β são números complexos que especificam a amplitude de probabilidade dos estados 0 e 1, de tal forma que quando normalizados, tem-se:

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2-3)$$

Sendo assim, a menor unidade de informação de um computador quântico, um q-bit, poderia ser representado em um computador clássico como um par de números (α, β) da seguinte forma:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \quad (2-4)$$

No entanto, no momento que uma solução é observada em um destes subespaços de soluções representado por (2-4), todos os outros estados e consequentemente possíveis soluções deixam de existir, não sendo possível recuperá-los. En-

¹ Em mecânica quântica, uma amplitude de probabilidade é um número complexo cujo módulo representa uma probabilidade ou densidade de probabilidade. Por exemplo, os valores tirados por uma função de onda normalizada ψ são amplitudes, uma vez que $|\psi(x)|^2$ dará a densidade de probabilidade na posição x . Amplitudes de probabilidade podem também corresponderem a probabilidades de resultados discretos.

tre outras palavras, a normalização dos estados garantirá que através de algum processo de medição do estado, deverá se obter um ou outro estado, não sendo possível recuperar a informação na forma quântica.

Conforme Abs da Cruz [2], a representação das amplitudes de probabilidade para um q-bit qualquer pode ser feita através de um círculo de raio unitário, para o caso em que α e β são reais, conforme figura 2.1.

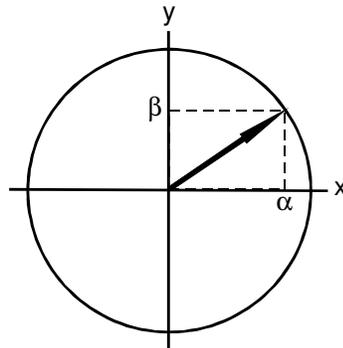


Figura 2.1 – Representação gráfica de um círculo de raio unitário com as amplitudes de probabilidade de se observar os valores 0 e 1 para um q-bit qualquer, no caso em que α e β são reais.

Na mecânica quântica, uma representação geométrica do espaço de estados num sistema de dois níveis como o apresentado em (2-1), pode ser obtida conforme figura a seguir. Proposta por Felix Bloch, e assim nomeada “esfera de Bloch”, trata-se de uma esfera abstrata com extremos opostos representando os estados $|0\rangle$ e $|1\rangle$.

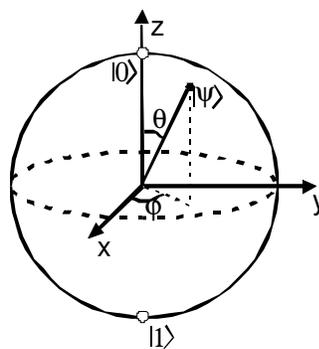


Figura 2.2 – Esfera de Bloch.

Outra característica de um computador quântico é a necessidade de algoritmos quânticos, os quais o desenvolvimento não é fácil. “E se não existem tais algoritmos, estes computadores serão como hardware sem softwares, e, portanto, inutilizáveis” (Han e Kin [4]).

Computação evolutiva, por sua vez, tem ajudado pesquisadores a desenvolver novas formas e métodos de programação. Segundo Han e Kin [4], estudos na área de computação evolutiva e computação quântica surgiram na década de 90, e se concentram em duas áreas específicas:

- (1) na geração de novos algoritmos quânticos usando técnicas de programação automática como programação genética;
- (2) no estudo de algoritmos inspirados na física quântica, para computadores clássicos.

Esta última área de concentração é um ramo de estudo da computação evolutiva com o objetivo de criar algoritmos evolutivos com inspiração na física quântica, que neste trabalho de dissertação passará a ser mais interessante.

2.2. Algoritmos com Inspiração Quântica

O termo “inspiração quântica” é utilizado para distinguir procedimentos computacionais puramente quânticos, os quais são fielmente baseados e fundados nos princípios e conceitos de mecanismos quânticos, de outros somente inspirados nestes mesmos conceitos e princípios, mas que de outra forma são mais práticos e úteis para se programar (Moore e Narayanan [24]). Na prática, procedimentos com inspiração quântica servem para distinguir os que funcionariam unicamente em computadores clássicos, dos próprios para computadores clássicos. Esta “inspiração” é similar ao que ocorre em outras áreas da inteligência computacional, tais como redes neurais e algoritmos genéticos. Redes neurais são inspiradas no funcionamento do cérebro, enquanto que algoritmos genéticos na biologia evolutiva darwiniana.

Neste trabalho, o que será apresentado é um algoritmo evolutivo inspirado na física quântica. Segundo Moore e Narayanan [24], computação inspirada na física quântica pode ser caracterizada por dois aspectos principais:

- Inspiração em alguns princípios de ondas estacionárias, interferência e coerência;
- O uso de algoritmos clássicos para checar se as soluções candidatas geradas pelo algoritmo com inspiração quântica são soluções válidas.

Alguns exemplos podem ser vistos em Moore e Narayanan [24] como uma extensão inspirada na física quântica do algoritmo Shore (algoritmo puramente quântico), desenvolvido para o problema de fatoração de números primos. Observa-se também em Moore e Narayanan a apresentação de outros dois algoritmos inspirado na física quântica: o algoritmo de ordenação; e o enigma 15.

O algoritmo de ordenação consiste em ordenar ascendentemente ou descendentemente uma lista de pelo menos dois elementos numéricos. Desta forma, uma ordenação natural em um computador clássico consistiria em:

- a) Considerar uma lista L1 para ser sorteada e inicializar uma segunda lista vazia L2 de elementos sorteados;
- b) No caso de uma ordenação ascendente, encontrar o menor elemento em L1 e copiá-lo em L2, para depois excluir este elemento de L1;
- c) Repetir os passos a) e b) até que L1 esteja vazia, e L2 contenha todos os elementos em ordem crescente.

