SBrT 2023 1570917956

Um algoritmo de amostragem para redes de difusão adaptativas multitarefas

Daniel G. Tiglea, Renato Candido e Magno T. M. Silva

***Resumo*— Neste artigo é proposto um algoritmo de amostragem voltado a redes de difusão adaptativas multitarefas *clusterizadas*. O algoritmo proposto busca manter os nós amostrados quando o erro quadrático em seus *clusters* é elevado e deixa de amostrá-los, caso contrário. As simulações mostram que a solução proposta alcança um custo computacional semelhante ao de um algoritmo monotarefa, mas com um melhor desempenho em regime permanente e uma taxa de convergência comparável ao de um algoritmo multitarefa com todos os nós amostrados. Além disso, o algoritmo proposto evita que efeitos adversos em um *cluster* específico afetem a amostragem em toda a rede.**

***Palavras-Chave*— Redes de difusão adaptativas, aprendizagem multitarefa, amostragem, processamento distribuído de sinais.**

***Abstract*— In this paper, a sampling algorithm for clustered multitask adaptive diffusion networks is proposed. The proposed algorithm seeks to keep the nodes sampled when the squared error in their clusters is high and stops sampling them otherwise. Simulations show that the proposed solution achieves a computational cost similar to that of a single-task algorithm, but with better performance and a convergence rate comparable to that of a multitask algorithm with all nodes sampled. Furthermore, the proposed algorithm prevents adverse effects on a specific cluster from affecting the sampling across the entire network.**

***Keywords*— Adaptive diffusion networks, multitask learning, sampling, distributed signal processing, impulsive noise.**

I. Introdução

Redes de difusão adaptativas têm se estabelecido como uma ferramenta eficaz para o processamento distribuído de sinais nas últimas décadas. Essas redes consistem em nós espalhados geograficamente e conectados por uma topologia, e têm se mostrado úteis em várias aplicações, como rastreamento de alvos, detecção de espectro em redes móveis, modelagem de redes sociais, entre outras [1]–[12]. No entanto, em algumas situações, o custo associado à medição e processamento de dados em cada nó é muito alto, o que torna necessária a utilização de um mecanismo de amostragem [13], [14]. Além disso, nas redes de sensores sem fio, o consumo de energia é frequentemente uma limitação crítica. Nessas situações, é comum usar técnicas de *censura* para limitar a quantidade de transmissões entre os nós [11], [12].

Com relação à diversidade de objetivos da rede, as redes de difusão adaptativas podem ser classificadas em duas categorias: as monotarefas (*single-task*), em que todos os nós compartilham do mesmo objetivo, e as multitarefas (*multitask*), em que o objetivo pode variar de nó para nó [15]–[17]. Mesmo neste último caso, a cooperação entre nós pode se mostrar interessante se os seus objetivos estiverem relacionados entre si, ainda que distintos [17]. Além disso, essa formulação pode ser usada para modelar variações regionais em parâmetros de interesse em problemas de estimação [15], [16].

Considerando-se as limitações energéticas e na capacidade de processamento das redes de difusão adaptativas, foi proposto em [18] um algoritmo de amostragem para essas redes que também pode ser usado como uma técnica de censura. Ele controla o número de nós amostrados ou censurados com base no erro quadrático na vizinhança de cada nó. Assim, reduz-se o número de nós amostrados quando a magnitude do erro é suficientemente baixa, permitindo uma rápida convergência e uma redução significativa no custo computacional e/ou consumo energético em regime permanente. Ao mesmo tempo, a amostragem dos nós não cessa permanentemente, o que preserva a capacidade dos algoritmos de detectar mudanças no ambiente. Posteriormente, foram propostas melhorias para ele em [19] a fim de facilitar a escolha de seus parâmetros e melhorar a resposta do algoritmo a mudanças abruptas no ambiente. No entanto, em ambos os trabalhos, focou-se exclusivamente em redes monotarefas. Neste artigo, uma versão do algoritmo de [18] é proposta para redes multitarefas. Simulações computacionais mostram o bom desempenho da versão proposta e uma redução significativa do custo computacional associado às tarefas de aprendizagem.

Este trabalho está organizado da seguinte forma. Na Seção II, é apresentada a formulação do problema de estimação em redes de difusão adaptativas multitarefas. Na Seção III, o algoritmo AS-dNLMS (*adaptive sampling diffusion normalized least-mean-squares*) de [18] é revisitado. Na Seção IV, são propostas modificações para o algoritmo a fim de adaptá-lo às redes multitarefas. Resultados de simulação e as conclusões são apresentados nas Seções V e VI, respectivamente.

Notação. São utilizadas letras em fontes convencionais para escalares e minúsculas em negrito para vetores. Além disso, $|\cdot|$ denota a cardinalidade, $\cap$ a intersecção entre dois conjuntos, $/$ a diferença entre dois conjuntos, $(\cdot)^{\mathrm{T}}$ transposição, $\mathrm{E}\{\cdot\}$ a esperança matemática e $\|\cdot\|$ a norma euclidiana.

