
PROJETO E ANÁLISE DE UMA REDE NEURAL PARA RESOLVER PROBLEMAS DE PROGRAMAÇÃO DINÂMICA

Ivan N. da Silva UNESP/FE/DEE CP 473, CEP 17033-360 Bauru – SP	Lúcia V. R. Arruda CPGEI/CEFET-PR CEP 80230-901 Curitiba – PR	Wagner C. do Amaral UNICAMP/FEEC/DCA CP 6101, CEP 13081-970 Campinas – SP	Mário E. Bordon UNESP/FE/DEE CP 473, CEP 17033-360 Bauru – SP
--	---	---	---

Resumo: Redes Neurais Artificiais são sistemas dinâmicos que possuem altas taxas de computação por utilizarem um número elevado de elementos processadores simples com alta taxa de conectividade entre si. Redes neurais com conexões retroalimentadas fornecem um modelo computacional capaz de resolver vários tipos de problemas de otimização. Este artigo apresenta uma nova abordagem para resolver problemas de programação dinâmica utilizando redes neurais artificiais. Mais especificamente, uma rede de Hopfield modificada é desenvolvida cujos parâmetros internos são computados utilizando a técnica de subespaço-válido de soluções. Estes parâmetros garantem a convergência da rede em direção aos pontos de equilíbrio que representam as soluções (não necessariamente ótimas) para o problema de programação dinâmica. Resultados de simulações são apresentados para validar a abordagem proposta.

Palavras Chaves: Redes neurais artificiais, programação dinâmica, redes de Hopfield, otimização de sistemas.

Abstract: Systems based on artificial neural networks have high computational rates due to the use of a massive number of simple processing elements and the high degree of connectivity between these elements. Neural networks with feedback connections provide a computing model capable of solving a large class of optimization problems. This paper presents a novel approach for solving dynamic programming problems using artificial neural networks. More specifically, a modified Hopfield network is developed and its internal parameters are computed using the valid-subspace technique. These parameters guarantee the convergence of the network to the equilibrium points which represent solutions (not necessarily optimal) for the dynamic programming problem. Simulated examples are presented and compared with other neural networks. The results demonstrate that proposed method gives a significant improvement.

Keywords: Artificial neural networks, dynamic programming, Hopfield networks, system optimization.

1 INTRODUÇÃO

A habilidade de Redes Neurais Artificiais (RNA) em solucionar problemas complexos e variados tem se tornado uma abordagem interessante que pode ser aplicada em diversas áreas de engenharia e ciências. Os sistemas baseados em RNA oferecem um método alternativo para solucionar problemas relacionados à otimização de sistemas, com um potencial expressivo para implementação em hardware. Entre os tipos de problemas de otimização que podem ser resolvidos por redes neurais artificiais destacam-se: problemas de otimização combinatorial (Hopfield e Tank, 1985), problemas de programação linear (Zak *et alii*, 1995), problemas de programação não-linear (Kennedy e Chua, 1988; Romero, 1993; Tagliarini *et alii*, 1991), problemas de roteamento em redes de comunicações (Rauch e Winarske, 1988), e problemas de otimização irrestrita (Jeffery e Rosner, 1986; Levy e Adams, 1987).

Normalmente, a resolução de problemas de otimização através da técnica de programação dinâmica envolve a utilização de relações dinâmicas de recorrência desenvolvidas principalmente por Richard Bellman (Hillier e Lieberman, 1980). Embora a programação dinâmica seja uma técnica poderosa para resolver uma variedade muito grande de problemas de otimização, sua utilização tem sido limitada devido a relativa ineficiência computacional (e.g. tempo de processamento, memória) exigida pelo algoritmo tradicional (Kirk, 1970). As redes neurais artificiais tornam-se portanto uma abordagem alternativa que pode ser aplicada eficientemente neste tipos de problemas. Entre as principais vantagens em se utilizar a abordagem de redes neurais artificiais em programação dinâmica, destacam-se: i) a capacidade intrínseca de operação em paralelo; ii) a simplicidade de implementação em hardware; e iii) o alcance de altas taxas de computação por intermédio de elementos simples de processamento.

Vários problemas práticos de engenharia que modelam sistemas do mundo real podem ser resolvidos através da aplicação de programação dinâmica. Entre as principais abordagens de redes neurais artificiais aplicadas em problemas de programação dinâmica para engenharia elétrica pode-se destacar: a abordagem neuro-nebulosa utilizada no controle de potência reativa de uma subestação (Hsu e Lu, 1998); a rede neural projetada para alocação dinâmica de largura de banda

Artigo Submetido em 05/02/99

1a. Revisão em 17/06/99; 2a. Revisão em 05/01/00; 3a. Revisão em 20/09/00

Aceito sob recomendação da Ed. Consultora Profa. Dra. Sandra A. Sandri

em redes de comunicações (Bolla *et alii*, 1998); a técnica neuro-fuzzy usada em controle de tráfego e transporte (Henry *et alii*, 1998); e o projeto de redes neurais que implementam um modelo de programação dinâmica nebulosa (Kacprzyk *et alii*, 1999).

Um problema típico de programação dinâmica pode ser modelado como um conjunto constituído de um nó fonte e um nó destino, com n estágios intermediários e m estados em cada estágio, conforme ilustrado na Figura 1 para $n = 4$ e $m = 3$. O custo associado da transição do i -ésimo estado pertencente ao x -ésimo estágio para o j -ésimo estado pertencente ao estágio seguinte, $x+1$, é dado pela variável constituída de índices duplos denotada por $d_{xi, (x+1)j}$, onde x é o índice dos estágios, i e j são os índices dos estados em cada estágio. Como ilustração, para a Figura 1, o custo d seria denotado por $d_{11,23}$. O objetivo então é encontrar um caminho composto de um e somente um estado em cada estágio, partindo da fonte em direção ao destino (abordagem forward), e que satisfaça algum critério de otimização.

Assim, o objetivo do problema de programação dinâmica considerado neste artigo é encontrar o menor caminho entre todos os caminhos possíveis (problema do caminho mínimo – shortest path problem), que se inicializa no nó fonte e finaliza no nó destino, passando-se por um único estado em cada estágio. Admite-se que cada nó de estado possa ser convenientemente considerado como um elemento processador (neurônio) do i -ésimo estado pertencente ao x -ésimo estágio.

Para este tipo de problema, as principais abordagens neurais utilizadas são o modelo baseado em redes neurais primais/duais proposto por Wang (1998), e a arquitetura neural recorrente desenvolvida por Chiu *et alii* (1991). Nestas redes, os termos de otimização e de restrições associados com o problema de programação dinâmica são tratados em um único estágio. Consequentemente, torna-se necessário especificar corretamente os parâmetros de ponderação associados a cada termo de restrição, e também estipular mecanismos com o objetivo de evitar a interferência dos termos de restrições com o de otimização. Para contornar estes problemas, que dificultam o processo de convergência da rede, desenvolve-se a rede de Hopfield modificada que é aplicada na solução de problemas de programação dinâmica.

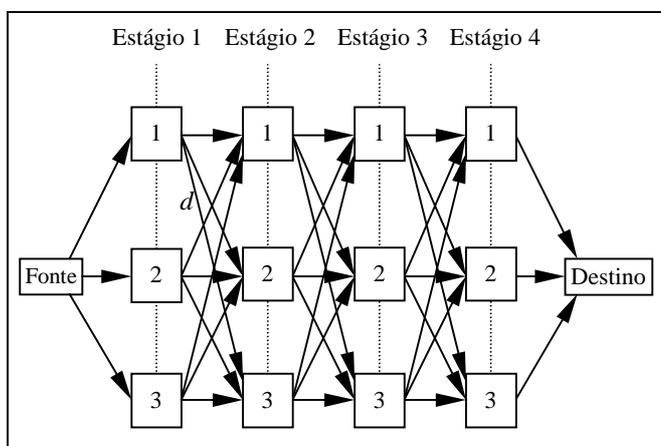


Figura 1. O Problema de Programação Dinâmica.

A rede de Hopfield modificada foi desenvolvida com o objetivo de que seus pontos de equilíbrio correspondem à solução do problema de programação dinâmica considerado. As principais características desta rede são: i) inexistência de

constantes de ponderação; ii) todas as restrições estruturais envolvidas ao problema de programação dinâmica são agrupadas em um único termo de restrição; iii) não há interferência entre termo de otimização e termo de restrição; e iv) nenhum parâmetro de inicialização (controle) é requerido para a execução da simulação. Uma análise do comportamento dinâmico da rede será também realizada para ilustrar o processo de convergência da rede em direção às soluções ótimas (pontos de equilíbrio).

