# COMPUTATION OF PROBLEMS RELATED WITH TENSORS ARISING FROM HYPERGRAPHS

CHANG JINGYA

PhD

The Hong Kong Polytechnic University

2018

The Hong Kong Polytechnic University

Department of Applied Mathematics

# Computation of Problems Related with Tensors Arising from Hypergraphs

Chang Jingya

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

Apr. 2018

# Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________(Signature)

Chang Jingya (Name of student)

# Abstract

Because the edge of a hypergraph can join more than two vertices, hypergraphs are much more powerful than graphs in storing multi-dimensional information, modelling complex relationship in real world. Tensors arising from a hypergraph, such as adjacency tensor, Laplacian tensor and signless Laplacian tensor, provide us a fundamental instrument to analyse the structure of a hypergraph, enrich the hypergraph theory, resolve the application problems in hypergraph.

Eigenvalues of tensors associated with hypergraphs are the core objects in spectral hypergraph theory. Although many achievements have been made in computation of tensor eigenvalues, the problem of computing eigenvalues of tensors arising from hypergraphs has not been completely settled yet, especially for large scale hypergraphs. By taking advantage of the sparsity of these tensors, we avoid saving the tensor information and propose a fast computational framework for the most time consuming operations. For the eigenvalue problem, we introduce a first-order optimization method called CEST (Computing Eigenvalues of large Sparse Tensors arising from hypergraphs) to solve it. It is proved that the sequence of function values and iterate points produced by the CEST method converge to the extremal eigenvalue and its associated eigenvector with a high probability. The CEST method is capable of calculating eigenvalues of tensors from hypergraphs with millions of vertices.

The $p$-spectral radius of a hypergraph is a concept that covers many important invariants and connected with Turán-type problems in the extremal hypergraph theory. The existing results about $p$-spectral radius problem are mainly based on theoretical analysis. Upper bounds or lower bounds of $p$-spectral radius of several structured hypergraphs are given, and as far as we know, there is not any computational method designed particularly for the $p$-spectral radius problem. By using adjacency tensors, we reformulate the original $p$-spectral radius problem to a spherical constraint maximization model and propose a method, named CSRH (Computing $p$-Spectral Radii of Hypergraphs), to solve it. The CSRH method can calculate the $p$-spectral radius when $p$ is greater than 1, and estimate the 1-spectral radii of uniform hypergraphs with high accuracy. As an application, we link the $p$-spectral radius model to data rank problems and successfully apply the CSRH method to rank 10305 authors according to their publication information from real life data set.

# Acknowledgements

First and foremost, I would like to express my deep gratitude to my supervisor, Professor Qi Liqun, for his patient guidance, generous assistance, and invaluable advice on both my research work and my life. I offer my sincere appreciation for the opportunity to learn from Professor Qi. Without his continued encouragement and support, this thesis would not have been completed or written. I also thank my co-supervisor, Dr Li xun, who gave me useful comments on my study.

I would especially like to thank my academic sisters Ouyang Chen and Liu Jinjie for their help during my study and life.

Finally I would like to thank my family, whose devotion, accompany and unconditional love always motivate me to move ahead in life.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Chapter 1

# Introduction

## 1.1 Background and motivation

In this subsection, we review the background of hypergraphs and tensors associated with hypergraphs. We explain our motivation on tackling the computational problems arising from hypergraph related tensors by displaying current state and illustrating the importance of this topic.

### 1.1.1 Why hypergraph?

It was in 1735 that Euler studied the“ seven bridges of Königsberg” problem and the graph theory arose subsequently (Gross and Yellen, 2004). Graph then became a useful tool in modeling a variety of problems in conceivable areas, such as social networks (Easley and Kleinberg, 2010), communication networks (Newman, 2010), software design applications (Agnarsson and Greenlaw, 2007), biological networks (Aldous and Wilson, 2003). Despite the advantages and widely applications, graph is confined to describe pairwise relationships via graph edges. Meanwhile, the complex relational objects in real life possess higher-order than binary relations. Here we cite the problem of grouping a set of articles raised in Zhou et al. (2007) to illustrate the limitation of graph in modeling complex relations. Suppose there are 3 authors $(Au_1, Au_2, Au_3)$ who collaborate 7 articles $(Ar_1, \dots, Ar_7)$ as shown in Table 1.1. If

$Au_i$ is an author of the article $Ar_j$, then the entry $(Au_i, Ar_j)$ in Table 1.1 is set to 1 and otherwise 0.

Table 1.1: Cooperations among 3 authors on 7 articles.

| | $Ar_1$ | $Ar_2$ | $Ar_3$ | $Ar_4$ | $Ar_5$ | $Ar_6$ | $Ar_7$ |
|---|---|---|---|---|---|---|---|
| $Au_1$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $Au_2$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $Au_3$ | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

In order to model the connections of objects in Table 1.1, one may construct a graph such that the two articles $(Ar_i, Ar_j)$ are joined by an edge if they have at least one common author or a graph such that the two authors $(Au_i, Au_j)$ are contained in an edge if they have collaborations as presented in Figure 1.1 and Figure 1.2.

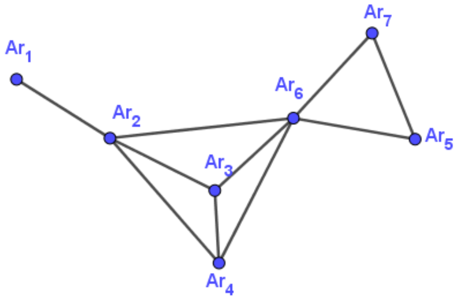


Figure 1.1: Articles that have common authors.

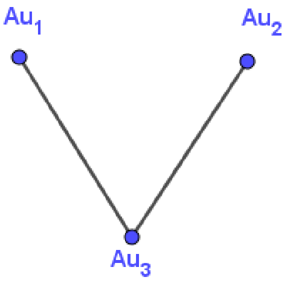


Figure 1.2: Authors that have collaborations.

The above two graphs demonstrate the information in Table 1.1 incompletely even if we consider both of them jointly. A natural way of remedying the information absence is to illustrate the data by a hypergraph instead in Figure 1.3. Also, it is shown that the pairwise similarity measure for separating points of two intersecting circles is inadequacy (Govindu, 2005; Chen et al., 2017). Some other examples in CVPR domain are given to show that hypergraphs are much more powerful than normal graphs in representing connections of objects in real world (Bunke H., 2008; Ducournau et al., 2012). Hypergraph as an important tool to extract valuable information and analyze massive data in our social life, is applied in science and engineering, such as chemistry (Klamt et al., 2009; Konstantinova and Skorobogatov, 2001), molecular chemistry (Konstantinova and Skorobogatov, 1995, 1998), computer science (Gunopulos et al., 1997; Karypis et al., 1999; Pliakos and Kotropoulos, 2015), image processing (Bretto and Gillibert, 2005; Chen et al., 2016b; Gao et al., 2012) and scientific computing (Fischer et al., 2010; Kayaaslan et al., 2012).

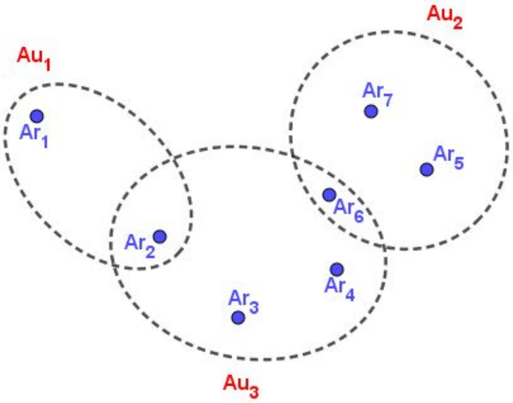


Figure 1.3: Hypergraph representation of author-article relationship.

### 1.1.2 Hypergraph related tensors

Matrices, such as adjacency matrices, incidence matrices and Laplacian matrices are commonly used to represent graphs, simplify the graph computation and study the graph properties (Bondy and Murty, 1976). Similarly, tensors are connected with hypergraphs (Shashua et al., 2006; Lim, 2005). The adjacency tensor of a uniform hypergraph is proposed in Friedman and Wigderson (1995); Lim (2008), and Cooper and Dutle (2012) proved some basic results in spectral hypergraph theory analogous to those in spectral graph theory via adjacency tensors. On account of the discretization of higher order Laplace-Beltrami operator, the Laplacian tensor of a uniform hypergraph is given in Hu and Qi (2012). Other definitions of Laplacian tensors of hypergraph are introduced in Li et al. (2013); Xie and Chang (2013a,c) soon afterwards. In formalism, all these Laplacian tensors are not as succinct as Laplacian matrices. Therefore, Qi (2014) defined the Laplacian tensor and signless Laplacian tensor of a uniform hypergraph as a natural extension of the Laplacian matrix in specral graph theory.

The hypergraph related tensors play important roles in at least the following two aspects of hypergraph study. First, the spectral hypergraph theory, which focuses on eigenvalues of adjacency tensors, Laplacian tensors and signless Laplacian tensors of hypergraphs, generalizes many classical conclusions from graph to hypergraph, estimates or gives bounds for hypergraph invariants, and is used to explore properties of the relevant hypergraphs. Lu and Man (2014, 2016a) directly extended the Smith's theorem from graph to hypergraph. Bulo and Pelillo (2009) generalized Motzkin-Straus Theorem to a class of hypergraphs, and linked the clique number to spectral hypergraph theory. Xie and Qi (2015) gave bounds for the clique number and the independence number of a uniform hypergraph via H-eigenvalues of the adjacency tensor, the Laplacian tensor, the signless Laplacian tensor of the related hypergraph. Hu and Qi (2012) introduced the algebraic connectivity of an even uniform hypergraph by Z-eigenvalues of its Laplacian tensor, and Li et al. (2017) p-resented bounds on analytic connectivity later. In terms of the bipartite property of a uniform hypergraph, Bu et al. (2016) showed that a connected even order uniform hypergraph is odd-bipartite if and only if the eigenvalues of its Laplacian and signless Laplacian tensors are the same, while Hu and Qi (2014) showed that any even order hypergraph is odd-bipartite if and only if zero is its signless Laplacian H-eigenvalue. Furthermore, the structures of uniform hypergraphs are discussed based on largest $\mathrm{H}^+$ eigenvalue of Laplacian tenors and spectral radius of adjacency tensors in Hu et al. (2015); Hu and Qi (2015); Fan et al. (2016a). Second, many problems related to hypergraph in science and engineering can be reformulated and solved by tensors. The Lagrangian of hypergraphs, as well as the $p$-spectral radii of hypergraphs in the extremal hypergraph theory are related to the adjacency tensors of the corresponding hypergraphs (Chang et al., 2018; Nikiforov, 2014). To solve the hypergraph matching problem in computer vision, a similarity tensor is constructed in Nguyen et al. (2015); Duchenne et al. (2011); Lee et al. (2011). The adjacency tensor of a hypergraph is involved to establish a spiked model for community detection in Kim et al. (2017). Ghoshdastidar and Dukkipati (2017) utilized the adjacency tensor of the corresponding hypergraph to tackle the uniform hypergraph partitioning problem, which is often encountered in computer vision, while Chen et al. (2017) introduced a Laplacian tensor spectral method to partition vertices of a hypergraph.

### 1.1.3 Computational problems of tensors arising from hypergraph

Due to the significance of tensors arising from hypergraph, many achievements have been made in this area (Qi and Luo, 2017). Most of the research emphasize on the theoretical analysis, such as spectral properties of adjacency tensors, Laplacian tensors, signless Laplacian tensors (Hu et al., 2015; Lin et al., 2016a,b; Yuan et al., 2015), the bounds for clique number (Xie and Qi, 2015), analytic connectivity (Hu and Qi, 2012; Li et al., 2017), the geometric or bipartite property of hypergraphs (Fan et al., 2016c; Hu et al., 2013; Fan et al., 2015, 2016b; Qi et al., 2014) and so on.

Although the computational problems, such as computing eigenvalues of adjacency tensors, Laplacian tensors, signless Laplacian tensors, computing the hypergraph invariants, computing the tensor models in hypergraph applications, are of important, few works have been done on these topics. Cui et al. (2016) proposed a feasible trust region algorithm to compute the analytic connectivity for a uniform hypegraph, and a quadratic penalty method for the tensor model in hypergraph matching was developed in Cui et al. (2018). Also, an optimization algorithm for the aforementioned Laplacian tensor spectral method was given in Chen et al. (2017) to solve the hypergraph partitioning problem. For other computational problems related to tensors arising from hypergraph, they are still open to our knowledge. This motivates us to work on the subject of hypergraph related tensor computation.

The tensor we study in this thesis refers to a hypermatrix or a tentrix (Gao, 2016).

## 1.2 Main results and outline of the thesis

In this subsection we briefly introduce the contributions of this thesis and give the outline of the following parts. The results in the thesis are based on the works in Chang et al. (2016) and Chang et al. (2018).

In Chapter 2, we give notations that will be used, and introduce results related to our work in previous tensor and hypergraph research.

Hypergraphs generated from real life data are usually large scale, while on the other hand the existing algorithms cannot solve problems of computing eigenvalues of large scale adjacency tensors, Laplacian tensors and signless Laplacian tensors completely. We deal with such problems arising from even uniform hypergraphs by exploiting the uniform hypergraph structure, considering the tensor optimization model on a unit sphere and introducing an efficient iterative algorithm called CEST.

With the aid of Łojasiewica inequality, the algorithm CEST is proved to be convergent, and numerical experiments are given to demonstrate that our method is highly efficient. Detailed introduction of this topic is shown in Chapter 3.

The $p$-spectral radii of uniform hypergraphs are significant in the extremal hypergraph theory, which cover parameters such as Lagrangians and spectral radii of hypergraphs. In Chang et al. (2018), we formulate a spherically constrained tensor optimization problem, and generate an algorithm called CSRH to calculate the $p$-spectral radii of uniform hypergraphs. The globally convergence property of the approach CSRH is proved. Furthermore, the $p$-spectral model is first suggested to be applied in network analysis. The numerical results provide examples of the C-SRH algorithm ranking real-life data via $p$-spectral radii of uniform hypergraphs. We describe the specific results on $p$-spectral radii of uniform hypergraphs in Chapter 4.

## 1.3 Notations

| | | |
|---|---|---|
| (1) | $\mathbb{R}$ | Set of real numbers; |
| (2) | $\mathbb{C}$ | Set of complex numbers; |
| (3) | $\mathbb{R}^n$ | Set of $n$-dimensional real vectors; |
| (4) | $\mathbb{C}^n$ | Set of $n$-dimensional complex vectors; |
| (5) | $T^{r,n}$ | Set of $r$th order $n$-dimensional tensors; |
| (6) | $\mathbb{C}^{r,n}$ | Set of $r$th order $n$-dimensional complex tensors; |
| (7) | $\mathbb{R}^{r,n}$ | Set of $r$th order $n$-dimensional real tensors; |
| (8) | $S^{r,n}$ | Set of $r$th order $n$-dimensional real symmetric tensors; |
| (9) | $\mathcal{A} \otimes \mathcal{B}$ | Tensor outer product; |
| (10) | $\mathcal{A} \bullet \mathcal{B}$ | Tensor inner product; |
| (11) | $\mathcal{A} \circ \mathcal{B}$ | Tensor Hadamard product; |
| (12) | $(\mathbf{x}^{[r-1]})_i$ | $x_i^{r-1}$ : the $i$th entry of the vector $\mathbf{x}^{[r-1]}$. |

# Chapter 2

# Preliminaries

In this Chapter, we introduce the concepts of tensor, hypergraph, etc., and review the relevant results in spectral hypergraph theory .

## 2.1 Tensor

### 2.1.1 Structured tensors and tensor multiplication

A tensor $\mathcal{T}$ is a multi array with its entry $t_{i_1 \cdots i_r} \in \mathbf{F}$, for $i_j = 1, \ldots, n_j$, and $j = 1, \ldots, r$. When $n_1 = n_2 = \ldots = n_r = n$, we say $\mathcal{T}$ is $r$th order $n$-dimensional, and denote it as $T^{r,n}$. If $\mathbf{F} = \mathbb{R}$, the set of $r$th order $n$-dimensional tensor is $\mathbb{R}^{r,n}$.

**Unit tensor:** A unit tensor $\mathcal{I}$ is defined as

$$d_{i_1 \cdots i_r} = \begin{cases} 1 & \text{if } i_1 = i_2 = \cdots = i_r = i, \\ 0 & \text{otherwise.} \end{cases}$$

**Reducible and irreducible tensor:** The tensor $\mathcal{C} \in \mathbb{C}^{r,n}$ is reducible (Chang et al., 2008) if there is a nonempty set $I \in \{1, \ldots, n\}$ such that

$$c_{i_1 \cdots i_r} = 0, \forall i_1 \in I \ \text{ and } \ i_2, \ldots, i_r \notin I.$$

Otherwise, it is an irreducible tensor.

**Nonnegative and $M$-tensor:** If all entries of a tensor are nonnegative, then it is an nonnegative tensor. As a natural extensional of $M$-matrix, a tensor $\mathcal{A} \in T^{r,n}$ is an $M$-tensor (Zhang et al., 2014) if there have a nonnegative tensor $\mathcal{B}$ and a positive real number $\eta$ satisfying

$$\mathcal{A} = \eta I - \mathcal{B}.$$

**Tensor outer product:** The outer product of two tensors $\mathcal{A} = (a_{i_1 \cdots i_{r_1}}) \in \mathcal{T}^{r_1,n}$ and $\mathcal{B} = (b_{j_1 \cdots j_{r_2}}) \in \mathcal{T}^{r_2,n}$ is defined as

$$\mathcal{A} \otimes \mathcal{B} = (a_{i_1 \cdots i_{r_1}} b_{j_1 \cdots j_{r_2}}) \in \mathcal{T}^{r_1+r_2,n}.$$

**Tensor inner product:** The inner product of two tensors $\mathcal{A} = (a_{i_1 \cdots i_r}) \in \mathcal{T}^{r,n}$ and $\mathcal{B} = (b_{i_1 \cdots i_r}) \in \mathcal{T}^{r,n}$ is defined as

$$\mathcal{A} \bullet \mathcal{B} = \sum_{i_1 \cdots i_r = 1}^{n} a_{i_1 \cdots i_r} b_{i_1 \cdots i_r}.$$

**Tensor Hadamard product:** The Hadamard product of two tensors $\mathcal{A} = (a_{i_1 \cdots i_r}) \in \mathcal{T}^{r,n}$ and $\mathcal{B} = (b_{i_1 \cdots i_r}) \in \mathcal{T}^{r,n}$ is defined as

$$\mathcal{A} \circ \mathcal{B} = a_{i_1 \cdots i_r} b_{i_1 \cdots i_r} \in \mathcal{T}^{r,n}.$$

**$k$-mode product:** The notations of several frequently used $k$-mode products between a tensor $\mathcal{A} = (a_{i_1 \cdots i_r}) \in \mathbb{R}^{r,n}$ and a vector $\mathbf{x}^\top = (x_1 \cdots x_n) \in \mathbb{R}^n$ are as follows:

$$\begin{aligned}
\mathcal{A}\mathbf{x}^{r-2} &= \sum_{i_3 \cdots i_r = 1}^{n} a_{iji_3 \cdots i_r} x_{i_3} \cdots x_{i_r} \in \mathbb{R}^{n \times n}, \\
\mathcal{A}\mathbf{x}^{r-1} &= \sum_{i_2 \cdots i_r = 1}^{n} a_{ii_2 \cdots i_r} x_{i_3} \cdots x_{i_r} \in \mathbb{R}^{n}, \\
\mathcal{A}\mathbf{x}^{r} &= \sum_{i_1 \cdots i_r = 1}^{n} a_{i_1 \cdots i_r} x_{i_1} \cdots x_{i_r} \in \mathbb{R}.
\end{aligned}$$

### 2.1.2 Eigenvalues of tensors

The eigenvalues of tensors are crucial in the computational problems we studied. The spectral hypergraph theory emphasises on eigenvalues of adjacency tensors, Laplacian tensors and signless Laplacian tensors of the corresponding hypergraph. The $p$-spectral radius of a hypergraph is related to eigenvalues of the adjacency tensor of the corresponding hypergraph when $p$ equals two or the order of this hypergraph.