II. Formulação do Problema

Consideremos uma rede com V nós e uma topologia predefinida. Dois nós são considerados vizinhos se eles podem trocar informações. Nesse caso, a vizinhança do nó k é dada pelo conjunto de seus vizinhos, incluindo o próprio nó k, e é denotada por $\mathcal{N}_k$. A cada iteração, cada nó k tem acesso a um

Os autores estão com Depto. de Engenharia de Sistemas Eletrônicos, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, e-mails: {dtiglea, renatocan}@lps.usp.br e magno.silva@usp.br. Este trabalho foi financiado pela CAPES (88887.512247/2020-00 e Código de Financiamento 001), FAPESP (2021/02063-6) e CNPq (303826/2022-3).

sinal de entrada $u_k(n)$ e a um sinal de referência [15], [16]

$$d_k(n) = \mathbf{u}_k^{\mathrm{T}}(n)\mathbf{w}_k^{\mathrm{o}} + v_k(n), \tag{1}$$

em que $\mathbf{u}_k(n) = [u_k(n) \;\; u_k(n-1) \cdots u_k(n-M+1)]^{\mathrm{T}}$ é um vetor regressor de comprimento M e $v_k(n)$ é o ruído de medição no nó k. Esse ruído é considerado independente de qualquer outra variável e estacionário no sentido amplo com média nula e variância $\sigma_{v_k}^2$. Por fim, $\mathbf{w}_k^{\mathrm{o}}$ é o sistema que se pretende estimar no nó k, também chamado na literatura de "sistema ótimo" do referido nó. O objetivo da rede é que cada nó k obtenha uma estimativa $\mathbf{w}_k$ do vetor $\mathbf{w}_k^{\mathrm{o}}$ de maneira distribuída. As redes de difusão monotarefas [1]–[3] podem ser entendidas como um caso particular das multitarefas, em que $\mathbf{w}_k^{\mathrm{o}} = \mathbf{w}^{\mathrm{o}}$ para todo nó k, $k = 1, \cdots, V$, em que $\mathbf{w}^{\mathrm{o}}$ representa um sistema ótimo único para toda a rede [15], [16]. Além disso, é possível que haja agrupamentos (*clusters*) de nós próximos entre si que tenham um sistema ótimo comum. Essas redes são às vezes denominadas de multitarefas *clusterizadas* na literatura [15]. O caso em que $\mathbf{w}_k^{\mathrm{o}}$ é diferente para cada nó também pode ser entendido como um caso particular dessas redes, considerando-se que haja V *clusters* na rede, cada um contendo um nó. Por generalidade, neste artigo adota-se a abordagem para redes multitarefas *clusterizadas* de [15]. Denotando por $\mathcal{C}(k)$ o *cluster* a que pertence o nó k, o objetivo da rede como um todo é minimizar a função custo global

$$\begin{aligned} J(\mathbf{w}_1, \cdots, \mathbf{w}_V) = & \sum_{k=1}^{V} \mathrm{E}\{[d_k(n) - \mathbf{u}_k(n)^{\mathrm{T}}\mathbf{w}_k]^2\} \\ & + \eta \sum_{k=1}^{V} \sum_{i \in \mathcal{N}_k/\mathcal{C}(k)} \rho_{ki} \|\mathbf{w}_i - \mathbf{w}_k\|^2, \end{aligned} \tag{2}$$

em que $\eta \geqslant 0$ é um fator de regularização e os ρ_{ki} satisfazem $\rho_{ki} \geqslant 0$ para todo i e k, $\rho_{ki} = 0$ se $i \notin \mathcal{N}_k/\mathcal{C}(k)$ e $\sum_{i \in \mathcal{N}/\mathcal{C}(k)} \rho_{ki} = 1$. Caso se escolha $\eta = 0$, a Eq. (2) corresponde à função custo do caso monotarefa aplicada a cada *cluster* individualmente, sem que haja cooperação entre *clusters* distintos. Em contrapartida, a adoção de η muito elevado força a similaridade entre as estimativas de diferentes *clusters*, mas pode prejudicar o aprendizado dos $\mathbf{w}_k^{\mathrm{o}}$ ao diminuir a importância relativa da primeira parcela em (2). Portanto, se for sabido *a priori* que os sistemas ótimos dos diferentes *clusters* estão fortemente relacionados entre si, faz sentido adotar valores maiores para η. Caso contrário, é mais adequado tornar η mais próximo de zero.

