

UMA ABORDAGEM EVOLUCIONÁRIA PARA A FORMAÇÃO DE MAPAS TOPOLÓGICOS AUTO-ORGANIZÁVEIS

J. E. Bessa Maia, Guilherme A. Barreto, André L. V. Coelho

Universidade Estadual do Ceará - UECE, jmaia@uece.br

Universidade Federal do Ceará - UFC, guilherme@deti.ufc.br

Universidade de Fortaleza - UNIFOR, acoelho@unifor.br

Resumo – Este trabalho aborda a formação de mapas topológicos (MTs) auto-organizáveis por otimização direta de uma nova função de aptidão, propondo para este fim um algoritmo evolucionário customizado. Este algoritmo induz à formação de um mapeamento de características topologicamente correto e com baixo erro de quantização. A abordagem proposta adota uma função de aptidão que é uma combinação linear ponderada de figuras de mérito que medem dois aspectos da qualidade do mapa: (i) o erro de quantização e (ii) a correlação entre distâncias correspondentes nos dois espaços. O trabalho avalia principalmente o uso do coeficiente de correlação de Pearson para guiar o ordenamento topológico do mapa. Os mapas resultantes são comparados com o algoritmo SOM de Kohonen, com o algoritmo genético proposto em [1] e com o algoritmo da subida da encosta iterado. Os experimentos mostram que a abordagem proposta produz mapas com melhores índices de qualidade que aqueles obtidos pelas outras três abordagens.

Palavras-chave – Formação de mapas topológicos, algoritmo SOM, algoritmos evolucionários, preservação de topologia, subida da encosta.

Abstract – This paper approaches the formation of self-organized topographic maps through direct optimization of a novel fitness function, leading to the proposal of a customized evolutionary algorithm. The proposed algorithm induces the formation of a feature map which is topologically ordered and presents low quantization errors. The approach introduces a fitness function comprised of a linear weighted combination of two figures of merit that measure the quality of the ordered map: (i) the quantization error and (ii) the correlation between distances in the input and output spaces. We then evaluate pros and cons in using Pearson's correlation coefficient (CCP) to guide the topological ordering of the map. The resulting maps are compared with Kohonen's self-organizing map (SOM), Curry & Morgan's genetic algorithm [1] and the iterated hill-climbing algorithm. The obtained results suggest that the proposed approach generates maps with better quality indexes than those achieved by the other three approaches.

Keywords – Topographic map formation, SOM algorithm, evolutionary algorithms, topology preservation, hill-climbing.

1. Introdução

As primeiras tentativas de construir modelos de redes neurais auto-organizáveis foram motivadas essencialmente pelo propósito de simular computacionalmente a formação de mapas corticais topologicamente ordenados [2]. Fortemente inspirados pelo trabalho pioneiro de Hubel & Wiesel [3] sobre o processamento sensorial no córtex visual de gatos, estes modelos buscavam emular o ordenamento topográfico das células nervosas corticais. A principal constatação daquele estudo foi a de que neurônios próximos no córtex representavam regiões próximas do campo visual; ou seja, o córtex visual cria uma espécie de mapa espacial do campo visual.

Kohonen [4] foi quem primeiro propôs um algoritmo computacionalmente eficiente, embora menos biologicamente plausível, para a formação de tais mapas, ao conceber o popular algoritmo SOM (*Self-Organizing Map*). Originalmente pensado como uma versão simplificada do modelo de von der Malsburg [5], o algoritmo SOM tornou-se um dos mais importantes modelos de redes neurais auto-organizáveis com inúmeras aplicações importantes em reconhecimento de padrões, especialmente como técnica de visualização e de redução de dimensionalidade [6]. Grosso modo, o algoritmo SOM é uma técnica de quantização vetorial¹ que tenta manter relações de distâncias (proximidade) entre os padrões de entrada em um arranjo uni-, bi- ou tridimensional de neurônios de saída. Em outras palavras, padrões que estão próximos um do outro no espaço de entrada original são mapeados em neurônios que estão, por sua vez, próximos um do outro no arranjo de saída, resultando assim em um mapeamento (ou projeção) que preserva a topologia do espaço de entrada.

O treinamento do algoritmo SOM compõe-se de dois mecanismos básicos. Primeiro, uma estratégia competitiva do tipo *Winner-take-all* (WTA) é usada para a seleção de um único neurônio vencedor para cada padrão de entrada; por exemplo, o neurônio vencedor é aquele cujo vetor de pesos é o mais próximo do padrão de entrada em termos de distância euclidiana. Segundo, uma estratégia cooperativa faz uso de uma função de vizinhança centrada na posição do neurônio vencedor no arranjo

¹Ou seja, os dados são representados por vetores-protótipos, que no contexto de redes neurais recebem o nome de vetores de peso.

de saída para atualizar também os vetores de pesos dos neurônios em torno do vencedor. A atualização dos vetores de pesos do neurônio vencedor e de seus vizinhos no arranjo de saída é sempre na direção do padrão de entrada.

A consequência da abordagem orientada à vizinhança é que os vetores de pesos daqueles neurônios que estão mais próximos ao neurônio vencedor são atualizados mais fortemente na direção do padrão de entrada atual. Por sua vez, observa-se que após um número suficiente de iterações de treinamento, o trabalho conjunto das estratégias competitivas e cooperativas subjacentes ao algoritmo SOM promove o surgimento de um mapa computacional em que a preservação de topologia é uma notável propriedade emergente. É importante destacar que a despeito da aparente simplicidade computacional, a análise matemática da convergência assim como do ajuste dos parâmetros do algoritmo SOM enquanto técnica de representação de dados são problemas de difícil formalização [7, 8], ainda com espaço para contribuições teóricas e/ou empíricas. Por exemplo, após mais de duas décadas de sua proposição, o mecanismo competitivo-cooperativo responsável pela formação do mapa no algoritmo SOM ainda não está totalmente entendido. Existem avanços teóricos importantes neste contexto, mas muito do conhecimento vem de estudos empíricos [9].

Com respeito à teoria de formação de MTs, várias abordagens já foram propostas, entre as quais podem ser mencionadas aquelas baseadas em Modelos Markovianos [10], Teoria da Informação [11, 12], Métodos de *Kernel* [13, 14], Técnicas de Otimização com Restrições [15], Algoritmo *Expectation-Maximization* [16, 17], Técnicas de Criação e Poda de sinapses [18, 19], além de técnicas recentes, como *Ativações Coletivas* [20]. Essencialmente, esses estudos procuram formalizar a geração de mapas topográficos como um problema de otimização. Assim, a idéia básica é formular uma função-custo que assuma um valor mínimo com respeito a parâmetros a serem determinados quando o estado desejado do mapeamento for atingido. Então a minimização da função-custo levará automaticamente ao conjunto ótimo de parâmetros. O problema é que esta abordagem requer uma função-custo diferenciável e convexa. Entretanto, já foi mostrado que uma função-custo global para o algoritmo SOM é altamente descontínua com muitos mínimos locais [7]. Neste caso, uma alternativa viável seria desenvolver mapas topográficos utilizando Algoritmos Evolucionários tais como Algoritmos Genéticos (AG) ou Estratégias Evolutivas (EE), abordagens essas que não foram ainda totalmente exploradas. Tendo isso em mente, este artigo aborda o problema da formação de MTs por otimização direta mediante o emprego de um algoritmo evolucionário (AE) customizado que mescla características tanto de Algoritmos Genéticos como de Estratégias Evolutivas. Mostra-se por simulação que, com a função de aptidão proposta, o processo de otimização direta sempre leva a um mapeamento com alto grau de preservação de topologia.

O restante deste artigo está organizado da seguinte forma. A Seção 2 traz uma revisão do estado da arte sobre algoritmos de formação de MTs agrupando-os segundo uma perspectiva baseada no mecanismo utilizado para correlacionar os espaços de entrada e de saída. A Seção 3 introduz a abordagem proposta baseada em Algoritmo Evolucionário para a formação do MT, descrevendo em detalhes todos os seus componentes. A Seção 4 apresenta e discute os resultados obtidos. O artigo é concluído na Seção 5.

