

Amostragem adaptativa aplicada a um algoritmo difuso voltado a grafos

Daniel Gilio Tiglia, Renato Candido e Magno T. M. Silva

Resumo—O processamento de sinais em grafos tem atraído a atenção da comunidade científica por ser uma ferramenta interessante para lidar com grandes quantidades de dados interrelacionados. Recentemente, foi proposto um algoritmo difuso para a filtragem adaptativa de sinais sobre grafos. Entretanto, esse algoritmo apresenta um custo computacional elevado, pois todos os nós do grafo são amostrados mesmo em regime permanente. Neste trabalho, é proposto um método adaptativo de amostragem para esse algoritmo que permite uma redução no custo computacional em regime permanente preservando-se o desempenho do algoritmo. Também é apresentada uma análise para facilitar a escolha de seus parâmetros.

Palavras-Chave—Processamento de sinais em grafos, amostragem em grafos, adaptação difusa, filtragem em grafos, combinações convexas.

Abstract—Graph signal processing has attracted attention in the signal processing community, since it is an effective tool to deal with large quantities of interrelated data. Recently, a diffusion algorithm for graph adaptive filtering was proposed. However, it suffers from high computational cost since all nodes in the graph are sampled even in steady state. In this paper, we propose an adaptive sampling method for this solution that reduces the computational cost in steady state, while maintaining convergence rate and steady-state performance. We also present an analysis to give insights about proper choices for its adaptation parameters.

Keywords—Graph signal processing, sampling on graphs, diffusion strategies, graph filtering, convex combination.

I. INTRODUÇÃO

Nos últimos anos, o processamento de sinais sobre grafos (GSP - *graph signal processing*) se tornou um tópico de intensa pesquisa, uma vez que os dados observados podem ser modelados por sinais definidos sobre grafos em diversas aplicações com estruturas de redes, tais como redes de sensores, *smart grids*, redes de transporte e de comunicações, entre outras [1]–[8]. Essas aplicações podem ser representadas por grafos com uma grande quantidade de nós conectados e, portanto, muitas técnicas para a amostragem de sinais sobre grafos têm sido propostas (ver, por exemplo, [6] e suas referências). Essas técnicas de amostragem usam informações de menos nós para fazer inferências sobre o sinal no grafo como um todo, frequentemente por meio de um mecanismo de amostragem aleatória [6].

Recentemente, em [7]–[9] foram aplicadas estratégias de adaptação difusa [10] para propor novas ferramentas para o processamento adaptativo de sinais sobre grafos, levando a soluções baseadas no algoritmo LMS (*least-mean-squares*). O

Daniel Gilio Tiglia, Renato Candido e Magno T. M. Silva. Departamento de Engenharia de Sistemas Eletrônicos da Escola Politécnica da Universidade de São Paulo. E-mails: daniel.tiglia@usp.br, renatocan@lps.usp.br e magno.silva@usp.br. Este trabalho foi financiado pelo CNPq (132586/2018-5 e 304715/2017-4) e pela FAPESP (2017/20378-9).

algoritmo distribuído de [8] foi proposto com foco na predição de sinais sobre grafos e utiliza uma estratégia distribuída eficiente para a amostragem de sinais sobre grafos baseada em uma abordagem probabilística. Em contrapartida, em [9] é proposto um algoritmo LMS difuso para a estimação de coeficientes de filtros invariantes ao deslocamento sobre grafos. Entretanto, na sua forma atual, o algoritmo distribuído de [9] requer as informações de todos os nós para o processamento. A questão que surge é se isso realmente é necessário, uma vez que a amostragem parcial dos nós reduz o custo computacional e de memória, o que é crucial em situações em que a medição e o processamento dos dados de todos os nós são muito custosos.

Neste trabalho é proposto um mecanismo de amostragem para o algoritmo difuso para grafos de [9] que altera a quantidade de nós amostrados de maneira adaptativa com base no erro quadrático médio (*mean squared error* - MSE) na vizinhança de cada nó. Dessa forma, o número de nós amostrados cai quando o MSE é suficientemente baixo, possibilitando uma alta velocidade de convergência ao mesmo tempo em que reduz significativamente o número de nós amostrados em regime permanente.

Este trabalho está organizado da seguinte forma. Na Seção II, formula-se o problema de processamento de sinais em grafos e se revisita o algoritmo distribuído de [9]. Na Seção III, é proposto um algoritmo de amostragem adaptativa e na Seção IV, é feita uma análise para facilitar a escolha dos seus parâmetros. Resultados de Simulação são mostrados na Seção V. Por fim, a Seção VI encerra o trabalho com as conclusões.