O tempo de processamento necessário para esta ordenação detalhada anteriormente segundo Moore e Narayanan *apud* Knuth (1973) seria da ordem de $(1/2)*N(N-1)$. Considerando uma lista de 16 números para ser sorteada com elementos (16, 14, 12, 3, 8, 4, 15, 7, 11, 6, 2, 5, 1, 10, 9, 13), é equivalente dizer que em média seriam necessárias 120 seleções até que toda a ordenação fosse feita. Uma extensão deste algoritmo agora com inspiração na física quântica forneceria algum ganho em eficiência e é apresentada por Moore e Narayanan.

- a) Seja $N = 16$, o número de elementos em uma lista; seja u o número de universos igual à raiz de N ; seja ℓ o número de elementos em cada universo². Separando a lista N em u universos, e ordenando-se em cada um, se obteria:

u_0	3	12	14	16
u_1	4	7	8	15
u_2	2	5	6	11
u_3	1	9	10	13

Tabela 2.1 – Matriz de elementos ordenados por universo.

² Se o número de elementos n não fosse quadrático, duplicaríamos elementos aleatoriamente até que se obtivesse um número de elementos total quadrático.

- b) Realiza-se uma interferência diagonal da seguinte forma: cada universo é reorganizado com a diagonal que começa a partir do elemento da primeira coluna, até que se complete ℓ elementos na última coluna. Desta forma, o primeiro universo irá conter a diagonal principal, o segundo a diagonal (4, 5, 10, 16), o terceiro a diagonal (2, 9, 14, 15), e o último (1, 12, 8, 11). Ordene crescentemente os elementos de cada universo e então se tem:

u_0	3	6	7	13
u_1	4	5	10	16
u_2	2	9	14	15
u_3	1	8	11	12

Tabela 2.2 – Matriz de elementos ordenados por universo após interferência diagonal.

- c) Em seguida, uma interferência vertical é executada. Todos os 4 elementos de cada coluna são transpostos, e ordenados crescentemente dentro de cada universo. Desta forma se obtém:

u_0	1	2	3	4
u_1	5	6	8	9
u_2	7	10	11	14
u_3	12	13	15	16

Tabela 2.3 – Matriz de elementos ordenados por universo após interferência vertical.

- d) No final de cada dupla interferência diagonal e vertical, uma verificação é feita se o último elemento de cada u_i é menor que o elemento de u_{i+1} . Se não para algum u_i , uma nova dupla de interferência é executada até que seja satisfeita a condição de ordenação.

Esta inspiração quântica existe, pois, como se pode observar, o algoritmo apresentado por Moore e Narayanan utiliza conceitos de múltiplos universos (u_1, u_2, u_3, u_4), paralelismo (cada universo existindo em paralelo), interferência (atuações na dinâmica de cada universo), e uma checagem clássica final (conforme passo d) para averiguar se o resultado obtido é uma solução válida.

O tempo de processamento deste algoritmo com inspiração quântica, por sua vez, é da ordem de

$$\sqrt{N} \left[N + 2(\sqrt{N} - 1) \right], \quad (2-5)$$

onde N , da mesma forma, igual ao número de elementos a ser sorteado. Neste caso, para os 16 elementos considerados, se necessitaria, em média, 88 seleções, ou

seja, 26% a menos com relação ao algoritmo de ordenação clássico. Este foi apenas um exemplo de como algoritmos tradicionais em computadores clássicos, se inspirados na física quântica, podem levar a um ganho de eficiência em tempo de processamento.

Tratando-se de computação evolutiva com inspiração na física quântica, faz-se algum destaque nas seções a seguir para dois algoritmos desenvolvidos até então. O primeiro é o AEIQ-B (Han e Kin [4]), onde a menor unidade de informação é representada por um Q-bit, numa analogia a um bit quântico de um computador quântico. E o segundo, o AEIQ-R (Abs da Cruz [2]), cuja menor unidade de informação será representada por um Q-real, conceito este ainda a ser apresentado, mas que diferentemente de um Q-Bit, os estados representados serão valores num domínio de uma função contínua. Mais detalhes de ambos os algoritmos em 2.2.1 e 2.2.2.

2.2.1. O Algoritmo AEIQ-B

Nosso objetivo nesta seção será detalhar algumas das funcionalidades do algoritmo AEIQ-B, tais como: forma de representação do cromossoma, operadores quânticos, e pseudocódigo.

A sigla AEIQ-B [8] é a denominação para “algoritmo evolutivo com inspiração quântica em representação binária”. Foi introduzido por Han e Kin como sendo um novo método de computação evolutiva para resolução do problema benchmark denominado “knapsack” [26]. Tal problema consiste basicamente de um caso de otimização combinatória. Dado um conjunto de itens, cada um com um valor e peso específico, o objetivo é selecionar um conjunto de itens em uma cesta, de tal forma que a soma total dos pesos dos itens selecionados fosse inferior a um limite especificado, e a soma dos valores fosse a máxima possível.

Desde a apresentação do algoritmo AEIQ-B em [8], Han et al têm publicado diversos outros estudos (ver [1], [3], [4]), sempre na otimização de problemas combinatórios como o “knapsack”, ou ainda, no reconhecimento de faces. Não somente Jang, Kin, Kim e Han, mas como também outros autores estão reconhecendo o potencial de aplicabilidade do algoritmo AEIQ-B, e novas aplicações têm surgido, tais como: caixeiro viajante em Talbi et al [6]; o enigma das n-rainhas em Talbi et al [6]; registro de imagens em Talbi et al [6]; segmentação de imagens em

Talbi et al [6]; emissão de poluentes em Arpaia et al [9]; o problema da mochila em Li et al [5] e Mahdabi et al [7].