2. Formação de Mapas Topologicamente Ordenados

Conforme já destacado, três aplicações típicas de MTs são quantização vetorial (via representação dos dados por vetores-protótipos), redução de dimensionalidade (via projeção dos dados em um sistema de coordenadas do arranjo de saída) e visualização de dados (via preservação da topologia). Tendo em mente esses diferentes níveis de abstração, busca-se construir um mapa que reflita simultaneamente a distribuição dos padrões de entrada, além de preservar, mesmo que de maneira aproximada, relações topológicas existentes nestes dados.

A seguir, descreve-se brevemente o algoritmo SOM, com o objetivo de estabelecer o referencial a partir do qual será proposta uma taxonomia para os algoritmos de formação de MTs. O algoritmo SOM é uma rede neural de aprendizagem competitiva não-supervisionada cujos neurônios são dispostos regularmente espaçados em um arranjo geométrico de saída. O número de neurônios e a dimensão do arranjo devem ser definidos previamente. Usualmente, o arranjo de saída é uni- ou bidimensional, mas arranjos tridimensionais são também encontrados na literatura, embora em número muito menor de aplicações.

Cada neurônio está associado a um vetor de pesos representando o centróide de uma região particular do espaço de padrões. Dá-se o nome de célula de Voronoi à região de domínio de um dado vetor de pesos². Os vetores de pesos têm a mesma dimensão dos vetores de entrada. Este artigo adota a seguinte notação matemática: N é o número de neurônios, P é a dimensão do espaço de entrada, L é o número de padrões de entrada, $\mathbf{w}_j \in R^P$ é o vetor de pesos do neurônio j , $\mathbf{w}_i \in R^P$ é o vetor de pesos do neurônio vencedor e $\mathbf{x} \in R^P$ é um padrão de entrada particular. Uma vez escolhido o número de neurônios e a dimensão do arranjo de saída, o treinamento do algoritmo SOM consiste dos seguintes passos:

Pseudo-código para o algoritmo SOM

1. Iniciar (aleatoriamente) os vetores de pesos: $\mathbf{w}_j(0)$, $j = 1, \dots, N$.
2. Selecionar os padrões de entrada \mathbf{x} de acordo com uma distribuição de probabilidade $p(\mathbf{x})$.
3. Escolher o neurônio vencedor $i(\mathbf{x})$ na iteração n , utilizando o critério da distância euclidiana mínima:

$$i(\mathbf{x}) = \arg \min_{\forall j} \|\mathbf{x}(n) - \mathbf{w}_j(n)\|, \quad j = 1, \dots, N. \quad (1)$$

²O termo campo receptivo também é comum no jargão da área.

4. Atualizar os vetores de pesos dos neurônios de acordo com a seguinte expressão:

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n)h_{i,j}(n) [\mathbf{x}(n) - \mathbf{w}_j(n)], \quad (2)$$

em que $\eta(n)$ é a taxa de aprendizagem, $h_{i,j}(n)$ é uma função vizinhança centrada em torno da coordenada \mathbf{r}_i do neurônio vencedor no arranjo de saída, sendo a função vizinhança do tipo gaussiana uma das mais usadas:

$$h_{i,j}(n) = \exp\left(-\beta(n) \|\mathbf{r}_i - \mathbf{r}_j\|^2\right), \quad (3)$$

em que $\beta(n) = 1/2\sigma^2$. É importante destacar que, para garantir a convergência do algoritmo SOM, as seguintes condições devem ser observadas:

$$\eta(n) \rightarrow 0 \text{ e } \sigma(n) \rightarrow 0 \text{ para } n \rightarrow \infty.$$

5. Repetir os passos 2-5 até que nenhuma mudança significativa seja observada.

A propriedade de preservação de topologia pode ser expressa de forma simplificada da seguinte maneira. Sejam \mathbf{x}_1 e \mathbf{x}_2 dois padrões de entrada separados entre si por uma distância $d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|$. Considere também que as coordenadas dos neurônios vencedores no arranjo de saída para \mathbf{x}_1 e \mathbf{x}_2 são representadas por \mathbf{r}_1 e \mathbf{r}_2 , respectivamente. A distância entre as duas coordenadas é dada por $d(\mathbf{r}_1, \mathbf{r}_2) = \|\mathbf{r}_1 - \mathbf{r}_2\|$. Assim, diz-se que o MT preserva relações de proximidade entre os padrões de entrada na projeção destes no arranjo de saída, se à medida que $d(\mathbf{x}_1, \mathbf{x}_2) \rightarrow 0$ for observado que $d(\mathbf{r}_1, \mathbf{r}_2) \rightarrow 0$.

A definição acima só é válida em um sentido (a saber, espaço de entrada espaço de saída). Em outras palavras, o fato de dois neurônios estarem próximos no arranjo de saída não implica necessariamente que os padrões neles mapeados também o estejam no espaço de entrada. A única situação em que a preservação de topologia é observada nos dois sentidos dá-se quando o espaço de entrada e o de saída tem a mesma dimensão (e.g., os padrões de entrada têm duas componentes e o arranjo de saída é bidimensional). Neste caso, diz-se que o espaço de entrada e o de saída são isomórficos.

A indução da propriedade de preservação da topologia entre a posição dos vetores de pesos no espaço de entrada e as coordenadas dos neurônios vencedores no arranjo de saída obriga a que, em princípio, qualquer algoritmo para a formação de MTs inclua um mecanismo que correlacione estes espaços entre si. A natureza e a semântica deste relacionamento, para além das relações de vizinhança geométrica, abrem possibilidades de aplicação do MT com variados níveis de abstração conceitual. O que se segue é uma revisão do estado da arte para algoritmos de formação de MTs baseada no mecanismo utilizado para estabelecer a correlação entre o espaço de padrões e o espaço de saída.

Na primeira classe estão os algoritmos cujo treinamento é baseado na separação do processamento em fases. No nível algorítmico, o mapa topográfico é construído com dois objetivos primários: quantização vetorial (o que significa escolher $N \ll L$ e mapeamento topológico entre os padrões de entrada e as coordenadas dos neurônios no arranjo de saída. Estes objetivos são em geral contraditórios, pois se observa que após certo número de iterações de treinamento do algoritmo SOM, a melhoria de um objetivo se dá à custa da degradação do outro [21, 22].

Assim, uma primeira idéia para a construção de mapas topográficos seria processar os dois objetivos supracitados em fases separadas, uma vez que são conhecidos algoritmos para quantização vetorial e também para ordenamento de dados. Em Kirk e Zurada [23] a formação do mapa é alcançada em duas etapas. Na primeira, aplica-se o algoritmo K-médias para posicionar os vetores-protótipos³ de modo a atingir o mínimo erro de quantização; enquanto na segunda etapa, com os vetores-protótipos congelados, utiliza-se um algoritmo genético (AG) para rearranjar as coordenadas dos protótipos no arranjo de saída, de modo que estas reflitam a ordenação topológica dos padrões de entrada. A função de aptidão do AG é a correlação positiva entre pares de distâncias $d(\mathbf{x}_1, \mathbf{x}_2)$ e $d(\mathbf{r}_1, \mathbf{r}_2)$ nos espaços de entrada e de saída, respectivamente. Este trabalho, entretanto, só apresenta resultados para mapas unidimensionais. Su e Chang [24] também adotam um procedimento de duas etapas. Na primeira etapa, o algoritmo busca por $N \times N$ padrões que melhor representam o conjunto dos dados; na segunda etapa, os $N \times N$ padrões selecionados são arranjados em um arranjo topologicamente ordenado. Junto com o procedimento, os autores propõem uma medida própria para quantificar a preservação de topologia, a qual é utilizada para buscar o ordenamento. Em Su e Chang [25] um procedimento em três etapas é adotado. A primeira etapa utiliza K -médias para selecionar N^2 elementos representativos do conjunto de padrões. Na segunda etapa, um método heurístico é utilizado para realizar um mapeamento que preserve vizinhanças. E na terceira e última etapa, o algoritmo SOM é empregado para refinamento caso a qualidade do mapa não seja satisfatória.