2 A REDE DE HOPFIELD MODIFICADA

Redes neurais artificiais fornecem um novo método para resolver problemas de programação dinâmica por explorar arquiteturas de processamento inerentemente paralelas e adaptativas (Haykin, 1994; Cichoki e Unbehaven, 1993). Neste trabalho, desenvolve-se uma rede de Hopfield modificada cujos pontos de equilíbrio correspondem à solução do problema. Como descrito em Hopfield e Tank (1985), as redes de Hopfield são redes de uma única camada com conexões realimentadas entre os nós. A equação nodal para uma rede de Hopfield com N neurônios é dada por:

$$\dot{u}_i(t) = -\eta u_i(t) + \sum_{j=1}^N T_{ij} v_j(t) + i_i^b \quad (1)$$

$$v_i(t) = g(u_i(t)) \quad (2)$$

onde:

$u_i(t)$ é o estado corrente do i -ésimo neurônio.

$v_i(t)$ é a saída do i -ésimo neurônio.

i_i^b indica a entrada correspondente ao i -ésimo neurônio.

$\eta u_i(t)$ é um termo de decaimento passivo.

T_{ij} é o peso de conexão entre o i -ésimo e j -ésimo neurônio.

Na equação (2), $g(u_i(t))$ é uma função monótona crescente que limita a saída de cada neurônio para um valor predefinido. Mostra-se em Tank e Hopfield (1986) e Silva *et alii* (1997) que se T é simétrica e $\eta=0$, os pontos de equilíbrio da rede correspondem aos valores de $v(t)$ que minimizam a função de energia associada à rede. Esta função de energia é dada por:

$$E(t) = -\frac{1}{2} v(t)^T T v(t) - v(t)^T i^b \quad (3)$$

Portanto, o mapeamento do problema de programação dinâmica utilizando uma rede de Hopfield consiste em determinar a matriz de pesos T e o vetor i^b para se obter os pontos de equilíbrio da rede que correspondem às soluções ótimas do problema. A função de energia $\{E^m(t)\}$ utilizada neste artigo será composta de dois termos de energia como definido a seguir:

$$E^m(t) = E^{conf}(t) + E^{ot}(t) \quad (4)$$

onde: $E^{conf}(t)$ é um termo de confinamento que agrupa as restrições estruturais (um e apenas um neurônio deve estar ativo em cada estágio) impostas pelo problema de programação dinâmica, e $E^{ot}(t)$ é um termo de otimização que conduz a saída da rede para os pontos de equilíbrio que correspondem à restrição de custo (comprimento total do caminho deve ser mínimo). Assim, a minimização de $E^m(t)$ da rede de Hopfield modificada é conduzida através de duas fases distintas:

i) minimização do termo $E^{conf}(t)$:

$$E^{conf}(t) = -\frac{1}{2} v(t)^T T^{conf} v(t) - v(t)^T i^{conf} \quad (5)$$

onde: $v(t)$ é a saída da rede, T^{conf} é a matriz de pesos e i^{conf} é o vetor de entradas pertencentes a E^{conf} . Esta operação

corresponde ao confinamento de $\mathbf{v}(t)$ para um subespaço-válido que confina as restrições estruturais impostas pelo problema.

ii) minimização do termo $E^{ot}(t)$:

$$E^{ot}(t) = -\frac{1}{2}\mathbf{v}(t)^T \mathbf{T}^{ot} \mathbf{v}(t) - \mathbf{v}(t)^T \mathbf{i}^{ot} \quad (6)$$

onde: \mathbf{T}^{ot} é a matriz de pesos e \mathbf{i}^{ot} é o vetor de entradas pertencentes ao termo E^{ot} . Esta operação corresponde em mover $\mathbf{v}(t)$ em direção a uma solução ótima representada pelos pontos de equilíbrio da rede.

Para uma grande variedade de problemas com restrições que podem ser resolvidos pela rede de Hopfield, através da decomposição da função de energia da rede (4) nas duas parcelas dadas nas equações (5) e (6), verifica-se que os pontos de equilíbrio da rede, correspondentes aos valores de \mathbf{v} que minimizam a função de energia E^{conf} dada pela equação (5), pertencem todos a um mesmo subespaço-comum (Aiyer *et alii*, 1990). Este subespaço comum, denominado subespaço-válido de soluções, possui equação definida por:

$$\mathbf{v}(t+1) = \mathbf{T}^{val} \mathbf{v}(t) + \mathbf{s} \quad (7)$$

onde:

- \mathbf{T}^{val} é uma matriz projeção (isto é: $\mathbf{T}^{val} \mathbf{T}^{val} = \mathbf{T}^{val}$) que projeta o vetor \mathbf{v} dentro do subespaço-válido ($\mathbf{v}^{val} = \mathbf{T}^{val} \mathbf{v}$, onde \mathbf{v}^{val} é a componente de \mathbf{v} projetada sobre o subespaço-válido).

- \mathbf{s} é um vetor que está relacionado com as restrições do problema a ser resolvido. Este vetor é ortogonal ao subespaço-válido ($\mathbf{T}^{val} \mathbf{s} = \mathbf{0}$).

Utilizando a abordagem de subespaço-válido, a operação da rede de Hopfield modificada consiste de três passos principais como descrito na Figura 2:

Passo (I): Minimização de E^{conf} : corresponde à projeção de $\mathbf{v}(t)$ no subespaço-válido gerado por todas as restrições estruturais impostas pelo problema. Este subespaço é definido por:

$$\mathbf{v}(t+1) = \mathbf{T}^{val} \mathbf{v}(t) + \mathbf{s} = \mathbf{T}^{conf} \mathbf{v} + \mathbf{i}^{conf} \quad (8)$$

onde: \mathbf{T}^{val} é uma matriz projeção ($\mathbf{T}^{val} \mathbf{T}^{val} = \mathbf{T}^{val}$) e $\mathbf{T}^{val} \mathbf{s} = \mathbf{0}$. Esta operação corresponde a uma minimização indireta de $E^{conf}(t)$, isto é, $\mathbf{T}^{conf} = \mathbf{T}^{val}$ e $\mathbf{i}^{conf} = \mathbf{s}$. Uma descrição detalhada da técnica do subespaço-válido é apresentada em (Aiyer *et alii*, 1990; Silva *et alii*, 1997).

Passo (II): Aplicação de uma função de ativação ‘rampa-simétrica’ restringindo $\mathbf{v}(t)$ dentro de um hiper-cubo:

$$g^r(v_i) = \begin{cases} 1, & \text{se } v_i > 1 \\ v_i, & \text{se } 0 \leq v_i \leq 1 \\ 0, & \text{se } v_i < 0 \end{cases} \quad (9)$$

onde $v_i \in [0,1]$. Este passo assegura que os elementos do vetor $\{\mathbf{v}\}$ de saída da rede estejam sempre entre 0 e 1. Após a convergência do loop interno (Figura 2), o qual é realizado através das aplicações sucessivas dos passos (I) e (II), a rede está pronta para executar um passo de otimização (Passo III).

Passo (III): Minimização de E^{ot} : consiste na alteração de $\mathbf{v}(t)$ em direção a uma solução ótima (definida por \mathbf{T}^{ot} e \mathbf{i}^{ot}) que é alcançada com a minimização de E^{ot} . Neste caso, o termo E^{ot} representa uma função custo para o problema de programação dinâmica. Utilizando a função de ativação ‘rampa-simétrica’ e assumindo $\eta=0$, a equação (2) torna-se:

$$\mathbf{v}(t) = g^r(\mathbf{u}(t)) = \mathbf{u}(t)$$

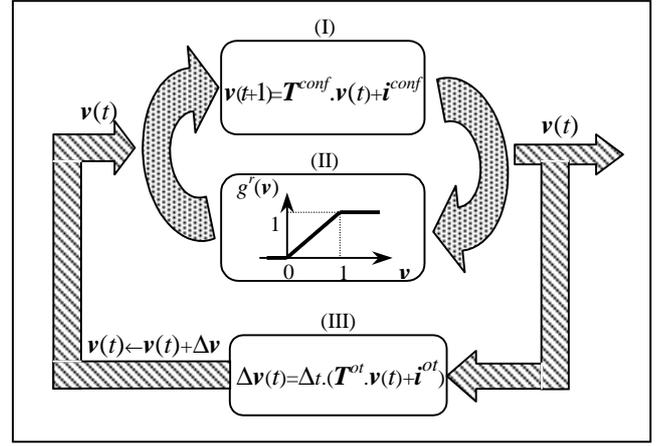


Figura 2. A Rede de Hopfield Modificada.

