SIAM J. SCI. COMPUT.
Vol. 38, No. 6, pp. A3618–A3643
© 2016 Society for Industrial and Applied Mathematics

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

COMPUTING EIGENVALUES OF LARGE SCALE SPARSE TENSORS ARISING FROM A HYPERGRAPH*

JINGYA CHANG†, YANNAN CHEN‡, AND LIQUN QI§

Abstract. The spectral theory of higher-order symmetric tensors is an important tool for revealing some important properties of a hypergraph via its adjacency tensor, Laplacian tensor, and signless Laplacian tensor. Owing to the sparsity of these tensors, we propose an efficient approach to calculate products of these tensors and any vectors. By using the state-of-the-art L-BFGS approach, we develop a first-order optimization algorithm for computing H- and Z-eigenvalues of these large scale sparse tensors (CEST). With the aid of the Łojasiewicz inequality, we prove that the sequence of iterates generated by CEST converges to an eigenvector of the tensor. When CEST is started from multiple random initial points, the resulting best eigenvalue could touch the extreme eigenvalue with a high probability. Finally, numerical experiments on small hypergraphs show that CEST is efficient and promising. Moreover, CEST is capable of computing eigenvalues of tensors related to a hypergraph with millions of vertices.

Key words. eigenvalue, hypergraph, Łojasiewicz inequality, Laplacian tensor, large scale tensor, L-BFGS, sparse tensor, spherical optimization

AMS subject classifications. 05C65, 15A18, 15A69, 65F15, 65K05, 90C35, 90C53

DOI. 10.1137/16M1060224

1. Introduction.

Since 1736, when Leonhard Euler posed a problem called "seven bridges of Königsberg," graphs and hypergraphs have been used to model relations and connections of objects in science and engineering, such as molecular chemistry [37, 38], image processing [21, 60], networks [34, 22], scientific computing [18, 32], and very large scale integration design [31]. For large scale hypergraphs, spectral hypergraph theory provides a fundamental tool. For instance, hypergraph-based spectral clustering has been used in complex networks [45], data mining [40], and statistics [54, 39]. In computer-aided design [64] and machine learning [23], researchers employed the spectral hypergraph partitioning. Other applications include the multilinear pagerank [24] and estimations of the clique number of a graph [8, 58].

Recently, spectral hypergraph theory is proposed to explore connections between the geometry of a uniform hypergraph and H- and Z-eigenvalues of some related symmetric tensors. Cooper and Dutle [13] proposed in 2012 the concept of an adjacency tensor for a uniform hypergraph. Two years later, Qi [52] gave definitions

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 5, 2016; accepted for publication (in revised form) September 15, 2016; published electronically November 15, 2016.

http://www.siam.org/journals/sisc/38-6/M106022.html

Funding: The work of the second author was supported by the National Natural Science Foundation of China (grant 11401539, 11571178), the Development Foundation for Excellent Youth Scholars of Zhengzhou University (grant 1421315070), and the Hong Kong Polytechnic University Postdoctoral Fellowship. The work of the third author was partially supported by the Hong Kong Research Grant Council (grant PolyU 501212, 501913, 15302114, and 15300715).

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; and School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China (jychang@zzu.edu.cn).

‡School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China and Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (ynchen@zzu.edu.cn).

§Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (maqilq@polyu.edu.hk).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of Laplacian and signless Laplacian tensors associated with a hypergraph. When an even-uniform hypergraph is connected, the largest H-eigenvalues of the Laplacian and signless Laplacian tensors are equal if and only if the hypergraph is odd bipartite [29]. This result gives a certification for checking whether a connected even-uniform hypergraph is odd bipartite or not.

We consider the problem of how to compute H- and Z-eigenvalues of the adjacency tensor, the Laplacian tensor, and the signless Laplacian tensor arising from a uniform hypergraph. Since the adjacency tensor and the signless Laplacian tensor are symmetric and nonnegative, an efficient numerical approach named the Ng–Qi–Zhou algorithm [46] could be applied to the problem of finding their largest H-eigenvalues and associated eigenvectors. Chang, Pearson, and Zhang [10] proved that the Ng–Qi–Zhou algorithm converges if the nonnegative symmetric tensor is primitive. Liu, Zhou, and Ibrahim [43] and Chang, Pearson, and Zhang [10] enhanced the Ng–Qi–Zhou algorithm and proved that the enhanced one converges if the nonnegative symmetric tensor is irreducible. Friedland, Gaubert, and Han [19] studied weakly irreducible nonnegative symmetric tensors and showed that the Ng–Qi–Zhou algorithm converges with an R-linear convergence rate for the largest H-eigenvalue problem of a weakly irreducible nonnegative symmetric tensor. Zhou, Qi, and Wu [62, 63] argued that the Ng–Qi–Zhou algorithm is Q-linear convergent. They refined the Ng–Qi–Zhou algorithm and reported that they could obtain the largest H-eigenvalue for any nonnegative symmetric tensors. A Newton's method with local quadratic rate of convergence is established by Ni and Qi [47].

With respect to the eigenvalue problem of general symmetric tensors, there are two sorts of methods. The first category could obtain all (real) eigenvalues of a tensor with only several variables. Qi, Wang, and Wang [53] proposed a direct approach based on the resultant. A semidefinite programming relaxation method coming from polynomial optimization was established by Cui, Dai, and Nie [14]. Chen, Han, and Zhou [11] preferred to use homotopy methods. Additionally, mathematical software Mathematica and Maple provides, respectively, subroutines `NSolve` and `solve` which could solve polynomial eigen-systems exactly. However, if we apply these methods to eigenvalue problem of a symmetric tensor with dozens of variables, the computational time is prohibitively long.

The second kind of methods turn to compute an (extreme) eigenvalue of a symmetric tensor, since a general symmetric tensor has plenty of eigenvalues [51] and it is NP-hard to compute all of them [28]. Kolda and Mayo [35, 36] proposed a spherical optimization model and established shifted power methods. By using fixed point theory, they proved that shifted power methods converge to an eigenvalue and its associated eigenvector of a symmetric tensor. For the same spherical optimization model, Hao, Cui, and Dai [27] preferred to use a subspace projection method. Han [26] constructed an unconstrained merit function that is indeed a quadratic penalty function of the spherical optimization. Preliminary numerical tests showed that these methods could compute eigenvalues of symmetric tensors with dozens of variables.

How to compute the (extreme) eigenvalue of the Laplacian tensor coming from an even-uniform hypergraph with millions of vertices? It is expensive to store and process a huge Laplacian tensor directly.

In this paper, we propose to store a uniform hypergraph by a matrix, whose row corresponds to an edge of that hypergraph. Then, instead of generating the large scale Laplacian tensor of the hypergraph explicitly, we give a fast computational framework for products of the Laplacian tensor and any vectors. The computational cost is linear in the size of edges and quadratic in the number of vertices of an edge, so it is cheap.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Other tensors arising from a uniform hypergraph, such as the adjacency tensor and the signless Laplacian tensor, could be processed in a similar way. These computational methods compose our main motivation.

Since products of any vectors and large scale tensors associated with a uniform hypergraph could be computed economically, we develop an efficient first-order optimization algorithm for computing H- and Z-eigenvalues of adjacency, Laplacian, and signless Laplacian tensors corresponding to the even-uniform hypergraph. In order to obtain an eigenvalue of an even-order symmetric tensor, we minimize a smooth merit function in a spherical constraint, whose first-order stationary point is an eigenvector associated with a certain eigenvalue. To preserve the spherical constraint, we derive an explicit formula for iterates by using the Cayley transform. Then, the algorithm for a spherical optimization looks like an unconstrained optimization. In order to deal with large scale problems, we explore the L-BFGS approach to generate a gradient-related direction and the backtracking search to facilitate the convergence of iterates. Based on these techniques, we obtain the novel algorithm (CEST) for computing eigenvalues of even-order symmetric tensors. Due to the algebraic nature of tensor eigenvalue problems, the smooth merit function enjoys the Łojasiewicz inequality. By using this property, we confirm that the sequence of iterates generated by CEST converges to an eigenvector corresponding to an eigenvalue. Moreover, if we start CEST from multiple initial points sampled uniformly from a unit sphere, it can be proved that the resulting best merit function value could touch the extreme eigenvalue with a high probability.

Numerical experiments show that CEST is dozens of times faster than the power method for eigenvalues of symmetric tensors related to small hypergraphs. Finally, we report that CEST could compute H- and Z-eigenvalues and associated eigenvectors of symmetric tensors involved in an even-uniform hypergraph with millions of vertices.

The outline of this paper is drawn as follows. We introduce some latest developments on spectral hypergraph theory in section 2. A new optimization algorithm based on L-BFGS and the Cayley transform is proposed in section 3. The convergence analysis of this algorithm is established in section 4. Section 5 addresses the computational issues on products of a vector and large scale sparse tensors arising from a uniform hypergraph. Numerical experiments reported in section 6 show that the new algorithm is efficient and promising. Finally, we conclude this paper in section 7.

2. Preliminary on spectral hypergraph theory. We introduce the definitions of eigenvalues of a symmetric tensor and then discuss developments in spectral hypergraph theory.

The conceptions of eigenvalues and associated eigenvectors of a symmetric tensor are established by Qi [51] and Lim [41] independently. Let

$$\mathbb{R}^{[k,n]} \equiv \mathbb{R}^{\overbrace{n \times \cdots \times n}^{k\text{-times}}}$$

denote the space of kth order n-dimensional real-valued tensors. Suppose

$$\mathcal{T} = (t_{i_1 \cdots i_k}) \in \mathbb{R}^{[k,n]} \qquad \text{for } \; i_j = 1, \ldots, n, j = 1, \ldots, k,$$

is a symmetric tensor. Here, the symmetry means that the value of $t_{i_1 \cdots i_k}$ is invariable under any permutation of its indices. For $\mathbf{x} \in \mathbb{R}^n$, we define a scalar

$$\mathcal{T}\mathbf{x}^k \equiv \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} t_{i_1 \cdots i_k} x_{i_1} \cdots x_{i_k} \in \mathbb{R}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Two column vectors $\mathcal{T}\mathbf{x}^{k-1} \in \mathbb{R}^n$ and $\mathbf{x}^{[k-1]} \in \mathbb{R}^n$ are defined with elements

$$(\mathcal{T}\mathbf{x}^{k-1})_i \equiv \sum_{i_2=1}^{n} \cdots \sum_{i_k=1}^{n} t_{ii_2\cdots i_k} x_{i_2} \cdots x_{i_k}$$

and $(\mathbf{x}^{[k-1]})_i \equiv x_i^{k-1}$ for $i = 1, \ldots, n$, respectively. Obviously, $\mathcal{T}\mathbf{x}^k = \mathbf{x}^\top(\mathcal{T}\mathbf{x}^{k-1})$.

If there exists a real λ and a nonzero vector $\mathbf{x} \in \mathbb{R}^n$ satisfying

$$\mathcal{T}\mathbf{x}^{k-1} = \lambda \mathbf{x}^{[k-1]}, \tag{2.1}$$

we call λ an H-eigenvalue of $\mathcal{T}$ and $\mathbf{x}$ its associated H-eigenvector. If the following system[1]

$$\begin{cases} \mathcal{T}\mathbf{x}^{k-1} = \lambda \mathbf{x}, \\ \mathbf{x}^\top \mathbf{x} = 1, \end{cases} \tag{2.2}$$

has a real solution $(\lambda, \mathbf{x})$, then λ is named a Z-eigenvalue of $\mathcal{T}$ and $\mathbf{x}$ is its associated Z-eigenvector. All of the H- and Z-eigenvalues of $\mathcal{T}$ are called its H-spectrum (Hspec($\mathcal{T}$)) and Z-spectrum (Zspec($\mathcal{T}$)), respectively.

These definitions on eigenvalues of a symmetric tensor have important applications in spectral hypergraph theory.

DEFINITION 2.1 (hypergraph). *We denote a hypergraph by $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the vertex set, $E = \{e_1, e_2, \ldots, e_m\}$ is the edge set, $e_p \subset V$ for $p = 1, \ldots, m$. If $|e_p| = k \geq 2$ for $p = 1, \ldots, m$ and $e_p \neq e_q$ when $p \neq q$, then G is called a uniform hypergraph or a k-graph. If $k = 2$, G is an ordinary graph.*

The k-graph $G = (V, E)$ is called odd bipartite if k is even and there exists a proper subset U of V such that $|e_p \cap U|$ is odd for $p = 1, \ldots, m$.

