

# Modelo Híbrido Neuro-Evolucionário para Problemas de Agrupamento utilizando Redes Neurais Spiking

Marco Silva

Marley M. B. R. Vellasco

Departamento de Engenharia Elétrica

Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

Rua Marquês de São Vicente, 225

Rio de Janeiro, RJ Brasil, 22451-900

Emails: mabs21@ele.puc-rio.br

marley@ele.puc-rio.br

Edson Cataldo

Departamento de Matemática Aplicada

Programa de Pós-Graduação em Engenharia Elétrica  
e de Telecomunicações

Universidade Federal Fluminense (UFF)

Rua Passos da Pátria, 156

São Domingos, Niterói

Rio de Janeiro, Brasil

Email: ecataldo@im.uff.br

**Resumo**—Este artigo apresenta um modelo híbrido para problemas de agrupamento em que os atributos de entrada e os parâmetros de uma rede neural *spiking* (SNN) são otimizados com algoritmos evolucionários. Uma abordagem evolutiva baseada no Algoritmo Evolutivo com Inspiração Quântica com representação Binário-Real (AEIQ-BR) foi utilizada. O modelo resultante foi nomeado *Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real Representation* (QbrSNN). O modelo neuro-evolucionário proposto é utilizado com o propósito de configurar corretamente a SNN e selecionar os atributos mais relevantes para agrupamento. O modelo QbrSNN foi avaliado em oito bases de dados *benchmark*, comparando seus resultados com modelos de agrupamento tradicionais (K-means e SNN clássica) e com uma SNN cujos parâmetros são otimizados por um Algoritmo Genético padrão. Os resultados mostraram a viabilidade do uso das SNNs para aprendizado não supervisionado, obtendo ótimos resultados.

**Keywords**—Redes Neurais Spiking, Agrupamento, Algoritmos Neuro-Evolutivos.

## I. INTRODUÇÃO

Redes Neurais Spiking (*Spiking Neural Network - SNN*) são apresentadas como um novo paradigma conexionista, sendo a terceira geração de modelos de redes neurais [1]. Estudos anteriores têm mostrado que redes neurais que utilizam neurônios *spiking* são, computacionalmente, superiores às redes neurais tradicionais, que utilizam neurônios com funções de ativação sigmoidal [1], [2], obtendo resultados promissores na solução de problemas importantes do mundo real.

Na construção de uma SNN, vários parâmetros que configuram a rede precisam ser estipulados, necessitando de uma combinação correta para que a rede neural tenha um bom desempenho, sendo o ajuste manual desses parâmetros uma tarefa desafiadora.

Diversas propostas que permitem otimização automática de parâmetros de uma SNN com treinamento supervisionado já foram apresentados [3], [4], [5], [6], [7]. Entretanto, propostas semelhantes ainda são escassas para SNN com treinamento não-supervisionado. Assim, este trabalho apresenta um modelo

híbrido, específico para problemas de agrupamento, cujos parâmetros de uma SNN são automaticamente determinados por um algoritmo evolutivo com inspiração quântica.

O modelo neuro-evolucionário proposto foi denominado *Quantum Inspired Evolution of Spiking Neural Networks with Binary-Real representation* (QbrSNN), sendo baseado no modelo do Algoritmo Evolutivo com Inspiração Quântica com representação Binário-Real (AEIQ-BR) [8]. Algoritmos evolutivos com inspiração quântica têm sido utilizados em problemas de otimização combinatória usando uma representação binária, apresentando um desempenho superior aos algoritmos evolutivos convencionais, encontrando soluções mais rápida e com menos indivíduos. Esse recurso reduz drasticamente o número de avaliações necessárias, e é um fator de desempenho importante quando o problema a ser solucionado envolve um grande custo computacional para o cálculo da função de avaliação de cada solução.

O trabalho está organizado em cinco seções adicionais. A Seção II apresenta um resumo sobre as SNNs tradicionais. A Seção III apresenta o algoritmo evolutivo com inspiração quântica utilizado neste trabalho. O modelo híbrido neuro-evolucionário proposto é descrito em detalhes na Seção IV. A Seção V apresenta os resultados obtidos, e finalmente a Seção VI discute as conclusões deste trabalho.