Em [15], a função custo de (2) é relaxada para permitir a sua otimização de maneira aproximada em cada nó k utilizando apenas as informações disponíveis em sua vizinhança. Devido à limitação de espaço, esses procedimentos não serão descritos em detalhes neste artigo, de forma que o leitor interessado é encorajado a consultar [15]. Neste caso, adotando-se uma estratégia do tipo LMS para minimizar (2), e uma configuração do tipo ATC (*adapt-then-combine*) para a difusão do aprendizado na rede, pode-se obter o algoritmo LMS de difusão (dLMS) multitarefa como nas Eqs. (3), mostradas no topo da página seguinte. Nessa configuração, a cada iteração realiza-se primeiramente a etapa de adaptação (3a) e, posteriormente, a etapa de combinação (3b). A ordem dessas etapas pode ser invertida, dando origem à configuração CTA (*combine-then-adapt*). Por simplicidade, neste artigo será considerada apenas a configuração ATC, mas os resultados podem ser estendidos à CTA. Na Equação (3a), μ_k é um passo de adaptação e $\boldsymbol{\psi}_k$ e $\mathbf{w}_k$ representam respectivamente as estimativas local e combinada de $\mathbf{w}_k^{\mathrm{o}}$ no nó k. Além disso, os c_{ik} são pesos que satisfazem $c_{ik} \geqslant 0$ para todo i e k, $c_{ik} = 0$ se $i \notin \mathcal{N}_k \cap \mathcal{C}(k)$ e $\sum_{i \in \mathcal{N} \cap \mathcal{C}(k)} c_{ik} = 1$. Possíveis escolhas para $\{c_{ik}\}$ incluem as regras Uniforme, Metropolis e do Grau Relativo [1], além de esquemas adaptativos como o algoritmo ACW (*adaptive combination weights*) [20]. Esse algoritmo é obtido resolvendo-se um problema de otimização com relação aos $\{c_{ik}\}$ e procura atribuir pesos maiores aos nós k em que $\sigma_{v_k}^2$ é comparativamente menor. Por fim, os b_{ik} são pesos sujeitos às mesmas regras que os c_{ik}.

III. Revisitando o algoritmo AS-dNLMS monotarefa

Para as redes monotarefas, diversos algoritmos foram propostos, sendo o dNLMS um dos mais utilizados [1]–[3]. Com base nele, foi proposto em [18] o algoritmo AS-dNLMS. Adotando-se uma configuração ATC, as suas etapas de adaptação e de combinação são dadas respectivamente por

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \bar{s}_k(n)\mu_k(n)\mathbf{u}_k(n)e_k(n) & \text{(4a)} \\ \mathbf{w}_k(n+1) = \sum_{i \in \mathcal{N}_k} c_{ik}\boldsymbol{\psi}_i(n+1), & \text{(4b)} \end{cases}$$

em que $\bar{s}_k(n) \in \{0,1\}$ é uma variável de amostragem. Além disso, $e_k(n) = d_k(n) - \mathbf{u}_k^{\mathrm{T}}(n)\mathbf{w}_k(n)$ é o erro de estimação e $\mu_k(n) = \tilde{\mu}_k / [\delta + \|\mathbf{u}_k(n)\|^2]$ é um passo de adaptação normalizado, com $0 < \tilde{\mu}_k < 2$ e um pequeno fator de regularização $\delta > 0$ [1]. Por fim, $\{c_{jk}\}$ são pesos de combinação que satisfazem $c_{jk} \geqslant 0$, $\sum_{j \in \mathcal{N}_k} c_{jk} = 1$ e $c_{jk} = 0$ se $j \notin \mathcal{N}_k$ [2], [3]. Assim, quando $\bar{s}_k(n) = 1$, $d_k(n)$ é amostrado e $e_k(n)$ é calculado. Em contrapartida, se $\bar{s}_k(n) = 0$, $d_k(n)$ não é amostrado, $\mathbf{u}_k^{\mathrm{T}}(n)\mathbf{w}_k(n)$, $e_k(n)$ e $\mu_k(n)$ não são calculados e $\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n)$.

Em vez de adaptar $\bar{s}_k(n)$ de forma direta, inspirado na combinação convexa de filtros adaptativos, introduz-se uma variável auxiliar $\alpha_k(n) \in [-\alpha^+, \alpha^+]$ tal que $\bar{s}_k(n) = 0$ para $\phi[\alpha_k(n)] < 0{,}5$ e $\bar{s}_k(n) = 1$ caso contrário, com $\phi[\cdot]$ definida como [18], [21]

$$\phi[\alpha_k(n)] \triangleq \frac{\mathrm{sgm}[\alpha_k(n)] - \mathrm{sgm}[-\alpha^+]}{\mathrm{sgm}[\alpha^+] - \mathrm{sgm}[-\alpha^+]}, \tag{5}$$

em que $\mathrm{sgm}[x] = (1 + e^{-x})^{-1}$ é a função sigmoide. Na literatura, usualmente se adota $\alpha^+ = 4$ [21]. Para deixar a notação mais compacta, $\phi[\alpha_k(n)]$ será escrito como $\phi_k(n)$. Cabe notar que $\phi[\alpha^+] = 1$, $\phi[0] = 0{,}5$ e $\phi[-\alpha^+] = 0$. Assim, $\bar{s}_k(n)$ pode ser relacionado a $\alpha_k(n)$ por meio de

$$\bar{s}_k(n) = \begin{cases} 1, \text{ se } \alpha_k(n) \geqslant 0, \\ 0, \text{ c.c.} \end{cases} . \tag{6}$$