Let $G = (V, E)$ be a k-graph. For each $i \in V$, its degree $d(i)$ is defined as

$$d(i) = |\{e_p : i \in e_p \in E\}|.$$

We assume that every vertex has at least one edge. Thus, $d(i) > 0$ for all i. Furthermore, we define Δ as the maximum degree of G, i.e., $\Delta = \max_{1 \leq i \leq n} d(i)$.

A 4-uniform hypergraph is illustrated in Figure 1. There are ten vertices $V = \{1, 2, \ldots, 10\}$ and three edges $E = \{e_1 = \{1, 2, 3, 4\}, e_2 = \{1, 5, 6, 7\}, e_3 = \{1, 8, 9, 10\}\}$. Hence, it is a 4-graph, and its degrees are $d(1) = 3$ and $d(i) = 1$ for $i = 2, \ldots, 10$. Also, we have $\Delta = 3$. Moreover, this hypergraph is odd bipartite since we could take $U = \{1\} \subset V$.

DEFINITION 2.2 (adjacency tensor [13]). *Let $G = (V, E)$ be a k-graph with n vertices. The adjacency tensor $\mathcal{A} = (a_{i_1\cdots i_k})$ of G is a kth order n-dimensional symmetric tensor, whose elements are*

$$a_{i_1\cdots i_k} = \begin{cases} \dfrac{1}{(k-1)!} & \text{if } \{i_1, \ldots, i_k\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

DEFINITION 2.3 (Laplacian tensor and signless Laplacian tensor [52]). *Let G be a k-graph with n vertices. We denote its degree tensor $\mathcal{D}$ as a kth order n-dimensional*

[1] Qi [51] pointed out that the tensor $\mathcal{T}$ should be regular, i.e., zero is the unique solution of $\mathcal{T}\mathbf{x}^{k-1} = \mathbf{0}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

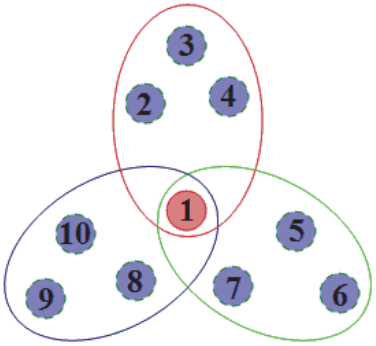

FIG. 1. *A 4-uniform hypergraph: sunflower.*

diagonal tensor whose ith diagonal element is $d(i)$. Then, the Laplacian tensor $\mathcal{L}$ and the signless Laplacian tensor $\mathcal{Q}$ of G are defined, respectively, as

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \qquad \textit{and} \qquad \mathcal{Q} = \mathcal{D} + \mathcal{A}.$$

Obviously, the adjacency tensor $\mathcal{A}$ and the signless Laplacian tensor $\mathcal{Q}$ of a hypergraph G are nonnegative. Moreover, they are weakly irreducible if and only if G is connected [50]. Hence, we could apply the Ng–Qi–Zhou algorithms [46, 10, 63] for computing their largest H-eigenvalues and associated H-eigenvectors. On the other hand, the Laplacian tensor $\mathcal{L}$ of a uniform hypergraph G is an M-tensor [61, 16]. Qi [52, Theorem 3.2] proved that zero is the smallest H-eigenvalue of $\mathcal{L}$. However, the following problems are still open.

- How to compute the largest H-eigenvalue of $\mathcal{L}$?
- How to calculate the smallest H-eigenvalues of $\mathcal{Q}$ and $\mathcal{A}$?
- How to obtain extreme Z-eigenvalues of $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$?

Many theorems in spectral hypergraph theory are given to address H- and Z-eigenvalues of $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$ when the involved hypergraphs have good geometric structures. For convenience, we denote the largest H-eigenvalue and the smallest H-eigenvalue of a tensor $\mathcal{T}$ related to a hypergraph G as $\lambda^H_{\max}(\mathcal{T}(G))$ and $\lambda^H_{\min}(\mathcal{T}(G))$, respectively. We also adopt similar notations for Z-eigenvalues of that tensor.

THEOREM 2.4. *Let G be a connected k-graph. Then the following assertions are equivalent.*

(i) *k is even and G is odd bipartite.*
(ii) $\lambda^H_{\max}(\mathcal{L}(G)) = \lambda^H_{\max}(\mathcal{Q}(G))$ *(from Theorem 5.9 of [29]).*
(iii) $\mathrm{Hspec}(\mathcal{L}(G)) = \mathrm{Hspec}(\mathcal{Q}(G))$ *(from Theorem 2.2 of [55]).*
(iv) $\mathrm{Hspec}(\mathcal{A}(G)) = -\mathrm{Hspec}(\mathcal{A}(G))$ *(from Theorem 2.3 of [55]).*
(v) $\mathrm{Zspec}(\mathcal{L}(G)) = \mathrm{Zspec}(\mathcal{Q}(G))$ *(from Theorem 8 of [7]).*

Khan, Fan, and Tan [33] studied a sort of non-odd-bipartite hypergraph and gave the following result.

THEOREM 2.5 (Corollary 3.6 of [33]). *Let G be a simple graph. For any positive integer k, we blow up each vertex of G into a set that includes k vertices and get a $2k$-graph $G^{2k,k}$. Then, $G^{2k,k}$ is not odd bipartite if and only if G is nonbipartite. Furthermore,*

$$\lambda_{\min}(\mathcal{A}(G)) = \lambda^H_{\min}(\mathcal{A}(G^{2k,k})) \quad \textit{and} \quad \lambda_{\min}(\mathcal{Q}(G)) = \lambda^H_{\min}(\mathcal{Q}(G^{2k,k})).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3. The CEST algorithm. The design of our CEST algorithm is based on a unified formula for the H- and Z-eigenvalues of a symmetric tensor [9, 17]. Let $\mathcal{I} \in \mathbb{R}^{[k,n]}$ be an identity tensor whose diagonal elements are all one and off-diagonal elements are zero. Hence, $\mathcal{I}\mathbf{x}^{k-1} = \mathbf{x}^{[k-1]}$. If k is even, we define a symmetric tensor $\mathcal{E} \in \mathbb{R}^{[k,n]}$ such that $\mathcal{E}\mathbf{x}^{k-1} = (\mathbf{x}^\top\mathbf{x})^{\frac{k}{2}-1}\mathbf{x}$. Obviously, tensors $\mathcal{I}$ and $\mathcal{E}$ are positive definite since $\mathcal{I}\mathbf{x}^k = \sum_{i=1}^n x_i^k$ and $\mathcal{E}\mathbf{x}^k = \|\mathbf{x}\|^k$ are positive for all $\mathbf{x} \neq \mathbf{0}$. By using tensors $\mathcal{I}$ and $\mathcal{E}$, we could combine systems (2.1) and (2.2) into

$$\mathcal{T}\mathbf{x}^{k-1} = \lambda\mathcal{B}\mathbf{x}^{k-1}, \tag{3.1}$$

where $\mathcal{B} = \mathcal{I}$ and $\mathcal{B} = \mathcal{E}$, respectively. In the remainder of this paper, we call a real λ and a nonzero vector $\mathbf{x} \in \mathbb{R}^n$ an eigenvalue and its associated eigenvector, respectively, if they satisfy (3.1). Now, we devote to computing such λ and $\mathbf{x}$ for large scale sparse tensors.

Let k be even. We consider the spherical optimization problem

$$\min\ f(\mathbf{x}) = \frac{\mathcal{T}\mathbf{x}^k}{\mathcal{B}\mathbf{x}^k} \qquad \text{s.t.}\ \ \mathbf{x} \in \mathbb{S}^{n-1}, \tag{3.2}$$

where the symmetric tensor $\mathcal{T}$ arises from a k-uniform hypergraph, so $\mathcal{T}$ is sparse and may be large scale. The symmetric positive definite tensor $\mathcal{B}$ has a simple structure such as $\mathcal{I}$ and $\mathcal{E}$. Without loss of generality, we restrict $\mathbf{x}$ to the unit sphere $\mathbb{S}^{n-1} \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top\mathbf{x} = 1\}$ because $f(\mathbf{x})$ is zero-order homogeneous. The compactness of $\mathbb{S}^{n-1}$ makes it easy to obtain and understand the convergence analysis of our algorithm CEST. Additionally, $\mathbf{x} \in \mathbb{S}^{n-1}$ is away from the original point.

The gradient of $f(\mathbf{x})$ [12] is

$$\nabla f(\mathbf{x}) = \frac{k}{\mathcal{B}\mathbf{x}^k}\left(\mathcal{T}\mathbf{x}^{k-1} - \frac{\mathcal{T}\mathbf{x}^k}{\mathcal{B}\mathbf{x}^k}\mathcal{B}\mathbf{x}^{k-1}\right). \tag{3.3}$$

The following theorem reveals the relationship between the spherical optimization (3.2) and the eigenvalue problem (3.1).

THEOREM 3.1. *Suppose that the order k is even and the symmetric tensor $\mathcal{B}$ is positive definite. Let $\mathbf{x}_* \in \mathbb{S}^{n-1}$. Then, $\mathbf{x}_*$ is a first-order stationary point, i.e., $\nabla f(\mathbf{x}_*) = 0$, if and only if there exists a scalar $\lambda_* \in \mathbb{R}$ such that $(\lambda_*, \mathbf{x}_*)$ satisfies (3.1). In fact, $\lambda_* = f(\mathbf{x}_*)$ is the H-eigenvalue (resp., Z-eigenvalue) and $\mathbf{x}_*$ is the associated H-eigenvector (resp., Z-eigenvector) if $\mathcal{B} = \mathcal{I}$ (resp., $\mathcal{B} = \mathcal{E}$).*

Proof. Since $\mathcal{B}$ is positive definite, $\mathcal{B}\mathbf{x}^k > 0$ for all $\mathbf{x} \in \mathbb{S}^{n-1}$. Hence, from (3.3), if $\mathbf{x}_* \in \mathbb{S}^{n-1}$ satisfies $\nabla f(\mathbf{x}_*) = 0$, $f(\mathbf{x}_*)$ is an eigenvalue and $\mathbf{x}_*$ is its associated eigenvector.

On the other hand, suppose that $\mathbf{x}_* \in \mathbb{S}^{n-1}$ is an eigenvector corresponding to an eigenvalue λ_*, i.e.,

$$\mathcal{T}\mathbf{x}_*^{k-1} = \lambda_*\mathcal{B}\mathbf{x}_*^{k-1}.$$

By taking inner products on both sides with $\mathbf{x}_*$, we get $\mathcal{T}\mathbf{x}_*^k = \lambda_*\mathcal{B}\mathbf{x}_*^k$. Because $\mathcal{B}\mathbf{x}_*^k > 0$, we have $\lambda_* = \frac{\mathcal{T}\mathbf{x}_*^k}{\mathcal{B}\mathbf{x}_*^k} = f(\mathbf{x}_*)$. Hence, we obtain $\nabla f(\mathbf{x}_*) = 0$ from (3.3). ∎

Next, we focus on numerical approaches for computing a first-order stationary point of the spherical optimization problem (3.2). First, we apply the limited memory BFGS (L-BFGS) approach to generate a search direction. Then, a curvilinear search technique is explored to preserve iterates in a spherical constraint.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.1. L-BFGS produces a search direction. The limited memory BFGS method is powerful for large scale nonlinear unconstrained optimization. In the current iteration c, it constructs an implicit matrix H_c to approximate the inverse of a Hessian of $f(\mathbf{x})$. At the beginning, we introduce the basic BFGS update.