Notação. Letras em fontes convencionais denotam escalares, letras minúsculas em negrito denotam vetores e letras maiúsculas em negrito denotam matrizes. O k -ésimo elemento do vetor \mathbf{x} é denotado por $[\mathbf{x}]_k$, e se \mathcal{X} é um conjunto, $|\mathcal{X}|$ denota a sua cardinalidade. Além disso, $(\cdot)^T$ denota transposição, $E\{\cdot\}$ a esperança matemática, $\|\cdot\|$ a norma euclidiana, $\text{col}\{\cdot\}$ o agrupamento de seus elementos de modo a formar um vetor coluna, e $\text{diag}\{\cdot\}$ uma matriz diagonal em que os seus argumentos são os elementos diagonais.

II. FILTRAGEM ADAPTATIVA DIFUSA EM GRAFOS

Seja $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ um grafo dotado de um conjunto de nós \mathcal{V} com rótulos $k = 1, 2, \dots, V$ e de um conjunto de arestas \mathcal{E} . Representa-se um sinal definido sobre \mathcal{G} pelo vetor coluna $\mathbf{x}(n) = [x_1(n), \dots, x_V(n)]^T \in \mathbb{R}^V$, em que $x_k(n)$ é o valor do sinal no nó k no instante de tempo n [1]. Além disso, seja \mathbf{A} uma matriz $V \times V$ que denota o operador deslocamento no grafo (*graph-shift operator*). Possíveis escolhas para \mathbf{A} incluem a matriz de adjacência [1] e a matriz laplaciana do grafo [2]. Em particular, o elemento (i, j) da matriz de

adjacência é diferente de zero apenas se há uma aresta conectando os nós i e j . Neste caso, ele é igual ao peso da aresta. Considerando um filtro linear e invariante ao deslocamento no grafo (*shift-invariant graph filter*), assume-se que o vetor $\mathbf{x}(n)$ é processado por este filtro para gerar o sinal no grafo filtrado [9]

$$\mathbf{y}(n) = \sum_{\ell=0}^{M-1} h_{\ell}^{\circ} \mathbf{A}^{\ell} \mathbf{x}(n-\ell) + \mathbf{v}(n), \quad (1)$$

em que $h_0^{\circ}, \dots, h_{M-1}^{\circ}$ são os M coeficientes do filtro, $\mathbf{v}(n) = [v_1(n), \dots, v_V(n)]^T \in \mathbb{R}^V$ é um ruído de média nula independente e identicamente distribuído (i.i.d.) com matriz de covariância $\mathbf{R}_v = \text{diag}\{\sigma_{v_k}^2\}_{k=1}^V$ que é assumido independente de qualquer outro sinal.

O vetor coluna \mathbf{h} que estima $\mathbf{h}^{\circ} = [h_0^{\circ}, \dots, h_{M-1}^{\circ}]^T$ pode ser obtido minimizando-se [9]

$$J(\mathbf{h}) = \sum_{k=1}^V J_k(\mathbf{h}), \quad \text{com } J_k(\mathbf{h}) \triangleq \mathbb{E}\{|y_k(n) - \mathbf{z}_k^T(n)\mathbf{h}|^2\}, \quad (2)$$

em que $y_k(n) = [\mathbf{y}(n)]_k$ e o vetor

$$\mathbf{z}_k(n) \triangleq \text{col}\{[\mathbf{A}^0 \mathbf{x}(n)]_k, \dots, [\mathbf{A}^{M-1} \mathbf{x}(n-M+1)]_k\} \quad (3)$$

agrega os k -ésimos elementos de todos os vetores $\{\mathbf{A}^{\ell} \mathbf{x}(n-\ell)\}_{\ell=0}^{M-1}$ e pode ser calculado localmente como mostrado em [9]. Seguindo uma estratégia adaptativa difusa para minimizar (2), em [9] são propostas duas versões de um algoritmo LMS difuso (dLMS): uma baseada em uma estratégia do tipo *adapt-then-combine* (ATC) e outra baseada em uma estratégia do tipo *combine-then-adapt* (CTA). Neste trabalho, considera-se apenas a versão ATC, mas os resultados podem ser estendidos para a versão CTA. Além disso, o algoritmo ATC difuso de [9] é estendido para uma versão normalizada:

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{h}_k(n) + \mu_k(n) \mathbf{z}_k(n) e_k(n) \\ \mathbf{h}_k(n+1) = \sum_{j \in \mathcal{N}_k} w_{kj} \boldsymbol{\psi}_j(n+1) \end{cases}, \quad (4)$$

em que

$$e_k(n) = y_k(n) - \mathbf{z}_k^T(n) \mathbf{h}_k(n), \quad (5)$$

é o erro de estimação no nó k , os vetores coluna $\boldsymbol{\psi}_k(n)$ e $\mathbf{h}_k(n)$ representam respectivamente as estimativas locais e combinadas de \mathbf{h}° no nó k , \mathcal{N}_k denota a vizinhança do nó k (incluindo o próprio nó k) e $\{w_{k,j}\}$ são pesos de combinação não negativos tais que

$$w_{kj} \geq 0, \quad \sum_{j \in \mathcal{N}_k} w_{kj} = 1, \quad \text{e } w_{kj} = 0 \text{ para } j \notin \mathcal{N}_k.$$

Neste trabalho, considera-se ainda um passo de adaptação local normalizado

$$\mu_k(n) = \tilde{\mu}_k / [\delta + \|\mathbf{z}_k(n)\|^2], \quad (6)$$

em que $0 < \tilde{\mu}_k < 2$ e δ é uma constante positiva usada para evitar divisão por zero.

