



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
<http://oatao.univ-toulouse.fr/19120>

To cite this version: Thouvenin, Pierre-Antoine and Dobigeon, Nicolas and Tourneret, Jean-Yves *Une approche distribuée asynchrone pour la factorisation en matrices non-négatives - Application au démixage hyperspectral*. (2017) In: 26eme Colloque GRETSI sur le Traitement du Signal et des Images (GRETSI 2017), 5 September 2017 - 8 September 2017 (Juan-les-Pins, France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Une approche distribuée asynchrone pour la factorisation en matrices non-négatives – Application au démixage hyperspectral

Pierre-Antoine THOUVENIN, Nicolas DOBIGEON, Jean-Yves TOURNERET

Université de Toulouse, IRIT/INP-ENSEEIH, 2 rue Camichel, 31071 Toulouse Cedex 7, France

Travail soutenu par la Direction Générale de l'Armement (DGA) et par le projet ERA-NET MED MapInvPlnt n°ANR-15-NMED-0002-02.

prénom.nom@enseeiht.fr

Résumé – Le démixage d'images hyperspectrales est un exemple particulier du problème de factorisation en matrices non-négatives (NMF) qui consiste à identifier les signatures spectrales d'un milieu imagé ainsi que leurs proportions dans chacun des pixels. Toutefois, le nombre important de pixels composant ces images peut s'avérer contraignant en termes de mémoire ou de temps de calcul, ce qui peut motiver le développement de techniques d'estimation distribuées (entre plusieurs processeurs et/ou plusieurs machines). Dans cette perspective, nous proposons une approche distribuée du problème de démixage, basée sur de récentes avancées en optimisation distribuée asynchrone inspirées de l'algorithme *proximal alternating linearized minimization* (PALM). L'intérêt d'une estimation asynchrone par rapport à une procédure synchrone est illustré dans ce contexte sur des données synthétiques.

Abstract – As a particular instance of nonnegative matrix factorization (NMF), hyperspectral unmixing consists in identifying the reference spectral signatures contained in a hyperspectral image and their abundance fractions in each pixel. In practice, the significant spatial dimension of the images under study results in increasing memory requirements and computation time, thus justifying the use of distributed estimation techniques. In this perspective, we propose an asynchronous distributed unmixing algorithm based on recent advances in asynchronous non-convex optimization, related to the proximal alternating linearized minimization (PALM) algorithm. The gain brought by an asynchronous PALM algorithm, when compared to its synchronous counterpart, is illustrated on synthetic data.

1 Contexte et contribution

Acquises dans un grand nombre de bandes spectrales contiguës, les images hyperspectrales peuvent être modélisées comme un mélange de signatures spectrales associées aux matériaux contenus dans la zone imagée. À ce titre, le démixage vise à estimer les spectres purs des différents matériaux, ainsi que leurs proportions dans chacun des pixels de l'image à l'aide d'un modèle de mélange prédéfini. Dans ce contexte, la dimension des données acquises peut s'avérer contraignante pour le traitement des données. En particulier, le nombre important de pixels permet d'envisager la mise en œuvre de procédures d'estimation distribuées, synchrones ou asynchrones. De façon générale, pour un algorithme donné, l'asynchronie permet d'exploiter au mieux les différences éventuelles de performance entre les machines et/ou processeurs utilisés, réduisant ainsi le temps de calcul nécessaire par rapport à une version synchrone du même algorithme. Un très grand nombre de méthodes d'estimation asynchrones a été récemment proposé dans la littérature, aussi bien dans le cadre général d'opérateurs monotones définis dans un espace de Hilbert [1, 2], que pour des problèmes d'optimisation convexes [3–5] et non-convexes [6–11]. En revanche, la littérature consacrée au problème de démixage ne couvre qu'un nombre restreint de méthodes distribuées synchrones [12, 13].