## II. MODELO DE REDE NEURAL Spiking PARA APRENDIZADO NÃO SUPERVISIONADO

O modelo de SNN utilizado neste estudo é baseado no modelo proposto por Hopfield [9], e estendido em [10]. Este modelo codifica os padrões de entrada em pulsos defasados no tempo, utilizando múltiplas sinapses e neurônios de saída modelados como SRM (*Spike Response Model*) e operando como neurônios RBF (*Radial Basis Function*). Os atributos de entrada são codificados de acordo com [11], em um grande número de disparos defasados no tempo.

Esse tipo de codificação permite que valores contínuos possam ser codificados em um trem de pulsos, utilizando  $m$

funções gaussianas sobrepostas para cada atributo. Assim, a  $i$ -ésima gaussiana codificando a variável  $n$ , é determinada pelo centro ( $C_i$ ) e seu desvio padrão (largura -  $\sigma$ ) conforme a seguir:

$$C_i = I_{min}^n + \left( \frac{2i - 3}{2} \right) \left( \frac{I_{max}^n - I_{min}^n}{m - 2} \right), \quad m > 2 \quad (1)$$

$$\sigma = \frac{1}{\gamma} \left( \frac{I_{max}^n - I_{min}^n}{m - 2} \right), \quad m > 2 \quad (2)$$

onde  $I_{min}$  e  $I_{max}$  são os valores mínimo e máximo, respectivamente, assumido pela variável de entrada  $n$ , com  $m$  gaussianas utilizadas. Portanto o número de neurônios na camada de entrada será composto por  $n.m$  neurônios.

Em um processo de aprendizado não supervisionado de uma rede *spiking*, o objetivo é fazer com que os neurônios RBF armazenem os centros de cada grupo, fazendo com que dispare o neurônio da camada de saída cujo centro esteja mais próximo do padrão de pulsos, recebidos na camada de entrada. Os pesos são inicializados, e, após análise dos vetores de entrada na rede, o neurônio da camada de saída que disparar primeiro será considerado neurônio vencedor (*Winner takes all*). Essa regra de aprendizagem modifica os pesos das conexões sinápticas usando uma variante temporal da regra de aprendizagem Hebbiana, expressada como (ver Figura 1):

$$\Delta w_{ij}^k = \eta L(\Delta t) = \eta(1 - b)e^{-((\Delta t - c)^2 / \beta^2)} - b \quad (3)$$

onde,  $\Delta w_{ij}^k$  é taxa de variação do peso da  $k$ -ésima conexão entre o neurônio  $i$  e o neurônio  $j$ ;  $\eta$  é a taxa de aprendizado;  $L$  uma função Hebbiana;  $b$  determina o valor com que os pesos sinápticos são incrementados e decrementados;  $c$  determina o centro da curva de aprendizado;  $\beta$  define a largura da parte positiva da curva de aprendizagem; e  $\Delta t$  é a diferença do tempo de recebimento de um pulso e o disparo do neurônio em questão. O gráfico da função Hebbiana é ilustrado na Figura 1

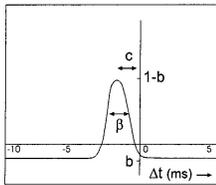


Figura 1. Gráfico da função de aprendizado.

A Figura 2 apresenta um exemplo de estrutura da SNN composta de múltiplas sinapses.

Conforme descrito anteriormente, diversos parâmetros influenciam no funcionamento da SNN, afetando seu desempenho. A próxima seção apresenta o modelo evolucionário com inspiração quântica utilizado neste trabalho para a determinação automática desses.