Busca-se então minimizar a seguinte função custo [18]

$$J_{\alpha_k}^{\mathrm{mono}}(n) = \phi_k(n)\beta\bar{s}_k(n) + [1 - \phi_k(n)] \sum_{i \in \mathcal{N}_k} c_{ik} e_i^2(n) \tag{7}$$

com relação a $\alpha_k(n)$, em que $\beta > 0$ é um parâmetro introduzido a fim de controlar a penalização da amostragem. Quando o erro quadrático na vizinhança do nó k é elevado, $J_{\alpha_k}^{\mathrm{mono}}(n)$ é minimizada fazendo-se $\phi_k(n)$ próximo de um, o que conduz à amostragem do nó k. O mesmo vale quando o nó k não é amostrado ($\bar{s}_k = 0$), o que garante que a sua amostragem seja retomada em algum momento. Em contrapartida, quando

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k \left\{ \sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} b_{ik} [d_i(n) - \mathbf{u}_i^{\mathrm{T}}(n)\mathbf{w}_k(n)]\mathbf{u}_i(n) + \eta \sum_{i \in \mathcal{N}_k / \mathcal{C}(k)} \rho_{ki} [\mathbf{w}_i(n) - \mathbf{w}_k(n)] \right\} & \text{(3a)} \\ \mathbf{w}_k(n+1) = \sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{ik} \boldsymbol{\psi}_i(n+1). & \text{(3b)} \end{cases}$$

o nó k é amostrado ($\bar{s}_k = 1$) e o erro quadrático é pequeno, $J_{\alpha_k}^{\text{mono}}(n)$ é minimizada fazendo-se $\phi_k(n)$ próximo de zero, o que faz com que o nó k deixe de ser amostrado [18].

A atualização de $\alpha_k(n)$ é feita tomando-se a derivada de $J_{\alpha_k}^{\text{mono}}(n)$ em relação a $\alpha_k(n)$ e utilizando-se o método do gradiente estocástico. Como não se tem acesso ao erro $e_i(n)$ quando o nó i deixa de ser amostrado, ele é substituído pela sua última medição, denotada por $\varepsilon_i(n)$. Assim, obtém-se [18]

$$\alpha_k(n+1) = \alpha_k(n) + \mu_s \phi_k'(n) \left[\sum_{i \in \mathcal{N}_k} c_{ik} \varepsilon_i^2(n) - \beta \bar{s}_k(n) \right], \quad (8)$$

em que $\mu_s > 0$ é um passo de adaptação e

$$\phi_k'(n) = \frac{d\phi[\alpha_k(n)]}{d\alpha_k(n)} = \frac{\text{sgm}[\alpha_k(n)]\{1 - \text{sgm}[\alpha_k(n)]\}}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}. \quad (9)$$

As Equações (4) e (8) são a base do algoritmo AS-dNLMS [18]. Cabe observar que o algoritmo requer que cada nó amostrado i transmita $\varepsilon_i^2(n) = e_i^2(n)$ para os seus vizinhos. No entanto, essa informação pode ser enviada juntamente com as estimativas locais $\boldsymbol{\psi}_i$ para não aumentar o número de transmissões. Na versão do algoritmo voltada à censura, denominada ASC-dNLMS (*adaptive-sampling-and-censoring diffusion* NLMS), a etapa de adaptação (4a) não é executada quando o nó não é amostrado. Dessa forma, $\boldsymbol{\psi}_k$ e ε_k permanecem inalterados nessas iterações. Considerando que os nós possam armazenar os dados enviados pelos vizinhos, isso permite reduzir o número de transmissões.

IV. O Algoritmo AS-dNLMS Multitarefa

Ao adaptar o mecanismo de amostragem do algoritmo AS-dNLMS ao cenário multitarefa, é conveniente que cada nó k seja influenciado apenas pelos vizinhos pertencentes ao mesmo *cluster*, já que são estes que partilham do mesmo sistema ótimo. Essa alteração em relação ao algoritmo monotarefa pode permitir, por exemplo, que *clusters* cujos nós já tenham alcançado estimativas suficientemente próximas de seus sistemas ótimos deixem de ser amostrados. Em contrapartida, em *clusters* em que a tarefa de aprendizado ainda esteja distante do desempenho em regime permanente, seria possível manter a adaptação dos nós. Tanto em um cenário como no outro, a adaptação do mecanismo de amostragem nos nós de um *cluster* não afeta aqueles pertencentes a outros *clusters*.