**Eigenvalues and eigenvectors:** For $\mathcal{A} \in T^{r,n}$, if we can find a number $\lambda \in \mathbf{C}$ and a vector $\mathbf{x} \in \mathbf{C}^n$ satisfying the following homogeneous polynomial equations

$$(\mathcal{A}\mathbf{x}^{r-1})_i = \lambda x_i^{r-1}, \quad \forall\, i = 1, \ldots, n, \tag{2.1}$$

the number $\lambda$ and the vector $\mathbf{x}$ are eigenvalue and eigenvector, or so-called eigenpair of $\mathcal{A}$ respectively. Moreover, denote a vector $\mathbf{x}^{[r-1]} \in \mathbb{R}^n$ such that its $i$th entry being $x_i^{r-1}$ for short. The equation in (2.1) can be written as $(\mathcal{A}\mathbf{x}^{r-1})_i = \mathbf{x}^{[r-1]}$. If $\lambda$ and $\mathbf{x}$ is an eigenpair of $\mathcal{A}$ and $\alpha$ is a real number, then $(\alpha\lambda, \mathbf{x})$ is also an eigenpair of $\mathcal{A}$.

**H-Eigenvalues and H-eigenvectors:** If an eigenvalue $\lambda$ of $\mathcal{A}$ associate with a real eigenvector $\mathbf{x}$, the eigenpair $(\lambda, \mathbf{x})$ is called H-eigenvalue and H-eigenvector of $\mathcal{A}$. Since $\mathbf{x}$ is not zero, there exists at least one index $i \in \{1, 2, \ldots, n\}$ such that

$x_i$ is nonzero. Therefore eigenvalue $\lambda = \frac{(\mathcal{A}\mathbf{x}^{r-1})_i}{x_i^{r-1}}$ is real. On the other hand, if the eigenvalue $\lambda$ is real, it may not be an H-eigenvalue, because its associated eigenvector $\mathbf{x}$ may not be a real vector. Such counter examples are presented in Qi (2005a). The H-eigenvalue set of $\mathcal{A}$ is called H-spectrum of $\mathcal{A}$, and we abbreviate it as Hspec($\mathcal{A}$).

**E-Eigenvalues and E-eigenvectors:** Although eigenvalues and H-eigenvalues have nice mathematical structures, they are not invariant under orthogonal transformation (Qi and Luo, 2017). In physics, some physical concepts are hoped to be invariant under orthogonal transformation in the laboratory coordinate system. The E-eigenvalues and Z-eigenvalues are then introduced to fill this gap.

If a number $\lambda \in \mathbf{C}$ and a vector $\mathbf{x} \in \mathbf{C}^n$ are solutions of the following equation system

$$\mathcal{A}\mathbf{x}^{r-1} = \lambda\mathbf{x}, \tag{2.2}$$

$$\mathbf{x}^\top\mathbf{x} = 1, \tag{2.3}$$

then $\lambda$ and $\mathbf{x}$ are E-eigenvalue and E-eigenvector of $\mathcal{A}$ respectively.

**Z-Eigenvalues and Z-eigenvectors:** If an E-eigenvalue of a tensor is associated with a real E-eigenvector, the E-eigenvalue and the E-eigenvector become Z-eigenvalue and Z-eigenvector respectively. It can be deduced from (2.2) that $\lambda = \mathcal{A}\mathbf{x}^r$, thus $\lambda$ is real when its associated eigenvector is real. Conversely, the condition that E-eigenvalue is real can not guarantee $\mathbf{x}$ being a real vector. If $\mathcal{A} \in S^{r,n}$, it always has Z-eigenvalues. The Z-eigenvalue set of $\mathcal{A}$ is called Z-spectrum of $\mathcal{A}$, Zspec($\mathcal{A}$) for short.

**$\mathcal{B}$-Eigenvalues and $\mathcal{B}$-eigenvectors of tensors:** In Chang et al. (2009), a generalized eigenvalue of tensor was proposed. If both $\mathcal{A}$ and $\mathcal{B}$ are $r$th order $n$-dimensional tensors on $\mathbb{R}$, and $(\lambda, \mathbf{x}) \in \mathbb{C} \times (\mathbb{C}^n \backslash \{0\})$ satisfies the following equation:

$$(\mathcal{A} - \lambda\mathcal{B})\mathbf{x}^{r-1} = 0,$$

we call $(\lambda, \mathbf{x})$ a $\mathcal{B}$ eigenpair of $\mathcal{A}$. Denote $\mathcal{I}$ as an identity tensor, whose elements are 0 except the diagonal elements being 1. If $\mathcal{B} = \mathcal{I}$, then the $\mathcal{B}$-eigenvalues of $\mathcal{A}$ are the eigenvalues of $\mathcal{A}$, and the real $\mathcal{B}$-eigenvalues of $\mathcal{A}$ with real eigenvectors are the H-eigenvalues of $\mathcal{A}$. Let $I_2$ be the $n \times n$ unit matrix. When the order $m = 2l$ is even and $\mathcal{B} = I_2^l$, a real $\mathcal{B}$-eigenvalue associated with a real eigenvector is in fact a Z-eigenvalue.

## 2.2 Hypergraph

First, we introduce the concepts of hypergraph, uniform hypergraph, adjacency tensor, Laplacian tensor, signless Laplacian of the corresponding hypergraph.

### 2.2.1 Hypergraph and structured hypergraphs

**Hypergraph and uniform hypergraph:** Hypergraph is an extension of ordinary graph. Each edge of a graph has two vertices, while the edge of a hypergraph can join any number of vertices. Let $V = \{1, 2, \ldots, n\}$ be the vertex set, and $E = \{e_1, e_2, \ldots, e_m\}$ be the edge set for $e_p \subset V, p = 1, \ldots, m$. Define $G = (V, E)$ as a hypergraph. If a positive number $s(e)$ is associated with each edge of the hyperpragh, then this hypergraph is a weighted hypergraph with $s(e)$ being the weight linked with edge $e$. When the weight of each edge is 1, a weighted hypergraph is an ordinary hypergraph. If the length of each edge in the hypergraph is the same, i.e.,$|e_p| = r$ for $p = 1, \ldots, m$, then the hypergraph $G$ is called a uniform hypergraph or an $r$-graph for short. When $r = 2$, a hypergraph is in fact an ordinary graph. For any vertex $i \in V$, the degree of $i$ is

$$d(i) = \mathrm{sum}\{s(e) : i \in e, e \in E\}.$$

A 4-graph is given in Figure 2.1 as an example. The vertex set of the sunflower hypergraph is $V = \{1, \cdots, 10\}$, and the edge set is $E = \{e_1 = \{1, 2, 3, 4\}, e_2 =$

$\{1, 5, 6, 7\}, e_3 = \{1, 8, 9, 10\}\}$. The order of this hypergraph is 4. The degree $d_i$ equals 1 for $i = 2, \cdots, 10$ and $d_1 = 3$.

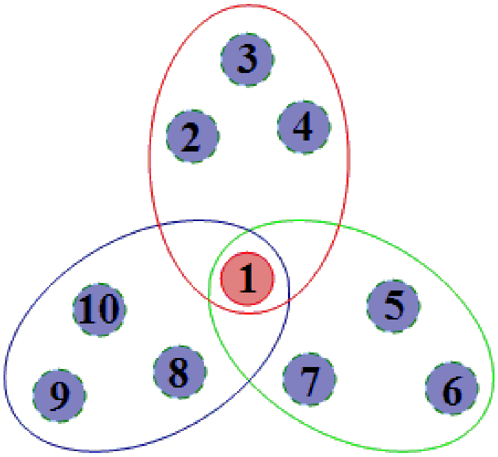


Figure 2.1: A 4-uniform hypergraph.

**Connected hypergraph:** If two vertices belong to the same edge, they are called adjacent. Two vertices $i$ and $j$ are connected, if they are adjacent or there exist a vertex subset $\{i, v_1, \ldots, v_k, j\} \subseteq V$ such that vertices $i$ and $v_1$, $v_k$ and $j$, $v_t$ and $v_{t+1}$ for $t = 1, \ldots, t-1$ are adjacent. The hypergraph $G$ is connected if all pair of vertices are connected.

**Odd partite hypergraph:** A hypergraph (Hu and Qi, 2014) $G$ is odd bipartite if there exist two nonempty subsets $V_1$ and $V_2$ such that

(1) $V = V_1 \cup V_2$,

(2) $V_1 \cap V_2 = \varnothing$,

(3) each edge of $G$ intersects $V_1$ in an odd number of vertices.

**Sunflower:** Suppose an $r$-graph $G_S = (V, E)$ has $\Delta$ edges and $n = (r-1)\Delta + 1$

vertices. If the vertex set $V$ can be decomposed as

$$V = V_0 \cup V_1 \cup \cdots \cup V_\Delta, \quad \text{with} \quad |V_0| = 1, \quad \text{and} \quad |V_i| = r - 1 \text{ for } i = 1, \ldots, \Delta,$$

and the edge set $E$ can be represented as

$$E = \{V_0 \cup V_i \,|\, i = 1, \ldots, \Delta\},$$

then $G_S$ is named a sunflower with its maximum degree being $\Delta$. For instance, the maximum degree of the sunflower shown in Figure 2.1 is $\Delta = 3$. A sunflower is also called a $\beta$-star. A class of 6-uniform $\beta$-stars are given in Figure 2.2 for reference.

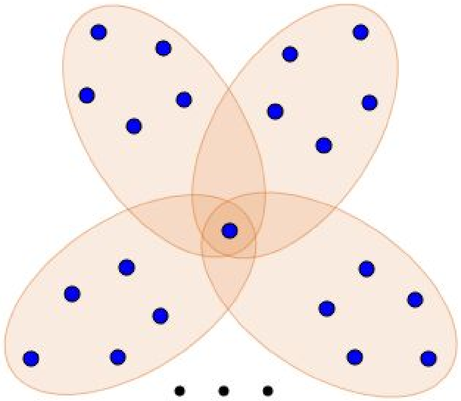


Figure 2.2: A class of 6-uniform $\beta$-stars.

**Loose path:** (Chang et al., 2018) An $r$-graph with $m$ edges is called a loose path if its vertex set is

$$V = \left\{i_{(1,1)}, \ldots, i_{(1,r)}, i_{(2,2)}, \ldots, i_{(2,r)}, \ldots, i_{(m,2)}, \ldots, i_{(m,r)}\right\}$$

and its edge set is

$$E = \left\{\{i_{(1,1)}, \ldots, i_{(1,r)}\}, \{i_{(1,r)}, i_{(2,2)}, \ldots, i_{(2,r)}\}, \ldots, \{i_{(m-1,r)}, i_{(m,2)}, \ldots, i_{(m,r)}\}\right\}.$$

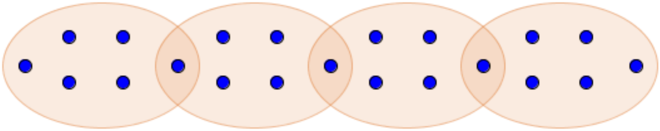


Figure 2.3: A loose path.

An $r$-uniform loose path with $m$ edges has $m(r-1)+1$ vertices. For instance, there are 21 vertices in the 6 order loose path in Figure 2.3.

**Complete hypergraph:** If an $r$-graph with $n$ vertices contains all possible edges, then it is called a complete hypergraph and denoted as $C_n^r$. For example, the tetrahedron in Figure 2.4 can be regarded as a complete 3-graph with 4 vertices, i.e., $C_4^3$.

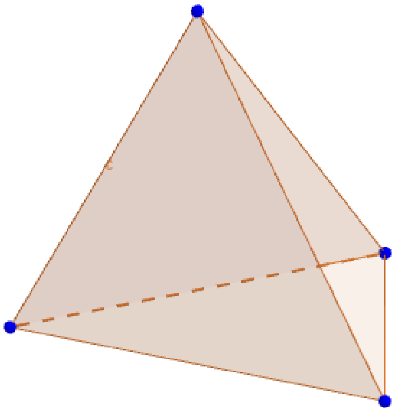


Figure 2.4: A 3-uniform complete hypergraph with 4 vertices: $C_4^3$.

### 2.2.2 Tensors arising from hypergraphs and $p$-spectral radius

Let $G$ be an $r$-graph with $n$ vertices. Next, we introduce the definitions of adjacency tensor $\mathcal{A}$, Laplacian tensor $\mathcal{L}$, degree tensor $\mathcal{D}$ and signless Laplacian tensor $\mathcal{Q}$ of $G$.

**Adjacency tensor:** The adjacency tensor $\mathcal{A} = (a_{i_1,\cdots,_{1_r}})$ is an $r$th order $n$-dimensional symmetric tensor as follows (Bulo and Pelillo, 2009; Cooper and Dutle, 2012; Xie and

Chang, 2013b)

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

**Degree tensor:** The degree tensor $\mathcal{D}$ of $G$ is an $r$th order $n$-dimensional diagonal tensor, whose entries are

$$d_{i_1\cdots i_r} = \begin{cases} d(i) & \text{if } i_1 = i_2 = \cdots = i_r = i, \\ 0 & \text{otherwise.} \end{cases}$$

**Laplacian and signless Laplacian tensor:** The notations of Laplacian and signless Laplacian tensors based on sums of $r$th powers in Hu and Qi (2012); Li et al. (2013); Xie and Chang (2013c) are not the natural extension from matrix forms. Hence, Qi (2014) introduced a simple definition for the Laplacian and signless Laplacian tensors of an $r$-graph according to the definition of Laplacian matrix. The Laplacian tensor is given as $\mathcal{L} = \mathcal{D} - \mathcal{A}$, while the signless Laplacian tensor is $\mathcal{Q} = \mathcal{D} + \mathcal{A}$. The signless Laplacian tensor $\mathcal{Q}$ of $G$ is symmetric nonnegative. Moreover, the Laplacian tensor is the limit of symmetric $M$-tensors. In the rest of this thesis, the symbols $\mathcal{D}$, $\mathcal{A}$, $\mathcal{L}$, and $\mathcal{Q}$ are specially referred to the degree tensor, adjacency tensor, Laplacian tensor and signless Laplacian tensor of a hypergraph respectively.

**$p$-spectral radius:** (Kang et al., 2015; Keevash et al., 2014; Nikiforov, 2014) The $p$-spectral radius of $G$ is denoted as

$$\lambda^{(p)}(G) = r! \max_{p \geqslant 1, \|\mathbf{x}\|_p = 1} \sum_{e=\{i_1,\ldots,i_r\}\in E} s(e)\, x_{i_1} \cdots x_{i_r} \tag{2.4}$$

where $\|\mathbf{x}\|_p = (\sum_{k=1}^{n} |\mathbf{x}_k|^p)^{\frac{1}{p}}$, and the vector $\mathbf{x}$ solving (2.4) is called a $p$-optimal weighting of $G$ (Caraceni, 2011).

**Lagrangian of a hypergraph:** The Lagrangian of a hyeprgraph is defined as

$$\lambda_L(G) = \begin{cases} \max & \sum_{e=\{i_1,\dots,i_r\}\in E} s(e)\, x_{i_1}\cdots x_{i_r} \\ \text{s.t.} & \sum\limits_{i=1}^{r} x_i = 1, \\ & x_i \geqslant 0, \quad \text{for} \quad i = 1,\dots,r. \end{cases} \tag{2.5}$$

From the definition of $p$-spectral radius in 2.4, we have $\lambda^{(1)}(G) = \lambda_L(G)$.

### 2.2.3 Some useful results in spectral hypergraph theory

Define $\lambda^H_{\max}(\mathcal{T}(G))$ and $\lambda^H_{\min}(\mathcal{T}(G))$ as the largest and smallest H-eigenvalue of a tensor $\mathcal{T}$ related to a hypergraph $G$ respectively. Similarly, the notation $\lambda^Z_{\max}(\mathcal{T}(G))$ and $\lambda^Z_{\min}(\mathcal{T}(G))$ are the largest and smallest Z-eigenvalue of $\mathcal{T}$ related to a hypergraph $G$ respectively.

**Theorem 2.1.** *For a connected $r$-graph $G$, the following conclusions are equivalent to each other*

*(1) $G$ is an even uniform and odd bipartite hypergraph.*

*(2) $\lambda^H_{\max}(\mathcal{L}(G)) = \lambda^H_{\max}(\mathcal{Q}(G))$ (Hu et al., 2015).*

*(3) $Zspec(\mathcal{L}(G)) = Zspec(\mathcal{Q}(G))$ (Bu et al., 2016).*

*(4) $Hspec(\mathcal{L}(G)) = Hspec(\mathcal{Q}(G))$ (Shao et al., 2015).*

*(5) $Hspec(\mathcal{A}(G)) = -Hspec(\mathcal{A}(G))$ (Shao et al., 2015).*

For a simple graph $G$, the generalized power of $G^{r,s}$ for $r \geqslant 2s$ is given by replacing each vertex of the simple graph by an $s$-subsect and adding $r-2s$ new vertices in each edge.

**Theorem 2.2.** *If $G$ is a simple graph, the $2r$-graph $G^{2r,r}$ obtained from $G$ is not odd bipartite if and only if $G$ is nonbipartite (Khan and Fan, 2015). Moreover, Khan et al. (2016) proved that*

$$\lambda_{\min}(\mathcal{A}(G)) = \lambda_{\min}(\mathcal{A}(G^{2r,r})), \quad \lambda_{\min}(\mathcal{Q}(G)) = \lambda_{\min}(\mathcal{Q}(G^{2r,r})).$$

The largest H-eigenvalue of the Laplacian tensor of an even order sunflower is proved to have a closed-form solution.

**Theorem 2.3.** *(Theorems 3.2 and 3.4 in Hu et al. (2015)) Assume $G$ is an $r$-graph with the order $r$ being an even number and greater than $4$. Let $\lambda_H^*$ be the unique real root of the equation $(1-\lambda)^{k-1}(\lambda-\Delta)+\Delta = 0$ lying in the interval $(\Delta, \Delta+1)$. Then we have*

$$\lambda_{\max}^H(\mathcal{L}) \geqslant \lambda_H^*.$$

*The equality holds if and only if $G$ is a sunflower.*

The following theorem gives the answer to the largest H-eigenvalues of adjacency tensors of loose paths with $m = 3$ or $m = 4$.

**Theorem 2.4.** *(Yue et al., 2016; Chang et al., 2018) Assume $G$ is an $r$-uniform loose path with $m$ edges. Then we have*

*(1) $\lambda_{\max}^H(G) = \left(\frac{1+\sqrt{5}}{2}\right)^{\frac{2}{r}}$ for $m = 3$,*

*(2) $\lambda_{\max}^H(G) = 3^{\frac{1}{r}}$ for $m = 4$.*

Although there is no algorithm or formula specially designed for the numerical or analytical solution of $p$-spectral radius of a general hypergraph, some progress has been made in obtaining the value of $p$-spectral radius of structured hypergraphs.