Comparando esta equação com (1) e (6), tem-se:

$$\frac{d\mathbf{v}(t)}{dt} = \dot{\mathbf{v}} = -\frac{\partial E^{ot}(t)}{\partial \mathbf{v}} \quad (10)$$

$$\Delta \mathbf{v} = -\Delta t \cdot \nabla E^{ot}(\mathbf{v}) = \Delta t \cdot (\mathbf{T}^{ot} \cdot \mathbf{v} + \mathbf{i}^{ot})$$

Como observado na Equação (10), a minimização de E^{ot} consiste então na alteração de $\mathbf{v}(t)$ na direção oposta ao gradiente de E^{ot} . Então, as aplicações sucessivas do passo (III), seguidas pelas execuções dos passos (I) e (II), levam a saída da rede para um ponto de equilíbrio que corresponde a uma solução localmente ótima do problema de programação dinâmica. Estes resultados são também válidos quando uma função de ativação do tipo ‘tangente hiperbólica’ é utilizada (Abe, 1993).

Assim, cada iteração da rede de Hopfield modificada possui dois estágios distintos. Primeiro, como descrito no Passo (III), \mathbf{v} é alterado utilizando apenas o gradiente do termo E^{ot} . Segundo, após cada iteração, \mathbf{v} é diretamente projetado no subespaço-válido. Este é um processo iterativo, em que \mathbf{v} é ortogonalmente projetado no subespaço-válido (Passo I), e em seguida os seus elementos são restringidos no domínio $[0,1]$ através da função de ativação ‘rampa-simétrica’ (Passo II). Portanto, o mapeamento de problemas de programação dinâmica através da rede de Hopfield modificada consiste em obter as matrizes de pesos \mathbf{T}^{ot} e \mathbf{T}^{conf} , e os vetores \mathbf{i}^{ot} e \mathbf{i}^{conf} . Estas expressões são determinadas na Seção 4. Na próxima seção, a análise da dinâmica da rede no contexto do subespaço-válido é apresentada.

3 ANÁLISE DA DINÂMICA DA REDE DE HOPFIELD MODIFICADA

Nesta seção, analisa-se a convergência da rede de Hopfield modificada cuja dinâmica de operação é implementada através dos passos (I)-(III) apresentados anteriormente. Em particular, considera-se que a região de operação na qual o vetor \mathbf{v} está contido é limitada pelo hiper-cubo definido pela função de ativação ‘rampa-simétrica’ (9). A equação nodal descrevendo o comportamento dinâmico desta rede é obtida a partir de (1) para $\eta = 0$ e $\mathbf{v}(t) = \mathbf{u}(t)$, ou seja:

$$\dot{v}_i(t) = \sum_{j=1}^N T_{ij} v_j(t) + i_i^b$$

A análise dinâmica da rede é portanto inicializada com a obtenção de uma equação para \dot{v}^{conf} , componente de $\dot{\mathbf{v}}$ que pertence ao subespaço-válido.

Como na rede de Hopfield modificada a saída \mathbf{v} é constantemente confinada ao subespaço-válido pela aplicação do passo (I) descrito na Seção 2, isto é $\{\mathbf{v} = \mathbf{T}^{conf} \cdot \mathbf{v} + \mathbf{i}^{conf}\}$, então qualquer componente de $\dot{\mathbf{v}}$ ortogonal a $\dot{\mathbf{v}}^{conf}$ é continuamente suprimida. Logo, a componente $\dot{\mathbf{v}}^{conf}$ (não $\dot{\mathbf{v}}$) caracteriza melhor a dinâmica global da rede. Assim, tem-se:

$$\begin{aligned} \dot{\mathbf{v}}^{conf} &= \mathbf{T}^{conf} \cdot \dot{\mathbf{v}} = \mathbf{T}^{conf}(\mathbf{T}^{ot} \cdot \mathbf{v} + \dot{\mathbf{i}}^{ot}) \\ &= \mathbf{T}^{conf}(\mathbf{T}^{ot}(\mathbf{T}^{conf} \cdot \mathbf{v} + \mathbf{i}^{conf}) + \dot{\mathbf{i}}^{ot}) \\ &= \mathbf{T}^{conf} \mathbf{T}^{ot} \mathbf{T}^{conf} \cdot \mathbf{v} + \mathbf{T}^{conf}(\mathbf{T}^{ot} \cdot \dot{\mathbf{i}}^{conf} + \dot{\mathbf{i}}^{ot}) \end{aligned} \quad (11)$$

A partir da equação (11), verifica-se que a componente $\dot{\mathbf{v}}^{conf}$ é constituída de duas partes, um termo constante $\{\mathbf{T}^{conf}(\mathbf{T}^{ot} \cdot \dot{\mathbf{i}}^{conf} + \dot{\mathbf{i}}^{ot})\}$; e um termo dependente de \mathbf{v} , $\{\mathbf{T}^{conf} \mathbf{T}^{ot} \mathbf{T}^{conf} \cdot \mathbf{v}\}$. Estas expressões podem ser simplificadas por:

$$\mathbf{T}^{conf} \mathbf{T}^{ot} \mathbf{T}^{conf} = \mathbf{A} \quad (12)$$

$$\mathbf{T}^{conf}(\mathbf{T}^{ot} \cdot \dot{\mathbf{i}}^{conf} + \dot{\mathbf{i}}^{ot}) = \mathbf{b} \quad (13)$$

Substituindo (12) e (13) em (11), obtém-se:

$$\dot{\mathbf{v}}^{conf} = \mathbf{A} \cdot \mathbf{v} + \mathbf{b} = \mathbf{A} \cdot \mathbf{v}^{conf} + \mathbf{b} \quad (14)$$

onde: $\mathbf{v}^{conf} = \mathbf{T}^{conf} \cdot \mathbf{v}$ e $\mathbf{T}^{conf} \cdot \mathbf{T}^{conf} = \mathbf{T}^{conf}$.

Com \mathbf{v} confinado ao subespaço-válido (isto é: $\mathbf{v} = \mathbf{T}^{conf} \cdot \mathbf{v} + \mathbf{i}^{conf}$ e $\mathbf{T}^{conf} \cdot \dot{\mathbf{i}}^{conf} = \mathbf{0}$), E^{ot} pode ser expressado como:

$$E^{ot} = -\frac{1}{2} \mathbf{v}^T \mathbf{A} \mathbf{v} - \mathbf{b}^T \mathbf{v} \quad (15)$$

Na equação (15) observa-se que a dinâmica de $\dot{\mathbf{v}} = \dot{\mathbf{v}}^{conf} = \mathbf{A} \cdot \mathbf{v} + \mathbf{b}$ simplesmente resulta num processo de otimização de E^{ot} em relação ao subespaço-válido. Logo, o objetivo consiste em obter uma solução válida que minimiza E^{ot} .

Corolário: A solução geral de um sistema linear invariante no tempo (autônomo) do tipo descrito por $\dot{\mathbf{v}}^{conf} = \mathbf{A} \cdot \mathbf{v}^{conf} + \mathbf{b}$, com $\mathbf{v}^{conf}(0) = \mathbf{v}_0^{conf}$ é dada por:

$$\mathbf{v}^{conf}(t) = e^{\mathbf{A}t} \mathbf{v}_0^{conf} + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{b} d\tau \quad (16)$$

onde \mathbf{v}_0^{conf} é um vetor de elementos aleatórios pequenos correspondendo ao valor de \mathbf{v}^{conf} no instante de tempo $t = 0$.

Definição: A exponencial de uma matriz quadrada \mathbf{A} é definida pela série de potências dada por:

$$e^{\mathbf{A}\zeta} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}\zeta)^k}{k!}$$

Então, rescrevendo a equação (16) como séries de potências, tem-se:

$$\begin{aligned} \mathbf{v}^{conf}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{conf} + \int_0^t \sum_{k=0}^{\infty} \frac{(t-\tau)^k}{k!} \mathbf{A}^k \mathbf{b} d\tau \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{conf} + \sum_{k=0}^{\infty} \frac{\mathbf{A}^k \mathbf{b}}{k!} \int_0^t (t-\tau)^k d\tau \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{conf} + \sum_{k=0}^{\infty} \frac{\mathbf{A}^k \mathbf{b}}{k!} \left(\frac{t^{k+1}}{k+1} \right) \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{conf} + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} \mathbf{A}^k \mathbf{b} \end{aligned} \quad (17)$$