Além disso, é interessante notar que uma diferença entre o algoritmo dLMS multitarefa da Eq. (3) e o algoritmo AS-dNLMS monotarefa da Eq. (4) consiste no fato de que, no primeiro caso, cada nó k utiliza informação de todos os vizinhos pertencentes ao mesmo *cluster* já na etapa de adaptação. Esses dados também poderiam ser incorporados ao mecanismo de amostragem em sua versão multitarefa. Dessa forma, em vez de utilizar apenas o erro de estimação na vizinhança do nó k, a função custo da Eq. (7) é alterada para

$$J_{\alpha_k}^{\text{multi}}(n) = \phi_k(n)\beta\bar{s}_k(n) + \left[1 - \phi_k(n)\right] \sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{ik}\xi_i(n), \quad (10)$$

em que

$$\xi_i(n) = \sum_{j \in \mathcal{N}_i \cap \mathcal{C}(i)} b_{ji} [d_j(n) - \mathbf{u}_j^{\mathrm{T}}(n)\mathbf{w}_i(n)]^2 \quad (11)$$

utiliza a informação de $d_j(n)$ e de $\mathbf{u}_j(n)$ dos vizinhos do nó i pertencentes ao mesmo *cluster*.

Utilizando-se um procedimento análogo àquele adotado para o mecanismo de amostragem no cenário monotarefa, pode-se obter a seguinte equação de atualização para $\alpha_k(n)$:

$$\alpha_k(n+1) = \alpha_k(n) + \mu_s \phi_k'(n) \left[\sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{ik}\zeta_i(n) - \beta\bar{s}_k(n) \right], \quad (12)$$

em que $\zeta_j(n)$ denota a última medida de $\xi_j(n)$ a que se tem acesso, já que a etapa de adaptação não é efetuada quando o nó k não é amostrado.

Como no AS-dNLMS monotarefa, a ideia no caso multitarefa é que, se o nó k não for amostrado, não se some o termo de correção relacionado ao gradiente à estimativa combinada $\mathbf{w}_k(n)$ na etapa de adaptação. Essa ideia está ilustrada nas Eqs. (13), exibidas no topo da página seguinte. Além disso, cada gradiente dentro do primeiro somatório na Eq. (13a) foi normalizado pela norma do vetor regressor $\mathbf{u}_i(n)$. Isto facilita a escolha dos passos de adaptação μ_k, uma vez que a seleção desse parâmetro depende da norma dos vetores regressores para as versões não normalizadas do algoritmo [1], [22].

V. Simulações Computacionais

Nesta seção, são apresentados resultados de simulação para verificar o desempenho do algoritmo AS-dNLMS multitarefa proposto na Seção IV, obtidos a partir de uma média de 100 realizações independentes. Para facilitar a visualização dos resultados, foi aplicado um filtro de média móvel com 64 coeficientes às curvas apresentadas.

Para as simulações, foi gerada de forma aleatória uma rede com 28 nós agrupados em quatro *clusters* diferentes, cada um com sete nós, numerados de $\mathcal{C}_1$ a $\mathcal{C}_4$. A rede resultante é apresentada na Fig. 1(a). Além disso, cada nó k está sujeito a um ruído de variância $\sigma_{v_k}^2$ e utiliza um valor de passo de adaptação μ_k, como mostrado na Fig. 1(b).

O sinal de entrada $u_k(n)$ é gaussiano com média nula e variância unitária para todos os nós k, $k = 1, \cdots, V$. O sistema ótimo $\mathbf{w}_{\mathcal{C}_i}^{\text{o}}$ associado ao *cluster* $\mathcal{C}_i$ foi gerado como

$$\mathbf{w}_{\mathcal{C}_i}^{\text{o}} = \mathbf{w}_{\text{comum}}^{\text{o}} + \mathbf{w}_{\text{local}_i}^{\text{o}},$$

em que $\mathbf{w}_{\text{comum}}^{\text{o}}$ é uma parcela comum a todos os *clusters* e $\mathbf{w}_{\text{local}_i}^{\text{o}}$ é específico para o *cluster* $\mathcal{C}_i$. Os coeficientes do vetor $\mathbf{w}_{\text{comum}}^{\text{o}}$ foram gerados segundo uma distribuição uniforme no intervalo $[-1,1]$. Em seguida, esse vetor foi normalizado, de forma a apresentar norma unitária. Por sua vez, os coeficientes dos vetores $\mathbf{w}_{\text{local}_i}^{\text{o}}$ foram gerados segundo uma distribuição uniforme no intervalo $[-2{\cdot}10^{-2}, 2{\cdot}10^{-2}]$. O comprimento dos sistemas ótimos é de $M = 10$. Além disso, para simular uma mudança abrupta no ambiente, na metade de cada realização a ordem dos coeficientes dos vetores $\mathbf{w}_{\mathcal{C}_i}^{\text{o}}$ é invertida.