Na segunda classe de modelos estão aqueles baseados no algoritmo SOM propriamente dito. Nesta classe estão todos os algoritmos que utilizam o conceito de função vizinhança em torno de uma unidade vencedora para atualização iterativa dos vetores de pesos. A função vizinhança pode ser entendida como uma versão simplificada da função chapéu mexicano [26], muito usada em modelos auto-organizáveis de inspiração neurobiológica para definir a natureza das interações laterais (se inibitória ou excitatória) entre neurônios vizinhos. De acordo com essa função, o neurônio mais ativo excita os neurônios em sua vizinhança imediata, enquanto neurônios mais distantes são inibidos. Kohonen eliminou a necessidade de conexões laterais inibitórias, ao permitir apenas iterações laterais excitatórias entre neurônios vizinhos implementadas por meio de uma função vizinhança centrada no neurônio vencedor.

Em suma, o uso da função vizinhança nada mais é do que uma forma prática de se emular a formação de respostas localizadas similares àquelas obtidas pelo uso de conexões laterais e função chapéu mexicano. Haykin [26] destaca que o principal propósito

³Equivalente aos vetores de pesos no algoritmo SOM.

da função vizinhança é correlacionar as direções de ajuste dos vetores de pesos dos neurônios que compõem a vizinhança do vencedor com a direção de ajuste deste. No início do treinamento, a vizinhança deve ser larga, de modo que praticamente todos os vetores de pesos são alterados na mesma direção. À medida que a largura da vizinhança diminui com o tempo, o número de neurônios cujas direções de atualização dos pesos são correlacionadas entre si também diminui. O efeito final é a emergência da propriedade de preservação de topologia.

A literatura apresenta inúmeras variações do algoritmo SOM alterando uma ou mais das propriedades do algoritmo original, especialmente o critério de definição do neurônio vencedor. Uma primeira abordagem diferencia-se quanto ao número de padrões de entrada usados na escolha do vencedor. Neste caso, têm-se tanto critérios locais, que consideram apenas o padrão de entrada atual, quanto globais, que consideram todos os padrões de entrada. Uma segunda abordagem caracteriza-se quanto ao número de neurônios de saída levados em conta para definir o vencedor. Neste segundo caso, critérios baseados em um neurônio (abordagem WTA) ou em vários neurônios (abordagem populacional) também são encontrados na literatura. Em quaisquer das abordagens acima, as métricas mais utilizadas de comparação entre padrões e vetores de pesos são baseadas em distâncias geométricas [16], em índices da teoria da informação [11], em teoria dos jogos [38], em inferência estatística [27] e em probabilidades (máxima verossimilhança, bayes, etc.) [28]. Para fins de referência, o algoritmo SOM original usa apenas o padrão de entrada atual, adota a abordagem WTA na escolha do vencedor e utiliza como métrica de seleção do vencedor a distância euclidiana para avaliar o grau de similaridade entre o padrão atual e os vetores de pesos da rede.

Na terceira classe estão algoritmos que não definem explicitamente uma função de vizinhança em torno de um vencedor por qualquer critério e operam de forma totalmente local. O princípio geral neste caso é baseado no conceito de campo receptivo de um neurônio. O ordenamento espacial ocorrerá desde que exista alguma superposição entre os campos receptivos dos neurônios envolvidos, refletida na ativação simultânea dos respectivos neurônios. A partir de interações locais entre os campos receptivos, emerge uma ordem global entre os vetores de pesos. Martinez e Schulten [29] utilizam aprendizagem hebbiana competitiva para construir MTs com base neste paradigma. Em van Hulle [30] é apresentado um algoritmo de formação de MTs baseado em interações puramente locais entre os neurônios. A correlação entre os espaços de entrada e saída é construída através da ativação simultânea das funções de saída geradas pela superposição dos campos receptivos e da regra de atualização dos parâmetros dos neurônios.

Na quarta classe estão os algoritmos construídos a partir da otimização de uma função-custo. Três tipos comuns de função-custo são àquelas funções-custo baseadas em conceitos de energia [7, 15], funções-custo baseadas em critérios da teoria da informação [31] e funções-custo compostas de índices de qualidade do mapa envolvendo o cálculo direto da correlação entre os dois espaços para guiar o ordenamento topológico e a quantização vetorial [21, 32]. Índices clássicos de correlação entre distâncias podem ser utilizados como também índices específicos, sintonizados com o objetivo de gerar um mapa capaz de representar adequadamente a topologia dos padrões de entrada no arranjo de saída.

3. Abordagem Proposta

O objetivo deste artigo é apresentar uma abordagem baseada em algoritmos evolucionários (AE) para a formação de MTs e compará-la com os algoritmos SOM e subida da encosta em termos da qualidade do mapa gerado. Uma vez que estes métodos não se baseiam em derivadas de uma função-custo, eles apresentam-se como opções adequadas para trabalhar com funções-custo descontínuas, além de serem menos susceptíveis a encontrar soluções subótimas (i.e. mínimos locais) que métodos baseados no gradiente de funções-custo não-convexas.

O uso de AE em conjunto com MTs tem sido abordado em diversos trabalhos [1, 21, 22, 33–36]. Contudo, alguns destes trabalhos [21, 22, 34, 35] não tratam explicitamente da formação de um mapa ordenado topograficamente, pois utilizam AE para evoluir o algoritmo SOM a fim de encontrar a topologia ótima (número de neurônios e dimensão do arranjo de saída), bem como valores ótimos dos parâmetros de treinamento (i.e. o ritmo de redução da taxa de aprendizagem e da largura da função vizinhança). O trabalho de Polani [22], que utiliza AGs, é representativo desse grupo. Já Curry e Morgan [1] propõem uma abordagem para formação direta de MTs baseada em AG. O termo “formação direta” refere-se ao fato de que o ajuste dos vetores de pesos e a ordenação topológica dos mesmos são realizados simultaneamente. Mais recentemente, uma outra abordagem meta-heurística, o algoritmo de otimização por Colônia de Formigas [37], também foi utilizada na formação de mapas topologicamente ordenados [36].

O presente artigo adota a metodologia do trabalho de Curry e Morgan, mas difere significativamente deste em vários pontos, por exemplo, quanto ao tipo de função de aptidão, aos operadores genéticos utilizados e aos parâmetros de desempenho utilizados para comparação. Em vez de um AG, o presente artigo trata do problema de formação de MTs por meio de uma abordagem por otimização direta utilizando AEs. A função de aptidão é uma combinação linear do Erro de Quantização com coeficiente de correlação de Pearson entre as distâncias correspondentes nos dois espaços. Os índices de mérito utilizados para comparação são o erro de quantização, o erro topográfico ponderado, e o coeficiente de correlação de Pearson, cujas definições são apresentadas nas próximas seções.

3.1 Algoritmo Evolucionário Básico

Algoritmos evolucionários compreendem uma família de algoritmos iterativos de busca estocástica que se baseiam no princípio da Seleção Natural (sobrevivência do mais apto) e em conceitos de genética populacional [38]. Tais algoritmos lançam mão de uma população de soluções candidatas, codificadas em cromossomos, as quais são avaliadas por uma função de aptidão, esta relacionada ao problema sob análise. A cada geração de um AE, indivíduos são escolhidos probabilisticamente para reprodução,

geralmente, com base em seus valores de aptidão, gerando descendentes (novas soluções) mediante o uso de operadores genéticos (mutação e recombinação). Finalmente, dentre pais e filhos, são selecionados aqueles que comporão a próxima geração, etapa esta conhecida como seleção para substituição. Geralmente, os indivíduos mais aptos são selecionados, muito embora essa escolha possa incorrer em convergência prematura.

Dentre as vantagens dos AE estão o seu elevado grau de flexibilidade e adaptabilidade à tarefa em mãos, sua robustez em lidar com problemas mal especificados e de elevada dimensionalidade, bem como a sua capacidade de realizar busca global sobre o espaço de soluções. Por outro lado, a chave para o sucesso na implementação de AE reside muito frequentemente na escolha de uma representação genética e de uma função de aptidão apropriadas, bem como na adoção de operadores genéticos inteligentes, os quais consigam manipular eficientemente as soluções sem ferir as restrições do problema.