BFGS is a quasi-Newton method which updates the approximation of the inverse of a Hessian iteratively. Let H_c be the current approximation,

$$\mathbf{y}_c = \nabla f(\mathbf{x}_{c+1}) - \nabla f(\mathbf{x}_c), \qquad \mathbf{s}_c = \mathbf{x}_{c+1} - \mathbf{x}_c, \quad \text{and} \quad V_c = I - \rho_c \mathbf{y}_c \mathbf{s}_c^\top, \tag{3.4}$$

where I is an identity matrix,

$$\rho_c = \begin{cases} \dfrac{1}{\mathbf{y}_c^\top \mathbf{s}_c} & \text{if } \mathbf{y}_c^\top \mathbf{s}_c \geq \kappa_\epsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

and $\kappa_\epsilon \in (0,1)$ is a small positive constant. We generate the new approximation H_c^+ by the BFGS formula [49, 56] as

$$H_c^+ = V_c^\top H_c V_c + \rho_c \mathbf{s}_c \mathbf{s}_c^\top. \tag{3.6}$$

For the purpose of solving large scale optimization problems, Nocedal [48] proposed the L-BFGS approach which implements the BFGS update in an economic way. Given any vector $\nabla f \in \mathbb{R}^n$, the matrix-vector product $-H_c \nabla f$ could be computed with only $\mathcal{O}(n)$ multiplications.

In each iteration c, L-BFGS starts from a simple matrix

$$H_c^{(0)} = \gamma_c I, \tag{3.7}$$

where $\gamma_c > 0$ is usually determined by the Barzilai–Borwein method [42, 5]. Then, we use BFGS formula (3.6) to update $H_c^{(\ell)}$ recursively,

$$H_c^{(L-\ell+1)} = V_{c-\ell}^\top H_c^{(L-\ell)} V_{c-\ell} + \rho_{c-\ell} \mathbf{s}_{c-\ell} \mathbf{s}_{c-\ell}^\top \qquad \text{for } \ell = L, L-1, \ldots, 1, \tag{3.8}$$

and obtain

$$H_c = H_c^{(L)}. \tag{3.9}$$

If $\ell \geq c$, we define $\rho_{c-\ell} = 0$ and L-BFGS does nothing for that ℓ. In a practical implementation, L-BFGS enjoys a cheap two-loop recursion. The computational cost is about $4Ln$ multiplications.

For the parameter γ_c, we have three candidates. The first two are suggested by Barzilai and Borwein [5]:

$$\gamma_c^{\mathrm{BB1}} = \frac{\mathbf{y}_c^\top \mathbf{s}_c}{\|\mathbf{y}_c\|^2} \qquad \text{and} \qquad \gamma_c^{\mathrm{BB2}} = \frac{\|\mathbf{s}_c\|^2}{\mathbf{y}_c^\top \mathbf{s}_c}. \tag{3.10}$$

The third one is their geometric mean [15]

$$\gamma_c^{\mathrm{Dai}} = \frac{\|\mathbf{s}_c\|}{\|\mathbf{y}_c\|}. \tag{3.11}$$

Also, we set $\gamma_c = 1$ if $\mathbf{y}_c^\top \mathbf{s}_c < \kappa_\epsilon$. Furthermore, we argue that $\mathbf{p}_c = -H_c \nabla f(\mathbf{x}_c)$ generated by L-BFGS is a gradient-related direction [49, 42]:

$$\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \leq -C_L \|\nabla f(\mathbf{x}_c)\|^2 \qquad \text{and} \qquad \|\mathbf{p}_c\| \leq C_U \|\nabla f(\mathbf{x}_c)\|, \tag{3.12}$$

where constants C_L and C_U satisfy $0 < C_L \leq 1 \leq C_U$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm L-BFGS The two-loop recursion for L-BFGS [48, 49, 56].

$\mathbf{q} \leftarrow -\nabla f(\mathbf{x}_c)$,
for $i = c-1, c-2, \ldots, c-L$ **do**
  $\alpha_i \leftarrow \rho_i \mathbf{s}_i^\top \mathbf{q}$,
  $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$,
end for
$\mathbf{p} \leftarrow \gamma_c \mathbf{q}$,
for $i = c-L, c-L+1, \ldots, c-1$ **do**
  $\beta \leftarrow \rho_i \mathbf{y}_i^\top \mathbf{p}$,
  $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{s}_i(\alpha_i - \beta)$,
end for
Stop with result $\mathbf{p} = -H_c \nabla f(\mathbf{x}_c)$.

3.2. Cayley transform preserves the spherical constraint. The Cayley transform [25] provides a smart approach for constructing orthogonal matrices, which play an important role in computation of the eigenvalue problem [20, 12] and optimization with orthogonal constraints [57, 30]. Let $W \in \mathbb{R}^{n\times n}$ be a skew-symmetric matrix. Then $(I + W)$ is invertible. The Cayley transform,

$$Q = (I + W)^{-1}(I - W) \in \mathbb{R}^{n\times n}, \tag{3.13}$$

produces an orthogonal matrix Q whose eigenvalues do not contain -1. Here, the matrix multiplication is commutative.

Suppose that $\mathbf{x}_c \in \mathbb{S}^{n-1}$ is the current iterate, $\mathbf{p}_c \in \mathbb{R}^n$ is a descent direction generated by Algorithm L-BFGS, and α is a damping factor. To preserve the spherical constraint, we aim to choose an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and generate the new iterate by

$$\mathbf{x}_{c+1} = Q\mathbf{x}_c. \tag{3.14}$$

At the same time, the new iterate $\mathbf{x}_{c+1}$ is required to satisfy the descent condition

$$\nabla f(\mathbf{x}_c)^\top (\mathbf{x}_{c+1} - \mathbf{x}_c) < 0.$$

By using the Cayley transform (3.13), we have $(I+W)\mathbf{x}_{c+1} = (I-W)\mathbf{x}_c$ and, hence, $\mathbf{x}_{c+1} - \mathbf{x}_c = -W(\mathbf{x}_{c+1} + \mathbf{x}_c)$. For simplicity, we take the skew-symmetric matrix W as

$$W = \mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top, \tag{3.15}$$

where $\mathbf{a}$ and $\mathbf{b}$ are two undetermined vectors. By some calculations, we have

$$\nabla f(\mathbf{x}_c)^\top (\mathbf{x}_{c+1} - \mathbf{x}_c) = -\nabla f(\mathbf{x}_c)^\top (\mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top)(\mathbf{x}_{c+1} + \mathbf{x}_c).$$

Because $f(\mathbf{x})$ in (3.2) is a zero-order homogeneous function, the projection of the gradient $\nabla f(\mathbf{x})$ on $\mathbf{x}$ is zero, i.e.,

$$\mathbf{x}^\top \nabla f(\mathbf{x}) = 0 \qquad \forall\, \mathbf{x} \neq \mathbf{0}. \tag{3.16}$$

This equality motivates us to choose $\mathbf{a} = \mathbf{x}_c$. Then, by (3.14), we get

$$\nabla f(\mathbf{x}_c)^\top (\mathbf{x}_{c+1} - \mathbf{x}_c) = \nabla f(\mathbf{x}_c)^\top \mathbf{b}(\mathbf{x}_c^\top Q\mathbf{x}_c + \mathbf{x}_c^\top \mathbf{x}_c).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Since Q is an orthogonal matrix and does not have -1 as its eigenvalue, we have $\mathbf{x}_c^\top Q\mathbf{x}_c + \mathbf{x}_c^\top \mathbf{x}_c > 0$ for all $\mathbf{x}_c \in \mathbb{S}^{n-1}$. It is as expected that when we pick $\mathbf{b} = \alpha\mathbf{p}_c$, the descent condition $\nabla f(\mathbf{x}_c)^\top(\mathbf{x}_{c+1} - \mathbf{x}_c) < 0$ holds because of (3.12).

Indeed, matrices W and Q do not need to be formed explicitly. The new iterate $\mathbf{x}_{c+1}$ could be generated from $\mathbf{x}_c$ and $\mathbf{p}_c$ directly with only about $4n$ multiplications.

LEMMA 3.2. *Suppose that the new iterate* $\mathbf{x}_{c+1}$ *is generated by* (3.13), (3.14), *and* (3.15). *Then, we have*

$$\mathbf{x}_{c+1}(\alpha) = \frac{[(1-\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2 - \|\alpha\mathbf{p}_c\|^2]\mathbf{x}_c + 2\alpha\mathbf{p}_c}{1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2} \tag{3.17}$$

and

$$\|\mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c\| = 2\left(\frac{\|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}{1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}\right)^{\frac{1}{2}}. \tag{3.18}$$

Proof. We employ the Sherman–Morrison–Woodbury formula: if A is invertible,

$$(A + UV^\top)^{-1} = A^{-1} - A^{-1}U(I + V^\top A^{-1}U)^{-1}V^\top A^{-1}.$$

It follows that

$$\begin{aligned}
(I+W)^{-1}\mathbf{x}_c &= \left(I + \begin{bmatrix} \mathbf{x}_c & -\alpha\mathbf{p}_c \end{bmatrix}\begin{bmatrix} \alpha\mathbf{p}_c^\top \\ \mathbf{x}_c^\top \end{bmatrix}\right)^{-1}\mathbf{x}_c \\
&= \left(I - \begin{bmatrix} \mathbf{x}_c & -\alpha\mathbf{p}_c \end{bmatrix}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \alpha\mathbf{p}_c^\top \\ \mathbf{x}_c^\top \end{bmatrix}\begin{bmatrix} \mathbf{x}_c & -\alpha\mathbf{p}_c \end{bmatrix}\right)^{-1}\begin{bmatrix} \alpha\mathbf{p}_c^\top \\ \mathbf{x}_c^\top \end{bmatrix}\right)\mathbf{x}_c \\
&= \frac{(1-\alpha\mathbf{x}_c^\top\mathbf{p}_c)\mathbf{x}_c + \alpha\mathbf{p}_c}{1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2},
\end{aligned}$$

where $1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2 \geq 1$ since $\mathbf{x}_c \in \mathbb{S}^{n-1}$ and $|\alpha\mathbf{x}_c^\top\mathbf{p}_c| \leq \|\alpha\mathbf{p}_c\|$. Then, the calculation of $\mathbf{x}_{c+1}$ is straightforward, i.e.,

$$\mathbf{x}_{c+1} = (I-W)\frac{(1-\alpha\mathbf{x}_c^\top\mathbf{p}_c)\mathbf{x}_c + \alpha\mathbf{p}_c}{1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2} = \frac{[(1-\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2 - \|\alpha\mathbf{p}_c\|^2]\mathbf{x}_c + 2\alpha\mathbf{p}_c}{1 + \|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}.$$

Hence, the iterate formula (3.17) is valid. Then, the proof of (3.18) is also straightforward. ∎

Jiang and Dai [30, Lemma 3.1] gave a similar update scheme to (3.17). If we define $\mathbf{q}_c \equiv (I - \mathbf{x}_c\mathbf{x}_c^\top)\mathbf{p}_c$ which is the projection of the descent direction $\mathbf{p}_c$ onto the tangent space $\{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^\top\mathbf{x}_c = 0\}$ of $\mathbb{S}^{n-1}$ at $\mathbf{x}_c$, we have

$$\mathbf{x}_{c+1}(\alpha) = \frac{(1-\alpha^2\|\mathbf{q}_c\|^2)\mathbf{x}_c + 2\alpha\mathbf{q}_c}{1 + \alpha^2\|\mathbf{q}_c\|^2} \in \mathbb{S}^{n-1}.$$

Hence, the update formula (3.17) is a retraction on the unit sphere [2]. Indeed, $\mathbf{x}_{c+1}(\alpha)$ is a geodesic rooted at $\mathbf{x}_c$ along the descent direction [57]. Whereafter, the damping factor α could be determined by an inexact line search owing to the following theorem.

THEOREM 3.3. *Suppose that* $\mathbf{p}_c$ *is a gradient-related direction satisfying* (3.12) *and* $\mathbf{x}_{c+1}(\alpha)$ *is generated by* (3.17). *Let* $\eta \in (0,1)$ *and* $\nabla f(\mathbf{x}_c) \neq 0$. *Then, there is an* $\tilde{\alpha}_c > 0$ *such that for all* $\alpha \in (0, \tilde{\alpha}_c]$,

$$f(\mathbf{x}_{c+1}(\alpha)) \leq f(\mathbf{x}_c) + \eta\alpha\mathbf{p}_c^\top\nabla f(\mathbf{x}_c). \tag{3.19}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm CEST Computing eigenvalues of sparse tensors.