**Theorem 2.5.** *(Nikiforov, 2014; Chang et al., 2018) Suppose the $r$-graph $G$ is a $\beta$-star with $m$ edges.*

*(1) If* $p > r-1$, *then* $\lambda^{(p)}(G) = r!r^{-\frac{r}{p}}m^{(1-\frac{r-1}{p})}$.

*(2) If* $p < r-1$, *then* $\lambda^{(p)}(G) = r!r^{-\frac{r}{p}}$.

*(3) If* $p = r-1$, *then* $\lambda^{(p)}(G) = (r-1)!r^{-\frac{1}{r-1}}$.

**Proposition 2.1.** *(Caraceni, 2011; Chang et al., 2018) Let* $G$ *be a complete* $r$*-graph with* $n$ *vertices, then the Lagrangian of* $G$ *is*

$$\lambda_L(G) = \begin{pmatrix} n \\ r \end{pmatrix} \frac{1}{n^r}. \tag{2.6}$$

# Chapter 3

# Computing eigenvalues of large scale tensors arising from even order uniform hypergraph

In this chapter, we consider problems of computing H- and Z-eigenvalues of the adjacency tensor, the Laplacian tensor, and the signless Laplacian tensor associated with a uniform hypergraph.

## 3.1 Introduction

Spectral hypergraph theory connects the geometry of a uniform hypergraph and H- and Z-eigenvalues of its corresponding tensors. For instance, an even-uniform connected hypergraph is odd-bipartite if and only if the largest H-eigenvalue of its Laplacian is equivalent to the largest H-eigenvalue of its signless Laplacian tensor (Hu et al., 2015). This result enables us to check the odd-bipartite property of a connected even-uniform hypergraph.

Since the adjacency tensor and the signless Laplacian tensor are symmetric and nonnegative, algorithms for eigenvalue problems of nonnegative symmetric tensors are available for eigenvalue problems of adjacency tensors, as well as signless Laplacian tensors. Therefore, an efficient algorithm called Ng-Qi-Zhou, for computing the largest eigenvalues of irreducible nonnegative tensors (Ng et al., 2009) is feasible for calculating the largest H-eigenvalues and associated eigenvectors of adjacency tensors and the signless Laplacian tensors. When apply the Ng-Qi-Zhou method to a primitive nonnegative symmetric tensor, the sequence generated by Ng-Qi-Zhou method is proved to be convergent (Chang et al., 2011). The convergence criteria is relaxed to irreducible nonnegative symmetric tensor in Liu et al. (2010). Further, the Ng-Qi-Zhou method is enhanced and it is demonstrated that this enhanced method converges when a nonnegative symmetric tensor is irreducible (Liu et al., 2010). In Friedland et al. (2013), the Ng-Qi-Zhou approach is shown to converge to the largest H-eigenvalue problem of a weakly irreducible nonnegative symmetric tensor R-linearly. Later, the Q-linear convergence property of the Ng-Qi-Zhou method is proved (Zhou et al., 2013a,b). Furthermore, Zhou et al. (2013b) refined the Ng-Qi-Zhou method, and showed that the refined one can acquire the largest H-eigenvalue of nonnegative symmetric tensors. Besides, a local quadratic convergent algorithm was proposed in Ni and Qi (2015).

Usually there are two kinds of approaches for eigenvalue problem of general symmetric tensors. One category can get all (real) eigenvalues of a small dimension tensor. With the aid of resultant, a direct method was given in Qi et al. (2009). Cui et al. (2014) established an SDP relaxation method coming from polynomial optimization. Two homotopy continuation type algorithms were proposed in Chen et al. (2016a). Also, subroutines "`NSolve`" and "`solve`" provided in softwares *Mathematica* and *Maple* can solve polynomial eigen-systems exactly. These methods cost prohibitively long time when finding eigenvalues of a symmetric tensor which has dozens of variables.

The other kind of methods focus on computing an (extreme) eigenvalue of a symmetric tensor. The reason for computing only one eigenvalue of a tensor is that, a general symmetric tensor has plenty of eigenvalues (Qi, 2005b), and it is NP-hard to calculate all the eigenvalues. (Hillar and Lim, 2013). A spherical optimization model was proposed by Kolda and Mayo. They introduce shifted power approaches for this model, which were proved to converge to an eigenvalue and its corresponding eigenvector of a symmetric tensor based on fixed point theory (Kolda and Mayo, 2011, 2014). Hao et al. (2015) introduced a subspace projection approach to solve the spherical optimization model. A quadratic penalty function of the spherical optimization problem was constructed in Han (2013) to find eigenvalues of even order symmetric tensors. Numerical experiments illustrated that these approaches are able to calculate eigenvalues of symmetric tensors which have dozens of variables.

In terms of a Laplacian tensor corresponding to an even-uniform hypergraph which has millions of vertices, how to calculate its (extreme) eigenvalue? Storing and processing a large scale tensor directly are difficult.

For the purpose of computing eigenvalues and eigenectors of a large scale sparse tensor, for instance the adjacency tensor, the Laplacian tensor, and the signless Laplacian tensor, related to a uniform hypergraph, we propose a fast computational method for products of these tensors and any vectors. In order to save a uniform hypergraph, we construct a matrix, whose row elements are the vertices in each edge of the hypergraph. Based on this matrix, our fast computational method avoid producing and saving the large scale tensor explicitly when calculate the product of a Laplacian tensor and a vector. The computational cost of the fast computational framework is linear in the size of edges and quadratic in the number of vertices of an edge. Moreover, other tensors associated with uniform hypergraphs, i.e., signless Laplacian tensors and adjacency tensors, can be treated in the same way. Next we propose an effective first-order optimization method for calculating H- and Z-eigenvalues of adjacency, Laplacian, and signless Laplacian tensors arising from the even-uniform hypergraph. By minimizing a smooth merit function on a unit sphere, we get the first-order stationary point of this optimization model, whcih is an eigenvector associated with an eigenvalue. By using Cayley transform, we develop a specific formula to deal with the spherical constraint. The spherical optimization model is then transferred into an unconstrained problem. In view of the large scale aspects of the problem, the L-BFGS method is explored to produce a direction which is relevant to the gradient, and a backtracking search is employed to guarantee that the sequence of iterates converges. Our algorithm (CEST) for calculating eigenvalues of even-order symmetric tensors is established on the basis of these techniques. Because the graph of the objective function is a semi-algebraic set, the objective function satisfies the Łojasiewicz inequality, which ensures that the sequence of iterative points given by the CEST method converges to an eigenvector. Furthermore, it is proved that if the CEST method is started from multiple initial points sampled uniformly from a unit sphere, the result could achieve the largest (smallest) eigenvalue with a probability close to one.

When compute eigenvalues of symmetric tensors corresponding to small hypergraphs, numerical tests demonstrate that our CEST method performs dozens of times faster than the power method. In addition, the CEST algorithm is able to calculate H- and Z-eigenvalues, as well as their associated eigenvectors of symmetric tensors arising from an even order uniform hypergraph which has millions of vertices.

## 3.2 The CEST method

We give the CEST method based on the unified equation for H- and Z-eigenvalues of symmetric tensors (Chang et al., 2009). Suppose $\mathcal{I} \in \mathbb{R}^{[r,n]}$ is an identity tensor. Then, $\mathcal{I}\mathbf{x}^{r-1} = \mathbf{x}^{[r-1]}$. When $r$ is even, let $\mathcal{E} \in \mathbb{R}^{[r,n]}$ be a symmetric tensor such that $\mathcal{E}\mathbf{x}^{r-1} = (\mathbf{x}^\top\mathbf{x})^{\frac{r}{2}-1}\mathbf{x}$. Because $\mathcal{I}\mathbf{x}^r = \sum_{i=1}^n x_i^r$ and $\mathcal{E}\mathbf{x}^r = \|\mathbf{x}\|^r$ are greater than zero for any nonzero vector $\mathbf{x}$, tensors $\mathcal{I}$ and $\mathcal{E}$ are positive definite. The systems (2.1) and (2.2) are combined into

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

in which $\mathcal{B}$ equals $\mathcal{I}$ and $\mathcal{E}$ respectively. In the rest of this chapter, if $\lambda \in \mathbb{R}$ and the nonzero vector $\mathbf{x} \in \mathbb{R}^n$ satisfy (3.1), $\lambda$ and $\mathbf{x}$ are called eigenvalue and its associated eigenvector of $\mathcal{T}$ respectively. Next, we concentrate on calculating such $\lambda$ and $\mathbf{x}$ for large scale sparse tensors. Suppose $r$ is even and the symmetric tensor $\mathcal{T}$ is associated with an $r$-uniform hypergraph. Hence $\mathcal{T}$ is sparse and may be large scale. We consider the spherical constraint optimization model

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

where $\mathbb{S}^{n-1} \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top\mathbf{x} = 1.\}$ Since $f(\mathbf{x})$ is zero-order homogeneous, we limit $\mathbf{x}$ on the unit sphere $\mathbb{S}^{n-1} \equiv \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top\mathbf{x} = 1\}$ without loss of generality. The compactness of $\mathbb{S}^{n-1}$ makes our algorithm CEST easy to obtain. Also, the spherical constraint guarantee $\mathbf{x} \in \mathbb{S}^{n-1}$ being away from the original point and the CEST algorithm being convergent. The structure of the symmetric positive definite tensor $\mathcal{B}$ is as simple as $\mathcal{I}$ and $\mathcal{E}$. The gradient of $f(\mathbf{x})$ Chen et al. (2016c) is

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

We reveal the connection between the optimization model (3.2) and the eigenvalue problem (3.1) in the following theorem. Owing to the freedom of the positive definite tensor $\mathcal{B}$ (3.1), the next theorem covers both the case of H-eigenvalue and the case of Z-eigenvalue as $\mathcal{B} = \mathcal{I}$ and $\mathcal{B} = \mathcal{E}$, respectively.

**Theorem 3.1.** *Let $\mathcal{B}$ be a positive definite symmetric tensor with its order $r$ being even. Let $\mathbf{x}_* \in \mathbb{S}^{n-1}$. Hence, $\mathbf{x}_*$ is a first-order stationary point, i.e., $\nabla f(\mathbf{x}_*) = 0$, if and only if there is a scalar $\lambda_* \in \mathbb{R}$ such that $(\lambda_*, \mathbf{x}_*)$ satisfies (3.1). Moreover,*

*$\lambda_* = f(\mathbf{x}_*)$ is the Z-eigenvalue (resp. H-eigenvalue) with $\mathbf{x}_*$ being the associated Z-eigenvector (resp. H-eigenvector) if $\mathcal{B} = \mathcal{E}$ (resp. $\mathcal{B} = \mathcal{I}$).*

*Proof.* Because $\mathcal{B}$ is positive definite and $\mathcal{B}\mathbf{x}^r$ is positive for any $\mathbf{x}$ in $\mathbb{S}^{n-1}$, we have $f(\mathbf{x}_*)$ is an eigenvalue and $\mathbf{x}_*$ is its related eigenvector from (3.3) if $\mathbf{x}_* \in \mathbb{S}^{n-1}$ satisfies $\nabla f(\mathbf{x}_*) = 0$. On the other hand, if $\mathbf{x}_* \in \mathbb{S}^{n-1}$ is an eigenvector associated with an eigenvalue $\lambda_*$ :

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

we obtain $\mathcal{T}\mathbf{x}_*^r = \lambda_* \mathcal{B}\mathbf{x}_*^r$ by multiplying $\mathbf{x}_*$ on both sides of the above equation. Since $\mathcal{B}\mathbf{x}_*^r > 0$, we have $\lambda_* = \frac{\mathcal{T}\mathbf{x}_*^r}{\mathcal{B}\mathbf{x}_*^r} = f(\mathbf{x}_*)$. Then, we get $\nabla f(\mathbf{x}_*) = 0$ from (3.3). □

In the following part of this subsection, we introduce a numerical method for calculating a first order stationary point of the maximization model (3.2). The limited memory BFGS (L-BFGS) method is explored to produce a descent direction. Afterwards, we apply a curvilinear search skill to keep iterative points on the unit sphere.

### 3.2.1 L-BFGS to produce a descent direction

The limited memory quasi-Newton approach is efficient in solving large scale nonlinear unconstrained optimization problems. If $c$ is the current iteration, L-BFGS generates an implicit matrix $H_c$ to estimate the inverse of a Hessian of $f(\mathbf{x})$. In the beginning, we review the primary BFGS method. The BFGS approach afresh the estimation of the Hessian's inverse iteratively. Suppose $H_c$ is the current approximation,

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

in which $I$ is an identity matrix,

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

and $\kappa_\epsilon \in (0,1)$ is a small positive parameter. The new estimation $H_c^+$ generated by BFGS method (Nocedal and Wright, 2006; Sun and Yuan, 2006) is

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

In order to finding solution of large scale optimization models, the L-BFGS approach was proposed (Nocedal, 1980), which implements the BFGS update economically. Given any vector $\nabla f \in \mathbb{R}^n$, the cost for the matrix-vector product $-H_c \nabla f$ is only $\mathcal{O}(n)$ multiplications. For the current iteration is $c$, the initial approximation matrix is as simple as

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

where $\gamma_c$ is a positive number determined by the Barzilai-Borwein method (Liu and Nocedal, 1989; Barzilai and Borwein, 1988). Then $H_c^{(\ell)}$ is updated by the BFGS formula (3.6) recursively

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

The approximation of Hessian matrix in iterate $c$ is

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

If $\ell \geqslant c$, the number $\rho_{c-\ell}$ is set to zero and L-BFGS does nothing for that $\ell$. In practice, the L-BFGS works well for a cheap two-loop recursion. The computational cost is about $4Ln$ multiplications.

**Algorithm 1** L-BFGS.

```
q ← $-\nabla f(\mathbf{x}_c)$,
for $i = c-1, c-2, \dots, c-L$ 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 $i = c-L, c-L+1, \dots, c-1$ 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 $\mathbf{p} = -H_c \nabla f(\mathbf{x}_c)$.
```

In terms of the parameter $\gamma_c$, we have three options. The first two are as follows (Barzilai and Borwein, 1988)

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

The third one is determined by the geometric mean of the above two canditates Dai (2014)

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

We set $\gamma_c = 1$ if $\mathbf{y}_c^\top \mathbf{s}_c < \kappa_\epsilon$.

In order to show L-BFGS produces a descent direction $\mathbf{p}_c = -H_c \nabla f(\mathbf{x}_c)$, first we consider the classical BFGS update (3.4)–(3.6) and establish the following two lemmas.

**Lemma 3.1.** *Suppose that $H_c^+$ is generated by BFGS (3.4)–(3.6). Then, we have*

$$\|H_c^+\| \leqslant \|H_c\| \left(1 + \frac{4M}{\kappa_\epsilon}\right)^2 + \frac{4}{\kappa_\epsilon}. \tag{3.12}$$

*Proof.* If $\mathbf{y}_c^\top \mathbf{s}_c < \kappa_\epsilon$, we get $\rho_c = 0$ and $H_c^+ = H_c$. Hence, the inequality (3.12) holds.

Next, we consider the case $\mathbf{y}_c^\top \mathbf{s}_c \geqslant \kappa_\epsilon$. Obviously, $\rho_c \leqslant \frac{1}{\kappa_\epsilon}$. From Lemma 3.7 and all iterates $\mathbf{x}_c \in \mathbb{S}^{n-1}$, we get

$$\|\mathbf{s}_c\| \leqslant 2 \quad \text{and} \quad \|\mathbf{y}_c\| \leqslant 2M. \tag{3.13}$$

Since

$$\|V_c\| \leqslant 1 + \rho_c\|\mathbf{y}_c\|\|\mathbf{s}_c\| \leqslant 1 + \frac{4M}{\kappa_\epsilon} \quad \text{and} \quad \|\rho_c\mathbf{s}_c\mathbf{s}_c^\top\| \leqslant \rho_c\|\mathbf{s}_c\|^2 \leqslant \frac{4}{\kappa_\epsilon},$$

we have

$$\|H_c^+\| \leqslant \|H_c\|\|V_c\|^2 + \|\rho_c\mathbf{s}_c\mathbf{s}_c^\top\| \leqslant \|H_c\|\left(1 + \frac{4M}{\kappa_\epsilon}\right)^2 + \frac{4}{\kappa_\epsilon}.$$

Hence, the inequality (3.12) is valid. □

**Lemma 3.2.** *Suppose that $H_c$ is positive definite and $H_c^+$ is generated by BFGS (3.4)–(3.6). Let $\mu_{\min}(H)$ be the smallest eigenvalue of a symmetric matrix $H$. Then, we get $H_c^+$ is positive definite and*

$$\mu_{\min}(H_c^+) \geqslant \frac{\kappa_\epsilon}{\kappa_\epsilon + 4M^2\|H_c\|}\mu_{\min}(H_c). \tag{3.14}$$

*Proof.* For any unit vector $\mathbf{z}$, we have

$$\mathbf{z}^\top H_c^+\mathbf{z} = (\mathbf{z} - \rho_c\mathbf{s}_c^\top\mathbf{z}\mathbf{y}_c)^\top H_c(\mathbf{z} - \rho_c\mathbf{s}_c^\top\mathbf{z}\mathbf{y}_c) + \rho_c(\mathbf{s}_c^\top\mathbf{z})^2.$$

Let $t \equiv \mathbf{s}_c^\top\mathbf{z}$ and

$$\phi(t) \equiv (\mathbf{z} - t\rho_c\mathbf{y}_c)^\top H_c(\mathbf{z} - t\rho_c\mathbf{y}_c) + \rho_c t^2.$$

Because $H_c$ is positive definite, $\phi(t)$ is convex and attaches its minimum at $t_* = \frac{\rho_c\mathbf{y}_c^\top H_c\mathbf{z}}{\rho_c + \rho_c^2\mathbf{y}_c^\top H_c\mathbf{y}_c}$. Hence,

$$\begin{aligned}
\mathbf{z}^\top H_c^+\mathbf{z} &\geqslant \phi(t_*) \\
&= \mathbf{z}^\top H_c\mathbf{z} - t_*\rho_c\mathbf{y}_c^\top H_c\mathbf{z} \\
&= \frac{\rho_c\mathbf{z}^\top H_c\mathbf{z} + \rho_c^2(\mathbf{y}_c^\top H_c\mathbf{y}_c\mathbf{z}^\top H_c\mathbf{z} - (\mathbf{y}_c^\top H_c\mathbf{z})^2)}{\rho_c + \rho_c^2\mathbf{y}_c^\top H_c\mathbf{y}_c} \\
&\geqslant \frac{\mathbf{z}^\top H_c\mathbf{z}}{1 + \rho_c\mathbf{y}_c^\top H_c\mathbf{y}_c},
\end{aligned}$$

where the last inequality holds because the Cauchy-Schwarz inequality is valid for the positive definite matrix norm $\|\cdot\|_{H_c}$, i.e., $\|\mathbf{y}_c\|_{H_c}\|\mathbf{z}\|_{H_c} \geqslant \mathbf{y}_c^\top H_c\mathbf{z}$. Therefore, $H_c^+$ is also positive definite. From (3.13), it is easy to verify that

$$1 + \rho_c\mathbf{y}_c^\top H_c\mathbf{y}_c \leqslant 1 + \frac{4M^2\|H_c\|}{\kappa_\epsilon}.$$

Therefore, we have $\mathbf{z}^\top H_c^+\mathbf{z} \geqslant \frac{\kappa_\epsilon}{\kappa_\epsilon+4M^2\|H_c\|}\mu_{\min}(H_c)$. Hence, we get the validation of (3.14). □

Second, we turn to L-BFGS. Regardless of the parameter $\gamma_c$ in (3.7) either from (3.10) or (3.11), we get the following lemma.