III. O ALGORITMO DE AMOSTRAGEM PROPOSTO

Em cada iteração, o algoritmo ATC (4) estima o vetor de coeficientes \mathbf{h}° a partir dos dados $\{y_k(n), \mathbf{z}_k(n)\}$. Para incorporar o conceito de amostragem nesse algoritmo, define-se a variável binária $\bar{s}_k(n)$, que assume os valores zero ou um para determinar se $e_k(n)$ deve ser calculado ou não em

cada iteração. Dessa forma, é possível reescrever a equação de adaptação em (4) como

$$\boldsymbol{\psi}_k(n+1) = \mathbf{h}_k(n) + \bar{s}_k(n) \mu_k(n) \mathbf{z}_k(n) e_k(n). \quad (7)$$

Se $\bar{s}_k(n) = 1$, $e_k(n)$ é calculado como em (5) e (7) coincide com a etapa de adaptação em (4). Em contrapartida, se $\bar{s}_k(n) = 0$, $y_k(n)$ não é amostrado e, portanto, $e_k(n)$, $\mathbf{z}_k(n)$ e $\mu_k(n)$ não são calculados e $\boldsymbol{\psi}_k(n+1) = \mathbf{h}_k(n)$.

Para obter a variável binária $\bar{s}_k(n) \in \{0, 1\}$ e selecionar os nós que devem ser amostrados, utiliza-se uma variável auxiliar $s_k(n) \in [0, 1]$ tal que $\bar{s}_k(n) = 0$ para $s_k(n) < 0.5$ e $\bar{s}_k(n) = 1$ caso contrário. Procura-se então minimizar a seguinte função custo com relação a $s_k(n)$:

$$J_{s,k}(n) = [s_k(n)] \beta s_k(n) + [1 - s_k(n)] \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} e_i^2(n), \quad (8)$$

em que $\beta > 0$ é um parâmetro introduzido para controlar a taxa de amostragem dos nós. Essa função custo é uma combinação convexa em $s_k(n)$ da norma ℓ_1 do próprio $s_k(n)$ e da média dos quadrados dos erros na vizinhança do nó k . Se o erro quadrático é elevado, $s_k(n)$ deve ser próximo de um para minimizar $J_{s,k}(n)$, o que implica a amostragem do nó k . Em contrapartida, se o erro quadrático é suficientemente pequeno na vizinhança, $J_{s,k}(n)$ é minimizada fazendo-se $s_k(n)$ próximo de zero, o que implica que o nó k não deveria ser amostrado. Dessa forma, economiza-se em processamento em regime permanente mantendo-se a velocidade de convergência durante o transitório.

Com base em combinações convexas de filtros adaptativos (ver, e.g., [11], [12] e suas referências), em vez de atualizar diretamente $s_k(n)$, propõe-se adaptar $\alpha_k(n)$, uma variável relacionada a $s_k(n)$ por meio de [12]

$$s_k(n) = \phi_{\alpha_k}(n) \triangleq \frac{\text{sgm}[\alpha_k(n)] - \text{sgm}[-\alpha^+]}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}, \quad (9)$$

em que $\text{sgm}[x] = (1 + e^{-x})^{-1}$ é a função sigmoideal e α^+ é o maior valor positivo que $\alpha_k(n)$ pode assumir. Cabe notar que $s_k(n)$ assume os valores 1 e 0 para $\alpha_k(n) = \alpha^+$ e $\alpha_k(n) = -\alpha^+$, respectivamente. Um valor comumente adotado na literatura é $\alpha^+ = 4$ [11].

Derivando-se (8) em relação a $\alpha_k(n)$, obtém-se a seguinte regra de atualização:

$$\alpha_k(n+1) = \alpha_k(n) + \mu_s \phi'_{\alpha_k}(n) \left[\frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} e_i^2(n) - \beta s_k(n) \right], \quad (10)$$

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

$$\phi'_{\alpha_k}(n) \triangleq \frac{ds_k(n)}{d\alpha_k(n)} = \frac{\text{sgm}[\alpha_k(n)] \{1 - \text{sgm}[\alpha_k(n)]\}}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}. \quad (11)$$