Em todos estes problemas, o algoritmo AEIQ-B era comparado com os algoritmos genéticos tradicionais, com alguma particularidade ou não na formulação para problemas multi-objetivos. Existiam muitas variações do algoritmo AEIQ-B na comparação dos estudos de algum autor em específico, sempre com a modificação de alguma ou outra funcionalidade, como é o caso Mahdabi et al em [7], com o algoritmo MSQIEA comparando-o com o QEA (denominação em inglês para AEIQ-B). Em Jang, Han e Kin [1], mencionam-se outros algoritmos classificadores como o ML, e o DFFS, específicos para o seu problema em estudo, a detecção de faces. Sobre a comparação do AEIQ-B no geral com outros algoritmos, conclui-se que a maioria dos estudos visa contribuições à própria literatura de AEIQ e AG no geral.

Em 2.1 concluiu-se que a menor unidade de informação que um computador quântico pode ter chama-se q-bit, e é representado por uma dupla de números complexos conforme (2-1) a (2-4). Desta forma se houvesse um sistema com m q-bits, este sistema poderia representar simultaneamente 2^m padrões simultaneamente, devido à representação probabilística e superposição de todos os estados. No entanto, o ato de observar um destes estados provocaria o colapso, ou perda total, de todos os demais estados. Sendo assim, analogamente a representação de um q-bit, um indivíduo quântico com m q-bits poderia ser representado por:

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix} \quad (2-6)$$

onde $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, 2, \dots, m$.

Desta forma, o indivíduo acima poderia representar todos os 2^m possíveis estados simultaneamente. Para fins de exemplificação, suponha que os valores dos q-bits de um indivíduo com 3 genes sejam dados conforme abaixo:

$$q = \begin{bmatrix} 1/\sqrt{2} ; 1; & \frac{1}{2} \\ 1/\sqrt{2} ; 0; & \sqrt{3}/2 \end{bmatrix} \quad (2-7)$$

Desta forma, o estado de um q-bit poderia ser representado por:

$$|\psi\rangle = \frac{1}{2\sqrt{2}} |000\rangle + \frac{\sqrt{3}}{2\sqrt{2}} |001\rangle + \frac{1}{2\sqrt{2}} |100\rangle + \frac{\sqrt{3}}{2\sqrt{2}} |101\rangle \quad (2-8)$$

E a probabilidade de se observar cada um dos estados é dado pela tabela 2.4.

Estado	Probabilidade
$ 000\rangle$	$(1/\sqrt{2})^2 \cdot (1)^2 \cdot (1/2)^2 = 1/8$
$ 100\rangle$	$(1/\sqrt{2})^2 \cdot (1)^2 \cdot (1/2)^2 = 1/8$
$ 010\rangle$	$(1/\sqrt{2})^2 \cdot (0)^2 \cdot (1/2)^2 = 0$
$ 001\rangle$	$(1/\sqrt{2})^2 \cdot (1)^2 \cdot (\sqrt{3}/2)^2 = 3/8$
$ 110\rangle$	$(1/\sqrt{2})^2 \cdot (0)^2 \cdot (1/2)^2 = 0$
$ 101\rangle$	$(1/\sqrt{2})^2 \cdot (1)^2 \cdot (\sqrt{3}/2)^2 = 3/8$
$ 011\rangle$	$(1/\sqrt{2})^2 \cdot (0)^2 \cdot (\sqrt{3}/2)^2 = 0$
$ 111\rangle$	$(1/\sqrt{2})^2 \cdot (0)^2 \cdot (\sqrt{3}/2)^2 = 0$

Tabela 2.4 – Exemplo de obtenção das probabilidades de um q-bit.

Isto significa dizer que um único q-bit com 3 genes representaria ao mesmo tempo todos os 8 possíveis estados de um suposto cromossoma binário, também com 3 genes.

A seguir, o pseudocódigo do algoritmo AEIQ-B:

Algoritmo AEIQ-B

Inicia

$t = 0$

Inicializa a população quântica $Q(t)$ com representação em q-bits.

Gera população clássica $P(t)$ com representação binária, observando $Q(t)$.

Avalia $P(t)$.

Faça $B(t) = P(t)$, onde $B(t)$ é a população dos melhores indivíduos clássicos.

Enquanto (condição de parada) faça

$t = t+1$

Faça $P(t)$ observando $Q(t-1)$.

Avalia $P(t)$.

$B(t) =$ Melhores indivíduos de $[P(t) \cup B(t-1)]$.

Atualiza $Q(t)$ usando operadores quânticos $U[t]$.

Fim

Fim

Figura 2.3 – Pseudocódigo do algoritmo evolutivo com inspiração quântica e representação binária.

O algoritmo AEIQ-B é similar a um algoritmo genético utilizando-se de: (1) representação de soluções candidatas em cromossomas; (2) operadores de cru-

zamento de indivíduos; e (3) otimização evolutiva. Sobre forma de representação, uma população de indivíduos quânticos de tamanho n , num instante de tempo t , poder-se-ia representada da seguinte forma:

$$Q(t) = [q_1^t \ q_2^t \ \dots \ q_n^t], \quad (2-9)$$

onde cada q_i^t , $i=1, \dots, n$, é dado conforme (2-6).

Na prática, a cada passo evolutivo t , q-bits de um indivíduo quântico q_i^t serão atualizados de tal forma que convergirão para uma probabilidade máxima igual a 1 para algum dos estados $|0\rangle$ ou $|1\rangle$, indicativamente o estado do melhor indivíduo possível. *Este processo de convergência é feita através de operadores quânticos $U[t]$, e funciona como uma atualização de cada gene, de cada indivíduo, em cada instante de tempo t .*

Segundo Abs da Cruz [2], um *operador* é um objeto matemático que age sobre uma função, transformando-a em outra função. Um exemplo é o operador ∂_x , que realiza a operação de diferenciar com respeito à variável x , transformando uma função $f(x)$ em sua derivada.