Para analisar o comportamento de \mathbf{v}^{conf} durante o processo de convergência da rede, considera-se os vetores \mathbf{v}^{conf} , \mathbf{v}_0^{conf} e \mathbf{b} escritos em termos de suas componentes que serão expressas no espaço coordenado gerado a partir dos autovetores normalizados de \mathbf{A} . Assim, considera-se que \mathbf{A} tem autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$, cujos autovetores normalizados são $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$. Para distinguir os autovalores nulos e não-nulos de \mathbf{A} , define-se o conjunto \mathbf{Z} tal que $\lambda_i = 0$ para $i \in \mathbf{Z}$ e $\lambda_i \neq 0$ para $i \notin \mathbf{Z}$. Decompondo-se \mathbf{v}^{conf} , \mathbf{v}_0^{conf} e \mathbf{b} ao longo dos autovetores de \mathbf{A} , tem-se:

$$\mathbf{v}^{conf} = \sum_{i=1}^n v_i \mathbf{u}_i, \quad \mathbf{v}_0^{conf} = \sum_{i=1}^n o_i \mathbf{u}_i, \quad \mathbf{b} = \sum_{i=1}^n b_i \mathbf{u}_i \quad (18)$$

onde v_i , o_i e b_i referem-se respectivamente aos valores da i -ésima componente de \mathbf{v}^{conf} , \mathbf{v}_0^{conf} e \mathbf{b} , representado no espaço coordenado gerado a partir dos autovetores de \mathbf{A} . Portanto, a partir de (18), obtém-se:

$$\mathbf{A}^k \mathbf{v}_0^{conf} = \sum_{i=1}^n o_i \lambda_i^k \mathbf{u}_i, \quad \mathbf{A}^k \mathbf{b} = \sum_{i=1}^n b_i \lambda_i^k \mathbf{u}_i \quad (19)$$

Substituindo a equação (19) em (17) resulta-se em:

$$\begin{aligned} \mathbf{v}^{conf}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \sum_{i=1}^n o_i \lambda_i^k \mathbf{u}_i + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} \sum_{i=1}^n b_i \lambda_i^k \mathbf{u}_i \\ &= \sum_{i=1}^n o_i \mathbf{u}_i \sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!} + \sum_{i \notin \mathbf{Z}} \frac{b_i \mathbf{u}_i}{\lambda_i} \sum_{k=0}^{\infty} \frac{t^{k+1} \lambda_i^{k+1}}{(k+1)!} + \sum_{i \in \mathbf{Z}} b_i \mathbf{u}_i \sum_{k=0}^{\infty} \frac{t^{k+1} 0^k}{(k+1)!} \\ &= \sum_{i=1}^n e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin \mathbf{Z}} \frac{b_i \mathbf{u}_i}{\lambda_i} \left(\sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!} - 1 \right) + \sum_{i \in \mathbf{Z}} b_i \mathbf{u}_i t \\ &= \sum_{i=1}^n e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin \mathbf{Z}} \frac{b_i \mathbf{u}_i}{\lambda_i} (e^{\lambda_i t} - 1) + \sum_{i \in \mathbf{Z}} b_i \mathbf{u}_i t \end{aligned} \quad (20)$$

A equação (20) é completamente geral para quaisquer valores arbitrários de \mathbf{A} , \mathbf{b} e \mathbf{v}_0^{conf} . Entretanto, esta equação pode ser simplificada se \mathbf{A} e \mathbf{b} forem definidas como nas equações (12) e (13). Neste caso, os autovetores de \mathbf{A} com autovalores zero serão confinados ao transpor o subespaço ortogonal ao subespaço-válido, enquanto \mathbf{b} sempre permanece no subespaço-válido $\{b_i = 0 \text{ para } i \in \mathbf{Z}\}$. Assim, a equação (20) torna-se:

$$\mathbf{v}^{conf}(t) = \sum_{i=1}^n e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin \mathbf{Z}} \frac{b_i \mathbf{u}_i}{\lambda_i} (e^{\lambda_i t} - 1) \quad (21)$$

Agora torna-se também importante examinar a equação (21) para valores pequeno e grande de t . Para t suficientemente pequeno, tem-se a seguinte aproximação:

$$e^{\lambda_i t} \approx 1 + \lambda_i t$$

Substituindo esta aproximação em (21), tem-se:

$$\mathbf{v}^{conf}(t) \approx \sum_{i=1}^n [o_i(1 + \lambda_i t) + b_i t] \mathbf{u}_i \quad (22)$$

Observa-se que para um \mathbf{v}_0^{conf} aleatoriamente pequeno, os termos o_i são frequentemente pequenos em comparação com os b_i . Logo, a equação (22) transforma-se em:

$$\mathbf{v}^{conf}(t) \approx t \sum_{i=1}^n b_i \mathbf{u}_i = t \mathbf{b} \quad (23)$$

Analisando a equação (23), nota-se que \mathbf{v}^{conf} inicialmente parte na direção do vetor \mathbf{b} . No limite, quando t é muito grande, a equação (21) indica que \mathbf{v}^{conf} tenderá em direção aos autovetores de A correspondente ao maior autovalor positivo. Neste caso, a partir dos resultados advindos de Aiyer *et alli* (1990) e Vidyasagar (1992), os pontos de equilíbrio da rede podem ser iterativamente computados visto que o estado da rede partindo de uma posição inicial arbitrária sempre convergirá para um ponto de equilíbrio estável. Este ponto de equilíbrio é sempre limitado pela função de ativação ‘rampa-simétrica’ definida em (9).

Embora o processo de convergência em direção aos pontos de equilíbrio seja garantido, deve-se observar que isto não implica na obtenção de um ponto de mínimo global em relação à função de energia $E^n(t)$ associada à rede de Hopfield modificada. Entretanto, para todas as simulações realizadas neste trabalho, a rede sempre converge para um ponto de energia mínima que corresponde a uma solução razoável (próxima da solução globalmente ótima) do problema de programação dinâmica.

4 MAPEANDO PROBLEMAS DE PROGRAMAÇÃO DINÂMICA ATRAVÉS DA REDE DE HOPFIELD MODIFICADA

Nesta seção, introduz-se inicialmente a notação empregada para vetores e matrizes envolvidos com o mapeamento de problemas de programação dinâmica, e também define-se as propriedades de produtos de Kronecker que serão usadas. Em seguida, utiliza-se esta notação na derivação das expressões para \mathbf{T}^{conf} , \mathbf{T}^{ot} , \mathbf{i}^{conf} e \mathbf{i}^{ot} .

Notação e Definições

Considera-se que o vetor $\mathbf{p} \in \mathfrak{R}^n$ representa uma solução de um problema de programação dinâmica com n estágios e m estados em cada estágio. O valor p_i fornece a posição do estado ativo no i -ésimo estágio. Assim, os elementos pertencentes a \mathbf{p} possuem valores inteiros definidos por:

$$p_i \in \{1, \dots, m\}, \text{ onde } i \in \{1..n\} \quad (24)$$

Como uma ilustração, considera-se o problema de programação dinâmica com 5 estágios e 4 estados em cada estágio ($n=5$ e $m=4$), cuja solução final é dada por: estado 3 no estágio 1, estado 2 no estágio 2, estado 4 no estágio 3, estado 4 no estágio 4, e estado 1 no estágio 5. O vetor \mathbf{p} representando esta solução seria dado por:

$$\mathbf{p} = [3 \ 2 \ 4 \ 4 \ 1]^T$$

Consequentemente, o vetor \mathbf{p} pode ser representado por um vetor \mathbf{v} , composto de uns e zeros, que representa a saída da rede. Esta representação será feita utilizando a notação por produtos de Kronecker (Graham, 1981). Utilizando esta notação, tem-se as seguintes definições:

• δ é uma matriz ($\delta \in \mathfrak{R}^{n \times n}$) definida por:

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (25)$$

• $\delta(k)$ é um vetor coluna ($\delta(k) \in \mathfrak{R}^n$) correspondente à k -ésima coluna da matriz δ . Os elementos deste vetor são definidos por:

$$\delta_i(k) = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases} \quad (26)$$

• $\mathbf{v}(\mathbf{p})$ é um vetor de dimensão $n.m$ representando a forma final do vetor \mathbf{v} , relacionado à saída da rede, e que corresponde à solução do problema fornecido por \mathbf{p} . O vetor $\mathbf{v}(\mathbf{p})$ é portanto definido por:

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \vdots \\ \delta(p_n) \end{bmatrix} \quad (27)$$

• $\mathbf{V}(\mathbf{p})$ é uma matriz de dimensão $n \times m$ definida por:

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} \delta(p_1)^T \\ \delta(p_2)^T \\ \vdots \\ \delta(p_n)^T \end{bmatrix} \quad (28)$$

onde $[\mathbf{V}(\mathbf{p})]_{ij} = [\delta(p_i)]_j$.