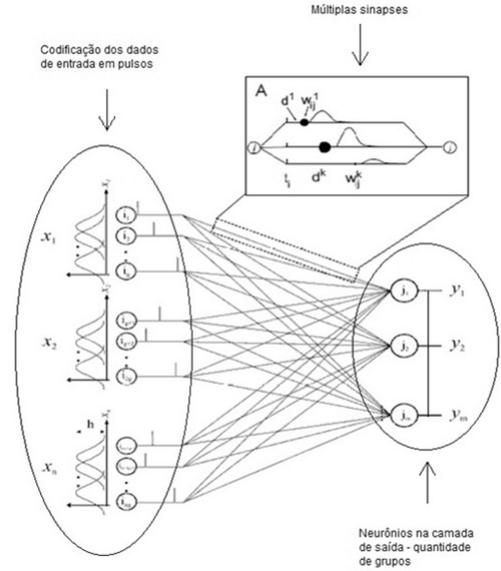


Figura 2. Estrutura da SNN - conexão composta de múltiplas sinapses. Cada atributo de entrada de um conjunto de dados é codificado separadamente em um trem de pulsos.

### III. ALGORITMOS EVOLUTIVOS COM INSPIRAÇÃO QUÂNTICA COM REPRESENTAÇÃO BINÁRIA REAL-AEIQ-BR

Han e Kim propuseram um algoritmo evolutivo com inspiração quântica (AEIQ) em 2002 [12], que foi inspirado no conceito de computação quântica. De acordo com o conceito de computação clássica, a informação é representada em bits, em que cada bit deve conter 0 ou 1. No entanto, na computação quântica, a informação é representada por um *qbit*, onde o valor de um *qbit* pode ser 0, 1, ou uma sobreposição de ambos. Essa superposição permite representação de todos os estados possíveis ao mesmo tempo com base na sua probabilidade. O estado quântico é definido como  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , onde  $\alpha$  e  $\beta$  são números reais, satisfazendo à condição de normalização  $|\alpha|^2 + |\beta|^2 = 1$ , onde  $|\alpha|^2$  indica a probabilidade do *qbit* ter valor 0 e  $|\beta|^2$  indica a probabilidade do *qbit* ter valor 1 quando observado. Um cromossomo quântico binário é representado como uma sequência de pares de números, que podem ser observados para gerar indivíduos clássicos. Uma notação geral para um indivíduo com  $l$  *qbits* pode ser representado com  $2^l$  possíveis padrões simultâneos.

Durante o processo evolutivo, um *qbit* de um indivíduo quântico é atualizado de tal forma a convergir para uma probabilidade máxima igual a 1 para os estados  $|0\rangle$  ou  $|1\rangle$  do melhor indivíduo possível. Esse processo de convergência é feito através de operadores quânticos e funciona como uma atualização de cada gene dos indivíduos. Para o algoritmo quântico binário, [12] utilizou o operador denominado porta de rotação para modificar os valores de  $\alpha$  e  $\beta$ , aumentando as chances de observação dos melhores indivíduos [12].

Este modelo de representação binária foi entendido para problemas de otimização numérica, com representação real [13].

A representação de uma parte numérica do cromos-

somo requer uma população de indivíduos para representar a superposição de estados possíveis que podem ser observados para os indivíduos clássicos. A população quântica  $Q(t)$ , em qualquer instante  $t$  do processo evolutivo, é formado por um conjunto de  $N$  indivíduos quânticos  $q_i (i = 1, 2, 3, \dots, N)$ , de forma que cada  $q_i$  é formada por  $G$  genes  $g_{ij} (j = 1, 2, 3, \dots, G)$ , que são formados por funções que representam uma densidade de probabilidade (fdp). O indivíduo quântico pode ser expresso como  $q_i = [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), \dots, g_{iG} = p_{iG}(x)]$ .

A função densidade de probabilidade (fdp) é usada para gerar os valores para os genes dos indivíduos clássicos, ou seja, a função  $p_{ij}$  deve ser integrável na região do domínio dentro do qual as variáveis que se deseja otimizar podem assumir valores [13]. Uma função simples usada como fdp é a da variável aleatória com função densidade de probabilidade uniforme, usada neste estudo. O gene quântico pode ser representado pelos valores do centro e da largura do pulso. A partir dessa distribuição, sorteia-se um conjunto de pontos que formam os cromossomas da população clássica.

Os modelos evolucionários com inspiração quântica com representação binária e real foram utilizados em conjunto em [8], proposição do modelo neuro-evolucionário AEIQ-BR para a configuração automática de uma rede Multi-Layer Perceptron para problemas de classificação. Uma descrição detalhada de cada passo de AEIQ-BR é fornecido em [8]. Os parâmetros de configuração do modelo AEIQ-BR são fornecidos na Tabela I.