Antes de se apresentar o operador quântico – ou mais comumente dito *porta quântica* – utilizado por Han no AEIQ-B, apresentar-se-á alguns conceitos de circuitos quânticos. A importância do estudo de *portas* lógicas em computação *quântica* baseia-se no fato de que toda matriz unitária 2×2 pode ser representada por um circuito quântico de 1 q-bit e vice-versa. Sendo assim, a evolução no tempo de um circuito quântico isolado, dado por um q-bit, pode ser representado tanto matematicamente (por uma transformação unitária) quanto logicamente (por um circuito quântico).

Sendo assim, conforme se pode observar em Portugal [27], considere a seguinte representação para um circuito quântico em que a entrada e a saída são um estado de 2 q-bits conforme figura 2.4.

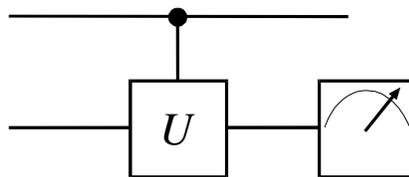


Figura 2.4 – Porta quântica U controlada.

- Entrada: pode ser o produto tensorial entre os q-bits de entrada ou um estado emaranhado (os q-bits não devem ser considerados individualmente).

- Linhas horizontais: não são necessariamente fios. Elas representam a evolução de um q-bit, podendo ser apenas a passagem do tempo ou, por exemplo, o deslocamento de um fóton.
- Sentido: o circuito descreve a evolução do sistema quântico no tempo, da esquerda para a direita. Com isso, não há sentido em aparecer retroalimentação que poderia ocorrer em um circuito clássico.
- Linhas verticais: o segmento vertical que aparece unindo os símbolos \bullet e \boxed{U} informa que o circuito atua simultaneamente nos dois q-bits. A linha vertical representa o sincronismo, e não o envio da informação. Portanto, não são permitidas nem junções, nem bifurcações de q-bits.
- Controle: o símbolo \bullet indica que o q-bit representado nessa linha é um q-bit de controle, ou seja, caso esteja no estado $|1\rangle$ a porta U realiza a operação, caso esteja no estado $|0\rangle$, a porta U não realiza operação alguma.
- Saída: os q-bits que compõem a saída do circuito podem ou não ser medidos. Como o q-bit inferior está sendo medido conforme símbolo de medida na figura 2.4, o resultado será 0 ou 1.

Existem diversos tipos de portas quânticas conforme se observa ainda por Portugal [27]. Algumas delas são: (1) porta NOT Quântica; (2) porta de Hadamard; (3) porta de Fase ou porta S; (4) porta $\pi/8$ ou porta T; (5) porta CNOT Quântica; (6) e a porta Toffoli Quântica. Prosseguir-se-á com a exemplificação da mais simples das portas apresentadas, a *porta NOT Quântica*.

Neste caso, a porta NOT Quântica trocará 1 por 0 e vice-versa. A generalização para o caso quântico é dada por um operador X que satisfaz

$$X|0\rangle = |1\rangle \text{ e } X|1\rangle = |0\rangle. \quad (2-10)$$

Com isso, verifica-se facilmente que a representação matricial do operador X é dada por

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (2-11)$$

Com a porta NOT Quântica, temos situações sem contrapartida no caso clássico, pois, se a entrada $|\psi\rangle$ for uma superposição dos estados $\alpha|0\rangle$ e $\beta|1\rangle$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

a saída será

$$X|\psi\rangle = \beta|0\rangle + \alpha|1\rangle. \quad (2-12)$$

A porta NOT Quântica com o operador X, anteriormente, é apenas uma das possíveis portas para 1 q-bit, já que há infinitas matrizes unitárias 2x2.

Han e Kin no algoritmo AEIQ-B em [8] utilizaram o operador do tipo *porta de Rotação* dado por:

$$U(\theta) = \begin{bmatrix} \cos(\theta) & (+/-)\text{seno}(\theta) \\ (+\text{ ou }-)\text{seno}(\theta) & \cos(\theta) \end{bmatrix}. \quad (2-13)$$

Onde θ um ângulo de rotação. E, pode-se demonstrar (ver Han e Kin [8]) que esta porta é uma matriz unitária. Esta última propriedade garantirá que o resultado obtido pela transformação do espaço de estados Ψ seja ainda normalizado em α e β .

No AEIQ-B esta porta quântica serve para atualizar o estado de um q-bit, conforme (2-1), de cada um dos genes de um indivíduo quântico. Segundo Han e Kin [8]:

“... este operador é mais apropriado, pois fará com que o Q-bit convirja gradativamente para os estados otimizados, sendo esta velocidade de convergência dada por θ .”

Isto não descarta o uso de outros tipos de portas quânticas. Por exemplo, segundo ainda Han e Kin, a porta NOT Quântica poderia ser utilizada para escapar de ótimos locais. Ou ainda, a porta NOT Controlado, que poderia ser utilizado para problemas onde há uma grande dependência de dois bits quânticos.

No AEIQ-B, a observação de um gene quântico para um computador clássico, por sua vez, acontecerá através de um evento de simulação. Uma vez que α e β são dados conforme (2-1), um experimento eficiente seria gerar um número aleatório entre 0 e 1, e caso este número esteja entre 0 e α^2 , atribui-se 0 à observação de um gene clássico. Caso contrário, atribui-se 1. Para isto, utilizar-se-ia genes equivalentes – ou seja, de mesma posição no cromossoma – de indivíduos da população $B(t)$.