Para a seleção dos pesos de combinação c_{jk} e b_{ik}, utiliza-se uma versão normalizada do algoritmo ACW, a fim de evitar divisão por zero. Suas equações são dadas por [18], [20]

$$c_{ik}(n) = \frac{[\delta_c + \hat{\sigma}_{ik}^2(n)]^{-1}}{\sum_{\ell \in \mathcal{N}_k} [\delta_c + \hat{\sigma}_{\ell k}^2(n)]^{-1}}, \text{ se } i \in \mathcal{N}_k \cap \mathcal{C}(k), \text{ ou } 0, \text{c.c.}, \quad (14)$$

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k \bar{s}_k(n) \left\{ \sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} \dfrac{b_{ik}[d_i(n) - \mathbf{u}_i^{\mathrm{T}}(n)\mathbf{w}_k(n)]\mathbf{u}_i(n)}{\delta + \|\mathbf{u}_i(n)\|^2} + \eta \sum_{i \in \mathcal{N}_k / \mathcal{C}(k)} \rho_{ki}[\mathbf{w}_i(n) - \mathbf{w}_k(n)] \right\} & \text{(13a)} \\ \mathbf{w}_k(n+1) = \sum_{i \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{ik} \boldsymbol{\psi}_i(n+1). & \text{(13b)} \end{cases}$$

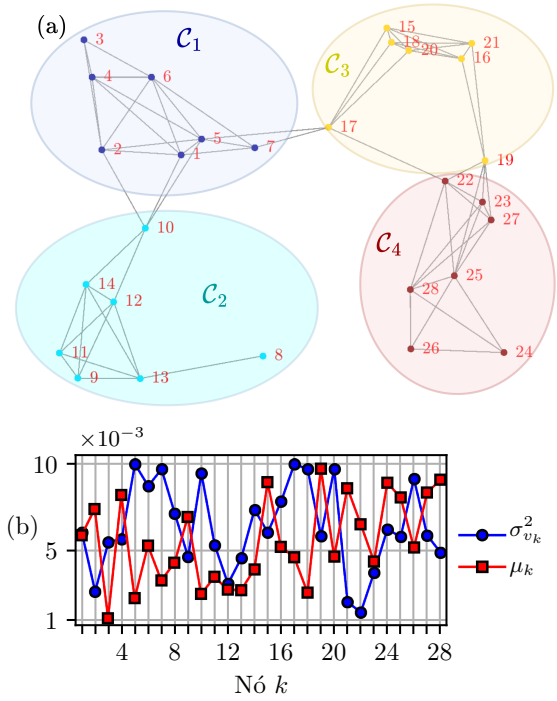

Fig. 1: (a) Rede considerada nas simulações, com seus nós agregados em quatro *clusters* distintos, numerados como $\mathcal{C}_1$ a $\mathcal{C}_4$. (b) Variância $\sigma^2_{v_k}$ do ruído e passo de adaptação μ_k para $k = 1, \cdots, V$.

em que $\delta_c > 0$ é um fator de regularização e

$$\hat{\sigma}^2_{ik}(n) = (1-\nu_k)\hat{\sigma}^2_{ik}(n-1) + \nu_k \|\boldsymbol{\psi}_i(n+1) - \mathbf{w}_k(n)\|^2, \quad (15)$$

com $\nu_k = 0{,}2$ para $k = 1, \cdots, V$ [20]. Além disso, foram adotados $\delta = \delta_c = 10^{-5}$ como fatores de regularização. A fim de simplificar a seleção dos parâmetros, utilizou-se $c_{ik} = b_{ik}$ para todo i e k. Para os pesos ρ_{ik}, foi utilizada a regra Uniforme. Diferentes valores de η foram testados no cenário descrito em simulações preliminares e se adotou $\eta = 10^{-3}$, que levou ao melhor desempenho por parte do algoritmo dNLMS multitarefa em regime permanente. Como métrica de desempenho, adota-se o NMSD (*network mean-square-deviation*), dado por

$$\mathrm{NMSD}(n) = \frac{1}{V} \sum_{k=1}^{V} \mathrm{E}\left\{ \left\| \mathbf{w}_k^{\mathrm{o}}(n) - \mathbf{w}_k(n) \right\|^2 \right\}. \quad (16)$$