Tabela I. PARÂMETROS DO AEIQ-BR

AEIQ-BR	Parâmetros
$NQ$	Números de indivíduos quânticos
$NC$	Número de indivíduos clássicos
$T$	Número de gerações
$C_{cb}$	Probabilidade de crossover clássico binário
$C_{cr}$	Probabilidade de crossover clássico real
$\Delta\theta$	Ângulo de rotação
$C_q$	Taxa de cruzamento quântico
$updateT$	Intervalo de execução dos operadores

A seção a seguir apresenta o modelo neuro-evolucionário com inspiração quântica baseado no AEIQ-BR utilizado para a configuração automática de uma SNN.

#### IV. QUANTUM-INSPIRED EVOLUTION OF SPIKING NEURAL NETWORKS WITH BINARY-REAL REPRESENTATION (QBR-SNN)

A escolha do modelo AEIQ-BR na implementação do primeiro modelo híbrido neuro-evolutivo de redes neurais *spiking* para aprendizado não supervisionado deve-se à possibilidade de se utilizar representação mista, numérica e binária, no cromossoma do algoritmo evolutivo. As principais vantagens do QbrSNN, proposto neste trabalho, são: capacidade de otimização global da rede neural *spiking*; representação probabilística do espaço de busca com alta diversidade populacional em cada indivíduo; poucos indivíduos populacionais com grande capacidade de exploração e aproveitamento do espaço de soluções, diminuindo o tempo computacional; e a seleção dos atributos de entrada relevantes para serem utilizados na entrada da rede neural.

O modelo QbrSNN representa a ligação entre o AEIQ-BR [8] e uma SNN para aprendizado não supervisionado. Os

parâmetros que configuram a SNN e o número de neurônios na saída (quantidade de grupos formados) definem a parte real do cromossomo com 10 genes. A Tabela II ilustra a descrição e os intervalos utilizados para os parâmetros da parte real do cromossomo:

Já a parte binária do modelo é formada por tantos genes quantos forem os atributos de entrada de cada base de dados a ser agrupada. A Figura 3 ilustra o cromossomo genérico do modelo QbrSNN.



Figura 3. Cromossomo usado para otimizar, simultaneamente, os parâmetros e os atributos de entrada de uma SNN.

Para cada indivíduo, os parâmetros reais são utilizados para configurar a estrutura do SNN e os valores binários definem os atributos selecionados para fornecer as informações para a SNN. O QbrSNN inicializa aleatoriamente a população quântica de acordo com as estratégias apresentadas anteriormente, sendo os genes da parte binária inicializados com valores de  $\alpha = \beta = \sqrt{\frac{1}{2}}$ , de forma que os bits 0 e 1 tenham igual probabilidade de serem observados; já os genes da parte real são inicializados com funções densidade de probabilidade uniforme na faixa de valores possíveis de cada parâmetro, definidas em função de seu centro e sua largura. Em função das faixas de valores de cada variável apresentadas na Tabela II, os genes do cromossoma quântico são inicializados conforme indicado abaixo. Em seguida, a SNN é treinada com base nessa configuração, determinando qual padrão melhor se adapta a cada grupo.

$$Q(t) = \left( \begin{array}{c} \left[ \begin{array}{c} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{array} \right] \dots \left[ \begin{array}{c} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{array} \right] \end{array} \right)_b$$

$$\left( \begin{array}{c} [14; 14; 0.55; 5; 5; 0.55; 5; 10; 5; 6] \\ [12; 12; 0.9; 4; 4; 0.9; 4; 10; 4; 8] \end{array} \right)_r$$

Um indivíduo é capaz de representar uma superposição de todas as possíveis soluções do domínio, no primeiro passo do algoritmo QbrSNN, conforme mostrado a seguir. A partir do segundo passo do algoritmo iniciará o ciclo de evolução, e assim se manterá até que atinja o número máximo de gerações.