Sendo assim, seja um q-bit q qualquer conforme (2-6), se através de alguma heurística se verificasse que a observação de genes clássicos tem sido mais promissora sempre que este mesmo q-bit gerar bits iguais a 0, o que se faria é atualizar q de tal forma que as chances de ser observar 0 sejam maior em $t+1$, numa nova geração da população quântica. Isto se faria através de:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{\text{nov}} = \begin{bmatrix} \cos(\theta) & \text{seno}(\theta) \\ -\text{seno}(\theta) & \cos(\theta) \end{bmatrix} \bullet \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2-14)$$

Uma heurística razoável para atualizar cada q-bit de um indivíduo, seria utilizar genes clássicos na mesma posição do cromossoma, de indivíduos selecionados da população $B(t)$. Caso o gene equivalente no indivíduo da população em $B(t)$ seja 1, atualizar-se-ia q de tal forma que a transformada em (2-1) atribua um peso maior ao número complexo β .

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{\text{nov}} = \begin{bmatrix} \cos(\theta) - \text{seno}(\theta) \\ \text{seno}(\theta) & \cos(\theta) \end{bmatrix} \bullet \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2-15)$$

À medida que um q-bit converge para uma probabilidade máxima, a diversidade representada no cromossoma desaparece. Através da relação trigonométrica $\cos^2(\theta) + \text{sen}^2(\theta) = 1$, pode-se provar que α_{nov} obtido de (2-14), o número complexo que especificará a probabilidade de se observar um gene igual a 0, será maior que α_{nov} obtido de (2-15). Enquanto que β_{nov} obtido em (2-15) será maior que β_{nov} em (2-14).

Tem-se então que, num instante inicial, a representação de α e β dado por $1/\sqrt{2}$ cada um, compreenderá a superposição de todos os estados, já que as probabilidades de se observar genes clássicos 0 ou 1 são iguais para todos os genes quânticos. Por estes motivos, diz-se que a representação em um q-bit é capaz de representar simultaneamente características de exploração com superposição de estados em instantes iniciais de t , e aproveitamento através da convergência das probabilidades para um estado máximo.

Sendo assim, através de um processo de observação de novos indivíduos, substituição do espaço de melhores soluções, e atualização e convergência da representação de indivíduos quânticos, obter-se-ia a solução otimizada para um determinado problema. Solução esta representada pelo melhor indivíduo em uma determinada população no passo t , após critério de parada de evolução ter sido alcançado.

2.2.2. O Algoritmo AEIQ-R

Abs da Cruz introduziu em [2] o algoritmo AEIQ-R, sigla para “Algoritmo Evolutivo com Inspiração Quântica e representação Real”. Motivado por limitações impostas pelo AEIQ-B quando da representação de genes *somente* binários,

Abs da Cruz desenvolveu um algoritmo com características semelhantes, mas que, por outro lado, seria fundamentado na representação de genes na forma real. Uma vez que problemas no mundo real muitas vezes são representados na forma de otimização numérica (e não combinatória), a proposição de Abs da Cruz viabilizaria um novo campo para aplicação de algoritmos com inspiração na física quântica.

Sendo assim, características como velocidade de convergência, compartilhamento de conhecimento entre indivíduos, representação probabilística do espaço de soluções, entre outras, seriam analogamente observadas também no algoritmo AEIQ-R.

Para analisar a eficácia do algoritmo proposto, Abs da Cruz utiliza-se de alguns problemas benchmark para controle, como é o caso do pêndulo invertido, o carro na montanha, e também *para o treinamento de redes neurais recorrentes para previsão de séries de dados temporais*. Outras aplicações envolviam a otimização de funções benchmark. Dependendo da aplicação, era disponibilizada uma comparação com outros algoritmos tais como: AEIQ-B; FEP; CEP; GA; RL-NFHB e RL-NFHP; Neural Q-Learning; CMAC Q-Learning; FQL; Q-Learning com MLP (Q-MLP); SARSA-CABA; SARSA-CMAC; SANE; CNE; NEAT; e ESP.

A aplicação por Abs da Cruz do algoritmo AEIQ-R no treinamento de redes neurais passa a ser mais interessante para nós, pois representava a primeira aplicação de algoritmos evolutivos com inspiração quântica no campo de neuroevolução e representação do cromossoma real. Na oportunidade, na otimização de redes do tipo Hopfield.

Nesta dissertação, diferentemente de Abs da Cruz, o modelo neuroevolutivo que se irá propor apresentará representação de cromossoma binário-real, aplicados à evolução de redes do tipo *alimentação adiante* e demais características a serem discutidas em capítulos seguintes.

Nesta seção, para se ter um comparativo paralelo com o algoritmo AEIQ-B introduzido na seção anterior, se discutirá brevemente também sobre a forma de representação, operadores quânticos, e pseudocódigo do algoritmo AEIQ-R. Se-que que o pseudocódigo pode ser representado conforme figura 2.5.

No AEIQ-B a representação de um gene era feita através de uma dupla de números complexos α e β . Esta representação era discreta, pois a observação de

um gene quântico só poderia gerar valores 0 ou 1. No algoritmo AEIQ-R esta representação deve ser contínua no domínio de cada gene numérico. Para que isto acontecesse, a solução encontrada por Abs da Cruz foi representar através de funções densidade de probabilidade cada gene quântico real do indivíduo quântico, que nesta dissertação serão denotados pelas siglas $p_{ij}(x)$ para a FDP e g_{ij} para o gene.

Algoritmo AEIQ-R

Inicia

$t = 1$

 Inicializa a população quântica $Q(t)$ com N indivíduos com G genes

Enquanto ($t \leq T$) **faça**

 Gera população clássica $P(t)$ com representação real, observando $Q(t)$.

Se ($t=1$) **faça**

 Faça $B(t) = P(t)$, $B(t)$ é a população dos melhores indivíduos clássicos.

Senão

$P(t)$ = Recombinação clássica entre $P(t-1)$ e $B(t-1)$

 Avalia $P(t)$.

$B(t)$ = Melhores indivíduos de [$P(t)$ U $B(t-1)$]

Fim

 Atualiza $Q(t)$ usando operadores quânticos mais apropriados.