As próximas seções descrevem os componentes da abordagem evolucionário proposta especificamente para a formação de MTs, a qual herda características tanto de AG como de EE. O pseudo-código do AE utilizado neste artigo é apresentado a seguir.

Pseudo-código para o Algoritmo Evolucionário

1. Gerar a população inicial.
2. Avaliar a aptidão de cada indivíduo da população.
3. Repetir
 - Seleção para reprodução.
 - Recombinação (crossover).
 - Operadores locais.
 - Mutação.
 - Avaliar a aptidão dos descendentes.
 - Seleção evolucionária (próxima geração).
4. Até terminar.

3.2 Função de Aptidão Proposta

Sejam $\widetilde{\mathbf{W}} = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$ e \mathbf{x}_l , respectivamente, o conjunto de todos os vetores de um mapa topologicamente ordenado e o l -ésimo padrão de entrada. Assim, os índices mais comumente utilizados para medir a qualidade do mapa formado são o Erro de Quantização (EQ) e o Erro Topográfico (ET), definidos respectivamente como

$$EQ(\widetilde{\mathbf{W}}) = \frac{1}{L} \sum_{l=1}^L \|\mathbf{w}_{i(\mathbf{x}_l)} - \mathbf{x}_l\|, \quad (4)$$

em que $i(\mathbf{x}_l)$ é o índice do neurônio vencedor para o padrão de entrada \mathbf{x}_l , e

$$ET(\widetilde{\mathbf{W}}) = \frac{1}{L} \sum_{l=1}^L \Psi(\mathbf{x}_l, \widetilde{\mathbf{W}}), \quad (5)$$

em que $\Psi(\mathbf{x}_l, \widetilde{\mathbf{W}}) \in [0, 1]$ é uma função dada por

$$\Psi(\mathbf{x}_l, \widetilde{\mathbf{W}}) = \Psi(\mathbf{x}_l, \mathbf{w}_1, \dots, \mathbf{w}_N) = \begin{cases} 1, & \text{se } \|\mathbf{r}_1(\mathbf{x}_l) - \mathbf{r}_2(\mathbf{x}_l)\| \leq \sqrt{2}, \\ 0, & \text{em caso contrário.} \end{cases} \quad (6)$$

em que $\mathbf{r}_1(\mathbf{x}_l)$ e $\mathbf{r}_2(\mathbf{x}_l)$ são, respectivamente, as coordenadas na grade (*array*) retangular de saída dos neurônios cujos vetores de pesos são o mais próximo e o segundo mais próximo do padrão de entrada \mathbf{x}_l .

É importante notar que a Eq. (6) considera apenas os vizinhos imediatos do neurônio vencedor, para uma vizinhança retangular. Assim, para os vizinhos imediatos que estão localizados ao norte, sul, leste e oeste do neurônio vencedor, tem-se $\|\mathbf{r}_1(\mathbf{x}_l) - \mathbf{r}_2(\mathbf{x}_l)\| = 1$ e para os neurônios localizados a nordeste, noroeste, sudeste e sudoeste do neurônio vencedor, tem-se $\|\mathbf{r}_1(\mathbf{x}_l) - \mathbf{r}_2(\mathbf{x}_l)\| = \sqrt{2}$. Logo, usar $\|\mathbf{r}_1(\mathbf{x}_l) - \mathbf{r}_2(\mathbf{x}_l)\| \leq \sqrt{2}$ na Eq. (6) abrange todos os oito vizinhos imediatos.

O índice EQ mede o quão bom é o mapa no quesito quantização vetorial. Já o índice ET é um indicador rudimentar da propriedade de preservação de vizinhança, pois quantifica a proporção de pontos cujos primeiro e segundo vetores de pesos mais próximos não são adjacentes no arranjo de saída.

A importância relativa dos índices EQ e ET no mapa final depende da aplicação e é praticamente impossível ser avaliada usando o algoritmo SOM original. É desejável, portanto, ter algum grau de controle desses índices durante a formação do MT, de forma que o usuário possa escolher qual deles é o mais importante para uma dada aplicação. A abordagem baseada em AE provê tal flexibilidade pela escolha adequada da função de aptidão.

Em princípio, qualquer dos índices de qualidade acima, ou sua combinação, podem ser utilizados como função de aptidão do AE, tendo em vista que esta é usada somente nas etapas de seleção para reprodução e para substituição. Por outro lado, a função de aptidão tem um efeito significativo sobre a qualidade do mapa e sobre a demanda computacional por parte do AE.

Kirk e Zurada [21], trabalhando com mapas unidimensionais, usam como índice de avaliação da preservação de topologia o Erro Topográfico Ponderado (ETP), definido como

$$ETP(\tilde{\mathbf{W}}) = \frac{1}{L} \sum_{l=1}^L \frac{|i_1(l) - i_2(l)| - 1}{N - 1}, \quad (7)$$

em que $i_1(l)$ and $i_2(l)$ são os índices dos neurônios cujos vetores de pesos são, respectivamente, o primeiro e o segundo mais próximos do l -ésimo padrão de entrada e $|\cdot|$ simboliza o operador valor absoluto. Os autores compararam os valores de ETP obtidos pela abordagem evolucionária por eles proposta com aqueles obtidos pelo algoritmo SOM original, com melhor desempenho por parte do AG.

Curry e Morgan [1] usam como função de aptidão a Distorção Local Ponderada (DLP), definida como

$$DLP(\tilde{\mathbf{W}}) = E \left[\sum_{l=1}^L h_{il} \|\mathbf{x}_l - \mathbf{w}_{i(\mathbf{x}_l)}\| \right], \quad (8)$$

em que $E[\cdot]$ denota o operador valor esperado tomado sobre todo o conjunto dos padrões de entrada, indexados por i , segundo a distribuição $p(\mathbf{x})$, normalmente assumida como sendo a uniforme. Uma comparação, em termos dos valores obtidos da DLP, entre a abordagem genética proposta por Curry e Morgan e o algoritmo SOM original favorece a primeira.

O presente artigo mostra que uma função de aptidão formada pela combinação simples do EQ e do Coeficiente de Correlação de Pearson (CCP) é tão boa quanto ETP e DLP para o propósito de formação de MTs. A função de aptidão proposta no presente artigo é dada por

$$\text{Aptidão}(\tilde{\mathbf{W}}) = \alpha \cdot CCP(\tilde{\mathbf{W}}) + \beta \cdot EQ(\tilde{\mathbf{W}}), \quad (9)$$

em que os parâmetros $\alpha, \beta \in [0, 10]$ ponderam a importância relativa entre os índices. O índice CCP é a correlação cruzada entre todos os pares de distâncias $[d(\mathbf{r}_m, \mathbf{r}_n), d(\mathbf{w}_m, \mathbf{w}_n)]$, sendo que $(\mathbf{r}_m, \mathbf{r}_n)$ são as coordenadas dos pares de neurônios no arranjo de saída e $(\mathbf{w}_m, \mathbf{w}_n)$ são os correspondentes pares de vetores de pesos. O índice CCP é calculado como

$$CCP(\tilde{\mathbf{W}}) = \frac{\sum_m \sum_n d(\mathbf{r}_m, \mathbf{r}_n) d(\mathbf{w}_m, \mathbf{w}_n)}{(N - 1) S_r S_w}, \quad (10)$$

em que S_r é o desvio-padrão das distâncias entre os neurônios no arranjo de saída e S_w é o desvio-padrão das distâncias entre os vetores pesos. As grandezas S_r e S_w normalizam as distâncias nos espaços de saída e entrada, respectivamente, tornando os valores de CCP independente de escala.

É importante frisar que CCP é um índice do tipo “quanto maior, melhor”, enquanto EQ é do tipo “quanto menor, melhor”. A maximização da função de aptidão mostrada na Eq. (9) via AE é a principal contribuição deste artigo. A abordagem proposta é comparada com o SOM, com um AE utilizando DLP como função de aptidão como em [1] e com um algoritmo subida da encosta em termos dos valores de EQ, ETP, DLP e CCP para um dado mapa formado.