O processo evolucionário utiliza a métrica Silhueta ( $S_i$ ) [15] como função de avaliação, a qual combina ambos os métodos de coesão e separação. Para descrever como calcular o método de silhueta para um dado objeto de um grupo, é necessário calcular a similaridade média entre todos os objetos do grupo ao qual ele pertence ( $a_i$ ); e calcular a similaridade média a todos os objetos dos outros grupos e encontrar o menor valor ( $b_i$ ). O coeficiente  $S_i$  para o objeto escolhido pode ser calculado através de  $S_i = (b_i - a_i) / \max(a_i, b_i)$

O valor do coeficiente  $S_i$  varia entre -1 e 1. Um valor negativo não é desejado, pois corresponde ao caso em que

Tabela II. PARÂMETROS DA PARTE REAL DO CROMOSSOMO

$n_G$	Número de Gaussianas	8-20
$\Delta$	Delay	8-20
$\eta$	Taxa de Aprendizizado	0.1-1
$\tau$	Tempo da curva do potencial da membrana	3-7
$t_{max}$	Tempo do cálculo do neurônio vencedor	3-7
$b$	Valor de incremento e decremento dos pesos	0.1-1
$c$	Valor do pico da curva de aprendizado	3-7
$\vartheta$	Valor de threshold (limiar)	5-15
$\beta$	Valor de largura da curva de aprendizado (vizinhança)	1-4
$n_{out}$	Número de neurônios na saída	2-10

o valor de similaridade média para os pontos no grupo ( $a_i$ ) é maior (isto é, menos similaridade) que a menor média de similaridade entre pontos de outro grupo ( $b_i$ ). O melhor é que o coeficiente de silhueta seja positivo ( $a_i < b_i$ ), e que  $a_i$  seja o mais próximo de 0 possível, desde que, o coeficiente ( $b_i$ ) assuma valor 1 quando  $a_i=0$  [14], [15].

Com a função de aptidão definida, cada indivíduo é avaliado, e este procedimento é repetido para todos os indivíduos da população. Com esses valores de aptidão, os indivíduos são classificados e selecionados, e os operadores de recombinação são aplicadas. Este processo é repetido até que um critério de parada seja atendido. A Figura 4 apresenta a arquitetura de uma QbrSNN. O modelo QbrSNN foi avaliado em oito bases benchmarks e os resultados obtidos são apresentados na próxima seção.

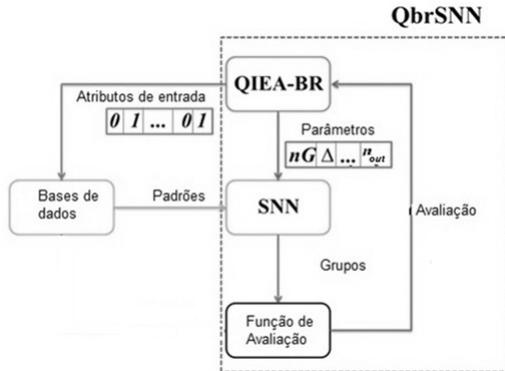


Figura 4. Arquitetura do QbrSNN.

## V. RESULTADOS

Para o teste de desempenho dos modelos foram utilizadas oito bases benchmark [16], [17]. Com o intuito de comprovar o desempenho da SNN, foi feita, inicialmente uma comparação dos resultados de uma SNN padrão, sem otimização de parâmetros e sem seleção dos atributos de entrada, com a abordagem clássica de agrupamento denominada K-means [18]. Para garantir a relevância estatística, foram executados 40 ensaios independentes para cada base de dados. Todas as bases de dados são resumidas na Tabela III, onde p, d, e k denotam os números de padrões, de atributos e de grupos, respectivamente.

Por serem bases que disponibilizam as classes, foi utilizado um índice que mede a relação entre a concordância e a discordância entre os grupos formados, denominado Correct Rand (cR) [19]. Este índice cR pode assumir valores entre [-1, 1], onde 1 denota um acordo perfeito entre partições e

Tabela III. BASES DE DADOS BENCHMARK

Bases de dados	p	d	k
Aggregation	788	2	7
Compound	399	2	6
Iris	150	4	3
Wine	178	13	3
Glass	214	9	7
Yeast	1484	8	10
Breast	699	10	2
Thyroid	215	5	2

valores negativos ou próximos de 0 representam algum tipo de discordância encontrada [19].