For a given uniform hypergraph G_r, we compute the degree vector $\mathbf{d}$.
Choose an initial unit iterate $\mathbf{x}_1$, a positive integer L, parameters $\eta \in (0,1)$, $\beta \in (0,1)$, and $c \leftarrow 1$.
while $\nabla f(\mathbf{x}_c) \neq \mathbf{0}$ **do**
  Compute $\mathcal{T}\mathbf{x}_c^{k-1}$ and $\mathcal{T}\mathbf{x}_c^{k}$ by using approaches proposed in section 5, where $\mathcal{T} \in \{\mathcal{A}, \mathcal{L}, \mathcal{Q}\}$.
  Calculate $\lambda_c = f(\mathbf{x}_c)$ and $\nabla f(\mathbf{x}_c)$ by (3.2) and (3.3), respectively.
  Generate $\mathbf{p}_c = -H_c \nabla f(\mathbf{x}_c)$ by Algorithm L-BFGS.
  Choose the smallest nonnegative integer ℓ such that $\alpha = \beta^\ell$ satisfies (3.19).
  Let $\alpha_c = \beta^\ell$ and update the new iterate $\mathbf{x}_{c+1} = \mathbf{x}_{c+1}(\alpha_c)$ by (3.17).
  Compute $\mathbf{s}_c, \mathbf{y}_c$, and ρ_c by (3.4) and (3.5), respectively.
  $c \leftarrow c + 1$.
end while

Proof. From (3.17), we obtain $\mathbf{x}_{c+1}(0) = \mathbf{x}_c$ and $\mathbf{x}'_{c+1}(0) = -2\mathbf{x}_c^\top \mathbf{p}_c \mathbf{x}_c + 2\mathbf{p}_c$. Hence, we have

$$\left.\frac{\mathrm{d}f(\mathbf{x}_{c+1}(\alpha))}{\mathrm{d}\alpha}\right|_{\alpha=0} = \nabla f(\mathbf{x}_{c+1}(0))^\top \mathbf{x}'_{c+1}(0) = \nabla f(\mathbf{x}_c)^\top(-2\mathbf{x}_c^\top \mathbf{p}_c \mathbf{x}_c + 2\mathbf{p}_c) = 2\mathbf{p}_c^\top \nabla f(\mathbf{x}_c),$$

where the last equality follows from (3.16). Since $\nabla f(\mathbf{x}_c) \neq 0$ and $\mathbf{p}_c$ satisfies (3.12), we have $\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) < 0$. Then, by Taylor's theorem, for a sufficiently small α, we obtain

$$f(\mathbf{x}_{c+1}(\alpha)) = f(\mathbf{x}_c) + 2\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + o(\alpha^2).$$

Owing to $\eta < 2$, there exists a positive $\tilde{\alpha}_c$ such that (3.19) is valid. ∎

Finally, we present the new Algorithm CEST formally. Roughly speaking, CEST is a modified version of the L-BFGS method for spherical constrained optimization. We use the Cayley transform explicitly to preserve iterates on a unit sphere. An inexact line search is employed to determine a suitable damping factor. Theorem 3.3 indicates that the inexact line search is well-defined.

4. Convergence analysis. First, we prove that the sequence of function values $\{f(\mathbf{x}_c)\}$ converges and every accumulation point of iterates $\{\mathbf{x}_c\}$ is a first-order stationary point. Second, by using the Łojasiewicz inequality, we show that the sequence of iterates $\{\mathbf{x}_c\}$ is also convergent. When the second-order sufficient condition holds at the limiting point, CEST enjoys a linear convergence rate. Finally, when we start CEST from plenty of random initial points, resulting eigenvalues may touch the extreme eigenvalue of a tensor with a high probability.

4.1. Basic convergence theory. If CEST terminates finitely, i.e., there exists an iteration c such that $\nabla f(\mathbf{x}_c) = 0$, we immediately know that $f(\mathbf{x}_c)$ is an eigenvalue and $\mathbf{x}_c$ is the corresponding eigenvector by Theorem 3.1. Therefore, in the remainder of this section, we assume that CEST generates an infinite sequence of iterates $\{\mathbf{x}_c\}$.

Since the symmetric tensor $\mathcal{B}$ is positive definite, the merit function $f(\mathbf{x})$ is twice continuously differentiable. Owing to the compactness of the spherical domain of $f(\mathbf{x})$, we obtain the following bounds [12].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 4.1. *There exists a constant $M > 1$ such that*

$$|f(\mathbf{x})| \le M, \quad \|\nabla f(\mathbf{x})\| \le M, \quad \textit{and} \quad \|\nabla^2 f(\mathbf{x})\| \le M \qquad \forall\, \mathbf{x} \in \mathbb{S}^{n-1}.$$

Because the bounded sequence $\{f(\mathbf{x}_c)\}$ decreases monotonously, it converges.

THEOREM 4.2. *Assume that CEST generates an infinite sequence of function values $\{f(\mathbf{x}_c)\}$. Then, there exists a λ_* such that*

$$\lim_{c\to\infty} f(\mathbf{x}_c) = \lambda_*.$$

Next, we establish bounds for damping factors generated by the inexact line search.

LEMMA 4.3. *There exists a constant $\alpha_{\min} > 0$ such that*

$$\alpha_{\min} \le \alpha_c \le 1 \qquad \forall\, c.$$

Proof. Let $0 < \alpha \le \hat{\alpha} \equiv \frac{(2-\eta)C_L}{(2+\eta)MC_U^2}$. Hence, $\alpha C_U M \le \frac{(2-\eta)C_L}{(2+\eta)C_U} < 1$. From (3.12) and Lemma 4.1, we obtain

$$-\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \le \alpha \|\mathbf{p}_c\| \|\nabla f(\mathbf{x}_c)\| \le \alpha C_U \|\nabla f(\mathbf{x}_c)\|^2 \le \alpha C_U M^2 < M$$

and

$$\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2 \le \alpha^2 \|\mathbf{p}_c\|^2 \le \alpha^2 C_U^2 \|\nabla f(\mathbf{x}_c)\|^2.$$

Then we have

$$\begin{aligned}
&2\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + 2M(\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) - \eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c)(1 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) \\
&\quad = (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2M - \eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c))(\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) \\
&\quad < (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2+\eta) M \alpha^2 C_U^2 \|\nabla f(\mathbf{x}_c)\|^2 \\
&\quad \le (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2-\eta) C_L \alpha \|\nabla f(\mathbf{x}_c)\|^2 \\
&\quad \le 0, \qquad (4.1)
\end{aligned}$$

where the last inequality follows from (3.12).

From the mean value theorem, Lemmas 4.1 and 3.2, and the equality (3.16), we have

$$\begin{aligned}
f(\mathbf{x}_{c+1}(\alpha)) - f(\mathbf{x}_c) &\le \nabla f(\mathbf{x}_c)^\top (\mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c) + \frac{1}{2} M \|\mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c\|^2 \\
&= \frac{2\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + 2M(\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2)}{1 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2} \\
&< \frac{\eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c)(1 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2)}{1 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2} \\
&= \eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c),
\end{aligned}$$

where the last inequality is deduced from (4.1). Due to the rule of the inexact search, the damping factor α_c satisfies $1 \ge \alpha_c \ge \beta \hat{\alpha} \equiv \alpha_{\min}$. ∎

The next theorem proves that every accumulation point of iterates $\{\mathbf{x}_c\}$ is a first-order stationary point.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 4.4. *Suppose that CEST generates an infinite sequence of iterates* $\{\mathbf{x}_c\}$. *Then,*

$$\lim_{c\to\infty} \|\nabla f(\mathbf{x}_c)\| = 0.$$

Proof. From (3.19) and (3.12), we get

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geq -\eta\alpha_c \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \geq \eta\alpha_c C_L \|\nabla f(\mathbf{x}_c)\|^2. \tag{4.2}$$

From Lemmas 4.1 and 4.3, we have

$$\begin{aligned} 2M \geq f(\mathbf{x}_1) - \lambda_* &= \sum_{c=1}^{\infty} [f(\mathbf{x}_c) - f(\mathbf{x}_{c+1})] \\ &\geq \sum_{c=1}^{\infty} \eta\alpha_c C_L \|\nabla f(\mathbf{x}_c)\|^2 \\ &\geq \sum_{c=1}^{\infty} \eta\alpha_{\min} C_L \|\nabla f(\mathbf{x}_c)\|^2. \end{aligned}$$

That is to say,

$$\sum_{c=1}^{\infty} \|\nabla f(\mathbf{x}_c)\|^2 \leq \frac{2M}{\eta\alpha_{\min} C_L} < +\infty.$$

Hence, this theorem is valid. ∎

4.2. Convergence of the sequence of iterates. The Łojasiewicz inequality was discovered by Łojasiewicz [44] for real-analytic functions in 1963. Absil, Mahony, and Andrews [1] proved that, if the objective function satisfies the Łojasiewicz inequality, iterates generated by a large class of optimization methods converge to a unique limit point without additional assumptions. Whereafter, Bolte, Daniilidis, and Lewis [6] extended the Łojasiewicz inequality to nonsmooth functions. Recently, the Łojasiewicz inequality was widely used to analyze proximal algorithms for nonconvex and nonsmooth optimization [4, 59].

The merit function $f(\mathbf{x}) = \frac{\mathcal{T}\mathbf{x}^k}{\mathcal{B}\mathbf{x}^k}$ is a semialgebraic function since its graph

$$\text{Graph}f = \{(\mathbf{x}, \lambda) : \mathcal{T}\mathbf{x}^k - \lambda\mathcal{B}\mathbf{x}^k = 0\}$$

is a semialgebraic set. Therefore, $f(\mathbf{x})$ satisfies the following Łojasiewicz inequality [1, 6].

THEOREM 4.5 (the Łojasiewicz inequality). *Let* $f(\mathbf{x})$ *be a real semialgebraic function with a closed domain* $\text{dom}f$, *and assume that* $f|_{\text{dom}f}$ *is continuous. Let* $\mathbf{x}_* \in \text{dom}f$. *Then, there exists an exponent* $\theta \in [0, 1)$ *and a positive constant* C_K *such that*

$$|f(\mathbf{x}) - f(\mathbf{x}_*)|^\theta \leq C_K \|\nabla f(\mathbf{x})\| \tag{4.3}$$

in some neighborhood $\mathscr{U}$ *of* $\mathbf{x}_*$.

Next, we will prove that the infinite sequence of iterates $\{\mathbf{x}_c\}$ converges to a unique accumulation point.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 4.6. *Suppose that CEST generates an infinite sequence of iterates* $\{\mathbf{x}_c\}$. *Then, there exists a single point* $\mathbf{x}_* \in \mathbb{S}^{n-1}$ *such that*

$$\lim_{c\to\infty} \mathbf{x}_c = \mathbf{x}_*.$$

Hence, the total sequence $\{\mathbf{x}_c\}$ *converges to a first-order stationary point* $\mathbf{x}_*$.

Proof. Since $f(\mathbf{x})$ in (3.2) satisfies the Łojasiewicz inequality, based on Theorem 3.2 in [1], we only need to prove that the sequence of iterates $\{\mathbf{x}_c\}$ satisfies the primary descent condition

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geq C_P \|\nabla f(\mathbf{x}_c)\| \|\mathbf{x}_c - \mathbf{x}_{c+1}\| \tag{4.4}$$

and the complementary descent condition

$$[f(\mathbf{x}_{c+1}) = f(\mathbf{x}_c)] \quad \Rightarrow \quad [\mathbf{x}_{c+1} = \mathbf{x}_c], \tag{4.5}$$

where C_P is a positive constant.

By combining (3.18) and (3.12), we have

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \leq 2\left(\|\alpha_c \mathbf{p}_c\|^2 - (\alpha_c \mathbf{x}_c^\top \mathbf{p}_c)^2\right)^{\frac{1}{2}} \leq 2\alpha_c \|\mathbf{p}_c\| \leq 2C_U \alpha_c \|\nabla f(\mathbf{x}_c)\|.$$

Then, from (4.2), we get

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geq \eta C_L \alpha_c \|\nabla f(\mathbf{x}_c)\|^2 \geq \frac{\eta C_L}{2C_U} \|\nabla f(\mathbf{x}_c)\| \|\mathbf{x}_c - \mathbf{x}_{c+1}\|.$$

Hence, (4.4) holds with $C_P = \frac{\eta C_L}{2C_U}$.