**Lemma 3.3.** *Suppose that $\gamma_c$ takes Barzilai-Borwein steps (3.10) or its geometric mean (3.11). Then, we have*

$$\frac{\kappa_\epsilon}{4M^2} \leqslant \gamma_c \leqslant \frac{4}{\kappa_\epsilon}. \tag{3.15}$$

*Proof.* If $\mathbf{y}_c^\top\mathbf{s}_c < \kappa_\epsilon$, we get $\gamma_c = 1$ which satisfies the bounds in (3.15) obviously.

Otherwise, we have $\kappa_\epsilon \leqslant \mathbf{y}_c^\top\mathbf{s}_c \leqslant \|\mathbf{y}_c\|\|\mathbf{s}_c\|$. Recalling (3.13), we get

$$\frac{\kappa_\epsilon}{2} \leqslant \|\mathbf{y}_c\| \leqslant 2M \qquad \text{and} \qquad \frac{\kappa_\epsilon}{2M} \leqslant \|\mathbf{s}_c\| \leqslant 2.$$

Hence, we have

$$\frac{\kappa_\epsilon}{4M^2} \leqslant \frac{\mathbf{y}_c^\top\mathbf{s}_c}{\|\mathbf{y}_c\|^2} \leqslant \frac{\|\mathbf{s}_c\|\|\mathbf{y}_c\|}{\|\mathbf{y}_c\|^2} = \frac{\|\mathbf{s}_c\|}{\|\mathbf{y}_c\|} = \frac{\|\mathbf{s}_c\|^2}{\|\mathbf{y}_c\|\|\mathbf{s}_c\|} \leqslant \frac{\|\mathbf{s}_c\|^2}{\mathbf{y}_c^\top\mathbf{s}_c} \leqslant \frac{4}{\kappa_\epsilon},$$

which means that three candidates $\gamma_c^{\text{BB1}}$, $\gamma_c^{\text{BB2}}$, and $\gamma_c^{\text{Dai}}$ satisfy the inequality (3.15). □

Third, based on Lemmas 3.1, 3.2, and 3.3, we obtain two lemmas as follows.

**Lemma 3.4.** *Suppose that the approximation of a Hessian's inverse $H_c$ is generated by L-BFGS (3.7)–(3.9). Then, there exists a positive constant $C_U \geqslant 1$ such that*

$$\|H_c\| \leqslant C_U.$$

*Proof.* From Lemma 3.3 and (3.7), we have $\|H_c^{(0)}\| \leqslant \frac{4}{\kappa_\epsilon}$. Then, by (3.9), (3.8) and Lemma 3.1, we get

$$\begin{aligned}
\|H_c\| &= \|H_c^{(L)}\| \\
&\leqslant \|H_c^{(L-1)}\| \left(1 + \frac{4M}{\kappa_\epsilon}\right)^2 + \frac{4}{\kappa_\epsilon} \\
&\leqslant \cdots \\
&\leqslant \|H_c^{(0)}\| \left(1 + \frac{4M}{\kappa_\epsilon}\right)^{2L} + \frac{4}{\kappa_\epsilon} \sum_{\ell=0}^{L-1} \left(1 + \frac{4M}{\kappa_\epsilon}\right)^{2\ell} \\
&\leqslant \frac{4}{\kappa_\epsilon} \sum_{\ell=0}^{L} \left(1 + \frac{4M}{\kappa_\epsilon}\right)^{2\ell} \equiv C_U.
\end{aligned}$$

The proof is complete. □

**Lemma 3.5.** *Suppose that the approximation of a Hessian's inverse $H_c$ is generated by L-BFGS (3.7)–(3.9). Then, there exists a constant $0 < C_L < 1$ such that*

$$\mu_{\min}(H_c) \geqslant C_L.$$

*Proof.* From Lemma 3.3 and (3.7), we have $\mu_{\min}(H_c^{(0)}) \geqslant \frac{\kappa_\epsilon}{4M^2}$. Moreover, Lemma 3.4 means that $\|H_c^{(\ell)}\| \leqslant C_U$ for all $\ell = 1, \ldots, L$. Hence, Lemma 3.2 implies

$$\mu_{\min}(H_c^{(\ell+1)}) \geqslant \frac{\kappa_\epsilon}{\kappa_\epsilon + 4M^2 C_U} \mu_{\min}(H_c^{(\ell)}).$$

Then, from (3.9) and (3.8), we obtain

$$\begin{aligned}
\mu_{\min}(H_c) &= \mu_{\min}(H_c^{(L)}) \\
&\geqslant \frac{\kappa_\epsilon}{\kappa_\epsilon + 4M^2 C_U}\mu_{\min}(H_c^{(L-1)}) \\
&\geqslant \cdots \\
&\geqslant \left(\frac{\kappa_\epsilon}{\kappa_\epsilon + 4M^2 C_U}\right)^L \mu_{\min}(H_c^{(0)}) \\
&\geqslant \frac{\kappa_\epsilon}{4M^2}\left(\frac{\kappa_\epsilon}{\kappa_\epsilon + 4M^2 C_U}\right)^L \equiv C_L.
\end{aligned}$$

We complete the proof. □

Finally, we get the following theorem from Lemmas 3.4 and 3.5.

**Theorem 3.2.** *Suppose that* $\mathbf{p}_c = -H_c \nabla f(\mathbf{x}_c)$ *is generated by L-BFGS. Then, there exist constants* $0 < C_L \leqslant 1 \leqslant C_U$ *such that*

$$\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \leqslant -C_L \|\nabla f(\mathbf{x}_c)\|^2 \qquad \textit{and} \qquad \|\mathbf{p}_c\| \leqslant C_U \|\nabla f(\mathbf{x}_c)\|. \tag{3.16}$$

### 3.2.2 Cayley transform to satisfy orthogonal constraints

Cayley transform (Golub and Van Loan, 2013) provides an efficient way to create orthogonal matrices, which is a useful tool for solving eigenvalue problem (Chen et al., 2016c; Friedland et al., 1987) and optimization problem with orthogonal constraints (Jiang and Dai, 2015; Wen and Yin, 2013). Suppose that $W \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix. Then $(I + W)$ is invertible, and we get an orthogonal matrix $Q$ by the Cayley transform,

$$Q = (I + W)^{-1}(I - W) \in \mathbb{R}^{n\times n}. \tag{3.17}$$

The eigenvalues of matrix $Q$ do not contain $-1$.

Let $\mathbf{x}_c \in \mathbb{S}^{n-1}$ be the current iterate, $\mathbf{p}_c \in \mathbb{R}^n$ be a descent direction produced by Algorithm 1 and $\alpha$ be a damping factor. In order to keep the new iterative point on the unit sphere, we want to find a suitable orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and obtain the next point via

$$\mathbf{x}_{c+1} = Q\mathbf{x}_c. \tag{3.18}$$

Also, the iterate $\mathbf{x}_{c+1}$ should be a descent condition, i.e.,

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

It can be deduced from the Cayley transform (3.17) that $(I + W)\mathbf{x}_{c+1} = (I - W)\mathbf{x}_c$ and $\mathbf{x}_{c+1} - \mathbf{x}_c = -W(\mathbf{x}_{c+1} + \mathbf{x}_c)$. To keep the skew-symmetric matrix $W$ simple, we construct it by

$$W = \mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top, \tag{3.19}$$

in which $\mathbf{a}$ and $\mathbf{b}$ are undetermined vectors. Further, we obtain

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

Since the function $f(\mathbf{x})$ in (3.2) is zero-order homogeneous, we have

$$\mathbf{x}^\top \nabla f(\mathbf{x}) = 0, \qquad \forall\ \mathbf{x} \neq \mathbf{0}. \tag{3.20}$$

Motivated by the equation above, we select $\mathbf{a}$ as $\mathbf{x}_c$. Further, we have

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

from (3.18). Since the matrix $Q$ is orthogonal and the number $-1$ is not an eigenvalue of $Q$, we get $\mathbf{x}_c^\top Q\mathbf{x}_c + \mathbf{x}_c^\top \mathbf{x}_c > 0$ for any $\mathbf{x}_c \in \mathbb{S}^{n-1}$. When set $\mathbf{b}$ equals $\alpha\mathbf{p}_c$, the desired descent property $\nabla f(\mathbf{x}_c)^\top(\mathbf{x}_{c+1} - \mathbf{x}_c) < 0$ is achieved from (3.16).

In actual calculation, we do not need to construct matrices $W$ and $Q$ explicitly. The new point $\mathbf{x}_{c+1}$ can be directly created via a formula based on $\mathbf{x}_c$ and $\mathbf{p}_c$ with the multiplications being about $4n$.

**Lemma 3.6.** *Since the new iterate* $\mathbf{x}_{c+1}$ *is produced by* (3.17), (3.18), *and* (3.19), *we can reformulate it as*

$$\mathbf{x}_{c+1}(\alpha) = \frac{[(1-\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2 - \|\alpha\mathbf{p}_c\|^2]\mathbf{x}_c + 2\alpha\mathbf{p}_c}{1+\|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}. \tag{3.21}$$

*Moreover we have*

$$\|\mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c\| = 2\left(\frac{\|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}{1+\|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}\right)^{\frac{1}{2}}. \tag{3.22}$$

*Proof.* First we review the the Sherman-Morrison-Woodbury formula:

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

where the matrix $A$ is invertible. Based on this formula we have

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

in which $1+\|\alpha\mathbf{p}_c\|^2 - (\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2 \geqslant 1$ because $|\alpha\mathbf{x}_c^\top\mathbf{p}_c| \leqslant \|\alpha\mathbf{p}_c\|$ and $\mathbf{x}_c \in \mathbb{S}^{n-1}$. Then, we have

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

The equation (3.21) is then obtained. Hence, we have

$$\begin{aligned}
&\|\mathbf{x}_{c+1}(\alpha)-\mathbf{x}_c\|^2\\
&= \left\|\frac{[2\alpha\mathbf{x}_c^\top\mathbf{p}_c(\alpha\mathbf{x}_c^\top\mathbf{p}_c-1)-2\|\alpha\mathbf{p}_c\|^2]\mathbf{x}_c+2\alpha\mathbf{p}_c}{1+\|\alpha\mathbf{p}_c\|^2-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}\right\|^2\\
&= \frac{[2\alpha\mathbf{x}_c^\top\mathbf{p}_c(\alpha\mathbf{x}_c^\top\mathbf{p}_c-1)-2\|\alpha\mathbf{p}_c\|^2][2\alpha\mathbf{x}_c^\top\mathbf{p}_c(\alpha\mathbf{x}_c^\top\mathbf{p}_c+1)-2\|\alpha\mathbf{p}_c\|^2]+4\|\alpha\mathbf{p}_c\|^2}{[1+\|\alpha\mathbf{p}_c\|^2-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2]^2}\\
&= \frac{(2\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2[(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2-1]-2\|\alpha\mathbf{p}_c\|^2(2\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2+4\|\alpha\mathbf{p}_c\|^4+4\|\alpha\mathbf{p}_c\|^2}{[1+\|\alpha\mathbf{p}_c\|^2-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2]^2}\\
&= \frac{(2\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2[(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2-1-\|\alpha\mathbf{p}_c\|^2]+4\|\alpha\mathbf{p}_c\|^2[-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2+\|\alpha\mathbf{p}_c\|^2+1]}{[1+\|\alpha\mathbf{p}_c\|^2-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2]^2}\\
&= \frac{4\|\alpha\mathbf{p}_c\|^2-4(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}{1+\|\alpha\mathbf{p}_c\|^2-(\alpha\mathbf{x}_c^\top\mathbf{p}_c)^2}.
\end{aligned}$$

Then the equality (3.22) holds. □

A similar update scheme as (3.21) was given in Jiang and Dai (2015). 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 $\mathbf{x}_c$ is $\mathbf{q}_c\equiv(I-\mathbf{x}_c\mathbf{x}_c^\top)\mathbf{p}_c$. Based on $\mathbf{q}_c$ we have

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

Therefore, we say that the update equation (3.21) is a retraction on the unit sphere (Wen and Yin, 2013). In fact, the new iterate $\mathbf{x}_{c+1}(\alpha)$ is a geodesic rooted at $\mathbf{x}_c$ along the descent direction (Absil et al., 2008). In terms of the damping parameter $\alpha$, it is given via the following inexact line search.

**Theorem 3.3.** *Assume that the gradient-related direction $\mathbf{p}_c$ satisfies (3.16) and $\mathbf{x}_{c+1}(\alpha)$ is given by (3.21). If $\eta\in(0,1)$ and $\nabla f(\mathbf{x}_c)\neq 0$, then we can find a constant $\tilde{\alpha}_c>0$ such that for all $\alpha\in(0,\tilde{\alpha}_c]$,*

$$f(\mathbf{x}_{c+1}(\alpha))\leqslant f(\mathbf{x}_c)+\eta\alpha\mathbf{p}_c^\top\nabla f(\mathbf{x}_c). \tag{3.23}$$

*Proof.* By the formula (3.21), we get $\mathbf{x}_{c+1}(0) = \mathbf{x}_c$ and $\mathbf{x}'_{c+1}(0) = -2\mathbf{x}_c^\top \mathbf{p}_c \mathbf{x}_c + 2\mathbf{p}_c$. Then with the aid of (3.20) we obtain

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

Since $\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \leqslant -C_L \|\nabla f(\mathbf{x}_c)\|^2$ and $\|\mathbf{p}_c\| \leqslant C_U \|\nabla f(\mathbf{x}_c)\|$, we have $\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) < 0$ from $\nabla f(\mathbf{x}_c) \neq 0$. If $\alpha$ is sufficiently small, we have

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

by Taylor's theorem. Since $\eta < 2$, then we could find a positive $\tilde{\alpha}_c$ satisfying (3.23).

□

Broadly speaking, our new algorithm CEST is a modified version of the L-BFGS method for orthogonal constrained optimization problem. The Cayley transform is employed to keep the iterative points satisfying the spherical constraint. Then we use an inexact line search to choose an appropriate damping factor. Theorem 3.3 ensures the availability of the inexact line search. Finally, We demonstrate our new method CEST explicitly, which means computing eigenvalues of sparse tensors, in Algorithm 2.

## 3.3 Convergence analysis

We demonstrate the convergent property of the CEST algorithm by three steps. First, we show that the sequence of objective function values $\{f(\mathbf{x}_c)\}$ is convergent and any cluster point of the sequence $\{\mathbf{x}_c\}$ is a first-order stationary point of the objective function. Second, we prove the convergence of iterate sequence $\{\mathbf{x}_c\}$ based on the Łojasiewicz inequality. Third, it is illustrated that our CEST method is able to get the largest or smallest eigenvalue of a tensor with great possibility when it is tested from a large number of initial points.

**Algorithm 2** Computing eigenvalues of sparse tensors(CEST).

```
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 3.4, 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 1.
    Choose the smallest nonnegative integer $\ell$ such that $\alpha = \beta^\ell$ satisfies (3.23).
    Let $\alpha_c = \beta^\ell$ and update the new iterate $\mathbf{x}_{c+1} = \mathbf{x}_{c+1}(\alpha_c)$ by (3.21).
    Compute $\mathbf{s}_c, \mathbf{y}_c$ and $\rho_c$ by (3.4) and (3.5) respectively.
    $c \leftarrow c + 1$.
end while
```

(Chang et al., 2016)

### 3.3.1 Convergence of sequences of function values and gradients

If the CEST method terminates in finite number of iterations, there exists an integer $c$ satisfying $\nabla f(\mathbf{x}_c) = 0$. Then from $f(\mathbf{x}_c)$ Theorem 3.1, we know $f(\mathbf{x}_c)$ is an eigenvalue with $\mathbf{x}_c$ being its associated eigenvector. Hence in our convergence analysis part, the sequence $\{\mathbf{x}_c\}$ created by the CEST algorithm is supposed to be infinite. Due to the positive definite property of tensor $\mathcal{B}$, the merit function $f(\mathbf{x})$ is twice continuously differentiable. Because the spherical domain of $f(\mathbf{x})$ is compactness, we get the following result.

**Lemma 3.7.** *For the objective function value $f(x)$, we can find a positive number $M > 1$ satisfying*

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

Further, since the sequence $\{f(\mathbf{x}_c)\}$ decreases monotonically, it converges.

**Theorem 3.4.** *Suppose the sequence of function values $\{f(\mathbf{x}_c)\}$ produced by the CEST algorithm is infinite. Then we have*

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

*where $\lambda_*$ is a constant.*

The next lemma shows that the damping factors $\alpha_c$ are bounded.

**Lemma 3.8.** *Suppose $\alpha_c$ is generated by the inexact line search in the CEST algorithm. Then we have*

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

*where $\alpha_{\min} > 0$ is a constant.*

*Proof.* Assume $0 < \alpha \leqslant \hat{\alpha} \equiv \frac{(2-\eta)C_L}{(2+\eta)MC_U^2}$. Then, $\alpha C_U M \leqslant \frac{(2-\eta)C_L}{(2+\eta)C_U} < 1$. By inequalities (3.16) and Lemma 3.7, we get

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

and

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

Further we obtain

$$\begin{aligned}
& 2\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + 2M(\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) - \eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c)(1 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) \\
= \;& (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2M - \eta \alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c))(\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2) \\
< \;& (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2+\eta) M \alpha^2 C_U^2 \|\nabla f(\mathbf{x}_c)\|^2 \\
\leqslant \;& (2-\eta)\alpha \mathbf{p}_c^\top \nabla f(\mathbf{x}_c) + (2-\eta) C_L \alpha \|\nabla f(\mathbf{x}_c)\|^2 \\
\leqslant \;& 0,
\end{aligned} \tag{3.24}$$

in which the last inequality is deduced from (3.16). Based on Lemma 3.7, Lemma

3.6, the equality (3.20), and the mean value theorem we have

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

in which the last inequality is given by (3.24). Therefore, the inequalities $1 \geqslant \alpha_c \geqslant \beta\hat{\alpha} \equiv \alpha_{\min}$ hold according to the rule of the inexact search. □

Next we show that each cluster point of the sequence $\{\mathbf{x}_c\}$ is the first-order stationary point.