As referências Curry e Morgan [1] e Kirk e Zurada [21] são representativas da abordagem para formação de mapas topológicos tratada no presente artigo uma vez que ambas aplicam otimização direta a uma função de aptidão que guia a formação do MT. No entanto, deve-se destacar algumas diferenças fundamentais entre os algoritmos ali utilizados e aquele ora proposto.

- No caso do trabalho de Curry e Morgan [1] a função de aptidão incorpora diretamente a função vizinhança h_{ij} proposta por Kohonen (ver Eq. (3)) para induzir o ordenamento topológico entre os espaços de entrada e de saída. Este procedimento introduz um forte viés na obtenção de mapas topologicamente ordenados. Esta abordagem é, no mínimo, incongruente, uma vez que para induzir a formação de mapas ordenados por AG, os autores lançam mão justamente do artifício original de Kohonen. Em suma, estes autores usam Kohonen dentro do AG por ele propostos, logo é natural que haja formação de mapas bem ordenados topologicamente. Além disso, a abordagem de Curry e Morgan utiliza busca global pelo neurônio vencedor para calcular o índice DLP; ou seja, é necessário rerepresentar todo o conjunto de dados ao mapa e encontrar o neurônio vencedor para cada padrão de entrada.
- No caso do trabalho Kirk e Zurada [21] não há o viés do uso explícito da função vizinhança h_{ij} , tal como ocorre no trabalho de Curry e Morgan. Porém, esta abordagem também utiliza-se de busca global pelos neurônios vencedores para calcular o índice ETP; ou seja, é necessário rerepresentar todo o conjunto de dados ao mapa e encontrar não só o neurônio vencedor, mas também o segundo vencedor, para cada padrão de entrada.
- Já a definição de CCP proposta na Eq. (10), não incorpora nenhum mecanismo explícito do tipo função de vizinhança como indutor do ordenamento topológico, nem demanda busca global pelos neurônios vencedores para calcular o índice CCP, que quantifica a preservação de topologia. Para calcular este índice para um dado mapa, basta correlacionar as distâncias $d(\mathbf{r}_m, \mathbf{r}_n)$ entre neurônios na grade de saída com as distâncias $d(\mathbf{w}_m, \mathbf{w}_n)$ dos seus respectivos vetores de pesos no espaço de entrada, não sendo necessária rerepresentar os padrões de entrada novamente.

3.3 Operadores genéticos

Operadores genéticos atuam sobre indivíduos da população chamados de cromossomos. A estrutura de dados escolhida para codificar os cromossomos costuma ter papel decisivo no desempenho do AE e na definição dos operadores. No AE considerado neste trabalho a codificação dos MTs em indivíduos foi realizada de maneira muito direta. Cada indivíduo é representado como uma matriz de números reais de dimensões $N \times P$ em que N denota o número de neurônios no arranjo de saída e P denota o número de componentes em cada vetor de pesos associado. Esta representação facilita a interpretação e a manipulação das soluções geradas e também a possibilidade de aplicação de operadores locais e operadores genéticos customizados. Os operadores genéticos de recombinação e mutação são descritos em seguida.

3.3.1 Operadores de recombinação

O papel do operador de recombinação é o de gerar novas soluções mediante o rearranjo de segmentos (blocos de genes) de pares de cromossomos escolhidos previamente para reprodução. Como mencionado anteriormente, essa seleção de indivíduos é geralmente feita probabilisticamente, com base nos valores de aptidão dos indivíduos da população atual. Um dado par selecionado tem probabilidade p_c (taxa de recombinação) de ter seu material genético efetivamente recombinado. Diferentes operadores de recombinação estão disponíveis na literatura, muitos deles projetados especificamente para atender a um determinado domínio de aplicação. Contudo, em geral, os mais utilizados são aqueles baseados em pontos de corte. Este tipo de operador toma como entrada um par de cromossomos-pais e um ou mais pontos de corte escolhidos aleatoriamente, gerando como saída um par de cromossomos-filhos a partir da troca de material genético entre tais pontos de corte.

Para o propósito de formação de MTs, estão propostos e testados neste trabalho dois operadores de recombinação. O primeiro, uma recombinação baseado em localização, escolhe aleatoriamente uma posição do arranjo de saída e troca os vetores de pesos dos neurônios nessa posição para dois mapas (cromossomos) escolhidos previamente. Já o segundo, uma recombinação baseado em vizinhança, estipula que os neurônios na mesma posição selecionada no arranjo de saída levam consigo, quando da recombinação, os seus vizinhos de raio 1. Este último operador também pode ser adaptado para arranjos uni- ou tridimensionais e para outros tipos de raio. Observa-se na Figura 1 que, para um arranjo bidimensional, dependendo da posição aleatória selecionada, três diferentes tipos de segmentos podem ser intercambiados pela aplicação do operador recombinação baseado em vizinhança: (i) um segmento em cruz, para posições no interior da grade; (ii) um segmento em 'L', para pontos localizados em um dos quatro vértices do mapa; e (iii) um segmento em 'T', para pontos localizados nas arestas laterais do mapa.

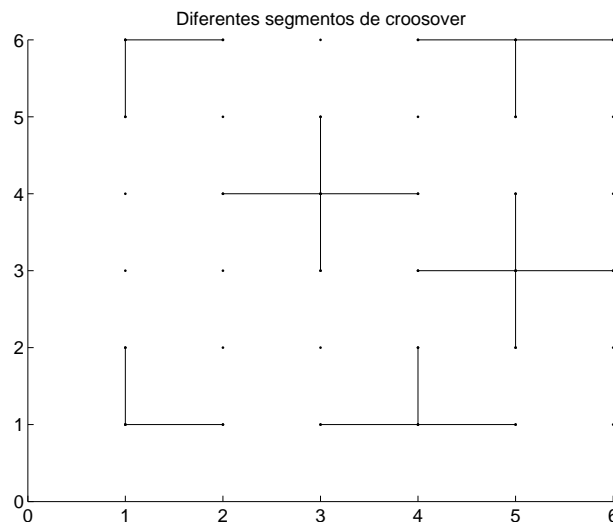


Figura 1: Três exemplos de segmentos intercambiados pelo operador de recombinação baseado em vizinhança numa grade 6x6.

3.3.2 Operador de mutação

Por outro lado, o papel do operador de mutação é o de preservar a diversidade na população e evitar que ela convirja a mínimos locais, sendo que sua aplicação é controlada por um parâmetro específico, chamado de taxa de mutação (p_m). A escolha desse operador também costuma ser dependente da representação adotada para os cromossomos. Neste artigo, o operador de mutação utilizado é aplicado a cada neurônio de cada cromossomo da população de acordo com a taxa p_m pré-fixada. Em sendo o caso, o operador simplesmente adiciona ao vetor de pesos selecionado um valor de passo que consiste de um vetor aleatório \mathbf{x} gerado a partir de uma distribuição gaussiana P -dimensional com vetor de médias nulo e matriz de covariância \mathbf{C} , ou seja, $\mathbf{x} \sim N(0, \mathbf{C})$, em que $\mathbf{C} = \sigma^2 \mathbf{I}$, com \mathbf{I} denotando a matriz identidade de dimensão adequada e σ^2 denotando a variância.

Uma vez que a representação aqui utilizada é uma matriz $N \times P$, outras estratégias de mutação podem ser implementadas. Por exemplo, a taxa de mutação p_m poderia ser adotada para cada componente dos vetores de pesos, ou poder-se-ia estipular

que apenas um componente de cada vetor de pesos escolhido fosse alterado a cada aplicação do operador. Outra possibilidade seria a de se adotar outras distribuições de probabilidade para o valor (passo) de mutação, tais como a distribuição uniforme. Na apresentação dos resultados, na seção 4, os valores utilizados para estes parâmetros serão especificados.

3.4 Operadores locais

Os operadores locais realizam operações determinísticas sobre indivíduos da população muito embora a seleção dos indivíduos para receber a operação possa ser aleatória. Como estes operadores não envolvem alteração ou troca de genes, eles não são considerados operadores genéticos. Em geral, operadores locais são sintonizados com o problema e são uma forma de se inserir conhecimento do domínio no arcabouço conceitual de AEs: (1) na fase de geração da população inicial; (2) na fase de avaliação dos indivíduos; e (3) quando da aplicação dos operadores genéticos. No caso do segundo escopo de aplicação, a adaptação do indivíduo geralmente é abordada como um fator de “aprendizado”.