For the complementary descent condition (4.5), we consider its contraposition. If $\mathbf{x}_{c+1} \neq \mathbf{x}_c$, we have $\|\nabla f(\mathbf{x})\| \neq 0$. Otherwise, CEST terminates finitely. By (4.2) and Lemma 4.3, we have

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geq \eta C_L \alpha_{\min} \|\nabla f(\mathbf{x}_c)\|^2 > 0.$$

Hence $f(\mathbf{x}_{c+1}) \neq f(\mathbf{x}_c)$. The proof is complete. ∎

Next, we estimate the convergence rate of CEST. The following lemma is useful.

LEMMA 4.7. *There exists a positive constant* C_m *such that*

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \geq C_m \|\nabla f(\mathbf{x}_c)\|. \tag{4.6}$$

Proof. Let $\langle \mathbf{a}, \mathbf{b} \rangle$ be the angle between nonzero vectors $\mathbf{a}$ and $\mathbf{b}$, i.e.,

$$\langle \mathbf{a}, \mathbf{b} \rangle \equiv \arccos \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \in [0, \pi].$$

In fact, $\langle \cdot, \cdot \rangle$ is a metric in a unit sphere and satisfies the triangle inequality

$$\langle \mathbf{a}, \mathbf{b} \rangle \leq \langle \mathbf{a}, \mathbf{c} \rangle + \langle \mathbf{c}, \mathbf{b} \rangle$$

for all nonzero vectors $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$.

From the triangle inequality, we get

$$\langle \mathbf{x}_c, -\nabla f(\mathbf{x}_c) \rangle - \langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c \rangle \leq \langle \mathbf{x}_c, \mathbf{p}_c \rangle \leq \langle \mathbf{x}_c, -\nabla f(\mathbf{x}_c) \rangle + \langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c \rangle.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Owing to (3.16), we know $\langle \mathbf{x}_c, -\nabla f(\mathbf{x}_c)\rangle = \frac{\pi}{2}$. Then we deduce that

$$\frac{\pi}{2} - \langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c\rangle \leq \langle \mathbf{x}_c, \mathbf{p}_c\rangle \leq \frac{\pi}{2} + \langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c\rangle.$$

Hence, we have

$$\sin\langle \mathbf{x}_c, \mathbf{p}_c\rangle \geq \sin\Big(\frac{\pi}{2} - \langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c\rangle\Big) = \cos\langle -\nabla f(\mathbf{x}_c), \mathbf{p}_c\rangle = \frac{-\mathbf{p}_c^\top \nabla f(\mathbf{x}_c)}{\|\mathbf{p}_c\|\|\nabla f(\mathbf{x}_c)\|} \geq \frac{C_L}{C_U},$$

where the last inequality holds because of (3.12). Recalling (3.18) and $\mathbf{x}_c \in \mathbb{S}^{n-1}$, we obtain

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| = 2\left(\frac{\|\alpha_c \mathbf{p}_c\|^2(1-\cos^2\langle \mathbf{x}_c, \alpha_c\mathbf{p}_c\rangle)}{1+\|\alpha_c\mathbf{p}_c\|^2(1-\cos^2\langle \mathbf{x}_c, \alpha_c\mathbf{p}_c\rangle)}\right)^{\frac{1}{2}} = \frac{2\alpha_c\|\mathbf{p}_c\|\sin\langle \mathbf{x}_c, \alpha_c\mathbf{p}_c\rangle}{\sqrt{1+\alpha_c^2\|\mathbf{p}_c\|^2\sin^2\langle \mathbf{x}_c, \alpha_c\mathbf{p}_c\rangle}}.$$

Since $\alpha_{\min} \leq \alpha_c \leq 1$ and $\|\mathbf{p}_c\| \leq C_U\|\nabla f(\mathbf{x}_c)\| \leq C_U M$, we obtain that

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \geq \frac{2\alpha_{\min}C_L C_U^{-1}}{\sqrt{1+C_U^2M^2}}\|\mathbf{p}_c\| \geq \frac{2\alpha_{\min}C_L}{C_U(1+C_UM)}\|\mathbf{p}_c\|.$$

From (3.12), we have $\|\mathbf{p}_c\|\|\nabla f(\mathbf{x}_c)\| \geq -\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \geq C_L\|\nabla f(\mathbf{x}_c)\|^2$. Hence, $\|\mathbf{p}_c\| \geq C_L\|\nabla f(\mathbf{x}_c)\|$. Therefore, this lemma is valid by taking $C_m \equiv \frac{2\alpha_{\min}C_L^2}{C_U(1+C_UM)}$. ∎

THEOREM 4.8. *Suppose that $\mathbf{x}_*$ is the stationary point of an infinite sequence of iterates $\{\mathbf{x}_c\}$ generated by CEST. Then, we have the following estimations.*

- *If $\theta \in (0, \frac{1}{2}]$, there exists a $\gamma > 0$ and $\varrho \in (0,1)$ such that*

$$\|\mathbf{x}_c - \mathbf{x}_*\| \leq \gamma\varrho^c.$$

- *If $\theta \in (\frac{1}{2}, 1)$, there exists a $\gamma > 0$ such that*

$$\|\mathbf{x}_c - \mathbf{x}_*\| \leq \gamma c^{-\frac{1-\theta}{2\theta-1}}.$$

Proof. Because of the validation of Lemma 4.7, the proof of this theorem is similar to [3, Theorem 2] and [12, Theorem 7]. ∎

Liu and Nocedal [42, Theorem 7.1] proved that L-BFGS converges linearly if the level set of $f(\mathbf{x})$ is convex and the second-order sufficient condition at $\mathbf{x}_*$ holds. We remark here that, if the second-order sufficient condition holds, the exponent θ is equal to $\frac{1}{2}$ in the Łojasiewicz inequality (4.3). According to Theorem 4.8, the infinite sequence of iterates $\{\mathbf{x}_c\}$ has a linear convergence rate, if θ is equal to $\frac{1}{2}$ in the Łojasiewicz inequality.

4.3. Probability of obtaining the extreme eigenvalue. For the target of computing the smallest eigenvalue of a large scale sparse tensor arising from a uniform hypergraph, we start CEST from plenty of random initial points. Then, we regard the resulting smallest merit function value as the smallest eigenvalue of this tensor. The following theorem reveals the successful probability of this strategy.

THEOREM 4.9. *Suppose that we start CEST from N initial points which are sampled from $\mathbb{S}^{n-1}$ uniformly and regard the resulting smallest merit function value as the*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

smallest eigenvalue. Then, there exists a constant $\varsigma \in (0,1]$ *such that the probability of obtaining the smallest eigenvalue is*

$$1-(1-\varsigma)^N. \tag{4.7}$$

Therefore, if the number of samples N *is large enough, we obtain the smallest eigenvalue with a high probability.*

Proof. Suppose that $\mathbf{x}^* \in \mathbb{S}^{n-1}$ is an eigenvector corresponding to the smallest eigenvalue and $\mathscr{U}$ is a neighborhood of $\mathbf{x}^*$ as defined in Theorem 4.5. By the proof of [1, Theorem 3.2], there exists a constant $\rho > 0$ such that the total sequence $\{\mathbf{x}_c\}$ would converge to $\mathbf{x}^*$ if the initial iterate $\mathbf{x}_1$ belongs to $\mathscr{V}(\mathbf{x}^*) \equiv \{\mathbf{x} \in \mathbb{S}^{n-1} : \|\mathbf{x}-\mathbf{x}^*\| < \rho\} \subseteq \mathscr{U}$. Next, we estimate the probability of the event that $\mathbf{x}_1 \in \mathscr{V}(\mathbf{x}^*)$, where $\mathbf{x}_1$ is sampled from $\mathbb{S}^{n-1}$ uniformly.

Let S and A be hypervolumes of $(n-1)$-dimensional solids $\mathbb{S}^{n-1}$ and $\mathscr{V}(\mathbf{x}^*)$, respectively.[2] (That is to say, the "area" of the surface of $\mathbb{S}^{n-1}$ in $\mathbb{R}^n$ is S and the area of the surface of $\mathscr{V}(\mathbf{x}^*) \subseteq \mathbb{S}^{n-1}$ in $\mathbb{R}^n$ is A. Hence, $A \le S$.) Then, S and A are positive. As a geometric probability model, the probability of getting one random initial point $\mathbf{x}_1 \in \mathscr{V}(\mathbf{x}^*)$ is

$$\varsigma \equiv \frac{A}{S} > 0.$$

In fact, once $\{\mathbf{x}_c\} \cap \mathscr{V}(\mathbf{x}^*) \neq \emptyset$, we could obtain the smallest eigenvalue.

We consider the complementary event. The probability of the event that CEST starts from one random initial point and does not reach $\mathbf{x}^*$ is $(1-\varsigma)$. Owing to the independence of N random initial points, the probability of the joint event that CEST starts from N random initial points and does not touch $\mathbf{x}^*$ is $(1-\varsigma)^N$. Conversely, we get the eigenvector $\mathbf{x}^*$ corresponding to the smallest eigenvalue. Hence, when starting from N initial points generated by a uniform sample on $\mathbb{S}^{n-1}$, the probability of obtaining the smallest eigenvalue is $1-(1-\varsigma)^N$. ∎

If we want to calculate the largest eigenvalue of a tensor $\mathcal{T}$, we only need to replace the merit function $f(\mathbf{x})$ in (3.2) with

$$\widehat{f}(\mathbf{x}) = -\frac{\mathcal{T}\mathbf{x}^k}{\mathcal{B}\mathbf{x}^k}.$$

All of the theorems for the largest eigenvalue of a tensor can be proved in a similar way.

5. Computational methods on sparse tensors arising from a hypergraph. The adjacency tensor $\mathcal{A}$, the Laplacian tensor $\mathcal{L}$, and the signless Laplacian tensor $\mathcal{Q}$ of a uniform hypergraph are usually sparse. For instance, $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$ of the 4-uniform sunflower illustrated in Figure 1 only contain 0.72%, 0.76%, and 0.76% nonzero elements, respectively. Hence, it is important to take advantage of the sparsity in tensors $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$. Now, we introduce a fast numerical approach based on MATLAB.

How to store a uniform hypergraph? Let $G = (V, E)$ be a k-graph with $|V| = n$ vertices and $|E| = m$ edges. We store G as an m-by-k matrix G_r whose

[2] The hypervolume of the $(n-1)$-dimensional unit sphere is $S = \frac{2\pi^{n/2}}{\Gamma(n/2)}$, where $\Gamma(\cdot)$ is the gamma function.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

rows are composed of the indices of vertices from corresponding edges of G. Here, the order of elements in each row of G_r is unimportant in the sense that we could permute them.

For instance, we consider the 4-uniform sunflower shown in Figure 1. The edge-vertex incidence matrix of this sunflower is a 3-by-10 sparse matrix

$$G_i = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$
$$\begin{array}{cccccccccc} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \quad \leftarrow \text{(the indices of vertices)}$$

From the perspective of scientific computing, we prefer to store the incidence matrix of the sunflower in a compact form

$$G_r = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 \\ 1 & 8 & 9 & 10 \end{bmatrix} \in \mathbb{R}^{3\times 4}.$$

Obviously, the number of columns of the matrix G_r is less than that of the original incidence matrix, since usually $k \ll n$.

How to compute products $\mathcal{T}\mathbf{x}^k$ and $\mathcal{T}\mathbf{x}^{k-1}$ when $\mathcal{T} = \mathcal{A}, \mathcal{L}$, and $\mathcal{Q}$? Suppose that the matrix G_r representing a uniform hypergraph and a vector $\mathbf{x} \in \mathbb{R}^n$ are available. Since $\mathcal{L} = \mathcal{D} - \mathcal{A}$ and $\mathcal{Q} = \mathcal{D} + \mathcal{A}$, it is sufficient to study the degree tensor $\mathcal{D}$ and the adjacency tensor $\mathcal{A}$.