**Theorem 3.5.** *The infinite sequence of iterates $\{\mathbf{x}_c\}$ generated by the CEST algorithm satisfies*

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

*Proof.* By (3.23) and (3.16), we obtain

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geqslant -\eta\alpha_c\mathbf{p}_c^\top\nabla f(\mathbf{x}_c) \geqslant \eta\alpha_c C_L\|\nabla f(\mathbf{x}_c)\|^2. \tag{3.25}$$

Further, we have

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

from Lemma 3.7 and Lemma 3.8. Then

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

and the reslut is acquired. □

### 3.3.2 Convergence of sequence of iterates

Lojasiewicz (1963) proposed the famous Łojasiewicz inequality for real-analytic functions. Further, Absil et al. (2005) showed that, if the objective function of an optimization model satisfies the Łojasiewicz inequality, the iterates given by a significant class of approaches for this optimization problem converge to a unique limit point without other hypothesis. The Łojasiewicz inequality was extended from real-analytic function to nonsmooth functions in Bolte et al. (2007) afterwards. Lately, many scholars have used the the Łojasiewicz inequality to study proximal algorithms for nonconvex and nonsmooth optimization (Attouch et al., 2010; Xu and Yin, 2013). The graph of a function $f(\mathbf{x})$ is

$$\operatorname{Graph} f := \{(\mathbf{x}, \lambda) \in \mathbb{R}^n \times \mathbb{R} : f(\mathbf{x}) = \lambda\}.$$

Then the graph of the objective function $f(\mathbf{x}) = \frac{\mathcal{T}\mathbf{x}^r}{\mathcal{B}\mathbf{x}^r}$ is

$$\operatorname{Graph} f = \{(\mathbf{x}, \lambda) \in \mathbb{R}^n \times \mathbb{R} \,:\, \mathcal{T}\mathbf{x}^r - \lambda\mathcal{B}\mathbf{x}^r = 0\}.$$

It can be seen that the graph is a semialgebraic set. Then the merit function $f(\mathbf{x})$ enjoys the semialgebraic property, and satisfies the following Łojasiewicz inequality (Absil et al., 2005; Bolte et al., 2007).

**Theorem 3.6** (The Łojasiewicz inequality)**.** *Suppose $f(\mathbf{x})$ is a real semialgebraic function with a closed domain $\operatorname{dom} f$, and $f|_{\operatorname{dom} f}$ is continuous. Let $\mathbf{x}_* \in \operatorname{dom} f$. Then in some neighborhood $\mathscr{U}$ of $\mathbf{x}_*$*

$$|f(\mathbf{x}) - f(\mathbf{x}_*)|^\theta \leqslant C_K \|\nabla f(\mathbf{x})\|, \tag{3.26}$$

*where $\theta \in [0, 1)$ and $C_K$ is a positive constant.*

The next theorem shows that the sequence of $\{\mathbf{x}_c\}$ converges to a unique limit point.

**Theorem 3.7.** *Let $\{\mathbf{x}_c\}$ be an infinite sequence of iterates generated by the CEST algorithm. Then we have*

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

*where $\mathbf{x}_* \in \mathbb{S}^{n-1}$ is unique.*

*Proof.* If $f(\mathbf{x})$ in (3.2) satisfies the Łojasiewicz inequality , the primary descent condition, i.e.,

$$f(\mathbf{x}_c) - f(\mathbf{x}_{c+1}) \geqslant C_P \|\nabla f(\mathbf{x}_c)\| \|\mathbf{x}_c - \mathbf{x}_{c+1}\|, \tag{3.27}$$

where $C_P$ is a positive constant and the complementary descent condition, i.e.,

$$[f(\mathbf{x}_{c+1}) = f(\mathbf{x}_c)] \quad \Rightarrow \quad [\mathbf{x}_{c+1} = \mathbf{x}_c], \tag{3.28}$$

then the reslut holds from Theorem 3.2 in Absil et al. (2005). Since the Łojasiewicz inequality of $f(\mathbf{x})$ is proved in Theorem 3.26 we only need to prove (3.27) and (3.28).

From (3.22) and (3.16), we obtain

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

Then we have

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

from (3.25). Therefore, when $C_P = \frac{\eta C_L}{2C_U}$ the inequality (3.27) is obtained.

We prove the complementary descent condition (3.28) by contradiction. Suppose $\mathbf{x}_{c+1} \neq \mathbf{x}_c$, then $\|\nabla f(\mathbf{x})\| \neq 0$. Otherwise, the CEST algorithm terminates in a finite number of iterations. Based on Lemma 3.8 and (3.25), we get

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

Therefore $f(\mathbf{x}_{c+1}) \neq f(\mathbf{x}_c)$. □

Hence, the iterative sequence $\{\mathbf{x}_c\}$ converges to a first-order stationary point $\mathbf{x}_*$ from Theorem 3.5. In order to estimate convergence rate of the CEST method, we first introduce the next result.

**Lemma 3.9.** *For the iterative sequence $\{\mathbf{x}_c\}$ generated by the CEST method, we have*

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \geqslant C_m \|\nabla f(\mathbf{x}_c)\|, \tag{3.29}$$

*where $C_m$ is a positive constant .*

*Proof.* Denote $\langle \mathbf{a}, \mathbf{b} \rangle$

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

as the angle between nonzero vectors $\mathbf{a}$ and $\mathbf{b}$. Then, for any vector $\mathbf{a} \neq \mathbf{0}, \mathbf{b} \neq \mathbf{0}$, and $\mathbf{c} \neq \mathbf{0}$, the following inequality

$$\langle \mathbf{a}, \mathbf{b} \rangle \leqslant \langle \mathbf{a}, \mathbf{c} \rangle + \langle \mathbf{c}, \mathbf{b} \rangle \tag{3.30}$$

holds, and $\langle \cdot, \cdot \rangle$ is a metric in a unit sphere. Based on (3.30), we obtain

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

From (3.20) we get $\langle \mathbf{x}_c, -\nabla f(\mathbf{x}_c) \rangle = \frac{\pi}{2}$. Hence, we have

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

Then we deduce that

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

in which the last inequality is given from (3.16). Since $\mathbf{x}_c \in \mathbb{S}^{n-1}$, we obtain

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

from (3.22). Based on the inequalities $\alpha_{\min} \leqslant \alpha_c \leqslant 1$ and $\|\mathbf{p}_c\| \leqslant C_U \|\nabla f(\mathbf{x}_c)\| \leqslant C_U M$, we have

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

Then it can be deduced from (3.16) that

$$\|\mathbf{p}_c\| \|\nabla f(\mathbf{x}_c)\| \geqslant -\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \geqslant C_L \|\nabla f(\mathbf{x}_c)\|^2.$$

Therefore, $\|\mathbf{p}_c\| \geqslant C_L \|\nabla f(\mathbf{x}_c)\|$ and the inequality (3.29) is acquired by setting $C_m \equiv \frac{2\alpha_{\min} C_L^2}{C_U(1+C_U M)}$. □

Then we have the following theorem based on Lemma 3.9. For the proof of this theorem, please refer to Theorem 2 in Attouch and Bolte (2009) and Theorem 7 in Chen et al. (2016c)

**Theorem 3.8.** *Let $\mathbf{x}_*$ be the stationary point of an infinite sequence of iterates $\{\mathbf{x}_c\}$ produce by the CEST algorithm.*

- *If $\theta \in (0, \frac{1}{2}]$, we have*

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

  *where $\gamma > 0$ and $\varrho \in (0, 1)$ are constants.*

- *If $\theta \in (\frac{1}{2}, 1)$, we have*

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

  *where $\gamma > 0$ is a constant.*

If the level set of $f(\mathbf{x})$ is convex and the second-order sufficient condition is satisfied at $\mathbf{x}_*$, the L-BFGS method is proved to be linearly convergent by Liu and Nocedal (1989). In fact, without assumption of the level set being convex , when the second-order sufficient condition holds, the exponent $\theta$ in the Łojasiewicz inequality (3.26) is equal to $\frac{1}{2}$, and the sequence $\{\mathbf{x}_c\}$ converges linearly from Theorem 3.8.

### 3.3.3 Probability of getting the extreme eigenvalue

In order to obtain the smallest eigenvalue of a tensor related to a uniform hypergraph, we run the CEST algorithm from a number of random initial points, and take the smallest objective function value as the smallest eigenvalue of the corresponding tensor in practical experiments. The next theorem give the probability of this strategy in getting the extreme eigenvalue.

**Theorem 3.9.** *If we run the CEST algorithm $N$ times from $N$ initial points which are sampled from $\mathbb{S}^{n-1}$ uniformly and take the smallest cost function value as the smallest eigenvalue, then the probability of getting the smallest eigenvalue is*

$$1 - (1 - \varphi)^N, \tag{3.31}$$

*in which $\varphi \in (0, 1]$ is a constant. Hence, the smallest eigenvalue can be acquired with a high probability when the number $N$ is large enough,.*

*Proof.* Let $\mathbf{x}^* \in \mathbb{S}^{n-1}$ be an eigenvector related to the smallest eigenvalue and $\mathscr{U}$ be a neighborhood of $\mathbf{x}^*$ as defined in Theorem 3.6. Then from the proof of Theorem 3.2 in Absil et al. (2005), we can find a constant $\rho > 0$ such that the initial iterate $\mathbf{x}_1$ belongs to the set $\mathscr{V}(\mathbf{x}^*) \equiv \{\mathbf{x} \in \mathbb{S}^{n-1} : \|\mathbf{x} - \mathbf{x}^*\| < \rho\} \subseteq \mathscr{U}$, and the iterative sequence $\{\mathbf{x}_c\}$ converges to $\mathbf{x}^*$. Given an initial point $\mathbf{x}_1$ which is sampled from $\mathbb{S}^{n-1}$ uniformly, next we consider the probability of the event that $\mathbf{x}_1 \in \mathscr{V}(\mathbf{x}^*)$.

Suppose $A$ and $S$ are hypervolumes of $(n-1)$-dimensional solids $\mathscr{V}(\mathbf{x}^*)$ and $\mathbb{S}^{n-1}$ respectively.[1] Therefore, the " area" of the surface of $\mathscr{V}(\mathbf{x}^*) \subseteq \mathbb{S}^{n-1}$ and $\mathbb{S}^{n-1}$ in $\mathbb{R}^n$ are $A$ and $S$ respectively, and $0 < A \leqslant S$. For this geometric probability model, the probability of $\mathbf{x}_1$ belonging to $\mathscr{V}(\mathbf{x}^*)$ is

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

[1] 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.

In fact, the smallest eigenvalue can be obtained once $\{\mathbf{x}_c\} \cap \mathscr{V}(\mathbf{x}^*) \neq \varnothing$.

If we run the the CEST algorithm $N$ times, the complementary event of the joint event is that the CEST starts from $N$ random initial points and does not touch the the eigenvector $\mathbf{x}^*$. The probability of this complementary event is $(1-\varphi)$. If the complementary event does not occur, we obtain the eigenvector $\mathbf{x}^*$ associated with the smallest eigenvalue. Therefore, the probability of getting the smallest eigenvalue is $1-(1-\varphi)^N$. □

In terms of the largest eigenvalue of a tensor $\mathcal{T}$, what we need do is changing the objective function $f(\mathbf{x})$ in (3.2) into

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

The results of computing largest eigenvalue by the CEST method can be deduced in a similar way as the corresponding ones in the smallest eigenvalue occasion.

## 3.4 Fast computational methods for product of vector and hypergraph related tensors

For computational problems related to adjacency, Laplacian and signless Laplacian tensor, a popular operation is tensor and vector product, which is always time consuming. On the other hand, such kind of tensors are usually sparse. For example, the adjacency tensor, Laplacian tensor, signless Laplacian tensor of the 4-uniform hypergraph in Figure 2.1 are usually have only 0.72%, 0.76%, and 0.76% nonzero elements respectively. Therefore, it is rational that we take advantage of the sparsity to propose a fast computational method for product of vector and hypergraph related tensors (Chang et al., 2016).

### 3.4.1 The way to store a uniform hypergraph economically

By saving the indices of vertices in each edge in a row of a matrix, we get an incidence matrix $G_r \in \mathbb{R}^{m\times r}$ to store the information of an $r$-graph $G = (V, E)$, which has $m$ edges. The order of an element in its corresponding row, as well as the location of a row in the matrix is insignificant because we could permute the order of elements in each row, or the location of a row in the matrix without changing the information of the hypergraph. Take the 4-graph in Figure 2.1 as an example. Its incidence matrix is:

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

In order to save storage space, we replace the sparse incidence matrix by a compact one as follows

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

It can be seen that, the quantity of columns in $G_r$ is greatly reduced when comparing with $G_i$, due to the fact that the condition $k \ll n$ holds in most cases. We call the matrix $G_r$ memory matrix of $G$.

### 3.4.2 Computing products of a tensor and a vector

Given an $r$-graph hypergraph $G$, we first obtain the incidence matrix $G_i$ and the memory matrix $G_r$ by the above mentioned approach. In practice, we only need to compute the product of $\mathcal{D}$ and a vector, and the product of $\mathcal{A}$ and a vector, because the Laplacian (Signless Laplacian) tensor equals $\mathcal{D}$ minus (plus) $\mathcal{A}$.

```
Gi=sparse(repmat((1:m)',r,1),Gr,1,m,n);

d =full(sum(Gi,1)');
```

Figure 3.1: Matlab codes producing $\mathcal{D}$ and $\mathbf{d}$.

Let us first study the product of the degree tensor $\mathcal{D}$ and a vector. Tensor $\mathcal{D}$ is diagonal with its $i$th diagonal entry being the degree of a vertex $i$, i.e., $d(i)$, for $i \in V$. We obtain the degree $d(i)$ by summarizing the $i$th column of $G_i$. The MATLAB codes in Figure 3.1 create a sparse matrix $G_i$, as well as the degree vector $\mathbf{d} \equiv [d(i)] \in \mathbb{R}^n$. Since the degree vector $\mathbf{d}$ is fixed for a given hypergraph, we save it from the start. Denote "$*$" as the component-wise Hadamard product. For a given vector $\mathbf{x} \in \mathbb{R}^n$, the computations of

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

are straightforward.

Next, we deal with the product of adjacency tensor $\mathcal{A}$ and vector $\mathbf{x}$. We combine the hypergraph and the vector information by constructing a matrix $X_{mat} = [x_{(G_r)_{\ell j}}]$ which has the same size as $G_r$. If the $(G_r)_{\ell j}$ equals $i$, then we set $(X_{mat})_{\ell j}$ as $x_i$. The product $\mathcal{A}\mathbf{x}^k$ is then rewritten as

$$\mathcal{A}\mathbf{x}^{r} = r \sum_{\ell=1}^{m} \prod_{j=1}^{r} (X_{mat})_{\ell j}. \tag{3.32}$$

In order to compute the vector $\mathcal{A}\mathbf{x}^{r-1}$, we represent the $i$th element of $\mathcal{A}\mathbf{x}^{r-1}$ by

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

where $i = 1, \ldots, n$. Using MATLAB codes in Figure 3.2, we construct a sparse matrix $M_j = [\delta(i, (G_r)_{\ell j})] \in \mathbb{R}^{n \times m}$ and a column vector $\mathbf{y}_j = [\prod_{s \neq j} (X_{mat})_{\ell s}] \in \mathbb{R}^m$

```
Mj=sparse(Gr(:,j),(1:m)',1,n,m);

yj=prod(Xmat(:,[1:j-1,j+1:r]),2);
```

Figure 3.2: Matlab codes producing $M_j$ and $\mathbf{y}_j$.

respectively for each $j = 1, \ldots, r$. Then, we can compute the vector

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

via a simple loop.

It costs about $mr^2$, $mr^2+nr$, and $mr^2+nr$ multiplications to compute products of tensors $\mathcal{A}$ , $\mathcal{L}$, and $\mathcal{Q}$ with any vector $\mathbf{x}$ respectively. Since $mr^2 < mr^2 + nr \leqslant 2mr^2$, it is cheap to employ our approach to compute products of $\mathcal{A}$ , $\mathcal{L}$, or $\mathcal{Q}$ and a vector, which are large scale sparse tensors arising from hypergraphs. Besides, the methods given above can be employed by parallel computing easily.

## 3.5 Numerical results

The experiments are compiled by using Matlab, while parameters involved are set as follows:

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

The termination criterion for the CEST algorithm is

$$\|\nabla f(\mathbf{x}_c)\|_\infty < 10^{-6} \tag{3.33}$$

or

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\|_\infty < 10^{-8} \quad \text{and} \quad \frac{|f(\mathbf{x}_{c+1}) - f(\mathbf{x}_c)|}{1 + |f(\mathbf{x}_c)|} < 10^{-16}. \tag{3.34}$$

When the number of iterations exceeds 5000, it is also stopped.

In this section, we compare the following four methods.

```
GradObj:on, LargeScale:off, TolX:1.e-8,

TolFun:1.e-16, MaxIter:5000,Display:*off
```

Figure 3.3: Settings of `fminunc`.

- Two adaptive shifted power methods (Kolda and Mayo, 2011, 2014) (Power M.), which can be obtained from Tensor Toolbox version 2.6 as `eig_sshopm` and `eig_geap` for Z- and H-eigenvalues of symmetric tensors respectively.

- An unconstrained optimization method (Han's UOA) (Han, 2013). The optimization model is solved by `fminunc` in Matlab whose settings are shown in Figure 3.3. Because this approach does not limit the iterative points on $\mathbb{S}^{n-1}$, its tolerance parameters are not the same as other methods.

- CESTde: An immature version of our CEST approach without using the fast computational technique proposed in Section 3.4.

- CEST: The approach introduced in this chapter.

In terms of problems involved in our numerical experiments, we run the related algorithms one hundred times from one hundred random initial points sampled from a unit sphere $\mathbb{S}^{n-1}$. By creating an $n$-dimensional vector with a standard Gaussian distribution and normalizing this vector, we obtain an initial point on the unit sphere. The one hundred initial points are generated in this way independently, which are also uniformly distributed on the unit sphere. Hence, we get one hundred estimated eigenvalues $\lambda_1, \ldots, \lambda_{100}$. Suppose $\lambda^*$ is the exact extreme eigenvalue of the related tensor, the accuracy rate of the method is counted by

$$\text{Accu.} \equiv \left| \left\{ i \, : \, \frac{|\lambda_i - \lambda^*|}{1 + |\lambda^*|} \leqslant 10^{-8} \right\} \right| \times 1\%. \tag{3.35}$$

By employing the global tactics in Section 3.3.3, the largest (smallest) result among the one hundred results is taken as the largest (smallest) eigenvalue computed by

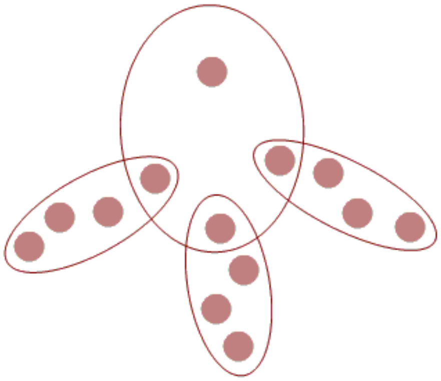


Figure 3.4: A 4-uniform squid: $G_S^4$.

the corresponding method. The CPU time presented in the following experiments are the summation of the one hundred runs in each case.

### 3.5.1 Eigenvalues of small-scale hypergraphs

In this subsection, we emphasis on computation of extreme eigenvalues of symmetric tensors arising from small sacle uniform hypergraphs.