Para configurar a SNN padrão foram utilizados os parâmetros sugeridos por [11]. Conforme mencionado, a função de avaliação utilizada foi a Si, calculando o cR para o melhor indivíduo. Os resultados referentes às 8 bases benchmark usando SNN e k-means, são mostrados na Tabela IV.

Tabela IV. RESULTADOS - SI E CR PARA O MELHOR INDIVÍDUO

Bases de Dados	K-means		SNN	
	Si	cR	Si	cR
Aggregation	0.78	0.77	0.80	0.78
Compound	0.75	0.68	0.76	0.74
Iris	0.80	0.89	0.82	0.92
Wine	0.37	0.50	0.45	0.53
Glass	0.48	0.51	0.51	0.56
Yeast	0.17	0.46	0.20	0.49
Breast	0.38	0.52	0.42	0.58
Thyroid	0.78	0.65	0.80	0.68
Média	0.56	0.62	0.60	0.66
Mediana	0.62	0.59	0.64	0.63

Os resultados da SNN padrão apresentaram um desempenho melhor na formação dos grupos em todas as bases de dados, obtendo o valor da Si e cR mais elevado em termos de média e mediana.

Em seguida, as mesmas bases foram utilizadas na avaliação de uma SNN otimizada por um algoritmo genético clássico (SNN-AG) e do modelo proposto (QbrSNN). Em ambos os casos, além da otimização dos parâmetros da SNN, os atributos de entrada também foram selecionados.

Assim, na proposição dos modelos SNN-AG e QbrSNN, a parte binária do cromossomo tem o tamanho definido pelo número de atributos de entrada ( $d$ ) de cada base de dados como na Tabela III. Os modelos QbrSNN e o SNN-AG foram executados em 40 ensaios independentes, evoluídos em um total de 3000 avaliações. A realização de 3000 avaliações implica em avaliar 3000 SNNs com configurações diferentes, sendo este valor escolhido após testes realizados com valores maiores de avaliações não terem apresentado melhoria nos resultados, e por ser computacionalmente viável.

Estas execuções foram realizadas em MATLAB (Ma-

tlab10), e os valores dos parâmetros utilizados na configuração dos modelos híbridos são definidos na Tabela V, com base nas recomendações de [8]. A configuração para o AG foi com valores *default*. Esses valores foram os mesmo utilizados para todas as bases de dados.

Tabela V. VALORES DOS PARÂMETROS: QBRNN E O SNN-AG

QbrSNN	Valores	SNN-AG	Valores
$NQ$	15	População	50
$NC$	60	Gerações	60
$T$	50	Torneio	4
$C_{cb}$	95%	Tx. Cruz.	80%
$C_{cr}$	5%	Tx. Mutação	20%
$\Delta\theta$	$0.050\pi$	Elitismo	2
$C_q$	10%	-	-
$updateT$	4	-	-

A Tabela VI apresenta a média dos resultados de 40 ensaios dos resultados de Si e cR para cada base benchmark utilizando uma SNN padrão, o SNN-AG clássico e o QbrSNN. Como esperado, os modelos neuro-evolutivos obtiveram resultados superiores à SNN padrão. O SNN-AG obteve o valor da Si 15% maior em média quando comparado com a SNN padrão, e o modelo QbrSNN obteve o valor da Si aproximadamente 35% mais elevado que a SNN padrão e 15% mais elevado que o SNN-AG. Isso confirma a vantagem do ajuste automático dos parâmetros das SNN, uma vez que se torna difícil fazer esses ajustes manualmente.

O teste de Wilcoxon [20] foi aplicado para comparar as medidas da Si entre as abordagens neuro-evolutivas. O modelo QbrSNN claramente superou as abordagens sem evolução em termos da mediana ( $valor - p < 0,01$ ). Portanto, realizando seleção de atributos e otimização de parâmetros com poucas avaliações pode-se melhorar muito a qualidade de agrupamento.