• $\mathbf{P} \otimes \mathbf{Q}$ denota o produto de Kronecker de duas matrizes. Se \mathbf{P} é uma matriz $n \times n$ e \mathbf{Q} é uma matriz $m \times m$, então $(\mathbf{P} \otimes \mathbf{Q})$ é uma matriz $(n.m) \times (n.m)$ dada por:

$$\mathbf{P} \otimes \mathbf{Q} = \begin{bmatrix} P_{11}\mathbf{Q} & P_{12}\mathbf{Q} & \dots & P_{1n}\mathbf{Q} \\ P_{21}\mathbf{Q} & P_{22}\mathbf{Q} & \dots & P_{2n}\mathbf{Q} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1}\mathbf{Q} & P_{n2}\mathbf{Q} & \dots & P_{nn}\mathbf{Q} \end{bmatrix} \quad (29)$$

Para o exemplo ilustrativo descrito anteriormente, admitindo que a solução final para um problema de programação dinâmica seja então dado por $\mathbf{p} = [3 \ 2 \ 4 \ 4 \ 1]^T$, os valores dos elementos do vetor $\mathbf{v}(\mathbf{p})$, $\delta(k)$ e da matriz $\mathbf{V}(\mathbf{p})$ seriam dados por:

$$\mathbf{v}(\mathbf{p}) = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$$

$$\begin{matrix} \delta(p_1)^T & \delta(p_2)^T & \delta(p_3)^T & \delta(p_4)^T & \delta(p_5)^T \end{matrix}$$

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

• $\mathbf{w} \otimes \mathbf{h}$ denota o produto de Kronecker de dois vetores. Se \mathbf{v} é um vetor de n elementos e \mathbf{h} é um vetor de m elementos, então $(\mathbf{w} \otimes \mathbf{h})$ é um vetor de $n.m$ elementos dado por:

$$\mathbf{w} \otimes \mathbf{h} = \begin{bmatrix} w_1 \mathbf{h} \\ w_2 \mathbf{h} \\ \vdots \\ w_n \mathbf{h} \end{bmatrix} \quad (30)$$

• $\text{vec}(\mathbf{U})$ é uma função que mapea uma matriz \mathbf{U} de dimensão $m \times n$ em um vetor \mathbf{v} de $n.m$ -elementos. Esta função é definida por:

$$\mathbf{v} = \text{vec}(\mathbf{U}) = [U_{11} \ U_{21} \dots \ U_{m1} \ U_{12} \ U_{22} \dots \ U_{m2} \ U_{1n} \ U_{2n} \dots \ U_{mn}]^T \quad (31)$$

As propriedades de produtos de Kronecker que serão utilizadas são (Graham, 1981):

$$(\mathbf{w} \otimes \mathbf{h})^T (\mathbf{x} \otimes \mathbf{g}) = (\mathbf{w}^T \mathbf{x}) (\mathbf{h}^T \mathbf{g}) \quad (32)$$

$$(\mathbf{P} \otimes \mathbf{Q})(\mathbf{w} \otimes \mathbf{h}) = (\mathbf{P}\mathbf{w} \otimes \mathbf{Q}\mathbf{h}) \quad (33)$$

$$(\mathbf{P}^T \otimes \mathbf{Q}^T) = (\mathbf{P} \otimes \mathbf{Q})^T \quad (34)$$

$$(\mathbf{P} \otimes \mathbf{Q})(\mathbf{E} \otimes \mathbf{F}) = (\mathbf{PE} \otimes \mathbf{QF}) \quad (35)$$

$$\text{vec}(\mathbf{Q} \cdot \mathbf{V} \cdot \mathbf{P}^T) = (\mathbf{P} \otimes \mathbf{Q}) \cdot \text{vec}(\mathbf{V}) \quad (36)$$

• \mathbf{o}^n e \mathbf{O}^n são respectivamente um vetor de n elementos e uma matriz de dimensão $n \times n$ com valores unitários, isto é:

$$\left. \begin{array}{l} [\mathbf{o}]_i = 1 \\ [\mathbf{O}]_{ij} = 1 \end{array} \right\} \text{para } i, j \in \{1..n\} \quad (37)$$

• \mathbf{R}^n é uma matriz projeção (isto é: $\mathbf{R}^n \cdot \mathbf{R}^n = \mathbf{R}^n$) de dimensão $n \times n$ definida por:

$$\mathbf{R}^n = \mathbf{I}^n - \frac{1}{n} \mathbf{O}^n \quad (38)$$

onde \mathbf{I}^n é a matriz identidade de ordem n .

A pós-multiplicação por \mathbf{R}^n transforma a soma dos elementos de cada linha de uma matriz para zero, enquanto a pré-multiplicação transforma a soma dos elementos de cada coluna para zero.

Definindo as Equações de \mathbf{T}^{conf} e \mathbf{i}^{conf}

As equações de \mathbf{T}^{conf} e \mathbf{i}^{conf} são desenvolvidas com o objetivo de forçar a validade das restrições estruturais. Para o problema de programação dinâmica, com n estágios e m estados, estas restrições implicam que um e apenas um estado em cada estágio pode estar ativado. Assim, cada linha da matriz $\mathbf{V}(\mathbf{p})$, obtida após a convergência da rede, deve conter um único valor 1. Matematicamente, define-se a matriz $\mathbf{V}(\mathbf{p})$ por:

$$\left. \begin{array}{l} [\mathbf{V}(\mathbf{p})]_{ij} \in \{1,0\} \\ \sum_{j=1}^m [\mathbf{V}(\mathbf{p})]_{ij} = 1 \end{array} \right\} \quad (39)$$

Neste caso, um subespaço-válido ($\mathbf{V} = \mathbf{T}^{conf} \cdot \mathbf{V} + \mathbf{i}^{conf}$) para o problema de programação dinâmica pode ser representado inicialmente por:

$$\mathbf{i}^{conf} = \mathbf{V} = \frac{1}{m} \mathbf{o}^n \cdot \mathbf{o}^m \quad (40)$$

A equação (40) garante que a soma dos elementos de cada linha da matriz \mathbf{V} possui valor igual a 1. Portanto, a matriz resultante da multiplicação de \mathbf{T}^{conf} por \mathbf{V} em $\{\mathbf{V} = \mathbf{T}^{conf} \cdot \mathbf{V} + \mathbf{i}^{conf}\}$ deve possuir elementos iguais a zero a fim de garantir que a soma dos elementos de cada linha de \mathbf{V} seja igual a 1. Usando as propriedades da matriz \mathbf{R} definida em (38), tem-se:

$$\mathbf{V} \cdot \mathbf{R}^m = \mathbf{T}^{conf} \cdot \mathbf{V} \quad (41)$$

$$\mathbf{I}^n \cdot \mathbf{V} \cdot \mathbf{R}^m = \mathbf{T}^{conf} \cdot \mathbf{V} \quad (42)$$

Substituindo as equações (40) e (42) na equação do subespaço-válido ($\mathbf{V} = \mathbf{T}^{conf} \cdot \mathbf{V} + \mathbf{i}^{conf}$), tem-se:

$$\mathbf{V} = \mathbf{I}^n \cdot \mathbf{V} \cdot \mathbf{R}^m + \frac{1}{m} \mathbf{o}^n \cdot \mathbf{o}^m \quad (43)$$

Aplicando o operador $\text{vec}(\cdot)$ dado por (36) na equação (43), obtém-se:

$$\text{vec}(\mathbf{V}) = \text{vec}(\mathbf{I}^n \cdot \mathbf{V} \cdot \mathbf{R}^m) + \frac{1}{m} \text{vec}(\mathbf{o}^n \cdot \mathbf{1} \cdot \mathbf{o}^m) \quad (44)$$

$$\text{vec}(\mathbf{V}) = (\mathbf{I}^n \otimes \mathbf{R}^m) \cdot \text{vec}(\mathbf{V}) + \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \quad (44)$$

Consequentemente, substituindo $\text{vec}(\mathbf{V})$ por \mathbf{v} na equação (44), tem-se:

$$\mathbf{v} = (\mathbf{I}^n \otimes \mathbf{R}^m) \cdot \mathbf{v} + \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \quad (45)$$

Comparando a equação (45) com a respectiva equação do subespaço válido $\{\mathbf{v} = \mathbf{T}^{conf} \cdot \mathbf{v} + \mathbf{i}^{conf}\}$, verifica-se (por identidade) que os parâmetros \mathbf{T}^{conf} e \mathbf{i}^{conf} são dados por:

$$\mathbf{T}^{conf} = (\mathbf{I}^n \otimes \mathbf{R}^m) \quad (46)$$

$$\mathbf{i}^{conf} = \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \quad (47)$$

As equações (46) e (47) também satisfazem as propriedades do subespaço-válido, isto é: $\mathbf{T}^{conf} \cdot \mathbf{T}^{conf} = \mathbf{T}^{conf}$ e $\mathbf{T}^{conf} \cdot \mathbf{i}^{conf} = \mathbf{0}$.