Cet article s'intéresse plus particulièrement à la mise en œuvre d'algorithmes asynchrones inspirés de [14, 15], i.e., de variantes de l'algorithme PALM. En effet, ce type d'approche s'avère particulièrement adapté au problème de démixage, tout en disposant de garanties de convergence appropriées. De plus, la

version standard des algorithmes [14, 15] se prête particulièrement à une distribution synchrone de l'estimation des coefficients d'abondance, fournissant ainsi un cadre approprié pour évaluer précisément l'apport de l'asynchronie pour le démixage. Bien que de nombreuses approches asynchrones aient été proposées pour résoudre des problèmes d'optimisation non-convexes [6, 7, 10], les analyses de convergence proposées dans ces articles conduisent à un pas bien trop conservatif pour une descente de gradient proximale. De fait, seule l'approche proposée dans [16] permet d'obtenir une accélération de la décroissance de la fonction objectif intéressante en pratique, tout en disposant d'une garantie de convergence appropriée *via* une étape de relaxation à coefficients décroissants. Nous proposons ainsi d'évaluer la pertinence de ce type d'approche dans le cadre du démixage d'images hyperspectrales, par rapport à une version distribuée synchrone de l'algorithme PALM. Par la généralité de l'approche proposée, les contextes applicatifs peuvent concerner aussi bien le démixage d'une seule image partitionnée en plusieurs imagerie de taille réduite ou le démixage d'une série temporelle d'images hyperspectrales acquises à des dates différentes mais partageant les mêmes spectres purs. Par ailleurs, l'étude proposée se généralise aisément à d'autres problèmes de factorisation en matrices non-négatives pour lesquels l'algorithme PALM est applicable, et dont le démixage est un exemple particulier.

L'article est organisé comme suit. Le modèle de démixage est rappelé dans la section 2 tandis que la section 3 décrit la procédure de démixage proposée. Les résultats des simulations conduites sur données synthétiques sont rapportés dans les sections 4 et 5. En particulier, la section 5 illustre l'intérêt

de l'algorithme proposé pour un modèle adapté à la présence de variabilité spatiale. Enfin, la section 6 conclut cette étude.

2 Formulation du problème

Modèle de mélange. Chacun des N pixels $\mathbf{y}_n \in \mathbb{R}^L$ composant une image – où L désigne le nombre de bandes spectrales – est représenté par une combinaison convexe de R spectres purs, notés \mathbf{m}_r (R supposé connu). Dans la perspective d'une estimation distribuée, les données peuvent être regroupées en $I < N$ groupes disjoints de J pixels. Le modèle peut alors s'écrire

$$\forall i \in \{1, \dots, I\}, \mathbf{Y}_i = \mathbf{M}\mathbf{A}_i + \mathbf{B}_i \quad (1)$$

où les colonnes de \mathbf{Y}_i représentent les spectres des pixels du bloc i , celles de \mathbf{M} correspondent aux spectres purs \mathbf{m}_r , et celles de \mathbf{A}_i représentent les proportions de chacun des spectres purs dans un pixel du bloc i . Enfin, les colonnes de \mathbf{B}_i correspondent à un bruit additif, supposé blanc et gaussien, résultant des erreurs d'acquisition et de modèle. Pour assurer l'interprétabilité physique des résultats, les contraintes suivantes sont habituellement considérées [17]

$$\mathbf{A}_i^T \mathbf{1}_R = \mathbf{1}_J, \quad \mathbf{A}_i \succeq \mathbf{0}_{R,J}, \quad \mathbf{M} \succeq \mathbf{0}_{L,R} \quad (2)$$

où \succeq représente une inégalité terme à terme. Le modèle (1) s'applique indifféremment au démixage d'une image hyperspectrale scindée en différents blocs, ou au démixage d'une séquence d'images partageant les mêmes spectres purs (les blocs \mathbf{Y}_i représentent alors une image). Ce dernier cas motive principalement ce travail.