Em sua forma atual, a Equação (10) não pode ser usada para controlar a amostragem porque requer que o erro na vizinhança do nó k (incluindo o próprio k) seja calculado para decidir se o nó k deve ser amostrado ou não, o que é contraditório. Por isso, aplica-se uma modificação heurística em (10), substituindo-se $e_i(n)$ por $\varepsilon_i(n)$, que denota a última medida de $e_i(n)$ à qual se tem acesso. Entretanto, essa alteração pode fazer com que o algoritmo deixe permanentemente de amostrar todos os nós,

o que deterioraria sua capacidade de rastreamento (*tracking*). Isto pode ser evitado substituindo-se $\beta s_k(n)$ em (10) por $\beta \bar{s}_k(n)$, conforme será explicado a seguir. Incorporando-se essas alterações em (10), obtém-se

$$\alpha_k(n+1) = \alpha_k(n) + \mu_s \phi'_{\alpha_k}(n) \left[\frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \varepsilon_i^2(n) - \beta \bar{s}_k(n) \right]. \quad (12)$$

Este algoritmo é denominado LMS difuso com amostragem adaptativa (AS-dLMS - *adaptive sampling diffusion LMS*). Cabe notar que $\mu_s \phi'_{\alpha_k}(n)$, o somatório de $\varepsilon_i^2(n)$ e $\beta \bar{s}_k(n)$ em (12) são sempre não negativos. Dessa forma, supondo que em uma dada iteração o nó k é amostrado ($\bar{s}_k = 1$) e que $\alpha_k = \alpha^+$, o Algoritmo (7) atualiza ψ_k para minimizar o MSE no grafo. Após algumas iterações, o MSE na vizinhança do nó k em (12) (primeiro termo entre colchetes) se torna menor que $\beta \bar{s}_k = \beta$. Portanto, em um ambiente estacionário, o termo de correção se torna negativo, o que diminui α_k até que ele também se torne negativo. Quando $\alpha_k < 0$, $\bar{s}_k = 0$ e o nó k não é mais amostrado. Neste caso, como $\beta \bar{s}_k = 0$, α_k cresce, tornando-se positivo novamente após algumas iterações, e o nó k é novamente amostrado. Dessa forma, o algoritmo não deixa de amostrar nenhum nó permanentemente, o que é essencial para detectar mudanças no ambiente. Além disso, quanto maior o MSE na vizinhança do nó k , mais cedo a amostragem desse nó recomeça.

Este mecanismo leva a uma diminuição no número de nós amostrados em regime permanente e conseqüentemente a uma economia em termos de custo computacional. Se β for escolhido apropriadamente, esta redução não ocorre durante o transitório e mantém-se a velocidade de convergência do algoritmo ATC sem o mecanismo de amostragem. Entretanto, cabe ressaltar que há um ligeiro aumento no custo durante o transitório. Por fim, diferentemente do algoritmo ATC sem o mecanismo de amostragem, o algoritmo AS-dLMS requer que cada nó amostrado transmita $e_i^2(n)$ aos seus vizinhos. Esta informação pode ser enviada junto com as estimativas ψ_i de modo a manter o número de transmissões inalterado.

IV. ESCOLHA DOS PARÂMETROS β E μ_s

O bom comportamento do algoritmo AS-dLMS depende de uma escolha apropriada para β e μ_s . Para determinar como essa escolha deve ser feita, nesta seção realiza-se uma análise do comportamento de $\alpha_k(n)$ enquanto o nó k é amostrado. Neste caso, pode-se substituir $\varepsilon_i^2(n)$ e $\beta \bar{s}_k(n)$ em (12) por $e_i^2(n)$ e β , respectivamente. Fazendo-se essas substituições, subtraindo $\alpha_k(n)$ de ambos os lados e tomando as esperanças, obtém-se

$$\mathbb{E}\{\Delta\alpha_k(n)\} = \mu_s \mathbb{E} \left\{ \phi'_{\alpha_k}(n) \left[\frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} e_i^2(n) - \beta \right] \right\}, \quad (13)$$

em que $\Delta\alpha_k(n) \triangleq \alpha_k(n+1) - \alpha_k(n)$. Para tornar a análise mais tratável, assume-se a independência estatística entre $\phi'_{\alpha_k}(n)$ e o termo entre colchetes em (13), o que leva a

$$\mathbb{E}\{\Delta\alpha_k(n)\} = \mu_s \mathbb{E}\{\phi'_{\alpha_k}(n)\} \left[\frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \mathbb{E}\{e_i^2(n)\} - \beta \right]. \quad (14)$$

Para que o nó k deixe de ser amostrado, α_k deve decrescer ao longo das iterações até se tornar negativo. Como $\mu_s \mathbb{E}\{\phi'_{\alpha_k}(n)\}$ é sempre positivo, para garantir que $\Delta\alpha_k(n)$ seja negativo na média, β deve satisfazer

$$\beta > \frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \mathbb{E}\{e_i^2(n)\}. \quad (15)$$

Assumindo que o algoritmo ATC estime apropriadamente o filtro ótimo, é razoável assumir que, em regime permanente, $\mathbb{E}\{e_i^2(n)\} \approx \sigma_{v_i}^2$, de modo que se tem

$$\frac{1}{|\mathcal{N}_k|} \sum_{i \in \mathcal{N}_k} \mathbb{E}\{e_i^2(n)\} \leq \sigma_{\max}^2 \triangleq \max_{i \in \mathcal{V}} \sigma_{v_i}^2. \quad (16)$$