Neste trabalho aplica-se um operador local que consiste em ordenar um segmento de vetores de pesos segundo a ordenação topológica de uma das coordenadas do arranjo de saída. O ponto inicial do segmento e o seu comprimento são escolhidos aleatoriamente. Um caso especial desse operador consiste em trocar a ordem de dois genes vizinhos no cromossomo. O *insight* para este operador advém do seguinte raciocínio: dado que se deseja um mapa topologicamente ordenado como produto final, se dois vetores de pesos estão “fora de ordem” no arranjo de saída, geralmente serão necessárias muitas operações de mutação e recombinação até que a ordem seja invertida. Um operador local de ordenamento de subsequências pode acelerar o processo de ordenamento do mapa.

3.5 Seleção para substituição

Uma vez avaliada a aptidão dos cromossomos-filhos, passa-se a selecionar, dentre pais e filhos, os indivíduos para a próxima geração. Nesse contexto, duas estratégias são comumente encontradas na literatura de Estratégias Evolutivas. Na estratégia- $(\lambda + \mu)$, μ pais geram λ descendentes, sendo que a seleção opera sobre todos os $\lambda + \mu$ indivíduos retornando os μ melhores. Neste caso, os pais sobrevivem até que os filhos gerados sejam mais aptos. Já na estratégia- (μ, λ) , μ pais geram λ ($\lambda > \mu$) filhos, sendo que os μ melhores filhos serão selecionados para ocupar o lugar dos pais. Assim, a seleção opera apenas sobre os descendentes. Para os experimentos reportados neste trabalho, adota-se a estratégia- $(\mu + \lambda)$.

3.6 Algoritmo de Subida da Encosta (*Hill Climbing*)

Para estudo e comparação com as técnicas evolucionárias descritas nos parágrafos anteriores, implementou-se também uma versão do algoritmo subida da encosta iterada (*Random-restart, steepest ascent hill climbing*) [38]. Ao contrário do AE proposto, o algoritmo subida da encosta iterado não é um algoritmo populacional. A versão implementada consiste em, partindo de uma solução inicial aleatória, gerar cinco descendentes por perturbação aleatória e avançar para aquele que gera maior ganho na função de aptidão. Para efeito de comparação adotou-se os mesmos valores para parâmetros equivalentes nos algoritmos evolucionário e subida da encosta, para cada corrida: o número de iterações no subida da encosta é igual ao tamanho da população utilizado no algoritmo evolucionário, as perturbações são realizadas em nível de gene com mesmo valor de passo e mesma probabilidade de mutação. Esta estratégia possibilitou um controle da implementação e também uma avaliação qualitativa do efeito obtido com os operadores local e de recombinação.

Para reduzir o tempo de processamento do algoritmo subida da encosta, foi adotada uma estratégia em etapas, que consiste em interromper o processo de busca naquelas soluções que, após um número pré-estabelecido de iterações, apresentarem resultados pouco promissores. O fator de redução utilizado foi 2 (dois), ou seja, em cada etapa são eliminadas metade das soluções. Para tal, o procedimento implementa o algoritmo em paralelo: inicialmente gera-se todas as soluções aleatórias iniciais; na primeira etapa evolui-se todas as soluções por N_i iterações e escolhe-se a metade melhor das soluções para continuar na etapa seguinte e assim sucessivamente até restar uma única solução. Por exemplo, começando com 64 soluções iniciais e com $N_i = 5000$, ter-se-á sete etapas com 64, 32, 16, 8, 4, 2 e 1 soluções cada, sendo que a solução apresentada no final operou por $NG = 35000$ iterações.

4 Resultados Computacionais

Na implementação do AE, cada indivíduo da população gera um número fixo de $D = 8$ descendentes; ou seja, a seleção para reprodução não é proporcional ao valor de aptidão. Adota-se este procedimento com o intuito de se obter ganhos computacionais. Os descendentes assim gerados são avaliados e posteriormente selecionados pela estratégia- $(\lambda + \mu)$. Três distribuições de dados foram usadas nas simulações: (i) uma distribuição uniforme sobre o quadrado unitário (conjunto de dados D1), (ii) uma distribuição uniforme sobre o triângulo com vértices em (0, 0), (1, 0) e (0,5, 1) (conjunto de dados D2) e (iii) uma distribuição no espaço tri-dimensional, uniformemente distribuída sobre a superfície $z = x^2$, com domínio no quadrado unitário do plano x-y $[0, 1] \times [0, 1]$ (conjunto de dados D3). Um total de 250 padrões de entrada foram gerados e uma população de $\mu = 32$ indivíduos foi adotada em cada geração. Os códigos foram todos escritos em *MATLAB*® usando funções da SOM toolbox [39].

Uma série de execuções do AE foi conduzida em que se adotou taxas de mutação no intervalo [0,01 - 0,50]. A iniciação da população foi determinada por um valor de semente para o gerador de números aleatórios, de forma a permitir uma eventual replicação dos experimentos. Os testes comparativos foram realizados com o algoritmo AE, com o algoritmo subida da encosta

iterado, com o Algoritmo Genético proposto em [1] e com o algoritmo SOM original para as seguintes topologias de mapa: 5×5 , 8×8 e 1×25 . Os parâmetros α e β na Eq. (9) assumiram valores no intervalo $[0,10 - 10,0]$. Os valores médios e os desvios padrões para 30 realizações do experimento para os índices EQ, ETP, DLP e CCP, para os conjuntos de dados D1, D2 e D3, estão mostrados na Tabela 1. Alguma variabilidade foi encontrada nas realizações para as três topologias de rede e para as três distribuições. Como esperado, constatou-se uma predominância de resultados favoráveis à abordagem AE proposta para todos os índices. Observa-se ainda que para redes menores o algoritmo subida da encosta é tão bom quanto o AE. Os dados da Tabela 1 e uma consulta aos resultados apresentados em [21] mostram que uma função de aptidão formada pela combinação simples do EQ e do Coeficiente de Correlação de Pearson (CCP) é tão boa quanto ETP e DLP para o propósito de formação de MTs.

A título de ilustração, MTs formados pelo AE proposto para os conjuntos D1, D2 e D3 são mostrados nas Figuras 2 e 3, a partir das quais pode-se observar que os MTs apresentam-se ordenados topologicamente. A Figura 4(a) mostra um exemplo de mapa 1-D gerado pelo algoritmo subida da encosta no qual o processamento prosseguiu até o número máximo de iterações e não atingiu um mapa bem formado. Neste caso, o erro de quantização resultou significativamente alto. A Figura 4(b) mostra um mapa 1-D gerado pelo AE proposto para o mesmo conjunto de dados e mesma quantidade de neurônios. Percebe-se claramente que o AE proposto tende a se “espalhar” mais por todo o *manifold* de dados a fim de gerar um menor erro de quantização que o mapa gerado pelo algoritmo subida da encosta.

Tabela 1: Valores médios (med) e desvios padrões (dp) de EQ, ETP, CCP e DLP para os quatro algoritmos (AE - algoritmo evolucionário, AG - com a função de aptidão de [1], HC - subida da encosta, SOM - self-organizing map), para os três conjuntos de dados (D1, D2, D3) e para as três dimensões diferentes do mapa.