Formulation du problème. Si l'on note $\mathbf{A} \triangleq [\mathbf{A}_1, \dots, \mathbf{A}_I]$ et $\mathbf{Y} \triangleq [\mathbf{Y}_1, \dots, \mathbf{Y}_I]$, et si le bruit est supposé blanc et gaussien, le modèle (1) et les contraintes (2) permettent de formuler le problème sous la forme

$$(\mathbf{A}^*, \mathbf{M}^*) \in \arg \min_{\mathbf{A}, \mathbf{M}} \{F(\mathbf{A}, \mathbf{M}) + g(\mathbf{A}) + \iota_{\mathcal{A}_{R,N}}(\mathbf{A}) + h(\mathbf{M}) + \iota_{\{\succeq \mathbf{0}\}}(\mathbf{M})\} \text{ avec} \quad (3)$$

$$F(\mathbf{A}, \mathbf{M}) = \sum_{i=1}^I f_i(\mathbf{M}, \mathbf{A}_i) = \frac{1}{2} \sum_{i=1}^I \|\mathbf{Y}_i - \mathbf{M}\mathbf{A}_i\|_F^2 \quad (4)$$

$$\mathcal{A}_{R,J} = \{\mathbf{A} \in \mathbb{R}^{R \times J} : \mathbf{a}_n \in \mathcal{S}_R, \forall n \in \{1, \dots, J\}\}$$

$$\mathcal{S}_R = \{\mathbf{x} \in \mathbb{R}^R : x_r \geq 0 \text{ et } \mathbf{x}^T \mathbf{1}_R = 1\}$$

où F est l'opposé de la log-vraisemblance du modèle, $\iota_{\mathcal{S}}$ est l'indicatrice de l'ensemble \mathcal{S} ($\iota_{\mathcal{S}} = 0$ si $\mathbf{x} \in \mathcal{S}$, $+\infty$ sinon), et g et h représentent des pénalités convexes appropriées. En choisissant les fonctions g et h séparables, le problème (3) pourra être distribué de façon efficace dans la section suivante. Nous étudions dans cet article le cas où $g = 0$ et h est définie comme la distance mutuelle entre les spectres purs utilisée dans [18].

3 Une approche distribuée asynchrone

De façon générale, le modèle (1) est caractérisé par le fait que $R \ll L \ll N$. Ainsi, distribuer l'estimation des colonnes

Algorithme 1 : Algorithme de démixage distribué, décrit du point de vue du processus maître.

Données : $\mathbf{A}^0, \mathbf{M}^0, \mu \in]0, 1[$ ($\mu = 10^{-6}$ pour les sections 4 et 5).

$\gamma^0 \leftarrow 1;$

$k \leftarrow 1;$

tant que critère d'arrêt non satisfait **faire**

- // Attente des données d'un processus i^k*
- Attendre l'arrivée d'un bloc $i^k \in \{1, \dots, I\}$, i.e. $\tilde{\mathbf{A}}_{i^k}^k$ tel que :
 - (a) $\tilde{\mathbf{A}}_{i^k}^k = \text{prox}_{\iota_{\mathcal{A}_{R,J}}} \left(\mathbf{A}_{i^k}^k - \frac{1}{\eta_{i^k}^k} \nabla_{\mathbf{A}_{i^k}} f_{i^k}(\mathbf{A}_{i^k}^k, \mathbf{M}^{k-d_{i^k}^k}) \right);$
 - // Mise à jour des abondances*
 - (b) $\mathbf{A}_i^{k+1} = \begin{cases} \mathbf{A}_i^k + \gamma^k (\tilde{\mathbf{A}}_{i^k}^k - \mathbf{A}_i^k), & i = i^k \\ \mathbf{A}_i^k, & i \neq i^k \end{cases};$
 - // Mise à jour des spectres purs*
 - (c) $\tilde{\mathbf{M}}^k = \text{prox}_{\iota_{\{\succeq \mathbf{0}\}}} \left(\mathbf{M}^k - \frac{1}{\nu^k} \nabla_{\mathbf{M}} [F(\mathbf{A}^{k+1}, \mathbf{M}^k) + g(\mathbf{M}^k)] \right);$
 - $\mathbf{M}^{k+1} = \mathbf{M}^k + \gamma^k (\tilde{\mathbf{M}}^k - \mathbf{M}^k);$
 - // Transmission des résultats au processus i^k*
 - (d) Transmettre $(\mathbf{M}^{k+1}, \mathbf{A}_{i^k}^{k+1})$ au processus $i^k;$
 - // Mise à jour des variables auxiliaires*
 - $d_i^k = \begin{cases} 0, & i = i^k \\ d_i^k + 1, & i \neq i^k \end{cases};$
 - $\gamma^{k+1} = \gamma^k (1 - \mu \gamma^k);$
 - $k \leftarrow k + 1;$