**Squid.** For an $r$-uniform hypergraph $G_S^r = (V, E)$, if it has $(r^2 - r + 1)$ vertices with edge set composed by legs $\{i_{1,1}, \dots, i_{1,r}\}, \dots, \{i_{r-1,1}, \dots, i_{r-1,r}\}$ and a head $\{i_{1,1}, \dots,$

$i_{r-1,1}, i_r\}$, we call it a squid. For example, the hypergraph illustrated in Figure 3.4 is a 4-uniform squid $G_S^4$. An even order squid $G_S^r$ is connected and odd-bipartite. Then the largest and smallest H-eigenvalue of the adjacency tensor of $G_S^r$ satisfy

$$\lambda_{\min}^H(\mathcal{A}(G_S^r)) = -\lambda_{\max}^H(\mathcal{A}(G_S^r))$$

from of Theorem 2.1(5). Due to the nonnegative and weakly irreducible property of the adjacency tensor $\mathcal{A}(G_S^r)$, we can obtain the largest H-eigenvalue $\lambda_{\max}^H(\mathcal{A}(G_S^r))$ via the Ng-Qi-Zhou algorithm. In terms of the smallest H-eigenvalue of $\mathcal{A}(G_S^r)$, the following tests are implemented. The parameter $L$ is suggested by Nocedal to lie between 3 and 7 for the L-BFGS algorithm to get a good performance. Therefore, we

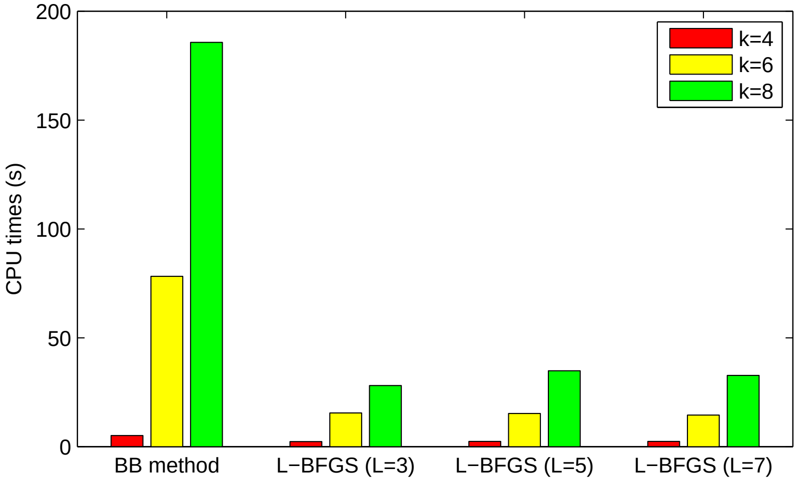


Figure 3.5: CPU time for computing smallest H-eigenvalue of $\mathcal{A}(G_S^k)$ by L-BFGS.

Table 3.1: Results for computing $\lambda_{\min}^H(\mathcal{A}(G_S^4))$.

| 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% |

compare L-BFGS for $L$ being $0, 3, 5, 7$. In fact, when $L = 0$ it becomes the Barzilai-Borwein method. For the parameter $\gamma_c$, we use $\gamma_c^{BB1}$, $\gamma_c^{BB2}$, and $\gamma_c^{Dai}$ randomly. We calculate the smallest H-eigenvalues of the adjacency tensors of $r$-uniform squids with $r$ being $4, 6, 8$. In Figure 3.5, we show the CPU time for one hundred runs. It can be seen that the L-BFGS method is approximately five times faster than the Barzilai-Borwein method. In our CEST method, the parameter $L$ is set to be 5 according to Nocedal's setting[2].

Next, we compute the smallest H-eignvalues of adjacency tensors of the $G_S^4$. See Figure 3.4. We obtain $\lambda_{\max}^H(\mathcal{A}(G_S^r)) = 1.3320$ by the Ng-Qi-Zhou algorithm as a reference. The results of four algorithms: Power M., Han's UOA, CESTde, and

[2] See `http://users.iems.northwestern.edu/ nocedal/lbfgs.html`.

CEST are presented in Table 3.1. It can be seen that, all algorithms get the smallest H-eigenvalue of $\mathcal{A}(G_S^4)$ with probability 1. The CESTde method cost 22% and 37% CPU time of Power M.'s and Han's UOA's CPU time respectively. Further, when the fast computational skill is explored, the CEST algorithm is forty times faster than the power method. Due to limitation of laptop's memory, a dense adjacency tensor for 6- and 8-uniform squid are unable to be stored. Therefore, we only report computational results of the the CEST method in Table 3.2.

Table 3.2: Performance of CEST computing the smallest H-eigenvalue of an adjacency tensor $\mathcal{A}(G_S^r)$.

| $r$ | $n$ | $\lambda^H_{\max}(\mathcal{A}(G_S^r))$ | $\lambda^H_{\min}(\mathcal{A}(G_S^r))$ | Iter. | time(s) | True Est. |
|---|---|---|---|---|---|---|
| 2 | 3 | 1.4142135624 | −1.4142135624 | 844 | 0.66 | 100% |
| 4 | 13 | 1.3320029867 | −1.3320029867 | 5291 | 2.43 | 100% |
| 6 | 31 | 1.2640653288 | −1.2640653288 | 26443 | 15.10 | 78% |
| 8 | 57 | 1.2202288301 | −1.2202288301 | 45993 | 37.19 | 9% |

**Blowing up the Petersen graph.** An ordinary graph $G_P$ called the Petersen graph is demonstrated in Figure 3.6. The Petersen graph is non-bipartite. The smallest eigenvalue of its signless Laplacian matrix being one. By blowing up each vertex of $G_P$ into an $r$-set, we get a $2r$-uniform hypergraph $G_P^{2r,r}$, which contains $10r$ vertices and 15 edges. From Theorem 2.2, the smallest H-eigenvalue of the signless Laplacian tensor $\mathcal{Q}(G_P^{2r,r})$ is the same as the smallest eigenvalue of the signless Laplacian matrix $\mathcal{Q}(G_P)$, i.e., $\lambda^H_{\min}(\mathcal{Q}(G_P^{2r,r})) = \lambda^H_{\min}(\mathcal{Q}(G_P)) = 1$.

The performances of four algorithms for computing the smallest H-eigenvalue of $\mathcal{Q}(G_P^{4,2})$ are reported in Table 3.3, where the method Han's UOA misses the smallest H-eigenvalue of $\mathcal{Q}(G_P^{4,2})$[3] and the other three methods touch the exact solution with a high probability. The CESTde method uses as much as 88% CPU time less than Power M. method. When the sparse structure of $\mathcal{Q}(G_P^{4,2})$ is explored, the CEST algorithm costs no more than 1% CPU time of CESTde. In Table 3.4, we show

[3] (*) means a failure.

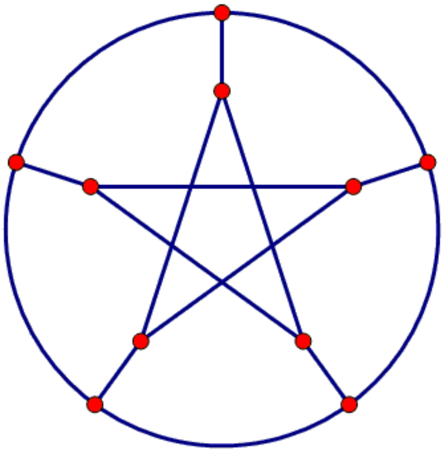


Figure 3.6: The Petersen graph $G_P$.

Table 3.3: Results for computing $\lambda^H_{\min}(\mathcal{Q}(G_P^{4,2}))$.

| Algorithms | $\lambda^H_{\min}(\mathcal{Q}(G_P^{4,2}))$ | 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 3.4: Accuracy rate of calculating $\lambda^H_{\min}(\mathcal{Q}(G_P^{2k,k}))$ by CEST.

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

the accuracy rates of the CEST method for finding the smallest H-eigenvalues of the signless Laplacian tensors of $2r$-uniform hypergraphs $G_P^{2r,r}$ with $r = 1, \ldots, 10$ respectively. The accuracy rate reduces along with the increase of the order $r$ of the hypergraph.

**Grid.** Given a square, we regard it as a 4-graph with 4 vertices and 1 edge. By subdividing a square $s$ times for $s \geqslant 0$, we get a grid, which is denoted as a 4-graph $G_G^s$ with $(2^s+1)^2$ vertices and $4^s$ edges. When the subdividing order $s$ is zero, the

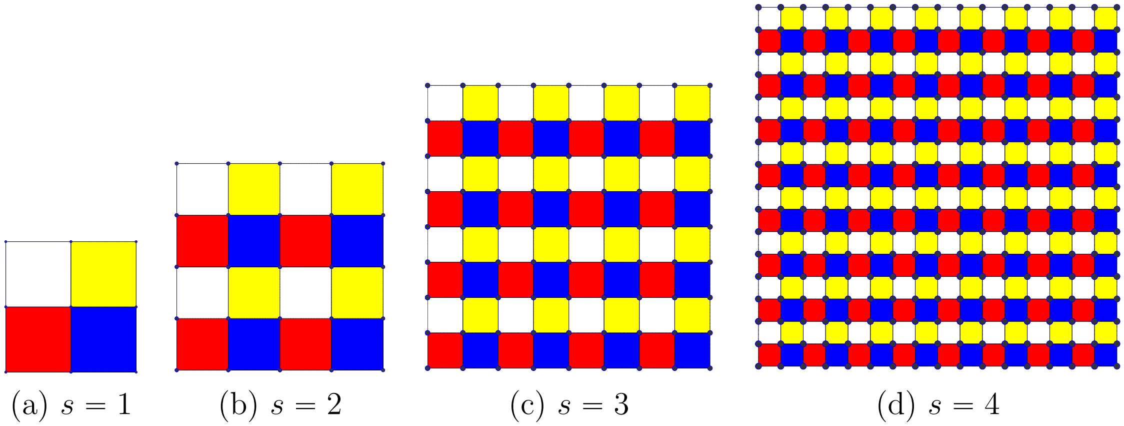

(a) $s = 1$ (b) $s = 2$ (c) $s = 3$ (d) $s = 4$

Figure 3.7: Some 4-uniform grid hypergraphs.

Table 3.5: Results of computing $\lambda^H_{\max}(\mathcal{L}(G^2_G))$.

| Algorithms | $\lambda^H_{\max}(\mathcal{L}(G^2_G))$ | 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% |

grid $G^0_G$ is in fact a square. We can also obtain $G^s_G$ by subdividing each edge of $G^{s-1}_G$ into four edges. For the $s = 1, 2, 3, 4$ order subdivision, please refer to the pictures presented in Figure 3.7. Since the hypergraph $G^2_G$ is connected and odd-bipartite, from Theorem 2.1(2) we have $\lambda^H_{\max}(\mathcal{L}(G^2_G)) = \lambda^H_{\max}(\mathcal{Q}(G^2_G))$. As a reference we give the solution $\lambda^H_{\max}(\mathcal{Q}(G^2_G)) = 6.5754$ obtained from the Ng-Qi-Zhou method. The results of computing $\lambda^H_{\max}(\mathcal{L}(G^2_G))$ by four methods: Power M., Han's UOA, CESTde, and CEST, are presented in Table 3.5. Each algorithm acquires $\lambda^H_{\max}(\mathcal{L}(G^2_G))$ with probability 100%. The Han's UOA and CESTde methods consume about 25% and 30% the CPU time of the Power M. method respectively, while the CEST method saves about 95% the CPU time of the Power Method. Furthermore, we employ the CEST method to compute $\lambda^H_{\max}$ of the grids in Figure 3.7. The results are shown in Table 3.6.

Table 3.6: Results of computing $\lambda^H_{\max}(\mathcal{L}(G^s_G))$.

| $s$ | $n$ | $m$ | $\lambda^H_{\max}(\mathcal{L}(G^s_G))$ | 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% |

### 3.5.2 Eigenvalues of large-scale hypergraphs

In this subsection, we list examples of the CEST method computing eigenvalues arising from large scale even-uniform hypergraphs. With respect to the large scale of the problems, it is rational that we slightly enlarge the the tolerance parameters and multiply the constants in (3.33) and (3.34) by $\sqrt{n}$ .

**Sunflower.** We employ the CEST method to compute $\lambda^H_{\max}$ of Laplacian tensors of even-uniform sunflowers. The relative error between our numerical result and the exact solution is computed by

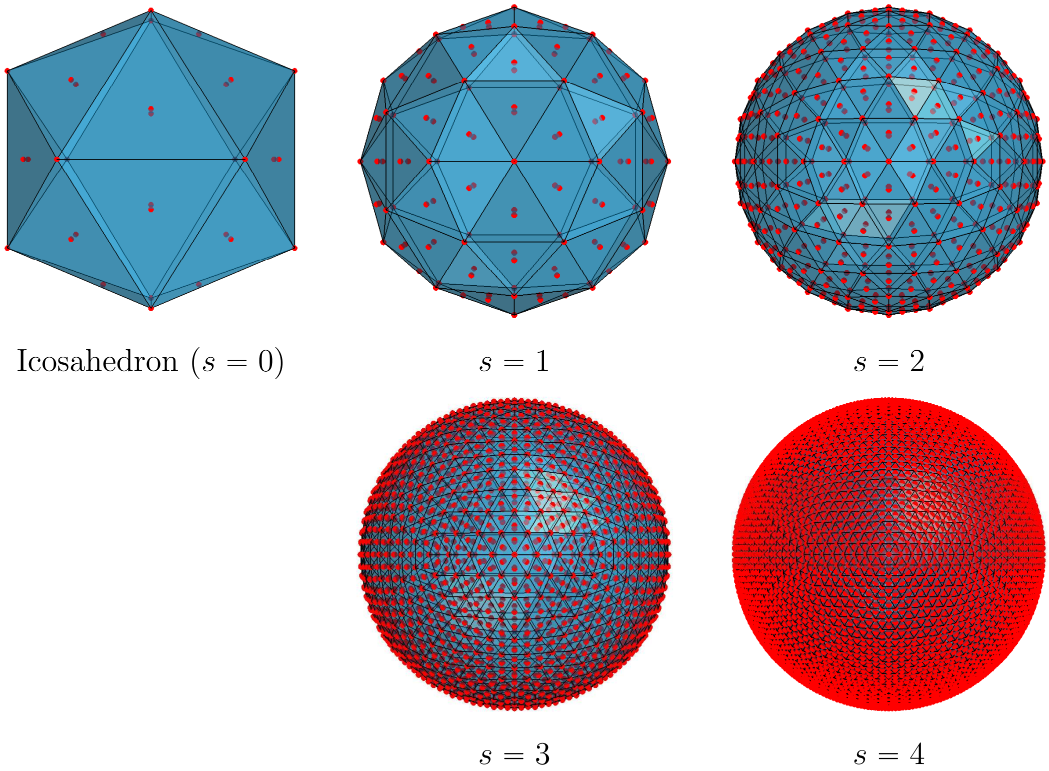
$$\mathrm{RE} = \frac{|\lambda^H_{\max}(\mathcal{L}(G_S)) - \lambda^*_H|}{\lambda^*_H},$$

in which $\lambda^*_H$ is given in Theorem 2.3. We present the detailed numerical results in Table 3.7. It can be seen that the CEST method find the largest H-eigenvalues of Laplacian tensors with high accuracy, and all relative errors are of magnitude less than $\mathcal{O}(10^{-10})$. The total CPU time of one hundred runs in each test is at most 78 minutes.

**Icosahedron.** An icosahedron has twelve vertices and twenty faces. As shown in Figure 3.8, we can approximate a unit sphere by subdividing an icosahedron as many times as possible. If we consider the three vertices of the triangle together with its center as an edge of a hypergraph, the $s$-order subdivision of an icosahedron is a 4-graph, named $G^s_I$, containing $20 \times 4^s$ edges.

Table 3.7: Results of computing $\lambda^H_{\max}(\mathcal{L}(G_S))$ by CEST.

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



Icosahedron ($s = 0$)

$s = 1$

$s = 2$

$s = 3$

$s = 4$

Figure 3.8: 4-uniform hypergraphs: subdivision of an icosahedron.

Table 3.8: Computing $\lambda^Z_{\max}(\mathcal{L}(G^s_I))$ and $\lambda^Z_{\max}(\mathcal{Q}(G^s_I))$ by CEST.

| | | | $\mathcal{L}(G^s_I)$ | | | $\mathcal{Q}(G^s_I)$ | | |
|---|---|---|---|---|---|---|---|---|
| $s$ | $n$ | $m$ | $\lambda^Z_{\max}$ | 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 |

The performance of the CEST method for computing the largest Z-eigenvalues of both Laplacian tensors and signless Laplacian tensors of the hypergraph $G^s_I$ is shown in Table 3.8. It can be seen that the CEST method costs at most twenty-one minutes for the one hundred runs in each test, even if dimensions reaches to almost two millions. Since the 4-graph $G^s_I$ is connected and odd bipartite, we have $\lambda^Z_{\max}(\mathcal{L}(G^s_I)) = \lambda^Z_{\max}(\mathcal{Q}(G^s_I))$ from the conclusion in Theorem 2.1(3), which is consistent with our numerical results. Besides, Bu et al. (2016) proved that

$$\lambda^Z_{\max}(\mathcal{L}(G^s_I)) = \lambda^Z_{\max}(\mathcal{Q}(G^s_I)) = \Delta \tag{3.36}$$

for $r$-uniform sunflowers when $3 \leqslant r \leqslant 2\Delta$. The results in our numerical experiments imply that the equation in (3.36) also holds for 4-graphs $G^s_I$. Moreover, whether the equality (3.36) is suitable for any connected odd-bipartite uniform hypergraph is still open.

## 3.6 Conclusion

In this section, we introduce an effective first-order optimization algorithm CEST to calculate the largest or smallest H- and Z-eigenvalues of sparse tensors corresponding to large scale uniform hypergraphs. Based on the algebraic nature of these tensors, the Łojasiewicz inequality is employed to prove the convergent property of the sequence of iterative points produced by the CEST method. Furthermore, we show that our method is able to obtain the the extreme eigenvalue of a symmetric tensor with a high probability, and compute the eigenvalues of tensors involved in hypergraphs with millions of vertices

By exploring the sparsity of tensors arising from a uniform hypergraph, a useful fast computational framework for products of a vector and these tensors is constructed. With the aid of this technique, we can store a large scale hypergraph economically, and improve the efficiency of a related algorithm greatly.

# Chapter 4

# Computing $p$-spectral radius of a uniform hypergraph with applications

In this section we introduce a first-order conjugate gradient based method (CSRH) to calculate the $p$-spectral radii of uniform hypergraphs. We also analyse the convergent property of the CSRH algorithm, and give the probability of the CSRH method getting the global optimization solution. Finally, we apply the $p$-spectral radius model, as well as the CSRH method in network analysis.

## 4.1 Introduction

Keevash et al. (2014) proposed the concept of $p$-spectral radius, which has important application in the extremal hypergraph theory. In 1941, Turán (1941) gave the famous Turán graph and Turán theorem, and it was regarded as the start of the extremal graph theory. In the extremal graph theory, people study the extremal problems about graph parameters such as order, size and girth. These problems attract a lot of attention and are widely studied in combinatorics. The Turán-type problems were generalized from graph to hypergraph that is to find the largest number of edges in a hypergraph which is $F$-free[1](Turán, 1961). In spite of the adequate study of Turán-type problems in graph field, it becomes much more difficult to deal with the Turán-type problems related to hypergraphs. A spectral Turán-type inequality was introduced, and extended the Turán theorem in Nikiforov (2007). Keevash et al. (2014) proposed the $p$-spectral format of Nikiforov's inequality, which is useful in resolving 'degenerate' Turán-type problems. Moreover, Nikiforov (2013) showed that the edge extremal problems asymptotically equals the extremal $p$-spectral radius problems.