Portanto, a condição

$$\beta > \sigma_{\max}^2 \quad (17)$$

é suficiente para garantir que na média todo nó deixará de ser amostrado em alguma iteração em regime permanente. A iteração em que isto ocorre é diferente para cada nó, uma vez que depende do MSE em sua vizinhança. Além disso, como explicado na Seção III, a interrupção na amostragem não é permanente. Cabe ressaltar que (17) não garante um bom desempenho em termos de MSE. Em particular, valores elevados de β podem levar a $\mathbb{E}\{\Delta\alpha_k(n)\} < 0$ durante o transitório, afetando a convergência e o rastreamento do algoritmo AS-dLMS. Resultados de simulação sugerem que se β for escolhido no intervalo $]\sigma_{\max}^2, 10\sigma_{\max}^2]$, o bom desempenho do algoritmo é preservado.

Escolhendo-se β apropriadamente, $\mathbb{E}\{\alpha_k(n)\} \approx \alpha^+$ durante o transitório e $\mathbb{E}\{\Delta\alpha_k(n)\} \leq 0$ quando $\bar{s}_k(n) = 1$ em regime permanente. Neste caso, pode-se encontrar um bom valor para μ_s estudando-se o quão rapidamente se obtém $\mathbb{E}\{\alpha_k(n)\} \leq 0$. Usando (14) e (16), obtém-se

$$\mathbb{E}\{\Delta\alpha_k(n)\} \leq \mu_s \mathbb{E}\{\phi'_{\alpha_k}(n)\} (\sigma_{\max}^2 - \beta). \quad (18)$$

Para simplificar (18), aproxima-se $\phi'_{\alpha_k}(n)$ no intervalo $[0, \alpha^+]$ por uma reta que cruza os pontos $(0, \phi'_0)$ e $(\alpha^+, \phi'_{\alpha^+})$, em que ϕ'_0 e ϕ'_{α^+} denotam os valores de ϕ'_{α_k} para $\alpha_k = 0$ e $\alpha_k = \alpha^+$, respectivamente. Esta aproximação é dada por

$$\phi'_{\alpha_k}(n) \approx \zeta \alpha_k(n) + \phi'_0, \quad (19)$$

em que $\zeta = [\phi'_{\alpha^+} - \phi'_0]/\alpha^+$. Para $\alpha^+ = 4$, esta é uma boa aproximação já que o erro quadrático médio no intervalo $[0, \alpha^+]$ é da ordem de 5×10^{-4} . Substituindo-se (19) em (18), obtém-se

$$\mathbb{E}\{\alpha_k(n+1)\} \lesssim \mathbb{E}\{\alpha_k(n)\} (1 + \zeta\theta) + \phi'_0\theta, \quad (20)$$

em que $\theta = \mu_s (\sigma_{\max}^2 - \beta)$. Como foi assumido que $\mathbb{E}\{\alpha_k(n)\} \approx \alpha^+$ durante o transitório, pode-se denotar a primeira iteração do regime permanente por n_0 e definir $n_0 + \Delta n \triangleq n + 1$. Então, considerando $\mathbb{E}\{\alpha_k(n_0)\} \approx \alpha^+$ em (20) e aplicando essa equação recursivamente, obtém-se

$$\mathbb{E}\{\alpha_k(n_0 + \Delta n)\} \lesssim \alpha^+ (1 + \zeta\theta)^{\Delta n} + \phi'_0\theta \sum_{\eta=0}^{\Delta n-1} (1 + \zeta\theta)^\eta. \quad (21)$$

Aplicando-se algumas manipulações algébricas, chega-se a

$$\mathbb{E}\{\alpha_k(n_0 + \Delta n)\} \lesssim [(\zeta\alpha^+ + \phi'_0)(1 + \zeta\theta)^{\Delta n} - \phi'_0]/\zeta. \quad (22)$$

Fazendo $E\{\alpha_k(n_0 + \Delta n)\}$ igual a zero em (22), obtém-se

$$\mu_s \approx \frac{\alpha^+}{(\beta - \sigma_{\max}^2)(\phi'_0 - \phi'_{\alpha^+})} \left[\left(\frac{\phi'_0}{\phi'_{\alpha^+}} \right)^{\frac{1}{\Delta n}} - 1 \right]. \quad (23)$$

Analisando-se (23), nota-se que quanto menor Δn , maior o valor de μ_s obtido, o que é razoável. É importante ressaltar que quanto maior β , pior se torna a aproximação em (23), já que valores elevados de β podem levar a $E\{\alpha_k(n_0)\}$ significativamente menor que α^+ , invalidando uma das hipóteses feitas para obter essa expressão. Entretanto, para $\beta \in]\sigma_{\max}^2, 10\sigma_{\max}^2]$, (23) concorda com os resultados de simulação e permite uma escolha mais embasada pra μ_s , como será mostrado a seguir.