$t = t+1$

Fim

Fim

Figura 2.5 – Pseudocódigo do algoritmo evolutivo com inspiração quântica e representação real.

Seja $Q(t)$ uma população quântica, em um instante de tempo t qualquer, formada por n indivíduos quânticos q_i ($i=1, \dots, n$). Cada indivíduo q_i desta população será formado por G genes g_{ij} ($j=1, \dots, G$), que por fim, serão representados por funções densidade de probabilidade. Sendo assim, um indivíduo q_i poderá ser descrito da seguinte forma:

$$q_i = [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), \dots, g_{iG} = p_{iG}(x)] \quad (2-16)$$

De tal forma que:

$$\int_{-\infty}^{+\infty} p_{ij}(x) dx = 1 \quad (2-17)$$

Sendo assim, através de $p_{ij}(x)$, uma função de densidade probabilidade (FDP) integrável em seu domínio para uma variável que se deseja otimizar, seria possível observar o gene desta representação quântica a partir da função distribuição cumulativa de probabilidade do gene. O que se faz para observar o gene de um indivíduo quântico é:

- (1) gerar um valor aleatório entre 0 e 1 (a imagem de qualquer FDP);
- (2) obter-se a inversa da função de distribuição cumulativa de probabilidade;
- (3) calcular a imagem da inversa (o domínio de x) para este número aleatório gerado entre 0 e 1.

Sendo assim, uma população $Q(t)$ com N indivíduos quânticos poderia ser descrita por:

$$Q(t) = [q_1^t, q_2^t, q_3^t, \dots, q_N^t] \quad (2-18)$$

Da mesma forma que em AEIQ-B, em AEIQ-R conforme figura 2.5, a atualização do gene quântico de um indivíduo em $Q(t)$ é feita utilizando indivíduos de $B(t)$. Isto fará com que cada gene de cada indivíduo q_i^t represente melhor regiões de busca mais promissoras.

Como dito sobre (2-17), $p_{ij}(x)$ será uma representação de um gene através de uma FDP. A função a ser escolhida dependerá do programador do algoritmo. No caso de Abs da Cruz, a função escolhida foi a *pulso quadrático* por motivos de simplicidade, já que esta encontra fundamentação na FDP do tipo uniforme, uma distribuição bivariada com geometria simples.

Contudo, poder-se-ia utilizar outros tipos de FDPs como, por exemplo, a *distribuição normal de probabilidade*. Se assim acontecesse, novos operadores de atualização quântica deveriam ser generalizados, pois os operadores utilizados por Abs da Cruz [2] não seriam aplicáveis.

Na função pulso quadrático temos que o domínio de x é definido no intervalo de a até b . Considerando-se um parâmetro de centro dado por $\mu=(a+b)/2$ e outro de largura dado por $\sigma=(b-a)$, a função pulso quadrático pode ser reescrita resumidamente conforme a seguir.

$$p_{ij}(x) = \begin{cases} \frac{1}{\sigma_{ij}}, & \text{se } (\mu_{ij} - \frac{\sigma_{ij}}{2}) \leq x \leq (\mu_{ij} + \frac{\sigma_{ij}}{2}) \\ 0 & \text{caso contrário} \end{cases} \quad (2-19)$$

Tecnicamente, isto significa dizer que cada gene quântico poderia ser representado através de μ (medida de centróide) e σ (medida de largura). Desta forma, a observação de um gene quântico de um indivíduo conforme (2-16) seria dada por um valor real no intervalo de $(\mu - \sigma/2)$ a $(\mu + \sigma/2)$.

Conforme se observa em Abs da Cruz, um exemplo de um gene quântico obtido através de um pulso quadrado é mostrado na figura a seguir. Considerando $\mu = 0$ e $\sigma = 1$, vêem-se que:

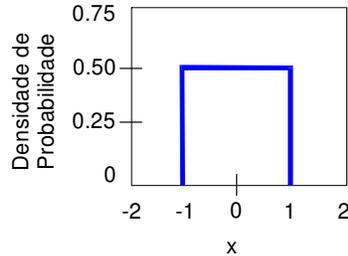


Figura 2.6 – Exemplo de um gene quântico do AEIQ-R.

Desta forma, uma atualização no gene quântico será equivalente a uma atualização nos parâmetros μ e σ do pulso quadrático. A heurística para determinar a atualização da largura σ de um gene seria dada pela regra do 1/5 conforme Abs da Cruz *apud* Michalewicz [2]: 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, tem-se que:

$$\sigma_{ij} = \begin{cases} \sigma_{ij} \cdot \delta & \varphi < 1/5 \\ \sigma_{ij} / \delta & \varphi > 1/5 \\ \sigma_{ij} & \varphi = 1/5 \end{cases} \quad (2-20)$$

Onde δ valor arbitrário no intervalo $]0,1[$ e φ é a taxa que indica quantos indivíduos da população clássica em t foram melhores do que os indivíduos na clássica em $t-1$.

No que diz respeito à medida de centro μ , o que se fazia era deslocá-la em direção a um ponto indicado pelo gene de um indivíduo clássico presente na população $B(t)$. Assim sendo, supondo-se que o centro do gene quântico na geração t seja dado por μ_{ij} e o valor do gene clássico seja dado por x_{ij} , então, a nova posição do gene quântico na geração $t+1$ seria dada por:

$$\mu_{ij}(t+1) = \mu_{ij}(t) + \lambda(x_{ij}(t) - \mu_{ij}(t)) \quad (2-21)$$

Onde λ um número em $[0,1]$, o percentual que se deseja deslocar o centro μ_{ij} na direção de x_{ij} . Com isto, o domínio de possíveis soluções para o problema a ser otimizado, representado através das funções distribuições de probabilidades dos genes quânticos, eram atualizadas a cada passo t do algoritmo AEIQ-R, de tal forma a representar melhor regiões com soluções mais promissoras para o problema.