Résultats : $\mathbf{M}, \mathbf{A}.$

de \mathbf{A} et des lignes de \mathbf{M} présente un intérêt notable. Toutefois, dans la perspective de l'analyse de données hyperspectrales multitemporelles (i.e., de séquences d'images hyperspectrales acquises à différents instants sur une même scène d'intérêt, susceptibles d'être composées des mêmes matériaux), il est raisonnable de considérer que des processus (ou machines) distincts ont accès à une partie seulement des pixels, partagés en I blocs disjoints composés de J pixels tels que, par simplicité de notations, $J I = N$. Ces processus sont alors chargés de la mise à jour des colonnes correspondantes de \mathbf{A} . La mise à jour de \mathbf{M} nécessite quant à elle de disposer de l'ensemble des informations agrégées lors de la mise à jour des abondances. Dans ce contexte, un processeur maître permet de répartir les tâches entre les différents processus, ainsi que d'agréger les informations qui lui sont transmises [8].

Sous des hypothèses similaires à celles de l'algorithme PALM [16], l'Algo. 1 décrit l'approche proposée, qui alterne entre la réception des données d'un processus i^k à l'instant k , la mise à jour des spectres purs (partagés entre les processus) et l'envoi des données mises à jour. L'asynchronie apparaît dans la mesure où l'état des spectres purs \mathbf{M} a potentiellement changé entre le moment où un processus a actualisé les données à sa disposition, et celui où les informations qu'il transmet sont agrégées par le processus maître (traduit par la notation des délais $d_{i^k}^k$ à l'étape (a) de l'Algo. 1). Pour un bloc i^k , les abondances sont mises à jour à l'aide de la règle suivante

$$\tilde{\mathbf{A}}_{i^k}^k = \text{prox}_{\iota_{\mathcal{A}_{R,J}}} \left(\mathbf{A}_{i^k}^k - \frac{1}{\eta_{i^k}^k} \nabla_{\mathbf{A}_{i^k}} f_{i^k}(\mathbf{A}_{i^k}^k, \mathbf{M}^{k-d_{i^k}^k}) \right) \quad (5)$$

$$\mathbf{A}_i^{k+1} = \mathbf{A}_i^k + \gamma^k (\tilde{\mathbf{A}}_{i^k}^k - \mathbf{A}_i^k) \quad (6)$$

où $\eta_{i^k}^k$ peut ici être pris égal à la constante de Lipschitz de la

TABLE 1 – Résultats de simulation sur données synthétiques.

	Sync.	Async.
aSAM(M) (°)	9.74e-01	1.04e+00
GMSE(A)	3.48e-04	5.25e-04
RE	1.05e-04	1.07e-04
aSAM(Y) (°)	2.23e-02	2.24e-02
temps (s)	1.39e+03	3.33e+02

fonction $\nabla_{\mathbf{A}_{i^k}} f_{i^k}(\cdot, \mathbf{M}^{k-d_{i^k}^k})$ [14, Remark 4 (iv)]. La mise à jour des spectres purs (et du paramètre ν^k associé) est conduite de façon analogue. Hormis les étapes de relaxation (b) et (c), notons que les paramètres sont mis à jour comme dans l’algorithme PALM [14, 15]. Par rapport à l’Algo. 1, la version synchrone utilisée pour les comparaisons de la section 4 consiste à attendre l’arrivée de l’ensemble des blocs i (étape (a)), à fixer $\gamma^k = 1$ et à transmettre les données mises à jour à l’ensemble des processus (étape (d)).