Definindo as Equações de \mathbf{T}^{ot} e \mathbf{i}^{ot}

Para os problemas de programação dinâmica, a função de energia E^{ot} da rede de Hopfield modificada é projetada para encontrar um caminho mínimo entre todos os caminhos possíveis. Quando E^{ot} é minimizada, a solução ótima corresponde a um estado de menor energia da rede. A função de energia E^{ot} é definida por:

$$\begin{aligned} E^{ot} = & \frac{1}{4} \left[\sum_{x=1}^{n-1} \sum_{i=1}^m \sum_{j=1}^m \underbrace{d_{xi,(x+1)j} \cdot v_{xi} \cdot v_{(x+1)j}}_{\text{1o. Termo}} + \right. \\ & \left. + \sum_{x=2}^n \sum_{i=1}^m \sum_{j=1}^m \underbrace{d_{(x-1)j,xi} \cdot v_{xi} \cdot v_{(x-1)j}}_{\text{2o. Termo}} \right] + \\ & \left[\sum_{x=1}^1 \sum_{i=1}^m \underbrace{d_{fonte,xi} \cdot v_{xi}}_{\text{3o. Termo}} + \sum_{x=n}^n \sum_{i=1}^m \underbrace{d_{xi,destino} \cdot v_{xi}}_{\text{4o. Termo}} \right] \end{aligned} \quad (48)$$

Nesta equação, o primeiro termo define o peso (custo métrico) da conexão ligando o i -ésimo neurônio do estágio x ao j -ésimo neurônio do estágio posterior ($x+1$). O segundo termo define o peso da conexão ligando o i -ésimo neurônio do estágio x ao j -ésimo neurônio do estágio anterior ($x-1$). O terceiro termo fornece o peso da conexão ligando o nó fonte a todos os outros nós do primeiro estágio, enquanto o quarto termo fornece o peso da conexão ligando o nó destino a todos os outros nós do último estágio. Deve-se ressaltar que o primeiro e segundo termo são responsáveis pelo encadeamento entre todos os estágios do problema de programação dinâmica. Portanto, a otimização de E^{ot} corresponde a minimizar cada termo dado por (48) em relação a v_{xi} . Assim, derivando-se a função de energia E^{ot} , dada em (48), em relação a v_{xi} (gradiente de E^{ot}), tem-se:

$$\begin{aligned} -\frac{\partial E^{ot}}{\partial v_{xi}} = & -\frac{1}{2} \left[\sum_{j=1}^m \underbrace{d_{xi,(x+1)j} \cdot v_{(x+1)j}}_{x < n} + \sum_{j=1}^m \underbrace{d_{(x-1)j,xi} \cdot v_{(x-1)j}}_{x > 1} \right] - \\ & \underbrace{d_{fonte,xi}}_{x=1} - \underbrace{d_{xi,destino}}_{x=n} \end{aligned} \quad (49)$$

Para obter a matriz \mathbf{T}^{ot} e o vetor \mathbf{i}^{ot} , define-se a matriz $\mathbf{P}_{xi,yj}$ como sendo a matriz cujos elementos correspondam ao custo métrico de conexão ligando o i -ésimo neurônio pertencente ao estágio x ao j -ésimo neurônio pertencente ao estágio y , ou seja:

$$[\mathbf{P}]_{xi,yj} = \frac{1}{2} d_{xi,yj} \begin{cases} i, j \in \{1..m\} \\ x, y \in \{1..n\} \end{cases} \quad (50)$$

No problema de programação dinâmica abordado neste artigo, o encadeamento entre todos os estágios é realizado através da definição de uma matriz $\mathbf{Q} \in \mathfrak{R}^{n \times n}$, cujos elementos são especificados por:

$$[\mathbf{Q}]_{xy} = \delta_{(x+1)y} + \delta_{(x-1)y} \quad (51)$$

Como ilustração, para um problema de programação dinâmica com 5 estágios ($n = 5$), a matriz Q seria dada por:

$$Q = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Então, a matriz T^{ot} e o vetor i^{ot} , pertencente ao termo de energia E^{ot} , deduzidos a partir da equação (49), podem ser representados em função das matrizes definidas nas equações (50) e (51), ou seja:

$$[T^{ot}]_{pq} = - [P]_{xi,yj} \cdot [Q]_{xy} \quad (52)$$

$$i^{ot} = - \underbrace{[d_{fonte,11} \ d_{fonte,12} \ d_{fonte,1m}]}_m \underbrace{[0 \ 0 \dots 0 \ 0]}_{m(n-2)} \underbrace{[d_{n1,destino} \ d_{n2,destino} \dots d_{nm,destino}]}_m \quad (53)$$

onde: $T^{ot} \in \mathfrak{R}^{nm \times nm}$ e $i^{ot} \in \mathfrak{R}^{nm}$

$$p = m.(x-1) + i$$

$$q = m.(y-1) + j$$

$$x, y \in \{1..n\}$$

$$i, j \in \{1..m\}$$

A Figura 3 ilustra a rede de Hopfield modificada utilizada para resolver o problema de programação dinâmica composta por n estágios e m estados em cada estágio. A operação desta rede é dinâmica, isto é, depois de aplicar um conjunto de novas entradas, as saídas são calculadas e retroalimentadas com a finalidade de modificar as entradas do instante anterior. Após este processo de convergência, um sinal (passo) de otimização é aplicado e todo o processo é repetido novamente até a convergência total da rede.

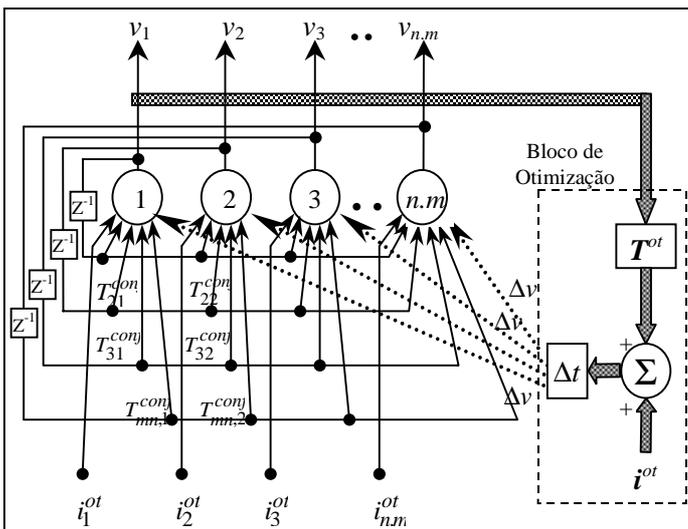


Figura 3. A Rede de Hopfield Modificada Utilizada Para Programação Dinâmica

Como observado na Figura 3, o número de neurônios utilizado para cada problema de programação dinâmica equivale ao número de estágios $\{n\}$ multiplicado pelo número de estados $\{m\}$. Para ilustrar o desempenho da rede neural proposta e comparar os resultados com outras abordagens, alguns exemplos de simulações são apresentados na próxima seção.

5 RESULTADOS DE SIMULAÇÃO

Os exemplos a seguir ilustram a aplicação da rede neural desenvolvida nas seções anteriores.

Exemplo 1

A Figura 4 mostra um problema simples a ser resolvido pela rede de Hopfield modificada. Os valores dos pesos (custo de conexão) estão também indicados na Figura 4.

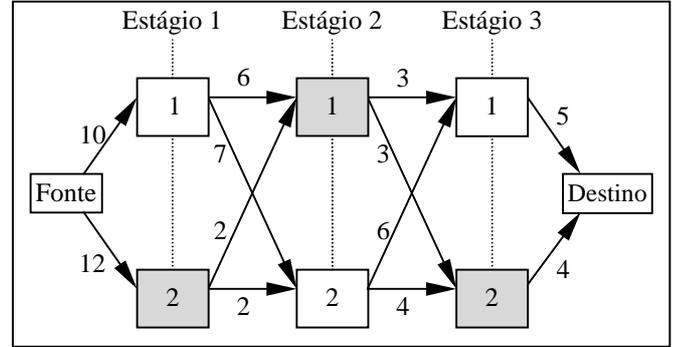


Figura 4. Ilustração do Problema (Exemplo 1).

Para este exemplo, as matrizes T^{ot} e T^{conf} , e os vetores i^{ot} e i^{conf} pertencentes aos termos de energia E^{ot} e E^{conf} são dados pelas seguintes expressões:

$$T^{ot} = \begin{bmatrix} 0 & 0 & 6 & 7 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 6 & 2 & 0 & 0 & 3 & 3 \\ 7 & 2 & 0 & 0 & 6 & 4 \\ 0 & 0 & 3 & 6 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 & 0 \end{bmatrix}$$

$$T^{conf} = \begin{bmatrix} 0.5 & -0.5 & 0 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & -0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & -0.5 \\ 0 & 0 & 0 & 0 & -0.5 & 0.5 \end{bmatrix}$$

$$i^{ot} = [10 \ 12 \ 0 \ 0 \ 5 \ 4]^T$$

$$i^{conf} = [0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5]^T$$

A rede de Hopfield modificada projetada para solucionar este problema é mostrada na Figura 5. A partir desta figura, observa-se que o número de neurônios necessários para a resolução do problema é da ordem de seis. Como o número de estados por estágios é igual a dois, tem-se que a cada sequência de dois neurônios corresponde ao número de estados de cada um dos estágios. Assim, se após a convergência da rede a saída do neurônio v_1 for igual a 1, então isto significa que o primeiro estado pertencente ao primeiro estágio estará ativado.