Detalhamentos mais técnicos que ajudarão a compreender os conceitos desta seção serão apresentados no capítulo 3, quando da apresentação do algoritmo AEIQ-BR. Veremos que o AEIQ-BR, apesar de apresentar diversas singularidades no seu funcionamento, inegavelmente terá grande interseção com os algoritmos AEIQ-B e AEIQ-R aqui discutidos, e, desta forma, alguns pontos serão retomados e detalhados com maior profundidade.

A seguir, apresentar-se-á um estado da arte para modelos neuroevolutivos. Uma área de estudo para algoritmos evolutivos que nesta disserta será considerada para aplicação do AEIQ-BR, originando assim o algoritmo NEIQ-BR.

2.3. Modelos Neuroevolutivos

Algoritmos evolutivos (AE) têm sido utilizados para se evoluir RNA em contrapartida aos métodos tradicionais de treinamento baseados no gradiente decrescente e retro-propagação do erro. AE quando utilizados para otimização dos pesos sinápticos apresentam alguma vantagens, pois, se bem ajustados os parâmetros de evolução, evitam problemas de convergência prematura a um mínimo local e supertreinamento (em inglês, *overfitting*) da RNA. Isto acontece porque em AE não existe uma regra de aprendizado da rede neural explicitamente, mas sim um processo evolutivo de adaptação ao problema

Isto acontece, pois AE em RNA apresentam uma característica ímpar que é a capacidade de se adaptar a ambientes dinâmicos, conforme se observa em Yao [28]. Ainda segundo Yao, esta capacidade de evolução e aprendizado adaptativo faz com que a rede neural seja mais eficiente em sistemas com evolução de arquiteturas e regras de aprendizado de RNA, sem que a intervenção humana seja ne-

cessária. Por estes motivos, esta área denominada por “*algoritmos neuroevolutivos*” tem atraído diversos pesquisadores.

Muitos estudos têm sido realizados no campo de algoritmos evolutivos (A-Es) aplicados em redes neurais artificiais (RNA) conforme se pode observar pela tabela 2.5. Nossa pesquisa bibliográfica aponta para alguns tipos de AEs tais como: algoritmos genéticos (AGs); programação evolutiva (PE); e os inspirado na física quântica, AEIQ-R. Cruz [2], por exemplo, cita outros algoritmos como: programação genética, evolução diferencial, e algoritmos culturais.

São muitas as finalidades em se utilizar AEs na modelagem de uma rede neural. A tabela 2.5 fornece um resumo das diversas considerações feitas por outros autores, quando da representação dos genes dos cromossomas de seus AEs, e os objetivos que estes eram submetidos. Paz e Kamath em [19] dividem o interesse em se utilizar RNA em três grandes objetivos:

- (1) Obtenção dos pesos de uma rede neural através de AE;
- (2) Seleção de padrões mais relevantes para treinamento da rede;
- (3) Desenho de topologia neural ótima.

A evolução de uma rede através de AEs parece estar explicitamente associada à obtenção dos pesos sinápticos da rede, uma vez que todos os autores assim consideraram na representação do cromossomo. Sabe-se que a otimização dos pesos através de AEs é uma alternativa para obtenção de um ótimo global, e sobre isto, ressalta-se a frase de Paz e Kamath:

”Treinar uma rede neural é uma tarefa de otimização com o objetivo de encontrar uma melhor configuração de pesos que minimize alguma medida de erro. O espaço de busca é de grande dimensão e, dependendo da medida de erro e dos padrões de entrada, deverá conter diversos ótimos locais.”

Autor	Considerações na Representação do Gene									Tipo Rede	Algoritmo Evolutivo	Problema	Aplicação	Fonte
	Padrões de Entrada	Topologia Camadas Escondidas	Topologia Camada de Entrada	Pesos dos Neurônios	Conjunto	Co-evolução	Funções e Parâmetros de Ativação	Função de Avaliação	Regras e Parâmetros Aprendizado					
Andre et al (2007)	Não	Sim	Não	Sim	Não	Não	Não	Não	Não	Hopfield	QIAE-R	Previsão, Controle	Finanças, Robótica	[2]
Lacerda et al (2005)	Não	Não	Não	Sim	Não	Não	Sim	Não	Não	RBF	AGs	Classificação	Finanças	[10]
Zhan et al (2004)	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Perceptron	AGs	Classificação	Finanças	[11]
Pedrajas et al (2002)	Não	Sim	Não	Sim	Não	Sim	Não	Não	Não	Perceptron	AGs	Classificação	Finanças	[12]
Pedrajas et al (2005)	Não	Sim	Não	Sim	Sim	Sim	Não	Não	Não	Perceptron	AGs	Classificação	Física, Medicina	[13]
Pedrajas et al (2003)	Não	Sim	Não	Sim	Não	Sim	Não	Não	Não	Perceptron	AGs	Classificação	Finanças, Medicina	[14]
Abraham (2004)	Não	Sim	Sim	Sim	Não	Não	Sim	Não	Sim	Perceptron	AGs	Previsão	Física	[15]
Gomez et al (2008)	Não	Não	Não	Sim	Não	Sim	Não	Não	Não	Hopfield	AGs	Controle	Robótica	[16]
Blanco et al (2001)	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Hopfield	AGs	Controle	Linguística	[17]
Delgado e Pegalajar (2005)	Não	Sim	Não	Sim	Não	Não	Não	Não	Não	Hopfield	AGs	Controle	Linguística	[18]
Paz e Kamath (2005)	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Hopfield e SOM	AGs	Classificação	Finanças, Medicina, Física, Jogos	[19]
Capi e Doya (2005)	Não	Sim	Não	Sim	Não	Sim	Não	Não	Não	Hibridização	AGs	Controle	Robótica	[20]
Kim e Cho (2008)	Não	Sim	Sim	Sim	Sim	Não	Não	Não	Não	Hibridização	AGs	Classificação	Finanças, Medicina	[21]
Pedrajas e Boyer (2007)	Não	Sim	Não	Sim	Não	Sim	Não	Sim	Não	Hibridização	AGs e PE	Classificação	³	[22]
Sexton e Dorsey (2001)	Não	Não	Não	Sim	Não	Não	Não	Sim	Não	Perceptron	AGs	Classificação	Jogos, Medicina, Física	[29]

Tabela 2.5 – Resumo da literatura de neuroevolução.