Garantie de convergence. L’Algo. 1 constitue un cas particulier de l’algorithme [16, Algo. 1] ce qui, sous des hypothèses similaires à celles de [15], assure que toute valeur d’adhérence de la séquence des itérés est presque sûrement un point critique de (3) [16, Th. 1] (dans la mesure où l’indice i^k du processus transmettant ses données à l’instant k , ainsi que le délai $d_{i^k}^k$ entre les éléments dont disposait ce processus par rapport à leur état courant pour le processus maître, sont représentés par des variables aléatoires discrètes, de distribution à support borné).

4 Simulation sur données synthétiques

Pour étudier l’intérêt éventuel de l’asynchronie pour le problème (1), nous comparons les performances de l’Algo. 1 par rapport à son analogue synchrone décrit au précédent paragraphe, pour $I = 3$ processus. Ces deux algorithmes, initialisés comme dans [18], ont été implantés en Julia [19]. Les données sont constituées de I images issues de mélanges linéaires de $R = 9$ spectres purs, acquis dans $L = 413$ bandes, et composées chacune de 100×100 pixels. Les images ainsi générées ont été corrompues par un bruit blanc additif gaussien, dont la variance assure un rapport signal-sur-bruit (RSB) de 30 dB.

Évaluation des performances. Les performances d’estimation rapportées dans la Table 1 sont évaluées en termes d’erreur de reconstruction (RE), d’angle spectral moyen (aSAM) pour les spectres purs, et d’erreur quadratique moyenne pour les abondances (GMSE). Ces mesures de qualité sont définies comme suit

$$\text{aSAM}(\widehat{\mathbf{M}}) = \frac{1}{R} \sum_{r=1}^R \arccos\left(\frac{\mathbf{m}_r^T \widehat{\mathbf{m}}_r}{\|\mathbf{m}_r\|_2 \|\widehat{\mathbf{m}}_r\|_2}\right)$$

$$\text{RE} = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F^2}{LN}, \quad \text{GMSE}(\widehat{\mathbf{A}}) = \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|_F^2}{RN}$$

où $\widehat{\mathbf{Y}}$ désigne la matrice des pixels reconstruits grâce aux paramètres estimés $\widehat{\mathbf{A}}, \widehat{\mathbf{M}}$. Enfin, les différents algorithmes ont la

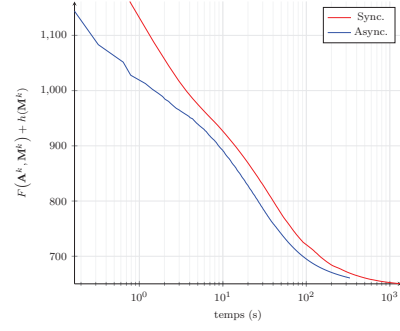


FIGURE 1 – Évolution de la fonction objectif au cours du temps, obtenue avec les différentes versions de l’algorithme PALM (jusqu’à convergence) pour le modèle (1).

même règle d’arrêt, correspondant à une décroissance relative de la fonction objectif inférieure à 10^{-5} .

Résultats. Les résultats obtenus en termes de temps de calcul, illustrés sur la figure 1, montrent l’intérêt de l’asynchronie pour réduire le temps nécessaire pour atteindre la convergence (gain d’un facteur 4 entre les deux versions). Cette modification, qui relève purement de l’implémentation de l’algorithme de démélange, laisse escompter un gain de temps d’autant plus important que la taille des données en jeu est susceptible d’augmenter. De plus, les estimés obtenus à convergence par les deux algorithmes sont relativement proches, ce qui est confirmé par les erreurs d’estimation indiquées dans la Table 1.