A solução ótima deste problema é fornecida pelos estados sombreados (Figura 4), isto é, estado 2 no estágio 1, estado 1 no estágio 2, e estado 2 no estágio 3. Em todas as simulações, a rede de Hopfield modificada convergiu para esta solução após três iterações. Os vetores p e $v(p)$, e a matriz $V(p)$ representando a solução obtida (após a convergência da rede) são dadas por:

$$p = [2 \ 1 \ 2]^T$$

$$v(p) = [0 \ 1 \ 10 \ 0 \ 1]^T$$

$$V(p) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

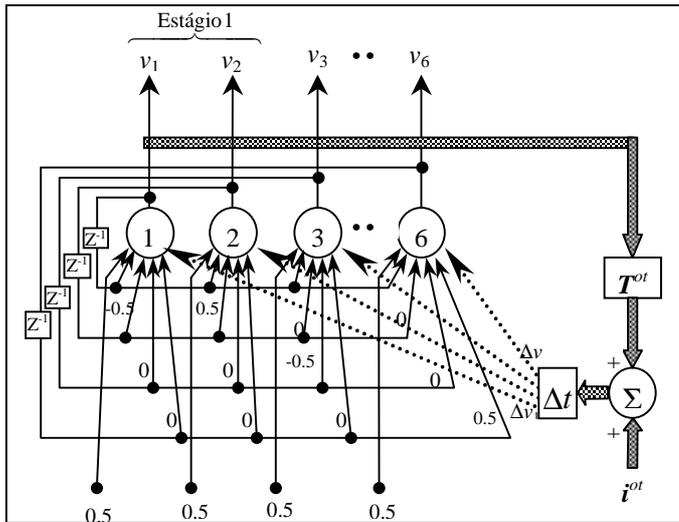


Figura 5. A Rede de Hopfield Modificada Utilizada no Exemplo 1.

A minimização do termo de energia E^{ot} garantiu que a solução obtida representa o caminho mínimo ($E^{ot} = 21$) entre todos os caminhos possíveis.

Exemplo 2

Para assegurar que a rede proposta pode ser eficientemente utilizada, alguns problemas de programação dinâmica foram simulados e comparados com os resultados obtidos pela rede proposta em Chiu *et alli* (1991).

Tabela 1. Resultados de Simulação.

Número de Estágios (n)	Número de Estados (m)	D^{RHM}	D^{RC}
2	2	3.13	3.25
4	4	2.03	3.12
8	8	1.34	2.00
16	16	1.06	1.85
32	32	1.03	1.61
64	64	1.02	1.39
16	2	3.14	3.21
16	4	1.79	2.98
16	8	1.26	1.85
16	32	1.13	1.79
2	16	1.17	1.53
4	16	1.02	1.60
8	16	1.09	1.76

Para uma comparação mais sistemática, utiliza-se a mesma metodologia de simulação usada em Chiu *et alli* (1991). Neste exemplo, o número de estágios e o número de estados em cada estágio foram incrementados passo a passo. Estes números foram obtidos a partir de valores pertencentes ao conjunto inteiro definido por $\{2, 4, 8, 16, 32, 64\}$. Os pesos das conexões ($d_{xi,yj}$) ligando os nós da rede foram aleatoriamente selecionados a partir de um conjunto de números inteiros definido por $\{1, 3, 5, 7, 9\}$.

O objetivo foi então encontrar o menor caminho possível que se inicia no nó fonte, visita um e apenas um nó em cada estágio, e termina no nó destino. A Tabela 1 mostra os resultados das simulações. Para as instâncias com n e m menor que 32, cada exemplo foi simulado vinte vezes com condições iniciais aleatórias. Outros casos foram simulados dez vezes. Nesta tabela, as colunas D^{RHM} e D^{RC} fornecem respectivamente os resultados do comprimento médio normalizado computados pela rede de Hopfield modificada e pela rede proposta em Chiu *et alli* (1991). O comprimento médio normalizado para cada instância da Tabela 1 é dado por:

$$D = \frac{S_c}{n_s \cdot (n + 1)} \quad (54)$$

onde S_c é a soma dos caminhos selecionados após a convergência da rede; n_s é o número de simulações; e n é o número de estágios.

Comparando os valores da Tabela 1 verifica-se que a rede de Hopfield modificada fornece melhores resultados, visto que as soluções obtidas (comprimento médio normalizado) são menores em relação àquelas computadas pela rede proposta em Chiu *et alli* (1991). Os fatores que podem influenciar no desempenho desta última rede são: i) termos de restrição e o termo de otimização são tratados em um único estágio, ii) interferência entre os termos de restrição e de otimização afeta a precisão dos pontos de equilíbrio, e iii) o processo de convergência da rede depende do ajuste correto das constantes de ponderação associadas aos vários termos de energia que compõem a rede.

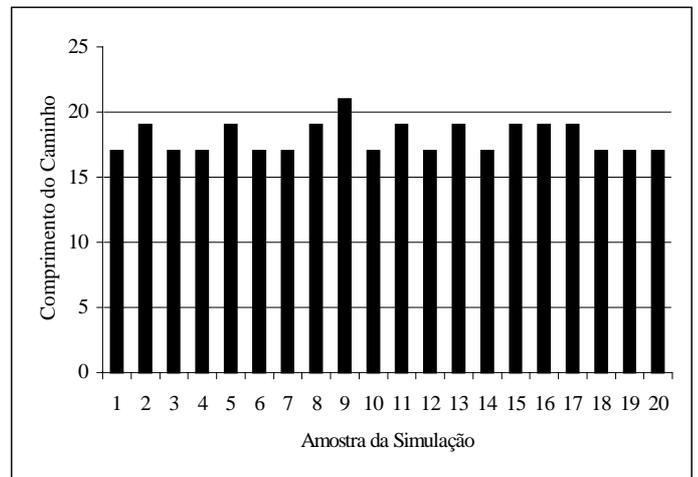


Figura 6. Gráfico dos Valores Calculados Pela RHM (16 Estágios e 16 Estados).

A Figura 6 mostra os valores obtidos pela rede de Hopfield modificada quando aplicada a um problema de programação dinâmica com 16 estágios ($n=16$) e 16 estados ($m=16$) em cada estágio. Para esta instância, a qual foram simuladas 20 situações distintas, a rede obteve 11 caminhos com comprimento total de 17 unidades, 8 caminhos com comprimento total de 19 unidades e 1 caminho com comprimento total de 21 unidades, resultando assim, num comprimento médio normalizado de 1.06 conforme apresentado na Tabela 1.

A rede de Hopfield modificada apresentada trata os termos de energia em diferentes estágios. Os parâmetros T^{conf} e i^{conf} (pertencentes a E^{conf}) foram desenvolvidos para forçar a validade das restrições estruturais associadas com o problema de programação dinâmica, enquanto que os parâmetros T^{ot} e i^{ot}

(pertencentes a E^{ot}) foram projetados para encontrar um caminho mínimo entre todos os caminhos possíveis. Assim, as principais vantagens em se utilizar a rede de Hopfield modificada para resolver problemas de programação dinâmica são: i) termos de restrições não interferem no termo de otimização, visto que são tratados em estágios distintos, ii) um único termo de energia (E^{conf}) agrupa todas as restrições impostas pelo problema, e iii) inexistência de constantes de ponderação para a sua inicialização.

Em todas as simulações, o vetor \mathbf{v} de saída da rede foi inicializado com valores aleatórios pequenos. Observa-se também que o aumento do número de estados e estágios não degrada o desempenho da rede.

6 CONCLUSÕES

Neste artigo, uma rede de Hopfield modificada foi desenvolvida para resolver problemas de programação dinâmica. Uma função de energia E^{ot} foi projetada com o objetivo de encontrar um caminho válido que tenha um comprimento total mínimo entre todos os caminhos possíveis. Resultados de simulações demonstram que a rede proposta é um método alternativo para resolver estes problemas eficientemente.

Além de fornecer uma nova abordagem para resolver problemas de programação dinâmica, a rede desenvolvida apresenta as seguintes características: i-) um único termo de energia definido por E^{conf} pode agrupar todas as restrições impostas pelo problema; ii-) os parâmetros internos T^{conf} e I^{conf} , pertencentes a E^{conf} , podem ser explicitamente obtidos pela técnica de subespaço-válido de soluções; iii-) a rede de Hopfield proposta não requer a definição de qualquer parâmetro de ponderação para ser inicializada; iv-) termos de otimização e termos de restrições envolvidos com o mapeamento do problema são tratados em estágios distintos; e v-) facilidade de implementação em hardware. Uma outra característica importante da abordagem proposta é que a partir de uma mesma sistemática de mapeamento é possível implementar várias outras classes de problemas de otimização (Silva, 1997). Entre os tipos de problemas que podem ser resolvidos pela rede de Hopfield modificada destacam-se: i-) problemas de otimização linear; ii-) problemas de programação não-linear; iii-) Problemas de otimização combinatorial, tais como o problema do caixeiro viajante (Travel Salesman Problem) e o problema das N -Rainhas; e iv-) problemas de identificação robusta (Silva *et alli*, 1997).