Método/Dados /Dimensão	EQ		ETP		CCP		DLP		NG	
	med	dp	med	dp	med	dp	med	dp	med	dp
AE/D1/5x5	0.0821	0.0044	0.0355	0.0142	0.9621	0.031	1.2633	0.0632	1930	47
AG/D1/5x5	0.0913	0.0067	0.0346	0.0192	0.9163	0.038	1.2908	0.0687	2434	96
HC/D1/5x5	0.0830	0.0031	0.0315	0.0120	0.9712	0.024	1.3045	0.0932	14750	290
SOM/D1/5x5	0.0833	0.0016	0.0303	0.0101	0.9116	0.007	1.2856	0.0695	23	0.7
AE/D1/8x8	0.0362	0.0042	0.0457	0.0172	0.9608	0.049	0.7127	0.0923	18550	1853
AG/D1/8x8	0.0368	0.0064	0.0484	0.0195	0.9116	0.036	0.7219	0.1300	23556	3008
HC/D1/8x8	0.0366	0.0071	0.0511	0.0121	0.9440	0.041	0.9213	0.2204	55923	2310
SOM/D1/8x8	0.0344	0.0010	0.0421	0.0133	0.9245	0.013	0.7355	0.1107	79	4
AE/D2/5x5	0.0797	0.0038	0.0354	0.0233	0.9545	0.067	1.3899	0.1212	2285	383
AG/D2/5x5	0.0734	0.0029	0.0314	0.0155	0.9606	0.035	1.3776	0.1165	1756	286
HC/D2/5x5	0.0813	0.0034	0.0354	0.0218	0.9572	0.046	1.0821	0.1734	12550	320
SOM/D2/5x5	0.0712	0.0015	0.0310	0.0101	0.8985	0.011	1.3826	0.1141	21	0.7
AE/D3/5x5	0.1314	0.0087	0.0365	0.0189	0.9604	0.061	1.4208	0.2831	1905	260
AG/D3/5x5	0.1388	0.0065	0.0388	0.0176	0.9336	0.044	1.4135	0.1954	2312	304
HC/D3/5x5	0.1201	0.0083	0.0326	0.0173	0.9502	0.087	1.0828	0.3655	15506	291
SOM/D3/5x5	0.0898	0.0025	0.0308	0.0113	0.9155	0.014	1.4811	0.1756	28	1.1
AE/D3/1x25	0.1643	0.0150	0.0390	0.0140	0.9401	0.121	0.2111	0.0376	2115	275
AG/D3/1x25	0.1422	0.0138	0.0383	0.0121	0.9455	0.140	0.2085	0.0545	1668	246
HC/D3/1x25	0.1714	0.0141	0.0367	0.0176	0.9434	0.098	0.2999	0.0783	21456	562
SOM/D3/1x25	0.0971	0.0016	0.0310	0.0126	0.9210	0.012	0.2103	0.0256	24	0.7

5. Conclusão

Este artigo introduziu uma nova função de aptidão para a formação de mapas de características mediante otimização direta via Algoritmos Evolucionários (AE). A principal motivação foi testar um procedimento de obtenção de mapas que leve em conta diferentes índices de qualidade. Uma segunda motivação foi a de possibilitar que o usuário possa ter controle sobre esse processo de formação calibrando parâmetros da função de acordo com seu interesse. Mapas de diferentes tamanhos e dimensões obtidos pelo algoritmo proposto foram comparados àqueles obtidos por um Algoritmo Genético que utiliza outra função de aptidão, pelo algoritmo SOM padrão e com um algoritmo subida da encosta iterado, usando três critérios de qualidade: o Erro de Quantização (EQ), o Erro Topológico Ponderado (ETP), Distorção Local Ponderada (DLP) e o Coeficiente de Correlação de Pearson (CCP). Os experimentos evidenciam valores compatíveis ou melhores desses índices para os mapas obtidos por otimização direta usando a abordagem proposta.

Em continuidade a este artigo, pode-se vislumbrar algumas direções de pesquisa futura. Uma linha de investigação interessante e que já se encontra em andamento é a de se investigar a possibilidade de se incluir outros índices de qualidade na função de aptidão (Eq. (9)), adotando para tanto uma abordagem evolucionária multi-objetivo. Segundo, planeja-se conceber outros operadores genéticos para o problema em mãos, avaliando o impacto de sua adoção. Finalmente, sendo a função de aptidão uma combinação linear ponderada do EQ com uma medida de correlação entre os espaços de entrada e saída, está sendo examinado um procedimento similar aquele adotado no algoritmo SOM para se alterar adaptativamente os parâmetros α e β em (9); por

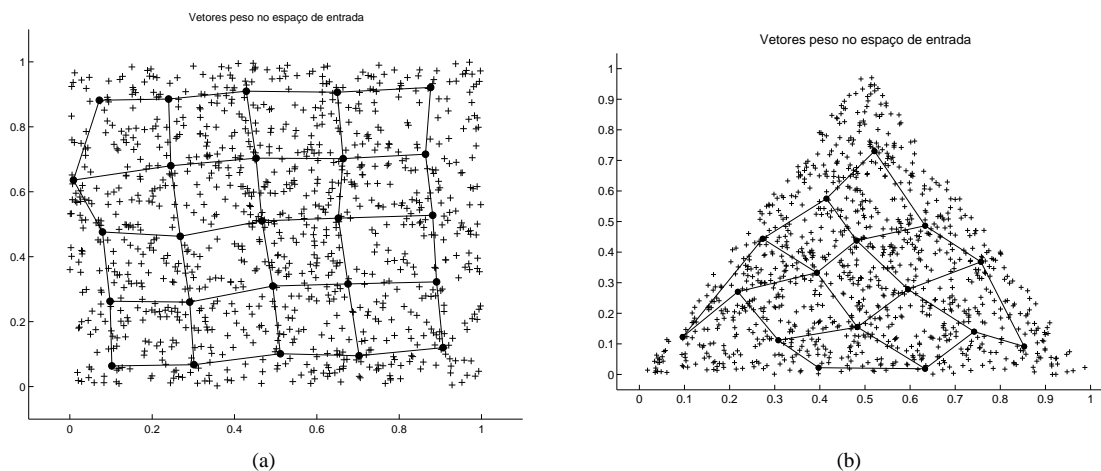


Figura 2: (a) Típico mapa 5×5 formado pelo AE proposto para o conjunto D1 ($\alpha = 1, \beta = 1$). (b) Típico mapa 4×4 formado pelo AE proposto para o conjunto D2 ($\alpha = \beta = 3$).

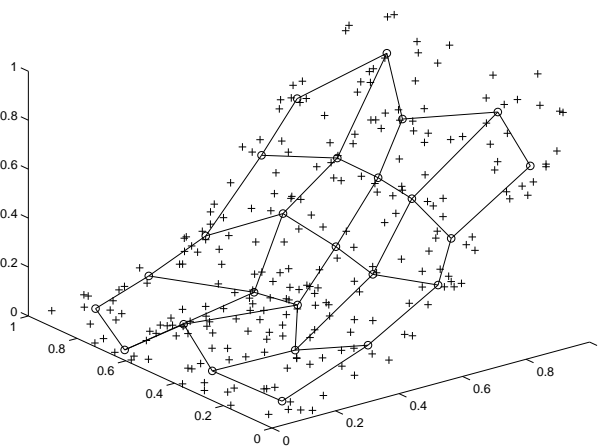


Figura 3: Típico mapa 5×5 formado pelo AE proposto para o conjunto D3 ($\alpha = 1, \beta = 2, 5$).

exemplo, iniciando-se com EQ mais fortemente ponderado em relação ao índice PCC ou vice-versa.

REFERÊNCIAS

- [1] B. Curry and P. H. Morgan. "Evaluating Kohonen's learning rule: An approach through genetic algorithms". *European Journal of Operational Research*, vol. 154, no. 1, pp. 191–205, 2004.
- [2] S. P. Luttrell. "A Bayesian analysis of self-organizing maps". *Neural Computation*, vol. 6, no. 5, pp. 767–794, 1994.
- [3] T. N. W. D. H. Hubel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". *Journal of Physiology*, vol. 160(1), pp. 106–154, 1962.
- [4] T. Kohonen. "Self-organized formation of topologically correct feature maps". *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.
- [5] C. von der Malsburg. "Self-organization of orientation sensitive cells in the striate cortex". *Kybernetik*, vol. 14, pp. 85–100, 1973.
- [6] A. Flexer. "On the use of self-organizing maps for clustering and visualization". *Intelligent Data Analysis*, vol. 5, pp. 373–384, 2001.
- [7] K. S. H. Ritter. "Kohonen's self-organizing maps: exploring their computational capabilities". *Proceedings of the IEEE International Conference on Neural Networks (ICNN'88)*, pp. 109–116, 1988.
- [8] J. C. Fort. "SOM's mathematics". *Neural Networks*, vol. 19, no. 6–7, pp. 812–816, 2006.