Primeiramente, é feita uma comparação entre os algoritmos AS-dNLMS multitarefa o algoritmo dNLMS multitarefa com V_s nós amostrados aleatoriamente a cada iteração, sendo $V_s \in \{7, 14, 21, 28\}$. Convém ressaltar que $V_s = 28$ corresponde ao caso em que todos os nós são amostrados. A título de referência, também é apresentado o resultado obtido com o algoritmo dNLMS monotarefa para o mesmo cenário de simulação com todos os nós amostrados. Para cada algoritmo, as curvas de NMSD, o número médio de nós amostrados e de multiplicações são mostrados ao longo das iterações nas Figuras 2(a), 2(b) e 2(c), respectivamente. Para o algoritmo proposto foram adotados $\beta = 0{,}09$ e $\mu_s = 0{,}055$. Esses valores foram selecionados porque permitiram igualar o número de multiplicações exigido por iteração em regime permanente ao do algoritmo dNLMS monotarefa, mas sem prejudicar o desempenho. Analisando-se a Fig. 2(a), observa-se que o algoritmo dNLMS monotarefa apresenta um desempenho em regime permanente inferior àquele alcançado pelas soluções multitarefas, o que ilustra a importância de se considerar variações regionais no sistema ótimo. Com relação aos algoritmos multitarefas, nota-se que a amostragem aleatória afeta significativamente a taxa de convergência. Em contrapartida, o algoritmo proposto apresenta aproximadamente a mesma taxa de convergência que o dNLMS multitarefa com todos os nós amostrados. Observando-se a Fig. 2(b), é possível perceber que isso ocorreu porque o algoritmo AS-dNLMS mantém todos os nós amostrados no transitório. Além disso, o algoritmo proposto foi capaz de detectar a mudança no ambiente e retomar a amostragem de todos os nós, o que lhe permite novamente alcançar uma taxa de convergência semelhante à do dNLMS multitarefa após a alteração abrupta. Analisando-se a Fig. 2, observa-se que o algoritmo proposto apresenta um custo computacional mais elevado do que o algoritmo dNLMS com todos os nós amostrados durante o transitório. No entanto, depois que o algoritmo AS-dNLMS multitarefa atinge o regime permanente, o custo computacional é reduzido significativamente com a diminuição no número de nós amostrados por iteração. Na média, o algoritmo AS-dNLMS executa aproximadamente 4211 multiplicações por iteração durante o regime permanente, contra 6436 por parte do algoritmo dNLMS multitarefas com todos os nós amostrados e 4268 por parte do algoritmo monotarefa.

A fim de verificar o efeito do mecanismo de amostragem em diferentes *clusters*, a simulação anterior foi repetida com o algoritmo AS-dNLMS multitarefa, mas considerando-se um cenário em que a variância do ruído nos nós pertencentes ao *cluster* $\mathcal{C}_4$ da Fig. 1(a) foi multiplicada por dez. Os resultados são mostrados na Fig. 3. Na Fig. 3(a), além do NMSD da rede como um todo, são mostradas as curvas com o desempenho médio dos nós em cada *cluster*. Na Fig. 3(b) é mostrado o número total de nós amostrados por iteração na rede, e na Fig. 3 (c), o número de nós amostrados em cada *cluster*. É possível observar que, como esperado, o nível de NMSD em regime permanente no *cluster* $\mathcal{C}_4$ é maior do que aquele verificado nos demais *clusters*. Analisando-se a Fig. 3(c), nota-se que, em decorrência da variância maior do ruído nesse *cluster*, a amostragem dos seus nós é mantida por mais iterações durante o transitório, em comparação com a amostragem dos nós dos demais *clusters*. Além disso, em regime permanente, o algoritmo de amostragem mantém mais nós amostrados no *cluster* $\mathcal{C}_4$ do que no restante da rede. Para os *clusters* $\mathcal{C}_1$ a $\mathcal{C}_3$, observa-se um comportamento semelhante àquele constatado para a rede inteira nas simulações da Fig. 2. Isto mostra que o funcionamento do mecanismo de amostragem em um *cluster*

é isolado dos demais no algoritmo proposto, o que impede que fenômenos adversos em uma determinada região afetem a amostragem na rede como um todo.

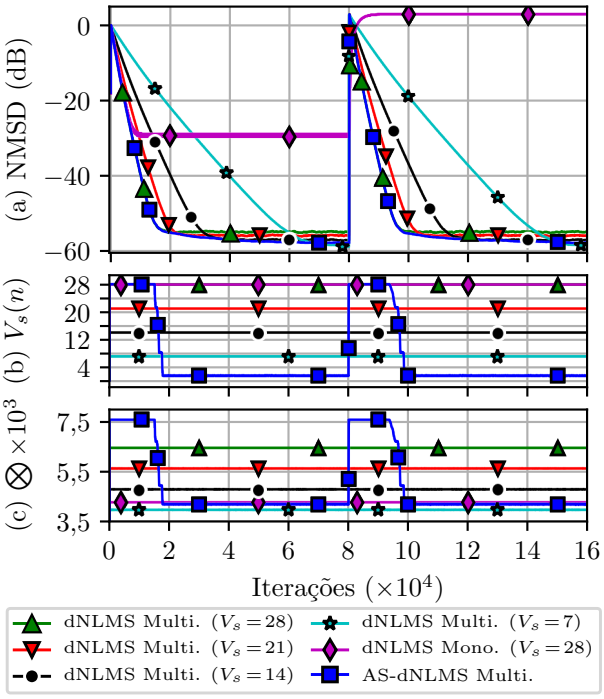

Fig. 2: Comparação entre os algoritmos dNLMS multitarefa com V_s nós amostrados aleatoriamente ($V_s \in \{7, 14, 21, 28\}$), AS-dNLMS multitarefa ($\beta = 0{,}09$ e $\mu_s = 0{,}055$) e dNLMS monotarefa com todos os nós amostrados. (a) curvas de NMSD, (b) Número de nós amostrados, e (c) Número de multiplicações por iteração.