A principal desvantagem da rede de Hopfield modificada é a característica off-line de sua arquitetura. Trabalhos futuros explorarão o uso da abordagem proposta na solução de problemas de programação dinâmica probabilística e também na resolução de problemas de controle ótimo.

AGRADECIMENTOS

Os autores expressam agradecimentos à FAPESP pela concessão de auxílio financeiro (Processo No. 98/08480) e aos revisores pelas sugestões que resultaram numa melhoria substancial deste artigo.

APÊNDICE

A rede proposta por Chiu *et alli* (1991), para resolver problemas de programação dinâmica, é também baseada num

modelo de rede recorrente. A função de energia desta rede, a ser minimizada, é dada por:

$$E(\mathbf{v}) = \frac{a}{2} E_1(\mathbf{v}) + \frac{b}{4} E_2(\mathbf{v}) + S_1(\mathbf{v}) + S_2(\mathbf{v})$$

onde:

$$E_1(\mathbf{v}) = \sum_x \sum_i \sum_{j \neq i} c_{xi} v_{xi} c_{xj} v_{xj} + \left(\sum_x \sum_i c_{xi} v_{xi} - n \right)^2$$

$$E_2(\mathbf{v}) = \sum_x \sum_i \sum_j d_{xi,(x+1)j} c_{xi} v_{xi} c_{(x+1)j} v_{(x+1)j} + d_{(x-1)j,xi} c_{xi} v_{xi} c_{(x-1)j} v_{(x-1)j}$$

$$S_1(\mathbf{v}) = \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} (1 - c_{xi}) v_{xi} (1 - c_{xj}) v_{xj} + \frac{a}{2} \left(\sum_x \sum_i (1 - c_{xi}) v_{xi} \right)^2 + \frac{b}{4} \sum_x \sum_i \sum_j d_{xi,(x+1)j} [c_{xi} v_{xi} (1 - c_{(x+1)j}) v_{(x+1)j} + (1 - c_{xi}) v_{xi} (1 - c_{(x+1)j}) v_{(x+1)j}] + d_{(x-1)j,xi} [c_{xi} v_{xi} (1 - c_{(x-1)j}) v_{(x-1)j} + (1 - c_{xi}) v_{xi} (1 - c_{(x-1)j}) v_{(x-1)j}]$$

$$S_2(\mathbf{v}) = a \left(\sum_x \sum_i c_{xi} v_{xi} - n \right) \left(\sum_x \sum_i (1 - c_{xi}) v_{xi} \right) + \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} c_{xi} v_{xi} (1 - c_{xj}) v_{xj} + \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} c_{xj} v_{xj} (1 - c_{xi}) v_{xi} + \frac{b}{4} \sum_x \sum_i \sum_j d_{xi,(x+1)j} c_{(x+1)j} v_{(x+1)j} (1 - c_{xi}) v_{xi} + \frac{b}{4} \sum_x \sum_i \sum_j d_{(x-1)j,xi} c_{(x-1)j} v_{(x-1)j} (1 - c_{xi}) v_{xi}$$

onde: \mathbf{v} é a saída da rede.

\mathbf{c} é o estado corrente da rede.

n é o número de neurônios da rede.

a e b são parâmetros de ponderação.

A especificação adequada para as constantes de ponderações a e b (relativos aos termos de restrições) necessárias para a convergência desta rede é uma das principais dificuldades em utilizar esta abordagem. De acordo com as constantes escolhidas, a interferência entre os termos de restrições e o termo de otimização afeta o número de estados mínimos da rede, refletindo-se assim na solução obtida por esta rede. Outra característica fundamental desta rede, é que a mesma não utiliza a abordagem do subespaço-válido para confinar as restrições estruturais.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abe, S. (1993). Global Convergence and Suppression of Spurious States of the Hopfield Neural Networks. *IEEE Trans. on Circuits and Systems*, Vol. CAS-40, pp. 246-257.
- Aiyer, S. V. B, M. Niranjan and F. Fallside (1990). A Theoretical Investigation into the Performance of the Hopfield Network. *IEEE Trans. on Neural Networks*, Vol. 1, pp. 53-60.

- Bolla, R., F. Davoli, P. Maryni and T. Parisini (1998). Adaptive Neural Network Admission Controller for Dynamic Bandwidth Allocation. *IEEE Trans. on Systems, Man, and Cybernetics (Part B: Cybernetics)*, Vol. 28, pp. 592-601.
- Chiu, C., C.Y. Maa and M. S. Shanblatt (1991). Energy Function Analysis of Dynamic Programming Neural Networks. *IEEE Trans. on Neural Networks*, Vol. 2, pp. 418-426.
- Cichoki, A. and R. Unbehaven (1993). *Neural Networks for Optimization and Signal Processing*. John Wiley, 1993.
- Graham, A. (1981). *Kronecker Products and Matrix Calculus*. Ellis Horwood Ltd., Chichester.
- Haykin, S. (1994). *Neural Networks – A Comprehensive Foundation*. McMillan, New York.
- Henry, J. J., J. L. Farges and J. L. Gallego (1998). Neuro-Fuzzy Techniques for Traffic Control. *Control Engineering Practice*, Vol. 6, pp. 755-761.
- Hillier, F. S. and G. J. Lieberman (1980). *Introduction to Operations Research*. Holden Day, San Francisco, California.
- Hopfield, J. J. and D. W. Tank (1985). 'Neural' Computation of Decisions in Optimization Problems. *Biological Cybernetics*, Vol.52, pp. 141-152.
- Hsu, Y.- Y. and F.- C. Lu (1998). Combined Artificial Neural Network-Fuzzy Dynamic Programming Approach to Reactive Power/Voltage Control in a Distribution Substation. *IEEE Trans. on Power Systems*, Vol. 13, pp. 1265-1271.
- Jeffery, W. and R. Rosner (1986). Optimization Algorithms: Simulated Annealing and Neural Network Processing. *Astrophys. Journal*, pp. 473-481.
- Kacprzyk, J., R. A. Romero and F. A. C. Gomide (1999). Involving Objective and Subjective Aspects in Multistage Decision Making and Control Under Fuzziness. *International Journal of Intelligent Systems*, Vol. 14, pp. 79-104.
- Kennedy, M. P. and L. O. Chua (1988). Neural Networks for Nonlinear Programming. *IEEE Trans. on Circuits and Systems*, Vol. 35, pp. 554-562.
- Kirk, D. E. (1970). *Optimal Control Theory: An Introduction*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Levy, B. C. and M. B. Adams (1987). Global Optimization with Stochastic Neural Networks. *Proc. of the IEEE First Int. Conf. Neural Networks*, Vol. III, pp. 681-689.
- Rauch, H. E. and T. Winarske (1988). Neural Networks for Routing Communication Traffic. *IEEE Control Syst. Magazine*, pp. 26-31.
- Romero, R. A. F. (1993). *Otimização de Sistemas Através de Redes Neurais Artificiais*. Tese de Doutorado, DCA/FEE/UNICAMP, Campinas – SP.
- Silva, I. N., L. V. R. de Arruda and W. C. do Amaral (1997). Robust Parametric Estimation for Nonlinear Models Using Artificial Neural Networks. *International Journal of Systems Science*, Vol. 28, pp. 447-455.
- Silva, I. N. (1997). *Uma Abordagem Neuro-Nebulosa Para Otimização de Sistemas e Identificação Robusta*. Tese de Doutorado, UNICAMP/FEEC/DCA, Campinas – SP.
- Tagliarini, G. A., J. F. Christ and E. W. Page. (1991). Optimization Using Neural Networks. *IEEE Trans. on Computers*, Vol. 40, pp. 1347-1358.
- Tank, D. W. and J. J. Hopfield (1986). Simple Neural Optimization Networks: An A/D Converter, Signal Decision Network, and a Linear Programming Circuit. *IEEE Trans. on Circuits Syst.*, Vol. CAS-33, pp. 533-541.
- Vidyasagar, M. (1992). *Nonlinear Systems Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Wang, J. (1998). Primal and Dual Neural Networks for Shortest-Path Routing. *IEEE Trans. on Systems, Man and Cybernetics (Part A: Systems and Humans)*, Vol. 28, pp. 864-869.
- Zak, S. H., V. Upatising and S. Hui (1995). Solving Linear Programming Problems With Neural Networks. *IEEE Trans. on Neural Networks*, Vol. 6, pp. 94-104.