No modelo QbrSNN, os melhores resultados foram encontrados mais rapidamente, apesar de ter exigido uma média de 40% a mais de esforço computacional. Portanto, o usuário deve considerar este trade-off ao aplicar QbrSNN para seleção de atributos e otimização de parâmetros.

Um número maior de atributos aumenta o tempo de processamento da rede, não significando obter melhores resultados ou uma maior precisão. Em média, 45% dos atributos de entrada não foram considerados para o agrupamento e, na maioria das bases de dados, o número de neurônios de saída (grupos formados) obtidos foram muito próximo do número de classe de cada base de dados. A Tabela VII ilustra a quantidade de atributos utilizados para a obtenção do melhor resultado de agrupamento, sendo diferente para algumas bases de dados.

Destaca-se o fato de que a base de dados Yeast apresentou um número de grupos ( $n_{out}$ ) diferente do fornecido pela base, para ambos os modelos. No modelo QbrSNN, das 10 classes disponíveis, 8 grupos foram formados (ver Tabela VIII). Com relação as outras bases de dados, a formação dos grupos foram concordantes com as classes sugeridas pelas bases.

## VI. CONCLUSÕES

Os testes utilizando bases benchmark foram fundamentais para o entendimento do funcionamento das redes utilizando neurônios *spiking*. Os resultados obtidos pelo modelo evoluído

para essas bases, que se apresentam como bases estáticas, demonstrou viabilidade, superando significativamente uma SNN padrão sem evolução. Ao comparar as duas abordagens o QbrSNN rendeu um agrupamento de maior qualidade em algumas bases de dados em relação ao SNN-AG.

Com relação à seleção de atributos relevantes, o algoritmo encontrou um número reduzido de atributos para a maioria das bases, o que determina um melhor rendimento dos modelos.

O QbrSNN comprovou uma maior velocidade de convergência quando comparado com o SNN-AG, em relação ao número de avaliações. Um menor número de avaliações se deve ao fato de que, no caso do QbrSNN, cada indivíduo da população quântica representa regiões promissoras do espaço de busca, ao invés de um mapeamento pontual, o que acelera a sua convergência.

No que diz respeito à precisão, o potencial do modelo com a otimização automática de parâmetros e seleção de atributos mostrou ser consistente, produzindo bons resultados com menos atributos de entrada. Esta abordagem se mostrou extremamente útil para a análise e modelagem de diferentes bases de dados.

## AGRADECIMENTOS

Os autores agradecem o apoio financeiro das agências brasileiras: CNPq e FAPERJ.

## REFERÊNCIAS

- [1] W. Maass, *Networks of spiking neurons: The third generation of neural network models*, vol. 10, no. 9, pp. 1659-1671 Neural Networks, Austria, 1997.
- [2] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, New York, 2002.
- [3] N. Kasabov and K. Dhoble and N. Nuntalid and G. Indiveri, *Dynamic evolving spiking neural networks for on-line spatio-temporal pattern recognition*, Neural Networks, vol. 41, pp. 188-201, New York, 2013.
- [4] N. Kasabov, *NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data*, Neural Networks, vol. 52, pp. 62-76, 2014.
- [5] H. N. A. Hamed and N. Kasabov and S. M. Shamsuddin, *Integrated Feature Selection and Parameter Optimization for Evolving Spiking Neural Networks using Quantum Inspired Particle Swarm Optimization*, IEEE Computer Society - International Conference of Soft Computing and Pattern Recognition, IEEE Press, pp. 695-698, 2009.
- [6] H. N. A. Hamed and N. Kasabov and S. M. Shamsuddin, *Quantum-inspired particle swarm optimization for feature selection and parameters optimization in evolving spiking neural networks for classification tasks*, Numerical Analysis and Scientific Computing, pp. 133-148, 2011.
- [7] S. Soltic and S. Wysoski and N. Kasabov, *Evolving Spiking Neural Networks for taste recognition*, IEEE World Congress on Computational Intelligent (WCCI) - Hon Kong, 2008.
- [8] A. G. Pinho and M. M. B. R. Vellasco and A. V. Abs da Cruz, *A New Model for Credit Approval Problems: A Quantum-Inspired Neuro-Evolutionary Algorithm with Binary-Real Representation*, World Congress on Nature Biologically Inspired Computing – NaBIC, pp. 445-450, 2009.
- [9] J. Hopfield, *Pattern Recognition Computation using Action Potential Timing for Stimulus Representation*, Nature, vol. 376, pp. 33-36, 1995.
- [10] T. Natschlagler and B. Ruf, *Spatial and Temporal Pattern Analysis via Spiking Neurons*, Network, vol. 9, no. 3, pp. 319-336, 1998.
- [11] S. M. Bohte and M.R. Meybodi and R. Safabakhsh, *Unsupervised Clustering With Spiking Neurons By Sparse Temporal Coding and Multilayer RBF Networks*, IEEE Transactions on Neural Networks, vol. 13, no. 2, 2002.