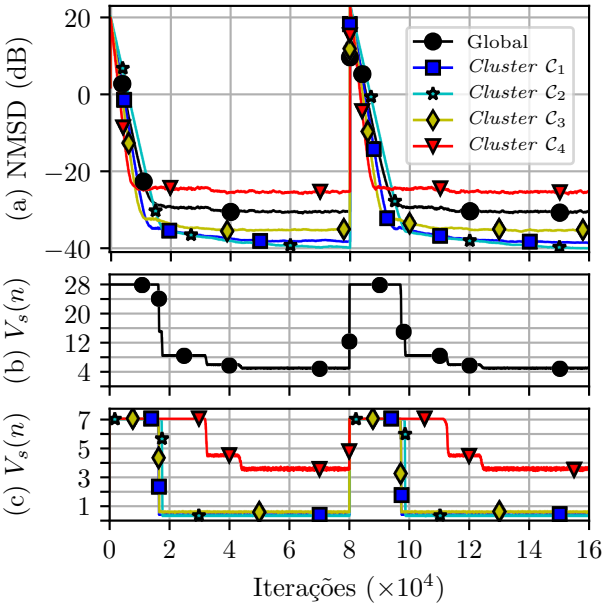

Fig. 3: Resultados obtidos com o algoritmo AS-dNLMS multitarefa ($\beta = 0{,}09$ e $\mu_s = 0{,}055$) considerando a rede da Fig. 1(a) como um todo e cada *cluster* individualmente. (a) curvas de NMSD e (b) Número de nós amostrados por iteração na rede como um todo e (c) em cada *cluster* separadamente.

VI. Conclusões

Neste artigo, foi proposto um algoritmo de amostragem para redes de difusão adaptativas multitarefas. Com a solução proposta, é possível obter um custo computacional semelhante àquele que seria exigido por um algoritmo monotarefa, mas com um desempenho melhor em regime permanente e com taxa de convergência semelhante à do algoritmo dNLMS multitarefa original. Além disso, foi possível constatar que o algoritmo proposto previne que efeitos adversos em um determinado *cluster* afetem a amostragem na rede como um todo. Em trabalhos futuros, pretende-se obter resultados teóricos que permitam prever o número médio de nós amostrados por iteração pelo algoritmo, e estender as melhorias propostas em [19] para a versão multitarefa do algoritmo aqui proposta.

Referências

[1] A. H. Sayed, *Adaptation, Learning, and Optimization over Networks*, vol. 7, Foundations and Trends in Machine Learning, now Publishers Inc., Hanover, MA, 2014.

[2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, pp. 3122–3136, 2008.

[3] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, pp. 1035–1048, 2009.

[4] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, pp. 4795–4810, 2010.

[5] C.-K. Yu and A. H. Sayed, "A strategy for adjusting combination weights over adaptive networks," in *Proc. ICASSP*, 2013, pp. 4579–4583.

[6] J. Fernandez-Bes, J. Arenas-García, M. T. M. Silva, and L. A. Azpicueta-Ruiz, "Adaptive diffusion schemes for heterogeneous networks.," *IEEE Trans. Signal Process.*, vol. 65, pp. 5661–5674, 2017.

[7] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. ICASSP*, 2008, pp. 3285–3288.

[8] X. Zhao and A. H. Sayed, "Single-link diffusion strategies over adaptive networks," in *Proc. ICASSP*, 2012, pp. 3749–3752.

[9] S. Xu, R. C. de Lamare, and H. V. Poor, "Adaptive link selection algorithms for distributed estimation," *EURASIP J. Advances in Signal Process.*, pp. 1–22, 2015.

[10] N. Takahashi and I. Yamada, "Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks," in *Proc. ICASSP*, 2010, pp. 3518–3521.

[11] R. Arroyo-Valles, S. Maleki, and G. Leus, "A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks," in *Proc. SPAWC*, 2013, pp. 155–159.

[12] J. Fernandez-Bes, R. Arroyo-Valles, J. Arenas-García, and J. Cid-Sueiro, "Censoring diffusion for harvesting WSNs," in *Proc. CAMSAP*, 2015, pp. 237–240.

[13] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, pp. 3584–3598, 2018.

[14] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Trans. Signal Process.*, vol. 65, pp. 4193–4208, 2017.

[15] J. Chen, C. Richard, and A. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, pp. 4129–4144, 2014.

[16] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, pp. 2733–2748, 2015.

[17] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and multitask wireless sensor networks—algorithms, applications, and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 11, pp. 450–465, 2017.

[18] D. G. Tiglea, R. Candido, and M. T. M. Silva, "A low-cost algorithm for adaptive sampling and censoring in diffusion networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 58–72, 2021.

[19] D. G. Tiglea, R. Candido, and M. T.M. Silva, "An adaptive algorithm for sampling over diffusion networks with dynamic parameter tuning and change detection mechanisms," *Digital Signal Process.*, vol. 127, pp. 103587, 2022.

[20] S.-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Proc. CAMSAP*, 2011, pp. 317–320.

[21] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "Adaptively biasing the weights of adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, pp. 3890–3895, 2010.

[22] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, NJ, 2008.