V. RESULTADOS DE SIMULAÇÃO

Nesta seção, são apresentados os resultados da simulação obtidos em uma média de 100 realizações. Para facilitar a visualização, as curvas foram filtradas por um filtro de média móvel com 64 coeficientes. Em cada realização, gerou-se um grafo aleatório com $V = 20$ nós. Para isso, foram atribuídas coordenadas X e Y seguindo distribuições uniformes no intervalo $[-1, 1]$ a cada nó. Se a distância euclidiana entre dois nós é menor ou igual a 0,9, considera-se que eles estão conectados e uma aresta com peso unitário é criada. Um exemplo é mostrado na Figura 1(a). Grafos com nós isolados são descartados e utiliza-se a matriz de adjacência do grafo como o operador de deslocamento, sendo que esta foi normalizada em cada caso pelo seu maior autovalor. Em cada nó k , considera-se um sinal $x_k(n)$ que consiste em um ruído colorido, gerado a partir de $x_k(n) = 0, 1x_k(n-1) + 0, 9r_k(n)$, em que $r_k(n)$ é um ruído branco gaussiano i.i.d. com variância $\sigma_{r_k}^2 = 1 \forall k$. Além disso, é assumido $v_k(n)$ é um ruído branco gaussiano i.i.d. com uma variância $\sigma_{v_k}^2$ diferente para cada nó $k = 1, \dots, V$, como mostrado na Figura 1 (b). Para simular uma mudança abrupta no sistema ótimo, considerou-se

$$\mathbf{h}^o(n) = \begin{cases} [1 & -\frac{1}{2} & \frac{1}{4} & -\frac{1}{8} & \frac{1}{16}]^T, & \text{se } n \leq \frac{N}{2}, \\ [\frac{1}{16} & -\frac{1}{8} & \frac{1}{4} & -\frac{1}{2} & 1]^T, & \text{c.c.} \end{cases},$$

em que N é o número de iterações. Os pesos de combinação são dados por $w_{kj} = 1/|\mathcal{N}_k|$ se $j \in \mathcal{N}_k$ e $w_{kj} = 0$ caso contrário. Para cada nó, considera-se um valor diferente de $\tilde{\mu}$, como mostrado na Figura 1 (c), e foi adotado $\delta = 10^{-5}$ em todas as simulações. Como um indicador de desempenho, adotou-se o desvio quadrático médio (*mean-square-deviation* - MSD), dado por $\frac{1}{V} \sum_{k=1}^V E \left\{ \|\mathbf{h}^o(n) - \mathbf{h}_k(n)\|^2 \right\}$.

Primeiramente, é feita uma comparação entre o comportamento do algoritmo AS-dLMS com o algoritmo de difusão LMS com uma técnica de amostragem em que V_s nós são amostrados aleatoriamente em cada iteração [8]. Para cada algoritmo, as curvas de MSD, a quantidade média de nós amostrados e o número de multiplicações são mostrados ao longo das iterações nas Figuras 2 (a), 2 (b) e 2 (c), respectivamente. Os valores de β e μ_s adotados para o algoritmo AS-dLMS foram aqueles que apresentaram os melhores resultados nas simulações. Pode-se observar que quanto mais nós são amostrados, mais rápida é a convergência. O algoritmo AS-dLMS foi capaz de detectar a mudança abrupta no sistema

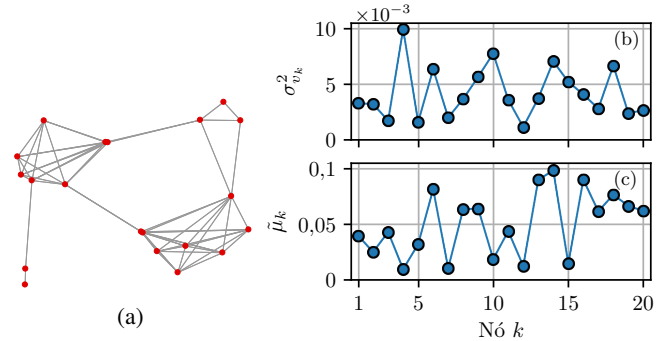


Fig. 1: (a) Exemplo de grafo gerado em uma realização utilizando o pacote GSPBOX [13]. (b) Valores de $\sigma_{v_k}^2$ e (c) $\tilde{\mu}_k$ para cada nó k .

ideal uma vez que todos os nós foram amostrados durante os transitórios, o que o levou a convergir tão rápido quanto o algoritmo dLMS de difusão com todos os nós amostrados. É interessante notar que quando o algoritmo AS-dLMS atinge o regime permanente, apenas quatro nós são amostrados na média. Por fim, pode-se observar também que, durante os transitórios, o custo computacional do algoritmo AS-dLMS é ligeiramente superior ao do dLMS com todos os nós amostrados, o que é amplamente compensado pela economia de custos computacionais em regime permanente.