³ Um total de 40 aplicações pôde ser encontrado em diversos campos da ciência. Sugerimos a leitura de [45] para mais detalhes.

Yao em [28] afirma que grandes bancos de dados aumentam o volume de dados necessário para se treinar uma rede, bem como o tempo necessário. Nestes casos, é possível que haja alguma redundância no banco de dados com variáveis apresentando valores iguais, ou bastantes semelhantes. Para que tais problemas possam ser contornados, é aconselhável que se faça um pré-processamento dos dados, com redução de dimensionalidade a partir de alguma técnica estatística. Ou ainda, é possível que no AE seja considerada no cromossomo uma parte para avaliar quais padrões mais relevantes devem ser considerados para evolução, e maximização/minimização de uma função de avaliação.

Sobre a camada escondida de uma rede, Paz e Kamath ressaltam que o número de neurônios ótimo estará estritamente associado com a capacidade da rede em aprender e generalizar. Contendo poucos processadores, a rede terá dificuldades no aprendizado, enquanto que com muitos, poderá causar super treinamento, com sérios problemas de generalização.

Outro ponto importante, que geralmente não é tratado separadamente na literatura, é a otimização da topologia da camada de entrada, com o objetivo de selecionar variáveis que de fato explicam o problema estudado. Sobre isto, observa-se que são muitos os autores que dizem otimizar a “topologia” de uma rede, porém são poucos que o fazem considerando também a entrada. A exemplo dos que otimizam a entrada, pode-se citar [15], [21], e [30].

Outros pesquisadores tratam também da evolução de conjuntos de redes neurais: [13], [14], [21], [28]. AEs são frequentemente utilizados para maximizar uma função de avaliação, ou minimizar uma função de erro. Tais objetivos são utilizados como procedimentos de otimização, e não como de aprendizado. Neste sentido, uma RNA obtida por AEs pode não estar associada a aspectos de generalização. Soma-se a isto que outros indivíduos numa determinada população podem ter conhecimento importante associado à generalização, e serem utilizados em conjunto com a RNA vencedora. Pensando nisto, alguns pesquisadores investem na idéia de evoluir e utilizar em conjunto RNAs para ajudar a aumentar a generalização.

Yao apresenta alguns objetivos adicionais encontrados na modelagem de RNA. Um deles é a evolução das regras de aprendizado, e parâmetros associados, de algoritmos com treinamentos de rede baseados no gradiente decrescente. A

justificativa, segundo Yao, é a necessidade da rede “aprender a aprender”. É o caso de Abraham em [15]. Dentre seus objetivos de otimização, a escolha do algoritmo de aprendizado entre o quais o backpropagation, gradiente conjugado, algoritmo quase-newton, e o de levenberg-marquadt, bem como os parâmetros associados a cada um, todos eram também otimizados, e, portanto, considerado no cromossoma.

Outro objetivo, segundo Yao, seria também a evolução da função de avaliação, ao contrário de utilizá-la como uma medida de desempenho. Isto pode ser verificado em Sexton e Dorsey em [29], quando estes decidiram codificar no cromossomo, o tipo de função que o modelo utilizaria, entre a soma dos erros quadráticos e a soma dos erros absolutos. O mesmo se observa em Pedrajas e Boyer [22]. Segundo Yao:

“... é muito custoso computacionalmente obter o valor de aptidão para alguns problemas de controle, pois é impraticável executar um sistema real para cada combinação de parâmetros. Com o objetivo de contornar este problema e fazer com que a evolução seja mais eficiente, os valores de aptidão podem ser aproximados ao invés de computados exatamente.”

Abraham também considerou o efeito de diferentes funções de ativação – sigmóide-logística ou sigmóide-hiperbólica – na representação do seu algoritmo evolutivo. Isto pode ser verificado também em Lacerda [10] otimizando os centros e variâncias de cada neurônio da camada escondida de sua RNA do tipo radial.

Nossa pesquisa teve como resultados artigos que se julgou mais relevantes, assim sendo, é provável que existam outros algoritmos evolutivos aplicados no treinamento de redes neurais, os quais não foram mencionados. Da mesma forma, há outros tipos de redes neurais além das analisadas. Como exemplo de outra rede, pode-se citar Yao *apud* Sarkar e Yegnanarayana, na otimização de uma rede neural probabilística através de programação evolutiva.

Algumas críticas feitas por Paz e Kamath sobre a forma com a qual resultados são apresentados na literatura serão consideradas para orientação dos resultados desta dissertação. Estas críticas sugerem que:

- Teste-se o modelo em bancos de dados os mais variados, com áreas de aplicação e níveis de dificuldade os mais diversos possíveis.

- Seja feita a separação dos resultados entre grupos de treinamento e teste, aspecto este que eventualmente não é esclarecido, ficando a critério do leitor o julgamento do grupo em que se estão apresentando os resultados.
- Sejam feitos testes estatísticos apropriados para identificar significativas diferenças em resultados apresentados.

Aspectos como o tempo de processamento, ou o custo computacional não foram investigados extensivamente por nenhum dos autores, mas sem dúvida são pontos que merecem a atenção *em RNA especificamente*. Sobre isto, conclui-se que algoritmos evolutivos com inspiração quântica apresentam um tempo de processamento inferior aos demais AE conforme se observa em Abs da Cruz [2].