We first consider the degree tensor $\mathcal{D}$. It is a diagonal tensor whose ith diagonal element is the degree $d(i)$ of a vertex $i \in V$. If the edge-vertex incidence matrix G_i is available, the degree $d(i)$ could be obtained by summarizing the ith column of G_i. We use MATLAB commands

```
Gi=sparse(repmat((1:m)',k,1),Gr,1,m,n);
d =full(sum(Gi,1)');
```

to generate the sparse matrix G_i and then produce the degree vector $\mathbf{d} \equiv [d(i)] \in \mathbb{R}^n$. The degree vector $\mathbf{d}$ could be saved from the start since it is fixed for a given hypergraph. For any vector $\mathbf{x} \in \mathbb{R}^n$, the computations of

$$\mathcal{D}\mathbf{x}^{k-1} = \mathbf{d} * (\mathbf{x}^{[k-1]}) \qquad \text{and} \qquad \mathcal{D}\mathbf{x}^k = \mathbf{d}^\top(\mathbf{x}^{[k]})$$

are straightforward, where "$*$" denotes the componentwise Hadamard product.

Second, we focus on the adjacency tensor $\mathcal{A}$. We construct a matrix $X_{mat} = [x_{(G_r)_{\ell j}}]$ which has the same size as G_r. Assume that the (ℓ, j)-th element of G_r is i. Then, the (ℓ, j)-th element of X_{mat} is defined as x_i. From this matrix, we rewrite the product $\mathcal{A}\mathbf{x}^k$ as

$$\mathcal{A}\mathbf{x}^k = k \sum_{\ell=1}^{m} \prod_{j=1}^{k} (X_{mat})_{\ell j}. \tag{5.1}$$

To compute the vector $\mathcal{A}\mathbf{x}^{k-1}$, we use the following representation

$$(\mathcal{A}\mathbf{x}^{k-1})_i = \sum_{j=1}^{k} \sum_{\ell=1}^{m} \left(\delta(i, (G_r)_{\ell j}) \prod_{\substack{s=1 \\ s\neq j}}^{k} (X_{mat})_{\ell s} \right) \qquad \text{for } \ i = 1, \ldots, n.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

For each $j = 1, \ldots, k$, we construct a sparse matrix $M_j = [\delta(i, (G_r)_{\ell j})] \in \mathbb{R}^{n \times m}$ and a column vector $\mathbf{y}_j = [\prod_{s \neq j} (X_{mat})_{\ell s}] \in \mathbb{R}^m$ by MATLAB commands

```
Mj=sparse(Gr(:,j),(1:m)',1,n,m);
yj=prod(Xmat(:,[1:j-1,j+1:k]),2);
```

respectively. Then, the vector

$$\mathcal{A}\mathbf{x}^{k-1} = \sum_{j=1}^{k} M_j \mathbf{y}_j$$

could be computed by a simple loop.

The computational costs for computing products of tensors $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$ with any vector $\mathbf{x}$ are about mk^2, $mk^2 + nk$, and $mk^2 + nk$ multiplications, respectively. Since $mk^2 < mk^2 + nk \leq 2mk^2$, the computational cost of the product of a vector and a large scale sparse tensor related to a hypergraph is cheap. Additionally, the numerical approaches proposed here could easily be extended to parallel computing.

6. Numerical experiments. The CEST algorithm is implemented in MATLAB and uses the following parameters

$$L = 5, \quad \eta = 0.01, \quad \text{and} \quad \beta = 0.5.$$

We terminate CEST if

$$\|\nabla f(\mathbf{x}_c)\|_\infty < 10^{-6} \tag{6.1}$$

or

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\|_\infty < 10^{-8} \quad \text{and} \quad \frac{|f(\mathbf{x}_{c+1}) - f(\mathbf{x}_c)|}{1 + |f(\mathbf{x}_c)|} < 10^{-16}. \tag{6.2}$$

If the number of iterations reaches 5000, we also stop.

We compare the following four algorithms in this section.

- An adaptive shifted power method [35, 36] (Power M.). In Tensor Toolbox 2.6,[3] it is implemented as `eig_sshopm` and `eig_geap` for Z- and H-eigenvalues of symmetric tensors, respectively.
- Han's unconstrained optimization approach (Han's UOA) [26]. We solve the optimization model by `fminunc` in MATLAB with settings `GradObj:on, LargeScale:off, TolX:1.e-8, TolFun:1.e-16, MaxIter:5000,Display:off`. Since iterates generated by Han's UOA are not restricted on the unit sphere $\mathbb{S}^{n-1}$, the tolerance parameters are different from other algorithms.
- CESTde: It is a raw version of the CEST method which means that the fast computational skills provided in section 5 are omitted.
- CEST: The novel method proposed and analyzed in this paper.

For tensors arising from an even-uniform hypergraph, each algorithm starts from one hundred random initial points sampled from a unit sphere $\mathbb{S}^{n-1}$. To obtain an initial point, we first generate an n-dimensional vector whose elements have a standard Gaussian distribution. Then, we normalize the vector and get an initial point on the unit sphere. In this way, we sample one hundred points independently. Hence, they have a uniform distribution on the unit sphere. Then, we obtain one hundred

[3] See http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

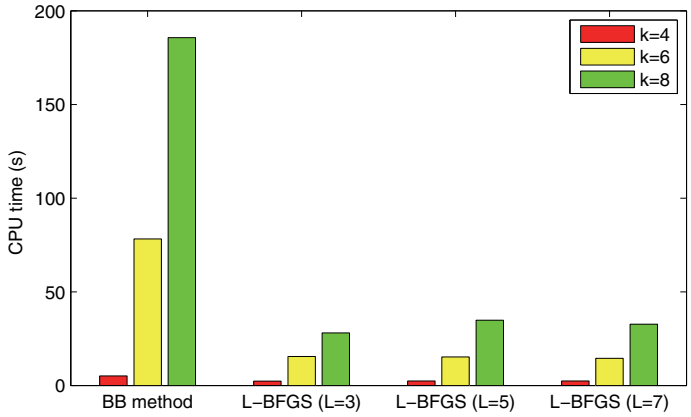

FIG. 2. *CPU time for L-BFGS with different parameters L.*

estimated eigenvalues $\lambda_1, \ldots, \lambda_{100}$. If the extreme eigenvalue λ^* of that tensor is available, we count the accuracy rate of this algorithm as

$$\text{Accu.} \equiv \left| \left\{ i \,:\, \frac{|\lambda_i - \lambda^*|}{1 + |\lambda^*|} \le 10^{-8} \right\} \right| \times 1\%. \tag{6.3}$$

After using the global strategy in section 4.3, we regard the best one as the estimated extreme eigenvalue. In the following experiments, we report the total CPU time for one hundred runs to find the best eigenvalue in each case.

6.1. Small hypergraphs. First, we investigate some extreme eigenvalues of symmetric tensors corresponding to small uniform hypergraphs.

Squid. A squid $G_S^k = (V, E)$ is a k-uniform hypergraph which has (k^2-k+1) vertices and k edges: legs $\{i_{1,1}, \ldots, i_{1,k}\}, \ldots, \{i_{k-1,1}, \ldots, i_{k-1,k}\}$ and a head $\{i_{1,1}, \ldots, *i_{k-1,1}, i_k\}$. When k is even, G_S^k is obviously connected and odd bipartite. Hence, we have $\lambda_{\min}^H(\mathcal{A}(G_S^k)) = -\lambda_{\max}^H(\mathcal{A}(G_S^k))$ because of Theorem 2.4(iv). Since the adjacency tensor $\mathcal{A}(G_S^k)$ is nonnegative and weakly irreducible, its largest H-eigenvalue $\lambda_{\max}^H(\mathcal{A}(G_S^k))$ could be computed by the Ng–Qi–Zhou algorithm [46]. For the smallest H-eigenvalue of $\mathcal{A}(G_S^k)$, we perform the following tests.

It is suggested by Nocedal that L-BFGS performs well when $3 \le L \le 7$. Hence, we compare L-BFGS with $L = 3, 5, 7$ and the Barzilai–Borwein method ($L = 0$). The parameter γ_c is chosen from γ_c^{BB1}, γ_c^{BB2}, and γ_c^{Dai} randomly. For k-uniform squids with $k = 4, 6, 8$, we compute the smallest H-eigenvalues of their adjacency tensors. The total CPU times for one hundred runs are illustrated in Figure 2. Obviously, L-BFGS is about five times faster than the Barzilai–Borwein method. By following Nocedal's setting,[4] we prefer to set $L = 5$ in CEST.

Next, we consider the 4-uniform squid G_S^4 illustrated in Figure 3. For reference, we get $\lambda_{\max}^H(\mathcal{A}(G_S^k)) = 1.3320$ by the Ng–Qi–Zhou algorithm. Then, we compare four kinds of algorithms: Power M., Han's UOA, CESTde, and CEST. Results are shown in Table 1. Obviously, all algorithms find the smallest H-eigenvalue of the adjacency tensor $\mathcal{A}(G_S^4)$ with probability 1. Compared with Power M., Han's UOA and CESTde save 78% and 63% CPU time, respectively. When the sparse structure

[4] See http://users.iems.northwestern.edu/~nocedal/lbfgs.html.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

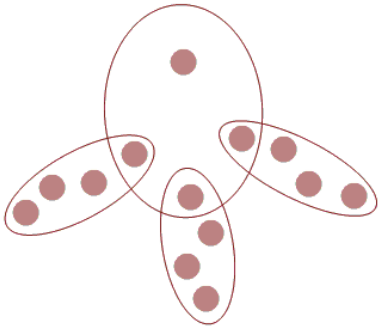

FIG. 3. *A 4-uniform squid* G_S^4.

TABLE 1
Results for finding the smallest H-eigenvalue of $\mathcal{A}(G_S^4)$.

Algorithms	$\lambda_{\min}^H(\mathcal{A}(G_S^4))$	Time(s)	Accu.
Power M.	−1.3320	97.20	100%
Han's UOA	−1.3320	21.20	100%
CESTde	−1.3320	35.72	100%
CEST	−1.3320	2.43	100%

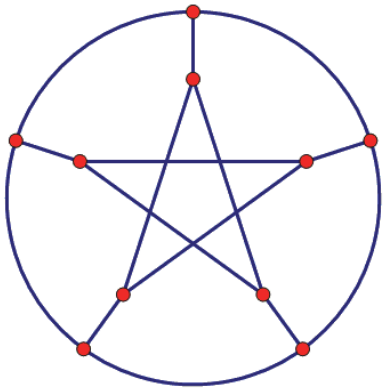

FIG. 4. *The Petersen graph* G_P.

of the adjacency tensor $\mathcal{A}(G_S^4)$ is explored, CEST is forty times faster than the power method.

Blowing up the Petersen graph. Figure 4 illustrates an ordinary graph G_P: the Petersen graph. It is nonbipartite and the smallest eigenvalue of its signless Laplacian matrix is one. We consider the $2k$-uniform hypergraph $G_P^{2k,k}$ which is generated by blowing up each vertex of G_P to a k-set. Hence, $G_P^{2k,k}$ contains $10k$ vertices and 15 edges. From Theorem 2.5, we know that the smallest H-eigenvalue of the signless Laplacian tensor $\mathcal{Q}(G_P^{2k,k})$ is exactly one.

Table 2 reports results of four sorts of algorithms for the 4-uniform hypergraph $G_P^{4,2}$. Here, Han's UOA misses the smallest H-eigenvalue of the signless Laplacian tensor. Power M., CESTde, and CEST find the true solution with a high probability. When compared with Power M., CESTde saves more than 88% CPU time. Moreover,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 2
Results for finding the smallest H-eigenvalue of $\mathcal{Q}(G_P^{4,2})$. (*) means a failure.

Algorithms	$\lambda_{\min}^H(\mathcal{Q}(G_P^{4,2}))$	Time(s)	Accu.
Power M.	1.0000	657.44	95%
Han's UOA	$1.1877^{(*)}$	93.09	
CESTde	1.0000	70.43	100%
CEST	1.0000	3.82	100%

TABLE 3
Accuracy rate of CEST for finding the smallest H-eigenvalues of $\mathcal{Q}(G_P^{2k,k})$.