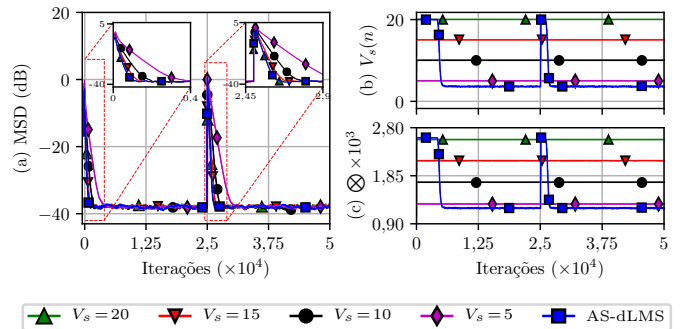


Fig. 2: Resultados de simulação obtidos com o algoritmo dLMS de [9] considerando-se diferentes quantidades de nós amostrados aleatoriamente e com o algoritmo AS-dLMS ($\beta = 0,02$, $\mu_s = 0,88$): (a) curvas de MSD, (b) Número de nós amostrados por iteração, e (c) Número de multiplicações por iteração.

Na Figura 3, são apresentadas as curvas MSD e a quantidade de nós amostrados ao longo das iterações para o algoritmo AS-dLMS, considerando valores diferentes de β e um passo de adaptação fixo ($\mu_s = 0,88$). Nas Figuras 3 (a) e 3 (c), considera-se o mesmo cenário da Figura 2, enquanto que nas Figuras 3 (b) e 3 (d) a potência do ruído foi igualada a $\sigma_{v_k}^2 = \sigma_{\max}^2 = 0,01$ para $k = 1, \dots, V$. No primeiro caso, nota-se que a Expressão (17) fornece uma estimativa conservadora para o valor mínimo de β . No segundo caso, $\beta = \sigma_{\max}^2$ mantém a amostragem de todos os nós, enquanto $\beta > \sigma_{\max}^2$ garante que alguns nós deixam de ser amostrados. Além disso, nota-se que quanto maior β , menor a quantidade de nós amostrados em regime permanente e mais rápido os nós deixam de ser amostrados, o que é benéfico em termos de custo computacional. No entanto, valores altos de β também podem comprometer o desempenho do algoritmo, já que em ambos os casos, $\beta = 1$ levou a baixas velocidades de convergência. Os resultados obtidos com $\beta = 0,03$ indicam uma taxa de convergência elevada antes da mudança abrupta

no sistema ótimo. Após essa alteração, entretanto, verifica-se uma convergência mais lenta. Uma possível interpretação para isto consiste no fato de que se tem $E\{\alpha_k(n)\} \ll \alpha^+$ logo antes da alteração e o valor relativamente elevado de β faz com que o algoritmo deixe de amostrar os nós pouco tempo após retomar a sua amostragem.

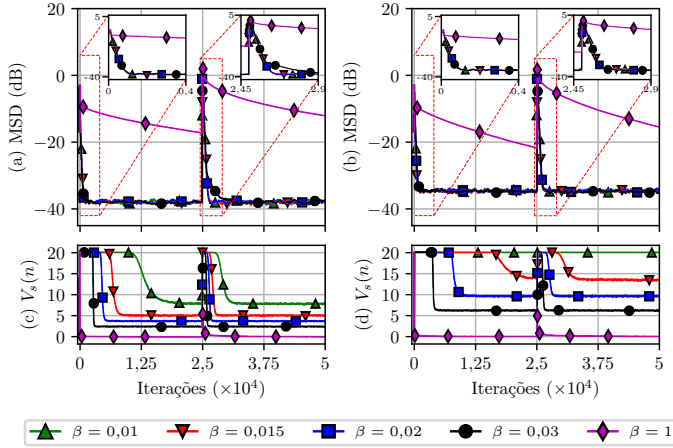


Fig. 3: Resultados de simulação obtidos com o algoritmo AS-dLMS ($\mu_s = 0,88$ e diferentes valores de β): (a) curvas de MSD e (c) número de nós amostrados por iteração para $\sigma_{v_k}^2$ como mostrado na Figura 1(b); (b) e (d) Resultados para $\sigma_{v_k}^2 = 0,01 \forall k$.

Nas simulações da Figura 4, utilizou-se a Expressão (23) para definir valores de μ_s para diferentes valores de β com $\Delta n = 5 \cdot 10^3$. Além do MSD e da quantidade de nós amostrados ao longo das iterações, na Figura 4 (c) também é mostrado o gráfico do MSE dado por $\frac{1}{V} \sum_{k=1}^V E\{e_k^2(n)\}$. Pode-se observar das Figuras 4 (b) e 4 (c) que, antes da mudança abrupta no sistema ótimo, o número de nós amostrados começa a cair aproximadamente ao mesmo tempo para todos os valores de β e que o número de iterações entre o início do regime de estado estacionário do MSE e o final deste processo é bastante próximo de Δn , o que corrobora a validade da Expressão (23). Comparando-se as Figuras 3 (a) e 4 (a), nota-se que escolher $\mu_s = 0,0089$ para $\beta = 1$ levou a um melhor desempenho em termos de MSD antes da mudança abrupta. No entanto, após essa alteração, o algoritmo deixa de amostrar os nós enquanto o MSD ainda está alto, levando novamente a uma convergência lenta. Isso não ocorre para $\beta \leq 0,02$, pois o algoritmo retoma a amostragem de cada nó no grafo e o regime permanente em termos de MSD é rapidamente atingido novamente. Esses resultados indicam que, mesmo com um passo de adaptação μ_s adequado, valores altos de β devem ser evitados, pois podem afetar o desempenho do algoritmo AS-dLMS.