5 Simulation avec variabilité spatiale

Modèle de mélange. Nous proposons d’appliquer un algorithme similaire à l’Algo. 1 au problème de démélange en présence de variabilité spatiale défini par le modèle [20]

$$\mathbf{y}_n = \sum_{r=1}^R (\mathbf{m}_r + \mathbf{d}\mathbf{m}_{r,n}) a_{r,n} + \mathbf{b}_n \quad (7)$$

où $\mathbf{d}\mathbf{m}_{r,n}$ représente la variabilité du spectre pur \mathbf{m}_r dans le pixel n . En plus des contraintes (2), nous proposons de restreindre l’énergie de la variabilité capturée en imposant

$$\|\mathbf{d}\mathbf{M}_n\|_F \leq \varepsilon \quad (8)$$

où les colonnes de $\mathbf{d}\mathbf{M}_n$ sont les perturbations $\mathbf{d}\mathbf{m}_{r,n}$. Des détails complémentaires sur ce modèle sont disponibles dans [18]. En regroupant les pixels comme dans la section 2, l’Algo. 1 est modifié de sorte que chaque processus estime alternativement le sous-ensemble des abondances \mathbf{A} et des perturbations $\mathbf{d}\mathbf{M}_n$ qui lui est assigné.

Résultats. Comme dans le cas précédent, l’évolution de la fonction objectif au cours du temps (Fig. 2) montre l’intérêt d’une version asynchrone de l’algorithme. Étant donné que les deux algorithmes convergent vers des solutions pour lesquelles la valeur de la fonction objectif est assez différente, le temps reporté dans la Table 2 correspond à la durée nécessaire pour atteindre la valeur de la fonction objectif obtenue à convergence

TABLE 2 – Résultats de simulation sur données synthétiques en présence de variabilité spectrale.

	Sync.	Async.
aSAM(M) (°)	1.28e+00	1.47e+00
GMSE(A)	6.73e-04	8.52e-04
GMSE(dM)	2.17e-05	2.56e-05
RE	4.08e-05	2.88e-05
aSAM(Y) (°)	1.92e-02	1.57e-02
temps (s)	9.05e+01	0.73

par l’algorithme synchrone. Par ailleurs, la performance d’estimation de l’Algo. 1 est similaire à celle de la version synchrone, comme l’attestent les résultats reportés dans la Table 2.

6 Conclusion

Cet article a permis d’illustrer l’apport d’une procédure de démixage asynchrone [16], basée sur l’algorithme PALM [14, 15], par rapport à sa version synchrone. En particulier, l’asynchronie présente un intérêt notable pour le démixage d’images hyperspectrales multitemporelles, i.e., de séquences d’images hyperspectrales acquises pour une même zone à différents instants, pour lequel le nombre d’images à disposition peut s’avérer rédhibitoire pour le démixage conjoint des données. L’étude proposée se généralise par ailleurs à d’autres problèmes de factorisation en matrices non-négatives pour lesquels l’algorithme PALM est applicable, et dont le démixage est un cas particulier. L’algorithme adopté, testé sur données synthétiques, garantit que toute valeur d’adhérence de la suite des itérés est (presque sûrement) un point critique du problème de démixage [16]. Il conviendrait, par la suite, d’envisager différentes stratégies de mises-à-jour et de partage des données pour étudier leur éventuelle pertinence pour le démixage, en regard du compromis entre la réduction du temps de calcul et le coût occasionné par l’échange de données entre les processus. Enfin, la possibilité de considérer des opérateurs proximaux approchés, à l’instar de [15], constituerait une extension intéressante.

Références

- [1] P. L. Combettes and J. Eckstein, “Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions,” *Math. Program., Ser. B*, pp. 1–28, 2016.
- [2] Z. Peng *et al.*, “Arock : an algorithmic framework for asynchronous parallel coordinate updates,” *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. 2851–2879, Sep. 2016.
- [3] P. Bianchi and J. Jakubowicz, “Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization,” *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 391–405, Feb. 2013.
- [4] J.-C. Pesquet and A. Repetti, “A class of randomized primal-dual algorithms for distributed optimization,” *Journal of nonlinear and convex analysis*, vol. 16, no. 12, pp. 2453–2490, Nov. 2015.
- [5] S. Sra *et al.*, “Adadelay : Delay adaptive distributed stochastic optimization,” in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, no. 16, Cadiz, Spain, 2016, pp. 957–965.