$2k$	2	4	6	8	10	12	14	16	18	20
Accu. (%)	100	100	100	100	99	98	86	57	20	4

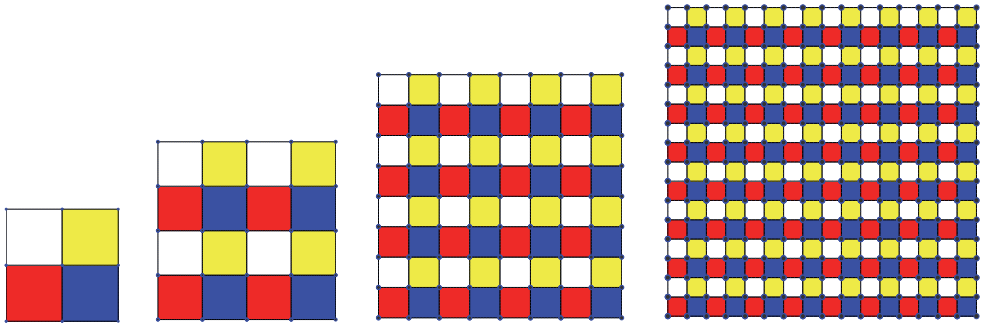

FIG. 5. *Some 4-uniform grid hypergraphs.*

CEST exploiting the sparsity of $\mathcal{Q}(G_P^{4,2})$ improves CESTde greatly and saves about 99% CPU time.

For $2k$-uniform hypergraph $G_P^{2k,k}$ with $k = 1, \ldots, 10$, we apply CEST to the computation of the smallest H-eigenvalues of their signless Laplacian tensors. Detailed results are shown in Table 3. For each case, CEST finds the smallest H-eigenvalue of the signless Laplacian tensor in at most one minute. With the increase of k, the accuracy rate decreases.

Grid hypergraphs. The grid G_G^s is a 4-uniform hypergraph generated by subdividing a square. If the subdividing order $s = 0$, the grid G_G^0 is a square with 4 vertices and only one edge. When the subdividing order $s \geq 1$, we subdivide each edge of G_G^{s-1} into four edges. Hence, a grid G_G^s has $(2^s + 1)^2$ vertices and 4^s edges. The 4-uniform grid G_G^s with $s = 1, 2, 3, 4$ is illustrated in Figure 5.

We study the largest H-eigenvalue of the Laplacian tensor of the 4-uniform grid G_G^2 in Figure 5(b). Obviously, the grid G_G^2 is connected and odd bipartite. Hence, we have $\lambda_{\max}^H(\mathcal{L}(G_G^2)) = \lambda_{\max}^H(\mathcal{Q}(G_G^2))$ from Theorem 2.4(ii). By using the Ng–Qi–Zhou algorithm, we calculate $\lambda_{\max}^H(\mathcal{Q}(G_G^2)) = 6.5754$ for reference. Table 4 shows the performance of four kinds of algorithms: Power M., Han's UOA, CESTde, and CEST. All of them find the largest H-eigenvalue of $\mathcal{L}(G_G^2)$ with probability one. Compared with Power M., Han's UOA and CESTde save about 75% and 70% CPU time, respectively. CEST is about fifty times faster than the power method.

In Table 5, we show the performance of CEST for computing the largest H-eigenvalues of the Laplacian tensors of grids illustrated in Figure 5.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 4
Results for finding the largest H-eigenvalue of $\mathcal{L}(G_G^2)$.

Algorithms	$\lambda^H_{\max}(\mathcal{L}(G_G^2))$	Time(s)	Accu.
Power M.	6.5754	142.51	100%
Han's UOA	6.5754	35.07	100%
CESTde	6.5754	43.35	100%
CEST	6.5754	2.43	100%

TABLE 5
Performance of CEST for finding the largest H-eigenvalues of Laplacian tensors $\mathcal{L}(G_G^s)$.

s	n	m	$\lambda^H_{\max}(\mathcal{L}(G_G^s))$	Iter.	Time(s)	Accu.
1	9	4	4.6344	2444	1.39	100%
2	25	16	6.5754	4738	2.43	100%
3	81	64	7.5293	12624	6.44	98%
4	289	256	7.8648	34558	26.08	65%

6.2. Large hypergraphs. Finally, we consider two large scale even-uniform hypergraphs.

Sunflower. A k-uniform sunflower $G_S = (V, E)$ with a maximum degree Δ has $n = (k-1)\Delta + 1$ vertices and Δ edges, where $V = V_0 \cup V_1 \cup \cdots \cup V_\Delta$, $|V_0| = 1$, $|V_i| = k-1$ for $i = 1, \ldots, \Delta$, and $E = \{V_0 \cup V_i \,|\, i = 1, \ldots, \Delta\}$. Figure 1 is a 4-uniform sunflower with $\Delta = 3$.

Hu, Qi, and Xie [29] argued in the following theorem that the largest H-eigenvalue of the Laplacian tensor of an even-uniform sunflower has a closed-form solution.

THEOREM 6.1 (Theorems 3.2 and 3.4 of [29]). *Let G be a k-graph with $k \geq 4$ being even. Then*

$$\lambda^H_{\max}(\mathcal{L}) \geq \lambda^*_H,$$

where λ^*_H is the unique real root of the equation $(1-\lambda)^{k-1}(\lambda - \Delta) + \Delta = 0$ in the interval $(\Delta, \Delta + 1)$. The equality holds if and only if G is a sunflower.

We aim to apply CEST to the computation of the largest H-eigenvalues of Laplacian tensors of even-uniform sunflowers. For $k = 4$ and 6, we consider sunflowers with maximum degrees from ten to one million. Since we deal with large scale tensors, we slightly enlarge tolerance parameters in (6.1) and (6.2) by multiplying them by $\sqrt{n}$. To show the accuracy of the estimated H-eigenvalue $\lambda^H_{\max}(\mathcal{L}(G_S))$, we calculate the relative error

$$\mathrm{RE} = \frac{|\lambda^H_{\max}(\mathcal{L}(G_S)) - \lambda^*_H|}{\lambda^*_H},$$

where λ^*_H is defined in Theorem 6.1. Table 6 reports detailed numerical results. Obviously, the largest H-eigenvalues of Laplacian tensors returned by CEST have a high accuracy. Relative errors are of the magnitude $\mathcal{O}(10^{-10})$. The CPU time costed by CEST in each test does not exceed 80 minutes.

Icosahedron. An icosahedron has twelve vertices and twenty faces. The subdivision of an icosahedron could be used to approximate a unit sphere. The s-order subdivision of an icosahedron has (20×4^s) faces and each face is a triangle. We regard three vertices of the triangle as well as its center as an edge of a 4-graph G_I^s. Then, the 4-graph G_I^s must be connected and odd bipartite. See Figure 6.

According to Theorem 2.4(v), we have $\lambda^Z_{\max}(\mathcal{L}(G_I^s)) = \lambda^Z_{\max}(\mathcal{Q}(G_I^s))$, although they are unknown. Experiment results are reported in Table 7. It can be seen that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 6
Performance of CEST for finding the largest H-eigenvalues of $\mathcal{L}(G_S)$.

k	n	$\lambda^H_{\max}(\mathcal{L}(G_S))$	RE	Iter.	Time(s)	Accu.
4	31	10.0137	5.3218×10^{-16}	4284	2.39	100%
	301	100.0001	7.3186×10^{-14}	4413	3.73	42%
	3,001	1,000.0000	1.2917×10^{-10}	1291	4.84	100%
	30,001	10,000.0000	5.9652×10^{-12}	1280	38.14	100%
	300,001	100,000.0000	9.6043×10^{-15}	1254	512.04	100%
	3,000,001	1,000,000.0000	0	1054	4612.28	100%
6	51	10.0002	2.4831×10^{-12}	4768	3.34	8%
	501	100.0000	2.4076×10^{-10}	1109	1.47	98%
	5,001	1,000.0000	3.2185×10^{-13}	1020	5.85	100%
	50,001	10,000.0000	5.7667×10^{-12}	927	44.62	100%
	500,001	100,000.0000	1.1583×10^{-13}	778	479.52	100%
	5,000,001	1,000,000.0000	2.3283×10^{-16}	709	4679.30	100%

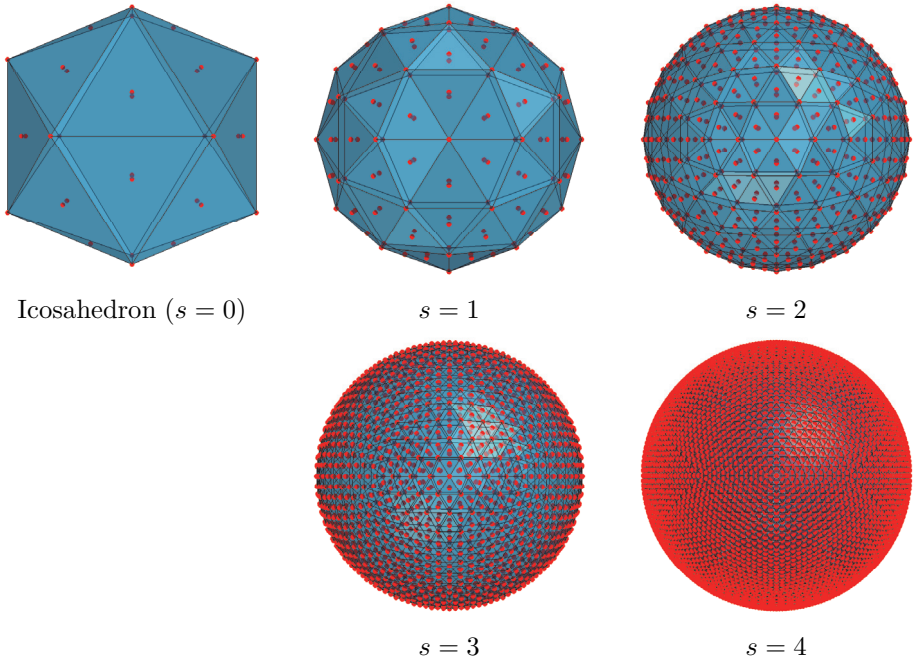

FIG. 6. *4-uniform hypergraphs: subdivision of an icosahedron.*

CEST could compute the largest Z-eigenvalues of both Laplacian tensors and signless Laplacian tensors of hypergraphs G_I^s with dimensions up to almost two million. In each case of our experiment, CEST costs at most twenty-one minutes.

Additionally, for 4-graphs G_I^s generated by subdividing an icosahedron, the following equality seems to hold:

$$\lambda^Z_{\max}(\mathcal{L}(G_I^s)) = \lambda^Z_{\max}(\mathcal{Q}(G_I^s)) = \Delta. \tag{6.4}$$

Bu, Fan, and Zhou [7] proved that (6.4) holds for a k-uniform sunflower with $3 \le k \le 2\Delta$. However, it is an open problem whether the equality (6.4) holds for a general connected odd bipartite uniform hypergraph.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 7
Performance of CEST for finding the largest Z-eigenvalues of $\mathcal{L}(G_I^s)$ and $\mathcal{Q}(G_I^s)$.

			$\mathcal{L}(G_I^s)$			$\mathcal{Q}(G_I^s)$		
s	n	m	$\lambda^Z_{\max}$	Iter.	time(s)	$\lambda^Z_{\max}$	Iter.	time(s)
0	32	20	5	1102	0.89	5	1092	0.75
1	122	80	6	1090	1.09	6	1050	0.75
2	482	320	6	1130	1.39	6	1170	1.23
3	1,922	1,280	6	1226	3.15	6	1194	2.95
4	7,682	5,120	6	1270	10.11	6	1244	10.06
5	30,722	20,480	6	1249	36.89	6	1282	35.93
6	122,882	81,920	6	1273	166.05	6	1289	161.02
7	491,522	327,680	6	1300	744.08	6	1327	739.01
8	1,966,082	1,310,720	6	574	1251.36	6	558	1225.87

TABLE 8
Results of optimization methods for finding the largest Z-eigenvalues of $\mathcal{L}(G_I^4)$ and $\mathcal{Q}(G_I^4)$.