The $p$-spectral radius of a hypergraph contains other concepts, for example, Lagrangian, and the spectral radius of a hypergraph (Lu and Man, 2016b). Moreover, the number of edges in the extremal problems is relevant to the $p$-spectral radii of hypergraphs. If the parameter $p$ equals one, the $p$-spectral radii of hypergraphs are, in fact, the Lagrangians of the corresponding hypergraphs. Motzkin and Straus (1965) introduced the Lagrangians of graph and hypergraph, which can be employed to give proof of Turán's theorem for a graph. The Lagrangian of a hypergraph was applied to disproving a conjecture in Erdös and Stone (1946); Frankl and Rödl (1984), and also to determine the non-jumping numbers for hypergraphs (Frankl et al., 2007; Peng, 2008; Peng and Zhao, 2008). Also, the Lagrangian of a hypergraph is helpful in determining Turán densities of hypergraphs (Brown and Simonovits, 1984; Keevash, 2011; Mubayi, 2006; Sidorenko, 1987), which is an asymptotic solution of the (non-degenerate) Turán problem. If the parameter $p$ is two, the $p$-spectral radius of a uniform hypergraph is related to the largest Z-eigenvalue (Qi, 2005a) of the adjacency tensor. If the parameter $p$ is even and equals the order of a hypergraph, the $p$-spectral radius is connected with the largest H-eigenvalue of the adjacency tensor. Hence, the $p$-spectral radii of hypergraphs are linked with the spectral radii

[1] A uniform hypergraph that does not have a subgraph isomorphic to the uniform hypergraph $F$ is said to be $F$-free.

of hypergrahs (Hu and Qi, 2015; Li et al., 2016; Lu and Man, 2016b). More theoretical results about $p$-spectral radii of hypergraphs are given in Kang et al. (2015); Nikiforov (2014).

In addition to the application in the extremal hypergraph theory, $p$-spectral radii could also evaluate the significance of items in networks. Quantifying the importance and rifeness of items is a hot topic in data mining, which is applied to rank web pages (Page, 1999; Kolda et al., 2005; Ding et al., 2003), estimate customer behaviours (Krohn-Grimberghe et al., 2012), and retrieve images (Huang et al., 2010), etc. We call elements of the vector related to the $p$-spectral radius of a hypergraph the $p$-optimal weighting, which reflect the importance of vertices. The sorted results of objects change over the parameter $p$. In the numerical experiment part of this chapter, we will give explanation on different ranking results. Moreover, we demonstrate consequences of ranking large amount of real-life data via the method proposed in this chapter.

Considering the computation of the $p$-spectral radius of a hypergraph, the methods for computing tensor eigenvalues, such as the shifted symmetric higher-order power method (Power M.) (Kolda and Mayo, 2011), the generalized eigenproblem adaptive power (GEAP) method (Kolda and Mayo, 2014), the Ng-Qi-Zhou method (Ng et al., 2009), and the CEST method can be utilized when $p = 2$ or $p$ is the order of an even uniform hypergraph, in which case the $p$-spectral radius is the Z- or H-eigenvalue of the adjacency tensor. With respect to positive $p$, when $p$ is an odd number or a fraction, the problem of computing $p$-spectral radii of a uniform hypergraph is still open to the best of our knowledge.

We translate the the original $p$-spectral radius problem into a spherically constrained optimization model. By employing a conjugate gradient method, we get an ascent direction based on the current iteration. In order to keep the unit length of each iterative point, the Cayley transform is applied to projecting the ascent direction onto the unit sphere. In addition, we illustrate that a positive constant can be found to satisfy Wolfe conditions in the line search. By using these strategies, an algorithm for Computing $p$-Spectral Radii of Hypergraphs (CSRH) with $p > 1$ is obtained. Under situation of $p = 1$, the CSRH method can estimate the Lagrangians (1-spectral radii) of hypergraphs. Moreover, the CSRH algorithm is proved to be convergent, and can find the global optimization solution with high probability. For calculating Z-eigenvalues and H-eigenvalues of adjacency tensors, the numerical tests indicate that the CSRH method is predominant when compared to existing approaches. Even if the dimensions of the hypergraph reaches to two millions, the CSRH method performs well in experiments. Besides, we discover the relations between the importance of hypergraph vertices and the sequence of entries of $p$-optimal weighting. By constructing a hypergraph model and sorting the vertices of the hypergraph, the CSRH method is further employed in network analysis. Given a small scale weighted hypergraph, the CSRH method calculates orders of the vertices from different point of view. According to the publication information of 10305 authors in real world, we construct the corresponding hypergraph model, and rank the authors from individual and corporate angle respectively. We give reasonable explanations for the ranking results, which are also consistent with the results in Ng et al. (2011).

If the elements in a set is repetitive, then this set is called a multiset. If a hypergraph contains at least one edge which is a multiset, this hypergraph is called a multi-hypergraph (Pearson and Zhang, 2014). The $p$-spectral radii problem can be generalized from hypergraph to muli-hypergraph. As a matter of convenience, we solve the $p$-spectral radii problems based on hypergraphs with no edge being multiset. However, all computational skills, as well as theoretical results, in the following part of this chapter is appropriate for $p$-spectral radii of multi-hypergraphs.

## 4.2 The CSRH method

For the $p$-spectral radius problem in (2.4), we reformulate it to a maximization problem with orthogonal constraint, and introduce an iterative algorithm to solve it.

### 4.2.1 A model with spherical constraint

By using the adjacency tensor of $G$, $\lambda^{(p)}(G)$ in (2.4) can be expressed as

$$\lambda^{(p)}(G) = \max_{\|\mathbf{x}\|_p=1} (r-1)!\mathcal{A}\mathbf{x}^r, \tag{4.1}$$

which equals the unconstrained format

$$\lambda^{(p)}(G) = \max_{\mathbf{x}\neq 0}(r-1)!\frac{\mathcal{A}\mathbf{x}^r}{\|\mathbf{x}\|_p^r}. \tag{4.2}$$

Then the $p$-spectral radius of $G$ in (2.4) is finally reformulated as

$$\lambda^{(p)}(G) = \begin{cases} \max f(\mathbf{x}) = (r-1)!\dfrac{\mathcal{A}\mathbf{x}^r}{\|\mathbf{x}\|_p^r} \\ \\ \text{ s.t. } \|\mathbf{x}\|_2 = 1. \end{cases} \tag{4.3}$$

If the parameter $p$ is greater than one and the vector $\mathbf{x}$ is nonzero, the merit function $f(\mathbf{x})$ is differentiable with its gradient being

$$\nabla f(\mathbf{x}) = \frac{r!}{\|\mathbf{x}\|_p^r}\left(\mathcal{A}\mathbf{x}^{r-1} - \mathcal{A}\mathbf{x}^r\|\mathbf{x}\|_p^{-p}\mathbf{x}^{\langle p-1\rangle}\right), \tag{4.4}$$

in which the vector $\mathbf{x}^{\langle p-1\rangle}$ is defined as

$$(\mathbf{x}^{\langle p-1\rangle})_i = |x_i|^{p-1}\text{sgn}(x_i), \quad \text{for} \quad i = 1, \ldots, n.$$

Further, due to the zero-order homogeneous property of $f(\mathbf{x})$, we obtain

$$\mathbf{x}^\top \nabla f(\mathbf{x}) = 0, \quad \forall\ 0 \neq \mathbf{x} \in \mathbb{R}^n. \tag{4.5}$$

If we consider $\lambda^{(p)}(G)$ as a function of $p$, the next proposition shows that $\lambda^{(p)}(G)$ is continuous in the variable $p$. Therefore, it enables us to estimate the $p$-spectral radius of a hypergraph, especially the Lagrangian (1-spectral radius) of a hypergraph, when we can not calculate it directly.

**Proposition 4.1.** *Assume $\{p_\vartheta\}$ is an infinite sequence satisfying*

$$\lim_{\vartheta\to\infty} p_\vartheta = p_*, \tag{4.6}$$

*where $p_\vartheta > 1$. Then we get*

$$\lim_{\vartheta\to\infty} \lambda^{(p_\vartheta)}(G) = \lambda^{(p_*)}(G). \tag{4.7}$$

*Proof.* Let $\hat{f}(\mathbf{x}, p)$ be a function of both $\mathbf{x} \in \mathbb{S}^{n-1}$ and $p > 1$ as follows

$$\hat{f}(\mathbf{x}, p) = (r-1)!\frac{\mathcal{A}\mathbf{x}^r}{\|\mathbf{x}\|_p^r} \qquad (\mathbf{x}, p) \in \mathbb{S}^{n-1} \times (1, +\infty).$$

Then $\hat{f}(\mathbf{x}, p)$ is continuous in $\mathbf{x}$ and $p$. For any $p_\vartheta \in \{p_\vartheta\}$, there exists at least an $\mathbf{x}^*_\vartheta$ such that

$$\lambda^{(p_\vartheta)}(G) = \hat{f}(\mathbf{x}^*_\vartheta, p_\vartheta). \tag{4.8}$$

Since $\{p_\vartheta\}$ is infinite and $\{\mathbf{x}^*_\vartheta\}$ is bounded, we have

$$\lim_{\vartheta\to\infty} \mathbf{x}^*_\vartheta = \mathbf{x}^*_0 \tag{4.9}$$

without loss of generality. From (4.3), we have

$$\hat{f}(\tilde{\mathbf{x}}, p_\vartheta) \leqslant \hat{f}(\mathbf{x}^*_\vartheta, p_\vartheta), \quad \forall \;\; \tilde{\mathbf{x}} \in \mathbb{S}^{n-1}. \tag{4.10}$$

Further we obtain

$$\lim_{\vartheta\to\infty} \hat{f}(\tilde{\mathbf{x}}, p_\vartheta) \leqslant \lim_{\vartheta\to\infty} \hat{f}(\mathbf{x}^*_\vartheta, p_\vartheta) \;\; \forall \;\; \tilde{\mathbf{x}} \in \mathbb{S}^{n-1}.$$

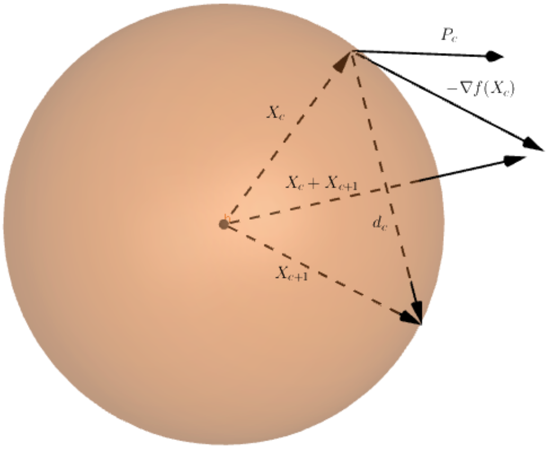


Figure 4.1: Process of searching a new point on the unit sphere.

Since $\hat{f}(\mathbf{x}, p)$ is continuous, we get

$$\hat{f}(\tilde{\mathbf{x}}, p_*) \leqslant \hat{f}(\mathbf{x}_0^*, p_*) \ \ \forall \ \ \tilde{\mathbf{x}} \in \mathbb{S}^{n-1}, \tag{4.11}$$

which means

$$\hat{f}(\mathbf{x}_0^*, p_*) = \max_{\mathbf{x} \in \mathbb{S}^{n-1}} \hat{f}(\mathbf{x}, p_*) = \lambda^{(p_*)}(G). \tag{4.12}$$

On the other hand, from (4.8) we obtain

$$\lim_{\vartheta \to \infty} \lambda^{(p_\vartheta)}(G) = \lim_{\vartheta \to \infty} \hat{f}(\mathbf{x}_\vartheta^*, p_\vartheta) = \hat{f}(\mathbf{x}_0^*, p_*). \tag{4.13}$$

Based on (4.12) and (4.13), the equation (4.7) is available. □

### 4.2.2 An algorithm for solving the model

In this subsection, we aim to design an iterative algorithm for the maximization problem (4.3). Let $\mathbf{x}_c$ be the current iterate. Then we want to seek a new feasible iterate $\mathbf{x}_{c+1}$ such that

1. $\mathbf{x}_{c+1}$ is a unit vector;

2. $\mathbf{d}_c = \mathbf{x}_{c+1} - \mathbf{x}_c$ is an ascent direction, i.e.,

$$\mathbf{d}_c^\top \nabla f(\mathbf{x}_c) > 0. \tag{4.14}$$

Consider the iteration on the unit sphere in Figure 4.1.

First, based on a given unit vector $\mathbf{x}_c$, the vector $\mathbf{x}_{c+1}$ is still on the unit sphere if and only if the vector $\mathbf{x}_{c+1} + \mathbf{x}_c$ is perpendicular to the vector $\mathbf{d}_c = \mathbf{x}_{c+1} - \mathbf{x}_c$, i.e.

$$(\mathbf{x}_{c+1} + \mathbf{x}_c)^\top \mathbf{d}_c = 0. \tag{4.15}$$

Given a skew-symmetric matrix $W_c = -W_c^\top$, we have

$$(\mathbf{x}_c + \mathbf{x}_{c+1})^\top W_c(\mathbf{x}_{c+1} + \mathbf{x}_c) = -(\mathbf{x}_{c+1} + \mathbf{x}_c)^\top W_c(\mathbf{x}_{c+1} + \mathbf{x}_c) = 0,$$

which implies that when

$$\mathbf{d}_c = W_c(\mathbf{x}_c + \mathbf{x}_{c+1}), \tag{4.16}$$

equation (4.15) and the first condition of $\mathbf{x}_{c+1}$ are satisfied. Next, to meet the second requirement, with the aid of optimization techniques we first construct an ascent direction $\mathbf{p}_c$ such that

$$\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) > 0. \tag{4.17}$$

Since from (4.5) we have $\mathbf{x}_c^\top \nabla f(\mathbf{x}_c) = 0$, then we set the direction $\mathbf{d}_c$ as a linear combination of $\mathbf{x}_c$ and $\mathbf{p}_c$ :

$$\mathbf{d}_c = a\mathbf{x}_c + b\mathbf{p}_c, \tag{4.18}$$

and get

$$\mathbf{d}_c^\top \nabla f(\mathbf{x}_c) = a\mathbf{x}_c^\top \nabla f(\mathbf{x}_c) + b\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) = b\mathbf{p}_c^\top \nabla f(\mathbf{x}_c). \tag{4.19}$$

Once the constant $b$ in (4.18) is positive, $\mathbf{d}_c$ remains an ascent direction from (4.17) and (4.19).

In brief, if $\mathbf{d}_c$ satisfies (4.16) and (4.18) with $b > 0$ the two criterions for the new iterative point are achieved. Then by (4.16) and (4.18), we generate the skew-symmetric matrix as

$$W_c = \frac{1}{2}\alpha(\mathbf{p}_c\mathbf{x}_c^\top - \mathbf{x}_c\mathbf{p}_c^\top) \in \mathbb{R}^{n\times n}, \tag{4.20}$$

in which $\alpha$ is a positive number. The equality (4.16) and (4.20) indicates that in (4.18)

$$b = \frac{1}{2}\alpha \mathbf{x}_c^\top(\mathbf{x}_c + \mathbf{x}_{c+1}) = \frac{1}{2}\alpha(1 + \mathbf{x}_c^T \mathbf{x}_{c+1}) \geqslant 0.$$

Here we claims that the constant $b$ is positive. Because if $b = 0$, we have $\mathbf{x}_{c+1} = -\mathbf{x}_c$. By substituting $\mathbf{x}_{c+1}$ for $-\mathbf{x}_c$ in (4.16) and from the equation

$$\mathbf{d}_c = \mathbf{x}_{c+1} - \mathbf{x}_c,$$

we obtain $\mathbf{d}_c = 0$ and $\mathbf{d}_c = -2\mathbf{x}_c$ which contradicts the fact $\mathbf{x}_c^\top \mathbf{x}_c = 1$. Therefore, when the skew symmetric matrix follows the form in (4.20) and $\mathbf{p}_c$ is an ascent direction, the equations (4.16) and (4.18) hold with $b > 0$, which means $\mathbf{x}_{c+1}$ is a feasible point and $\mathbf{d}_c$ is an ascent direction.

**Lemma 4.1.** *By (4.16) and (4.20), the new iterate $\mathbf{x}_{c+1}$ is reformulated as*

$$\mathbf{x}_{c+1}(\alpha) = \frac{[(2 - \alpha \mathbf{x}_c^\top \mathbf{p}_c)^2 - \|\alpha \mathbf{p}_c\|^2]\mathbf{x}_c + 4\alpha \mathbf{p}_c}{4 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2}, \tag{4.21}$$

*Moreover, we get*

$$\|\mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c\| = 2\left(\frac{\|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_k^\top \mathbf{p}_c)^2}{4 + \|\alpha \mathbf{p}_c\|^2 - (\alpha \mathbf{x}_c^\top \mathbf{p}_c)^2}\right)^{\frac{1}{2}}. \tag{4.22}$$

*Proof.* From (4.16), we obtain

$$\mathbf{x}_{c+1} = (I - W_c)^{-1}(I + W_c)\mathbf{x}_c.$$

Then the orthogonal transformation is in fact based on a Cayley transform to the skew-symmetric matric $W$. We omit the proof since it is similar to Lemma 3.6. □

To find the new feasible point $\mathbf{x}_{c+1}$ in (4.21), an important step to be dealt with is generating a direction $\mathbf{p}_c$, which is expected to be an ascent direction, i.e., satisfying

(4.17). A hypergraph or a tensor probably is large scale and it is usually time-consuming if we compute a problem involving a hypergraph or a tensor in practice. On the other hand, the nonlinear conjugate gradient approach is devised for solving large problems. Therefore, we adopt the nonlinear conjugate gradient approach to get an appropriate $\mathbf{p}_c$. Suppose the current iterate is $\mathbf{x}_c$, a new direction $\mathbf{p}_c$ based on $\mathbf{x}_c$ and the vector $\mathbf{d}_{c-1} = \mathbf{x}_c - \mathbf{x}_{c-1}$ under a conjugate gradient framework is computed by

$$\mathbf{p}_c = \nabla f(\mathbf{x}_c) + \beta_{c-1}\mathbf{d}_{c-1}. \tag{4.23}$$

An effective nonlinear conjugate gradient method called CG_DESCENT was proposed and proved to have a good descent property in Hager and Zhang (2005, 2006). With the aid of the formula of parameter $\beta_c$ in CG_DESCENT, we update the parameter $\beta_{c-1}$ by $\beta_{c-1} = \max(0, \tilde{\beta}_{c-1})$, where

$$\tilde{\beta}_{c-1} = \begin{cases} \left(\tau\mathbf{d}_{c-1}\dfrac{\|\mathbf{y}_{c-1}\|^2}{\mathbf{d}_{c-1}^\top\mathbf{y}_{c-1}} - \mathbf{y}_{c-1}\right)^\top \dfrac{\nabla f(\mathbf{x}_c)}{\mathbf{d}_{c-1}^\top\mathbf{y}_{c-1}} & \text{if } |\mathbf{d}_{c-1}^\top\mathbf{y}_{c-1}| \geqslant \epsilon\|\mathbf{d}_{c-1}\|\|\mathbf{y}_{c-1}\| \\ 0 & \text{otherwise,} \end{cases} \tag{4.24}$$

$\mathbf{y}_{c-1} = \nabla f(\mathbf{x}_c) - \nabla f(\mathbf{x}_{c-1})$, $\tau > 1/4$ and $\epsilon > 0$. We choose the gradient direction as the initial direction, i.e., $\mathbf{p}_0 = \nabla f(\mathbf{x}_0)$. The following lemma shows that the direction $\mathbf{p}_c$ generated via (4.23) and (4.24) satisfies the sufficient ascent condition.