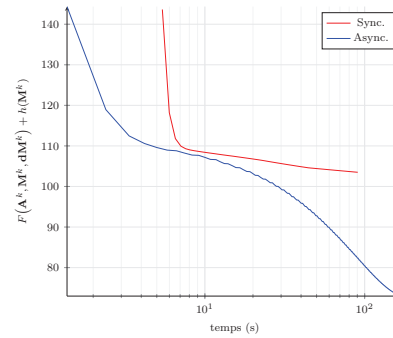


FIGURE 2 – Évolution de la fonction objectif au cours du temps pour les versions synchrone et asynchrone de l’algorithme PALM (jusqu’à convergence), pour le modèle (7).

- [6] M. Li *et al.*, “Communication efficient distributed machine learning with the parameter server,” in *Adv. in Neural Information Processing Systems*, 2014, pp. 91–27.
- [7] D. Davis, “The asynchronous PALM algorithm for nonsmooth nonconvex problems,” Apr. 2016, submitted. [Online]. Available : <https://arxiv.org/abs/1604.00526>
- [8] T.-H. Chang *et al.*, “Asynchronous distributed ADMM for large-scale optimization—part I : Algorithm and convergence analysis,” *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3118–3130, Jun. 2016.
- [9] F. Facchinei, G. Scutari, and S. Sagratella, “Parallel selective algorithms for nonconvex big data optimization,” *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874–1889, Apr. 2015.
- [10] Z. Peng *et al.*, “On the convergence of asynchronous parallel iteration with arbitrary delays,” Dec. 2016, arXiv preprint. [Online]. Available : <https://arxiv.org/abs/1612.04425>
- [11] G. Scutari *et al.*, “Parallel and distributed methods for nonconvex optimization-part I : Theory,” *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–2944, Apr. 2017.
- [12] S. A. Robila and D. Ricart, “Distributed algorithms for unmixing hyperspectral data using nonnegative matrix factorization with sparsity constraints,” in *Proc. IEEE Int. Conf. Geosci. Remote Sens. (IGARSS)*, Melbourne, Australia, Jul. 2013, pp. 2156–2159.
- [13] J. Sigurdsson *et al.*, “Sparse distributed hyperspectral unmixing,” in *Proc. IEEE Int. Conf. Geosci. Remote Sens. (IGARSS)*, Beijing, China, Jul. 2016, pp. 6994–6997.
- [14] J. Bolte, S. Sabach, and M. Teboulle, “Proximal alternating linearized minimization for nonconvex and nonsmooth problems,” *Mathematical Programming*, vol. 1-2, no. 146, pp. 459–494, Jul. 2013.
- [15] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, “A block coordinate variable metric forward-backward algorithm,” *J. Glob. Optim.*, vol. 66, no. 3, pp. 547–485, 2016.
- [16] L. Cannelli *et al.*, “Asynchronous parallel algorithms for nonconvex big-data optimization. Part I : Model and convergence,” Jul. 2016, arXiv preprint. [Online]. Available : <https://arxiv.org/abs/1607.04818>
- [17] J. M. Bioucas-Dias *et al.*, “Hyperspectral unmixing overview : Geometrical, statistical, and sparse regression-based approaches,” *IEEE J. Sel. Topics Appl. Earth Observ. in Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [18] P.-A. Thouvenin, N. Dobigeon, and J.-Y. Tourneret, “Estimation de variabilité pour le démixage non-supervisé d’images hyperspectrales,” in *Actes du XXVème Colloque GRETSI*, Lyon, France, Sep. 2015, in French.
- [19] J. Bezanson *et al.*, “Julia : A fresh approach to numerical computing,” *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.
- [20] P.-A. Thouvenin, N. Dobigeon, and J.-Y. Tourneret, “Hyperspectral unmixing with spectral variability using a perturbed linear mixing model,” *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 525–538, Jan. 2016.