	$\mathcal{L}(G_I^4)$				$\mathcal{Q}(G_I^4)$			
Algorithms	$\lambda^Z_{\max}$	Iter.	time(s)	Func.	$\lambda^Z_{\max}$	Iter.	time(s)	Func.
SD	6	518	24.21	4382	6	528	23.32	4357
CG	6	802	23.04	4201	6	808	22.60	4240
QN	6	1439	12.27	2984	6	1435	11.51	2913
CEST	6	1270	10.11	2456	6	1244	10.06	2564

Finally, the unit sphere is indeed a simple Stiefel manifold. By introducing an inner product on its tangent space, the Stiefel manifold becomes a Riemannian manifold [2]. Hence, many optimization algorithms developed in the Riemannian manifold could be applied. Here, we consider three sorts of methods. The first one is the steepest descent (SD) method. Second, we test the nonlinear Polak and Ribiére conjugate gradient (CG) method which is equipped with an inexact Wolfe line search [56, 49]. The last one is the limited memory BFGS quasi-Newton (QN) method with a backtracking line search. Although the objective function $f(\mathbf{x})$ is zero-order homogeneous, the gradient $\nabla f(\mathbf{x})$ in (3.3) would tend to zero if the norm of $\mathbf{x}$ is huge. To avoid premature termination , we normalize all iterates in the above three algorithms. Simply speaking, the Cayley transform is replaced by a projection onto the unit sphere.

To compute the largest Z-eigenvalues of a Laplacian tensor $\mathcal{L}(G_I^4)$ and a signless Laplacian tensor $\mathcal{Q}(G_I^4)$, we run SD, CG, QN, and CEST from one hundred random initial points separately. The total iterations, CPU time, and number of function evaluations are shown in Table 8. In the process of a Wolfe line search, we need to evaluate function and gradient values simultaneously. The backtracking line search is cheap since it only requires function values. Hence, SD and CG used more CPU time. Since the search direction generated by quasi-Newton methods (e.g., L-BFGS) enjoys a good initial step-size one, the backtracking line search works well for L-BFGS. Here, CEST performs slightly better because the Cayley transform is explored.

7. Conclusion. Motivated by recent advances in spectral hypergraph theory, we propose an efficient first-order optimization algorithm CEST for computing extreme H- and Z-eigenvalues of sparse tensors arising from large scale uniform hypergraphs. Due to the algebraic nature of tensors, the Łojasiewicz inequality is applied to analyze the convergence of the sequence of iterates generated by CEST. By using a simple global strategy, it is proved that the extreme eigenvalue of a symmetric tensor could be touched with a high probability.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We establish a fast computational framework for products of a vector and large scale sparse tensors arising from a uniform hypergraph. By using this technique, the storage of a hypergraph is economic and the computational cost of CEST in each iteration is cheap. Numerical experiments show that CEST could deal with uniform hypergraphs with millions of vertices.

Acknowledgment. The authors are grateful to the associate editor and two anonymous referees for helping us improve the original manuscript.

REFERENCES

[1] P. A. Absil, R. Mahony, and B. Andrews, *Convergence of the iterates of descent methods for analytic cost functions*, SIAM J. Optim., 16 (2005), pp. 531–547.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2009.

[3] H. Attouch and J. Bolte, *On the convergence of the proximal algorithm for nonsmooth functions involving analytic features*, Math. Program., 116 (2009), pp. 5–16.

[4] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457.

[5] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[6] J. Bolte, A. Daniilidis, and A. Lewis, *The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, SIAM J. Optim., 17 (2007), pp. 1205–1223.

[7] C. Bu, Y. Fan, and J. Zhou, *Laplacian and signless Laplacian Z-eigenvalues of uniform hypergraphs*, Front. Math. China, 11 (2016), pp. 511–520.

[8] S. R. Bulo and M. Pelillo, *New bounds on the clique number of graphs based on spectral hypergraph theory*, in Learning and Intelligent Optimization, Springer, Berlin, 2009, pp. 45–58.

[9] K.-C. Chang, K. J. Pearson, and T. Zhang, *On eigenvalue problems of real symmetric tensors*, J. Math. Anal. Appl., 350 (2009), pp. 416–422.

[10] K.-C. Chang, K. J. Pearson, and T. Zhang, *Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 806–819.

[11] L. Chen, L. Han, and L. Zhou, *Computing tensor eigenvalues via homotopy methods*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 290–319.

[12] Y. Chen, L. Qi, and Q. Wang, *Computing extreme eigenvalues of large scale Hankel tensors*, J. Sci. Comput., 68 (2016), pp. 716–738.

[13] J. Cooper and A. Dutle, *Spectra of uniform hypergraphs*, Linear Algebra Appl., 436 (2012), pp. 3268–3292.

[14] C.-F. Cui, Y.-H. Dai, and J. Nie, *All real eigenvalues of symmetric tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1582–1601.

[15] Y.-H. Dai, *A positive BB-like stepsize and an extension for symmetric linear systems*, in Workshop on Optimization for Modern Computation, Beijing, China, 2014, http://bicmr.pku.edu.cn/conference/opt-2014/slides/Yuhong-Dai.pdf.

[16] W. Ding, L. Qi, and Y. Wei, *M-tensors and nonsingular M-tensors*, Linear Algebra Appl., 439 (2013), pp. 3264–3278.

[17] W. Ding and Y. Wei, *Generalized tensor eigenvalue problems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1073–1099.

[18] E. Fischer, A. Matsliah, and A. Shapira, *Approximate hypergraph partitioning and applications*, SIAM J. Comput., 39 (2010), pp. 3155–3185.

[19] S. Friedland, S. Gaubert, and L. Han, *Perron-Frobenius theorem for nonnegative multilinear forms and extensions*, Linear Algebra Appl., 438 (2013), pp. 738–749.

[20] S. Friedland, J. Nocedal, and M. L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal., 24 (1987), pp. 634–667.

[21] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, *3-D object retrieval and recognition with hypergraph analysis*, IEEE Trans. Image Process., 21 (2012), pp. 4290–4303.

[22] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. J. Newman, *Random hypergraphs and their applications*, Phys. Rev. E (3), 79 (2009), 066118.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[23] D. Ghoshdastidar and A. Dukkipati, *A provable generalized tensor spectral method for uniform hypergraph partitioning*, in Proceedings of The 32nd International Conference on Machine Learning, International Machine, Learning Society, Princeton, NJ, 2015, pp. 400–409.
[24] D. F. Gleich, L.-H. Lim, and Y. Yu, *Multilinear pagerank*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1507–1541.
[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 2013.
[26] L. Han, *An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors*, Numer. Algebra Control Optim., 3 (2013), pp. 583–599.
[27] C.-L. Hao, C.-F. Cui, and Y.-H. Dai, *A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors*, Numer. Linear Algebra Appl., 22 (2015), pp. 283–298.
[28] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM, 60 (2013), 45.
[29] S. Hu, L. Qi, and J. Xie, *The largest Laplacian and signless Laplacian H-eigenvalues of a uniform hypergraph*, Linear Algebra Appl., 469 (2015), pp. 1–27.
[30] B. Jiang and Y.-H. Dai, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program., 153 (2015), pp. 535–575.
[31] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, *Multilevel hypergraph partitioning: Applications in VLSI domain*, IEEE Trans. Very Large Scale Integration (VLSI) Systems, 7 (1999), pp. 69–79.
[32] E. Kayaaslan, A. Pinar, Ü. Çatalyürek, and C. Aykanat, *Partitioning hypergraphs in scientific computing applications through vertex separators on graphs*, SIAM J. Sci. Comput., 34 (2012), pp. A970–A992.
[33] M. Khan, Y. Fan and Y. Tan, *The H-spectra of a class of generalized power hypergraphs*, Discrete Math., 339 (2016), pp. 1682–1689.
[34] S. Klamt, U.-U. Haus, and F. Theis, *Hypergraphs and cellular networks*, PLOS Computational Biology, 5 (2009), e1000385.
[35] T. G. Kolda and J. R. Mayo, *Shifted power method for computing tensor eigenpairs*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1095–1124.
[36] T. G. Kolda and J. R. Mayo, *An adaptive shifted power method for computing generalized tensor eigenpairs*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1563–1581.
[37] E. V. Konstantinova and V. A. Skorobogatov, *Molecular hypergraphs: The new representation of nonclassical molecular structures with polycentric delocalized bonds*, J. Chem. Inform. Comput. Sci., 35 (1995), pp. 472–478.
[38] E. V. Konstantinova and V. A. Skorobogatov, *Molecular structures of organoelement compounds and their representation as labeled molecular hypergraphs*, J. Struct. Chem., 39 (1998), pp. 268–276.
[39] J. Lei and A. Rinaldo, *Consistency of spectral clustering in stochastic block models*, Ann. Statist., 43 (2015), pp. 215–237.
[40] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, *Context-aware hypergraph construction for robust spectral clustering*, IEEE Trans. Knowl. Data Eng., 26 (2014), pp. 2588–2597.
[41] L.-H. Lim, *Singular values and eigenvalues of tensors: a variational approach*, in 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2005, IEEE, Piscataway, NJ, 2005, pp. 129–132.
[42] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Program., 45 (1989), pp. 503–528.
[43] Y. Liu, G. Zhou, and N. F. Ibrahim, *An always convergent algorithm for the largest eigenvalue of an irreducible nonnegative tensor*, J. Comput. Appl. Math., 235 (2010), pp. 286–292.
[44] S. Łojasiewicz, *Une propriété topologique des sous-ensembles analytiques réels*, in Les Équations aux Dérivées Partielles, LNRS, Paris, 1963, pp. 87–89.
[45] T. Michoel and B. Nachtergaele, *Alignment and integration of complex networks by hypergraph-based spectral clustering*, Phys. Rev. E (3), 86 (2012), 056111.
[46] M. Ng, L. Qi, and G. Zhou, *Finding the largest eigenvalue of a nonnegative tensor*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1090–1099.
[47] Q. Ni and L. Qi, *A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map*, J. Global Optim., 61 (2015), pp. 627–641.
[48] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.
[49] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, Berlin, 2006.
[50] K. J. Pearson and T. Zhang, *On spectral hypergraph theory of the adjacency tensor*, Graphs Combin., 30 (2014), pp. 1233–1248.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[51] L. Qi, *Eigenvalues of a real supersymmetric tensor*, J. Symbolic Comput., 40 (2005), pp. 1302–1324.
[52] L. Qi, *H^+-eigenvalues of Laplacian and signless Laplacian tensors*, Commun. Math. Sci., 12 (2014), pp. 1045–1064.
[53] L. Qi, F. Wang, and Y. Wang, *Z-eigenvalue methods for a global polynomial optimization problem*, Math. Program., 118 (2009), pp. 301–316.
[54] K. Rohe, S. Chatterjee, and B. Yu, *Spectral clustering and the high-dimensional stochastic blockmodel*, Ann. Statist., (2011), pp. 1878–1915.
[55] J.-Y. Shao, H.-Y. Shan, and B.-F. Wu, *Some spectral properties and characterizations of connected odd-bipartite uniform hypergraphs*, Linear Multilinear Algebra, 63 (2015), pp. 2359–2372.
[56] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.
[57] Z. Wen and W. Yin, *A feasible method for optimization with orthogonality constraints*, Math. Program., 142 (2013), pp. 397–434.
[58] J. Xie and L. Qi, *The clique and coclique numbers bounds based on the H-eigenvalues of uniform hypergraphs*, Int. J. Numer. Anal. Model., 12 (2015), pp. 318–327.
[59] Y. Xu and W. Yin, *A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion*, SIAM J. Imaging Sci., 6 (2013), pp. 1758–1789.
[60] J. Yu, D. Tao, and M. Wang, *Adaptive hypergraph learning and its application in image classification*, IEEE Trans. Image Process., 21 (1012), pp. 3262–3272.
[61] L. Zhang, L. Qi, and G. Zhou, *M-tensors and some applications*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 437–452.
[62] G. Zhou, L. Qi, and S.-Y. Wu, *Efficient algorithms for computing the largest eigenvalue of a nonnegative tensor*, Front. Math. China, 8 (2013), pp. 155–168.
[63] G. Zhou, L. Qi, and S.-Y. Wu, *On the largest eigenvalue of a symmetric nonnegative tensor*, Numer. Linear Algebra Appl., 20 (2013), pp. 913–928.
[64] J. Y. Zien, M. D. F. Schlag, and P. K. Chan, *Multilevel spectral hypergraph partitioning with arbitrary vertex sizes*, IEEE Trans. Comput.-Aided Des. Integrated Circuits Systems, 18 (1999), pp. 1389–1399.

Downloaded 11/16/16 to 202.196.114.131. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.