Tabela VI. RESULTADOS - SI E CR PARA O MELHOR INDIVÍDUO

Bases de Dados	SNN		SNN-AG		QbrSNN	
	Si	cR	Si	cR	Si	cR
Aggregation	0.80	0.78	0.82	0.80	<b>0.87</b>	<b>0.97</b>
Compound	0.76	0.74	0.80	0.75	<b>0.89</b>	<b>0.88</b>
Iris	0.82	0.92	0.86	0.92	<b>0.88</b>	<b>0.82</b>
Wine	0.45	0.53	0.77	0.55	<b>0.98</b>	<b>0.76</b>
Glass	0.51	0.56	0.75	0.88	<b>0.92</b>	<b>0.92</b>
Yeast	0.20	0.49	0.62	0.50	<b>0.88</b>	<b>0.86</b>
Breast	0.42	0.58	0.79	0.52	<b>0.86</b>	<b>0.75</b>
Thyroid	0.80	0.68	0.88	0.77	<b>0.96</b>	<b>0.80</b>
Média	0.60	0.66	0.79	0.71	0.91	0.85
Mediana	0.64	0.63	0.80	0.76	0.89	0.84

Tabela VII. NÚMERO DE ATRIBUTOS (MÉDIA APROXIMADA)

Base de dados	$d$	QbrSNN
Aggregation	2	2
Compound	2	2
Iris	4	<b>3</b>
Wine	13	<b>8</b>
Glass	9	<b>4</b>
Yeast	8	<b>4</b>
Breast	10	<b>6</b>
Thyroid	5	<b>3</b>

Tabela VIII. NÚMERO DE GRUPOS (MÉDIA APROXIMADA)

Base de dados	$k$	QbrSNN
Aggregation	7	7
Compound	6	6
Iris	3	3
Wine	3	3
Glass	7	7
<b>Yeast</b>	<b>10</b>	<b>8</b>
Breast	2	2
Thyroid	2	2

- [12] K. Han and J. Kin, *Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization*, IEEE Transaction on Neural Networks, vol. 6, no. 6, pp. 580-593, 2002.
- [13] A. V. Abs da Cruz and M. M. B. R. Vellasco and M. A. C. Pacheco, *Quantum-inspired evolutionary algorithms for numerical optimization*, IEEE, pp. 2630-2637, 2006.
- [14] V. Kumar and P. Tan and M. Steinbach, *Cluster Analysis: Basic Concepts and Algorithms*, In Introduction to Data Mining, pp. 487-586, 2006.
- [15] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, John Wiley Sons, Inc. - New York, 1990.
- [16] K. Bache and M. Lichman, *UCI Machine Learning Repository* - <http://archive.ics.uci.edu/ml>, Irvine, CA: University of California, School of Information and Computer Science.
- [17] <http://cs.joensuu.fi/sipu/datasets/>.
- [18] T. Kanungo and D. M. Mount and N. S. Ntenyahu and C. D. Piatko and R. Silverman and A. Y. Wu, *An Efficient k-Means Clustering Algorithm: Analysis and Implementation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.
- [19] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.
- [20] J. Derrac and S. Garcia and D. Molina and F. Herrera, *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm and Evolutionary Computation, vol. 1, pp. 3-18, 2011.