VI. CONCLUSÕES

Neste trabalho, foram propostas modificações para o algoritmo LMS para grafos distribuído de [9] de modo a possibilitar o uso de técnicas de amostragem. Foi proposto ainda um mecanismo adaptativo de amostragem que utiliza a informação de mais nós quando o erro na rede é elevado e reduz a quantidade de nós amostrados caso contrário. Os resultados de simulação mostram que o algoritmo AS-dLMS é capaz de manter a taxa de convergência do algoritmo original de [9] durante o transitório e apresentar menor custo computacional durante o regime permanente. Além disso,

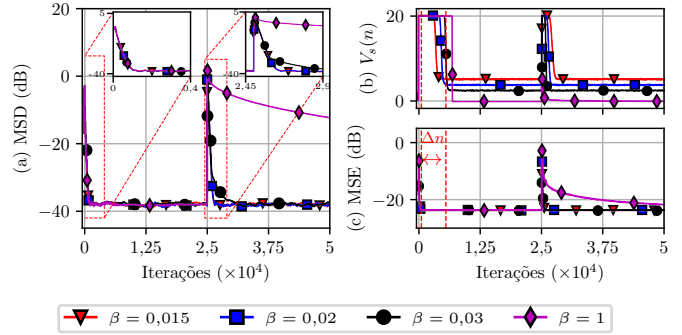


Fig. 4: Resultados de simulação obtidos com o algoritmo AS-dLMS (diferentes valores de β e μ_s ajustado por meio de (23) para cada caso): (a) curvas de MSD, (b) Número de nós amostrados por iteração, e (c) curvas de MSE.

foram obtidos resultados teóricos para auxiliar na escolha dos parâmetros de adaptação β e μ_s , os quais são corroborados pelos resultados de simulação. Ao mesmo tempo, foi mostrado que escolhas inadequadas de β e μ_s podem comprometer a taxa de convergência e o rastreamento do algoritmo AS-dLMS. Cabe notar que o resultado teórico referente à escolha do parâmetro β pressupõe o conhecimento apenas da ordem de grandeza da potência do ruído de medição. Em trabalhos futuros, pretende-se obter uma análise de desempenho do algoritmo proposto e testá-lo em outros cenários.

REFERÊNCIAS

- [1] A. Sandryhaila e J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, pp. 1644–1656, Apr. 2013.
- [2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, e P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, pp. 83–98, May 2013.
- [3] S. Chen, R. Varma, A. Sandryhaila, e J. Kovacevic, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, pp. 6510–6523, Dec. 2015.
- [4] A. Anis, A. Gadde, e A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, pp. 3775–3789, Jul. 2016.
- [5] M. Tsitsvero, S. Barbarossa, e P. Di Lorenzo, "Signals on graphs: Uncertainty principle e sampling," *IEEE Trans. Signal Process.*, vol. 64, pp. 4845–4860, Sep. 2016.
- [6] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, e G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, pp. 3584–3598, Jul. 2018.
- [7] F. Hua, R. Nassif, C. Richard, H. Wang, e A. H. Sayed, "A preconditioned graph diffusion LMS for adaptive graph signal processing," in *Proc. 26th EUSIPCO*, 2018, pp. 111–115.
- [8] P. Di Lorenzo, P. Banelli, S. Barbarossa, e S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Trans. Signal Process.*, vol. 65, pp. 4193–4208, Aug. 2017.
- [9] R. Nassif, C. Richard, J. Chen, e A. H. Sayed, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE ICASSP*, 2018, pp. 4129–4133.
- [10] A. H. Sayed, *Adaptation, Learning, e Optimization over Networks*, vol. 7, Foundations and Trends in Machine Learning, now Publishers Inc., Hanover, MA, 2014.
- [11] J. Arenas-Garcia, L. A. Azpicueta-Ruiz, M. T. M. Silva, V. H. Nascimento, e A. H. Sayed, "Combinations of adaptive filters: Performance e convergence properties," *IEEE Signal Process. Mag.*, vol. 33, pp. 120–140, Jan. 2016.
- [12] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, e J. Arenas-Garcia, "Adaptively biasing the weights of adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, pp. 3890–3895, Jul. 2010.
- [13] N. Perraudin, J. Paratte, D. I. Shuman, V. Kalofolias, P. Vandergheynst, e D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *arXiv*, vol. preprint:1408.5781, 2014.