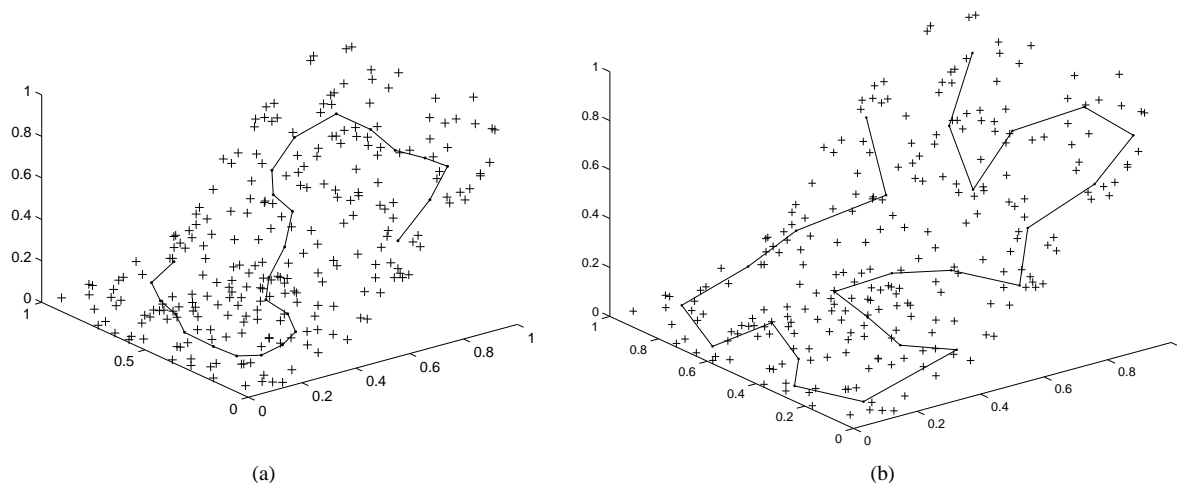


Figura 4: (a) Típico mapa 1×25 formado pelo algoritmo subida da encosta para o conjunto D3 ($\alpha = 1, \beta = 4, 0$). (b) Um típico mapa 1×25 formado pelo AE proposto para o conjunto D3 ($\alpha = 1, \beta = 3, 0$).

- [9] M. Cottrell, J. C. Fort and G. Pagès. “Theoretical aspects of the SOM algorithm”. *Neurocomputing*, vol. 21, no. 1-3, pp. 119–138, 1998.
- [10] H. Ritter and K. Schulten. “Convergence Properties of Kohonen’s Topology Conserving Maps: Fluctuations, Stability, and Dimension Selection”. *Biological Cybernetics*, vol. 60, pp. 59–71, 1988.
- [11] R. Kamimura. “Cooperative information maximization with Gaussian activation functions for self-organizing maps”. *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 909–918, 2006.
- [12] R. Linsker. “How to Generate Ordered Maps by Maximizing the Mutual Information between Input and Output Signals”. *Neural Computation*, vol. 1, no. 3, pp. 402–411, 1989.
- [13] P. András. “Kernel-Kohonen Networks”. *International Journal of Neural Systems*, vol. 12(2), pp. 117–136, 2001.
- [14] M. M. Van Hulle. “Maximum Likelihood Topographic Map Formation”. *Neural Computation*, vol. 17, no. 3, pp. 503–513, 2005.
- [15] L. Wiskott and T. J. Sejnowski. “Constrained Optimization for Neural Map Formation: A Unifying Framework for Weight Growth and Normalization”. *Neural Computation*, vol. 10, pp. 671–716, 1998.
- [16] T. Graepel, M. M. Burger and K. Obermayer. “Self-organizing maps: Generalizations and new optimization techniques”. *Neurocomputing*, vol. 21, pp. 173–190, 1998.
- [17] T. Heskes. “Self-organizing Maps, Vector Quantization and Mixture Modeling”. *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1299–1305, 2001.
- [18] S. A. Bamford, A. F. Murray and D. J. Willshaw. “Synaptic Rewiring for Topographic Map Formation”. In *Proceedings of the 2008 International Conference on Artificial Neural Networks (ICANN’08)*, volume LNCS-5164, pp. 218–227. Springer, 2008.
- [19] S. A. Bamford. “Synaptic Rewiring in Neuromorphic VLSI for Topographic Map Formation”. Ph.D. thesis, School of Informatics, University of Edinburgh, 2009.
- [20] R. Kamimura. “Collective Activations to Generate Self-Organizing Maps”. In *Advances in Neuro-Information Processing*, volume LNCS-5507, pp. 943–950. Springer, 2009.
- [21] J. S. Kirk and J. M. Zurada. “Motivation for a genetically-trained topography-preserving map”. In *Proceedings of the 2002 International Joint Conference on neural Networks (IJCNN’02)*, volume 1, pp. 394–399, 2002.
- [22] D. Polani. “On the optimization of self-organizing maps by genetic algorithms”. In *Kohonen Maps*, edited by E. Oja and S. Kaski, pp. 157–169. Elsevier, 1999.
- [23] J. M. Z. J. S. Kirk. “A two-stage algorithm for improved topography preservation in self-organizing maps”. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4, pp. 2527–32, 2000.

- [24] H. T. C. M.-C. Su. “Topological-Preserving Map Formation Achieved with a Genetic Algorithm and its Application in Cluster Analysis”. *Tamkang Journal of Science and Engineering*, vol. 2(1), pp. 29–36, 1999.
- [25] H. T. C. M.-C. Su. “Fast Self-Organizing Feature Map Algorithm”. *IEEE Transactions on Neural Networks*, vol. 11(3), pp. 721–733, 2000.
- [26] S. Haykin. *Neural Networks: A Comprehensive Foundation, 2Ed.* Prentice Hall PTR Upper Saddle River, NJ, USA, 1998.
- [27] J.-T. Y. J. Herbert. “GTSOM: Game Theoretic Self-Organizing Maps”. In *Trends in Neural Computation*, Springer, pp. 199–223, 2007.
- [28] H. Ritter and K. Schulten. “Topographic Map Formation as Statistical Inference”. In *L. Smith (ed), Neural Computation and Psychology*, pp. 86–96, 1995.
- [29] K. S. T. Martinez. “Topology Representing Networks”. *Neural Networks*, vol. 7(3), pp. 507–522, 1994.
- [30] M. M. V. Hulle. “Topology-preserving map formation achieved with a purely local unsupervised competitive learning rule”. *Neural Networks*, vol. 10(3), pp. 431–446, 1997.
- [31] R. Kamimura. “Unifying cost and information in information-theoretic competitive learning. Neural Networks”. *Neural Networks*, vol. 18(5-6), pp. 711–718, 2005.
- [32] A. S. Nissinen and H. Hyotyniemi. “Evolutionary self-organizing map”. In *Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT’98)*, volume 3, pp. 1596–600, 1998.
- [33] M. Chang, H. Yu and J. S. Heh. “Evolutionary self-organizing map”. In *Proceedings of the 1998 IEEE World Congress on Computational Intelligence (WCCI’98)*, volume 1, pp. 680–685, 1998.
- [34] S. Harp and T. Samad. “Genetic optimization of self-organizing feature maps”. *International Joint Conference on Neural Networks (IJCNN’91)*, vol. 1, pp. 341–346, 1991.
- [35] M. McInerney and A. Dhawan. “Training the self-organizing feature map using hybrids of genetic and Kohonen methods”. In *IEEE World Congress on Computational Intelligence (WCCI’04)*, volume 2, pp. 641–644, 1994.
- [36] J. Handl, J. Knowles and M. Dorigo. “Ant-Based Clustering and Topographic Mapping”. *Artificial Life*, vol. 12, no. 1, pp. 35–61, 2006.
- [37] M. Dorigo and M. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [38] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
- [39] J. Vesanto *et al.*. *SOM Toolbox for Matlab 5*. Helsinki University of Technology, 2000.