**Lemma 4.2.** *If $\mathbf{p}_c$ is a search direction given by (4.23) and (4.24), then we have*

$$\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) \geqslant \left(1 - \frac{1}{4\tau}\right)\|\nabla f(\mathbf{x}_c)\|^2. \tag{4.25}$$

*Further, there exists a number $M_0 > 1$ such that*

$$\|\mathbf{p}_c\| \leqslant M_0\|\nabla f(\mathbf{x}_c)\|. \tag{4.26}$$

*Proof.* For the trivial case of $\beta_{c-1} = 0$, the two inequalities hold immediately. Next we consider the condition of $\beta_{c-1} \neq 0$. From (4.23) and (4.24) we obtain

$$\begin{aligned}
\mathbf{p}_c^\top \nabla f(\mathbf{x}_c) &= \|\nabla f(\mathbf{x}_c)\|^2 + \tau \frac{\mathbf{d}_{c-1}^\top \nabla f(\mathbf{x}_c)}{\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1}} \frac{\|\mathbf{y}_{c-1}\|^2}{\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1}} \mathbf{d}_{c-1}^\top \nabla f(\mathbf{x}_c) - \frac{\mathbf{y}_{c-1}^\top \nabla f(\mathbf{x}_c)}{\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1}} \mathbf{d}_{c-1}^\top \nabla f(\mathbf{x}_c) \\
&= \frac{1}{4\tau} \|\nabla f(\mathbf{x}_c)\|^2 - \frac{\mathbf{d}_{c-1}^\top \nabla f(\mathbf{x}_c)}{\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1}} \mathbf{y}_{c-1}^\top \nabla f(\mathbf{x}_c) + \tau \frac{(\mathbf{d}_{c-1}^\top \nabla f(\mathbf{x}_c))^2}{(\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1})^2} \|\mathbf{y}_{c-1}\|^2 \\
&\quad + \left(1 - \frac{1}{4\tau}\right) \|\nabla f(\mathbf{x}_c)\|^2 \\
&\geqslant \left(1 - \frac{1}{4\tau}\right) \|\nabla f(\mathbf{x}_c)\|^2.
\end{aligned}$$

It is derived from

$$\|\mathbf{d}_{c-1} \cdot \mathbf{y}_{c-1}^\top\| = \|\mathbf{d}_{c-1}\| \cdot \|\mathbf{y}_{c-1}\| \quad \text{and} \quad \|\mathbf{d}_{c-1} \cdot \mathbf{d}_{c-1}^\top\| = \|\mathbf{d}_{c-1}\|^2,$$

that

$$\begin{aligned}
\|\beta_{c-1}\mathbf{d}_{c-1}\| &\leqslant \left\| \frac{\tau \|\mathbf{y}_{c-1}\|^2 \cdot \mathbf{d}_{c-1} \cdot \mathbf{d}_{c-1}^\top - \mathbf{d}_{c-1}^\top \mathbf{y}_{c-1} \cdot \mathbf{d}_{c-1} \cdot \mathbf{y}_{c-1}^\top}{(\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1})^2} \right\| \cdot \|\nabla f(\mathbf{x}_c)\| \\
&\leqslant \left[ \frac{\|\mathbf{d}_{c-1}\| \|\mathbf{y}_{c-1}\|}{|\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1}|} + \frac{\tau \|\mathbf{y}_{c-1}\|^2 \|\mathbf{d}_{c-1}\|^2}{(\mathbf{d}_{c-1}^\top \mathbf{y}_{c-1})^2} \right] \cdot \|\nabla f(\mathbf{x}_c)\| \\
&\leqslant \left[ \frac{1}{\epsilon} + \frac{\tau}{\epsilon^2} \right] \|\nabla f(\mathbf{x}_c)\|.
\end{aligned}$$

Then we get

$$\|\mathbf{p}_c\| \leqslant \|\nabla f(\mathbf{x}_c)\| + \|\beta_{c-1}\mathbf{d}_{c-1}\| \leqslant \left[1 + \frac{1}{\epsilon} + \frac{\tau}{\epsilon^2}\right] \|\nabla f(\mathbf{x}_c)\|.$$

By taking $M_0 = 1 + \frac{1}{\epsilon} + \frac{\tau}{\epsilon^2}$, the inequality (4.26) is obtained. □

### 4.2.3 An inexact curvilinear search

In this subsection, it is illustrated that we can find a proper $\alpha_c$ for the inexact curvilinear search such that the analogous Wolfe conditions hold.

**Lemma 4.3.** *Suppose $f'(\alpha)$ is the derivative of $f(\mathbf{x}_{c+1}(\alpha))$ at point $\alpha$. Then we get*

$$\alpha f'(\alpha) = -\nabla f(\mathbf{x}_{c+1}(\alpha))^\top \mathbf{x}_c. \tag{4.27}$$

*Proof.* From (4.21) we have

$$[4 + \alpha^2\|\mathbf{p}_c\|^2 - \alpha^2(\mathbf{x}_c^\top \mathbf{p}_c)^2]\mathbf{x}_{c+1}(\alpha) = [(2 - \alpha\mathbf{x}_c^\top \mathbf{p}_c)^2 - \alpha^2\|\mathbf{p}_c\|^2]\mathbf{x}_c + 4\alpha\mathbf{p}_c.$$

By taking derivative with respect to $\alpha$, we obtain

$$\begin{aligned}&2\alpha(\|\mathbf{p}_c\|^2 - (\mathbf{x}_c^\top \mathbf{p}_c)^2)\mathbf{x}_{c+1}(\alpha) + [4 + \alpha^2\|\mathbf{p}_c\|^2 - \alpha^2(\mathbf{x}_c^\top \mathbf{p}_c)^2]\mathbf{x}'_{c+1}(\alpha)\\ =&[-4\mathbf{x}_c^\top \mathbf{p}_c + 2\alpha(\mathbf{x}_c^\top \mathbf{p}_c)^2 - 2\alpha\|\mathbf{p}_c\|^2]\mathbf{x}_c + 4\mathbf{p}_c.\end{aligned} \tag{4.28}$$

It is derived from multiplying both sides of (4.28) by $\alpha$ and analyzing the result based on (4.21) that

$$\alpha\mathbf{x}'_{c+1}(\alpha) = \frac{-2\alpha^2(\|\mathbf{p}_c\|^2 - (\mathbf{x}_c^\top \mathbf{p}_c)^2)}{4 + \alpha^2\|\mathbf{p}_c\|^2 - \alpha^2(\mathbf{x}_c^\top \mathbf{p}_c)^2}\mathbf{x}_{c+1}(\alpha) + \mathbf{x}_{c+1}(\alpha) - \mathbf{x}_c. \tag{4.29}$$

Combing (4.29) with the equation $\nabla f(\mathbf{x}_{c+1}(\alpha))^\top \mathbf{x}_{c+1}(\alpha) = 0$, we obtain

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

□

Due to the fact $f(\mathbf{x})$ is twice continuously differentiable in the compact set $\mathbb{S}^{n-1}$, there exists a positive number $M$ such that

$$|f(\mathbf{x})| \leqslant M, \quad \|\nabla f(\mathbf{x})\| \leqslant M, \quad \text{and} \quad \|\nabla^2 f(\mathbf{x})\| \leqslant M. \tag{4.30}$$

If an optimization method enjoys a good ascent or descent property, Nocedal and Wright (2006)[Lemma 3.1] showed that we can find an interval of step lengths satisfying the Wolfe conditions. In the following theorem we prove that, an analogue Wolfe conditions are feasible for the curvilinear line search in (4.21).

**Theorem 4.1.** *Suppose the parameters $c_1$ and $c_2$ satisfy $0 < c_1 < c_2 < 1$. Then there exists a positive number $\alpha_c$ such that*

$$f\big(\mathbf{x}_{c+1}(\alpha_c)\big) \geqslant f(\mathbf{x}_c) + c_1\alpha_c \nabla f(\mathbf{x}_c)^\top \mathbf{p}_c, \tag{4.31}$$

$$\nabla f(\mathbf{x}(\alpha_c))^\top \mathbf{p}_c \leqslant c_2 \nabla f(\mathbf{x}_c)^\top \mathbf{p}_c. \tag{4.32}$$

*Proof.* Let $\mathbf{x}(\alpha) = \mathbf{x}_{c+1}(\alpha)$ and $f(\alpha) = f(\mathbf{x}_{c+1}(\alpha))$. Based on (4.21), we obtain $\mathbf{x}'_{c+1}(0) = -\mathbf{x}_c^\top \mathbf{p}_c \mathbf{x}_c + \mathbf{p}_c$, and

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

Define $l(\alpha)$ as a linear function $l(\alpha) = f(\mathbf{x}_c) + c_1\alpha\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c$. From $0 < c_1 < 1$ and $\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c > 0$ in (4.25), we have $f(0) = l(0) = f(\mathbf{x}_c)$ and $f'(0) > l'(0) > 0$. Because $f(\alpha)$ is bounded above, there is at least one intersection point of the line $l(\alpha)$ and $f(\alpha)$ when $\alpha > 0$. Let $\bar{\alpha}$ be the smallest intersection point. Hence we get

$$f(\mathbf{x}_{c+1}(\bar{\alpha})) = f(\mathbf{x}_c) + c_1\bar{\alpha}\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c. \tag{4.33}$$

By the mean value theorem, there exists a constant $\rho \in (0, \bar{\alpha})$ such that

$$\begin{aligned} f(\mathbf{x}_{c+1}(\bar{\alpha})) - f(\mathbf{x}_c) &= \bar{\alpha} f'(\rho) \\ \text{[By (4.27)]} \qquad &= -\frac{\bar{\alpha}}{\rho}\nabla f(\mathbf{x}_{c+1}(\rho))^\top \mathbf{x}_c. \end{aligned} \tag{4.34}$$

From (4.5) and (4.21) we obtain

$$\begin{aligned} \nabla f(\mathbf{x}_{c+1}(\rho))^\top \mathbf{x}_{c+1}(\rho) &= \frac{[(2-\rho\mathbf{x}_c^\top \mathbf{p}_c)^2 - \|\rho\mathbf{p}_c\|^2]\nabla f(\mathbf{x}_{c+1}(\rho))^\top \mathbf{x}_c}{4 + \|\rho\mathbf{p}_c\|^2 - (\rho\mathbf{x}_c^\top \mathbf{p}_c)^2} \\ &\quad + \frac{4\rho\nabla f(\mathbf{x}_{c+1}(\rho))^\top \mathbf{p}_c}{4 + \|\rho\mathbf{p}_c\|^2 - (\rho\mathbf{x}_c^\top \mathbf{p}_c)^2} \\ &= 0, \end{aligned}$$

which means

$$-[(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2]\nabla f(\mathbf{x}_{c+1}(\rho))^\top\mathbf{x}_c=4\rho\nabla f(\mathbf{x}_{c+1}(\rho))^\top\mathbf{p}_c. \tag{4.35}$$

By combining(4.34) and (4.35), we get

$$[(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2][f(\mathbf{x}_{c+1}(\bar{\alpha}))-f(\mathbf{x}_c)]=4\bar{\alpha}\nabla f(\mathbf{x}_{c+1}(\rho))^\top\mathbf{p}_c. \tag{4.36}$$

Further, it can be deduced from (4.33) that

$$\begin{aligned}&[(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2][f(\mathbf{x}_{c+1}(\bar{\alpha}))-f(\mathbf{x}_c)]\\=&[(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2]c_1\bar{\alpha}\nabla f(\mathbf{x}_c)^\top\mathbf{p}_c.\end{aligned} \tag{4.37}$$

From (4.36) and (4.37), we obtain

$$4\nabla f(\mathbf{x}_{c+1}(\rho))^\top\mathbf{p}_c=[(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2]c_1\nabla f(\mathbf{x}_c)^\top\mathbf{p}_c$$

Since

$$\begin{aligned}\mathbf{x}_c^\top\mathbf{p}_c&=\mathbf{x}_c^\top(\nabla f(\mathbf{x}_c)+\beta_{c-1}\mathbf{d}_{c-1})\\&=\beta_{c-1}\mathbf{x}_c^\top(\mathbf{x}_c-\mathbf{x}_{c-1})\\&=\beta_{c-1}(1-\mathbf{x}_c^\top\mathbf{x}_{c-1})\\&\geqslant 0\end{aligned}$$

and $|\mathbf{x}_c^\top\mathbf{p}_c|\leqslant\|\mathbf{p}_c\|$, we have

$$\begin{aligned}(2-\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2&=4-4\rho\mathbf{x}_c^\top\mathbf{p}_c+(\rho\mathbf{x}_c^\top\mathbf{p}_c)^2-\|\rho\mathbf{p}_c\|^2\\&\leqslant 4-4\rho\mathbf{x}_c^\top\mathbf{p}_c\\&\leqslant 4.\end{aligned}$$

Based on $\nabla f(\mathbf{x}_c)^\top\mathbf{p}_c\geqslant 0$, we obtain

$$\nabla f(\mathbf{x}_{c+1}(\rho))^\top\mathbf{p}_c\leqslant c_1\nabla f(\mathbf{x}_c)^\top\mathbf{p}_c. \tag{4.38}$$

By setting the parameter $\alpha_c=\rho$, the inequality (4.32) holds for $c_2>c_1$. Since $\rho\in(0,\bar{\alpha})$, we have $f(\alpha_c)>l(\alpha_c)$. Hence (4.31) is valid. □

**Algorithm 3** Computing $p$-spectral radius of a hypergraph(CSRH).

```
1: For a uniform hypergraph $G$, $p > 1$, choose parameters $0 < c_1 < c_2 < 1$, $\tau > 1/4$,
   $\epsilon > 0$, an initial unit point $\mathbf{x}_0$, and $k \leftarrow 0$. Calculate $\mathbf{p}_0 = \nabla f(\mathbf{x}_0)$.
2: while the sequence of iterates does not converge do
3:     Use interpolation method to find $\alpha_c$ such that (4.31) and(4.32) hold.
4:     Update the new iterate $\mathbf{x}_{c+1} = \mathbf{x}_{c+1}(\alpha_c)$ by (4.21).
5:     Compute $\mathbf{d}_c$, $\nabla f(\mathbf{x}_{c+1})$ , $\beta_c$, and $\mathbf{p}_{c+1}$ by (4.23) .
6:     $c \leftarrow c + 1$.
7: end while
```

(Chang et al., 2018)

We present the CSRH method for computing the $p$-spectral radius of a hypergraph in algorithm 3. First, we rewrite the original problem of $\lambda^{(p)}(G)$ as an equivalent spherical constrained maximization model (4.3). Then we generate an ascent direction $\mathbf{p}_c$ via (4.4), (4.24) and (4.23). Finally, by choosing a suitable $\alpha_c$ in the curvilinear search (4.21), the new iterate $\mathbf{x}_{c+1}$ is projected on the unit sphere and the analogue Wolfe conditions (4.31) and (4.32) are valid. The fast computational technique for products $\mathcal{A}\mathbf{x}^r$ and $\mathcal{A}\mathbf{x}^{r-1}$ in Section 3.4 is employed in the process of the CSRH method.

## 4.3 Convergence analysis

In this section we prove that the sequence of gradient norms approaches zero and the sequence of iterates converges to a stationary point of $f(\mathbf{x})$. In addition, we point out that our method is able to get the exact $p$-spectral radius with a high probability. For brevity, suppose the sequence $\{\mathbf{x}_c\}$ generated by the CSRH method is infinite without loss generality.

### 4.3.1 Primary convergence results

Since $f(\mathbf{x})$ is bounded above from (4.30) and increase monotonically from (4.14), we have the following result.

**Lemma 4.4.** *The sequence $\{f(\mathbf{x}_c)\}$ generated by the CSRH method converges.*

The convergence property of $\{\nabla f(\mathbf{x}_c)\}$ is given in the following theorem.

**Theorem 4.2.** *The sequence $\{\|\nabla f(\mathbf{x}_c)\|\}$ generated by the algorithm CSRH from any $\mathbf{x}_0 \in \mathbb{S}^n$ converges to zero, i.e.*

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

*Proof.* The result is achieved by two steps. First, we prove the Zoutendijk condition holds, i.e.,

$$\sum_{c=0}^{\infty} \cos^2 \varphi_c \|\nabla f(\mathbf{x}_c)\|^2 < \infty. \tag{4.39}$$

Here $\varphi_c$ is the angle between $\nabla f(\mathbf{x}_c)$ and $\mathbf{p}_c$ as follows:

$$\varphi_c \equiv \arccos \frac{\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c}{\|\nabla f(\mathbf{x}_c)\|\|\mathbf{p}_c\|}.$$

Since $\nabla^2 f(\mathbf{x})$ is bounded (4.30), the gradient function $\nabla f(\mathbf{x})$ is Lipschitz continuous on $\mathbb{S}^{n-1}$, i.e.,

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leqslant L\|\mathbf{x}_1 - \mathbf{x}_2\| \qquad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{S}^{n-1}, \tag{4.40}$$

where $L$ is a positive constant. From (4.20), we have

$$\|W\| = \|\frac{\alpha_c}{2}(\mathbf{x}_c\mathbf{p}_c^\top - \mathbf{p}_c\mathbf{x}_c^\top)\| \leqslant \frac{\alpha_c}{2}(\|\mathbf{x}_c\mathbf{p}_c^\top\| + \|\mathbf{p}_c\mathbf{x}_c^\top\|) \leqslant \alpha_c\|\mathbf{p}_c\|.$$

Hence by (4.16) we obtain

$$\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \leqslant \|W_c\|(\|\mathbf{x}_{c+1}\| + \|\mathbf{x}_c\|) \leqslant 2\alpha_c\|\mathbf{p}_c\|. \tag{4.41}$$

It can be deduce from (4.40) and (4.41) that

$$(\nabla f(\mathbf{x}_c) - \nabla f(\mathbf{x}_{c+1}))^\top \mathbf{p}_c \leqslant L\|\mathbf{x}_{c+1} - \mathbf{x}_c\|\|\mathbf{p}_c\| \leqslant 2L\alpha_c\|\mathbf{p}_c\|^2.$$

From (4.32), we get

$$(\nabla f(\mathbf{x}_{c+1}) - \nabla f(\mathbf{x}_c))^\top \mathbf{p}_c \leqslant (c_2 - 1)\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c. \tag{4.42}$$

By using the above two relations, we have

$$(1 - c_2)\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c \leqslant 2L\alpha_c \|\mathbf{p}_c\|^2,$$

which implies

$$\alpha_c \geqslant \frac{1 - c_2}{2L} \frac{\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c}{\|\mathbf{p}_c\|^2}. \tag{4.43}$$

Then based on (4.31), we obtain

$$f(\mathbf{x}_{c+1}) - f(\mathbf{x}_c) \geqslant \frac{c_1(1 - c_2)}{2L} \frac{(\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c)^2}{\|\mathbf{p}_c\|^2} = \frac{c_1(1 - c_2)}{2L} \cos^2 \varphi_c \|\nabla f(\mathbf{x}_c)\|^2,$$

which deduces the following inequality

$$f(\mathbf{x}_{c+1}) - f(\mathbf{x}_0) = \sum_{i=0}^{c} f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i) \geqslant \frac{c_1(1 - c_2)}{2L} \sum_{i=0}^{c} \cos^2 \varphi_i \|\nabla f(\mathbf{x}_i)\|^2.$$

From (4.30), the inequality (4.39) holds.

Next, we show that the angle $\varphi_c$ is bounded away from $\frac{\pi}{2}$. By combining (4.25) and (4.26), we get

$$\frac{\nabla f(\mathbf{x}_c)^\top \mathbf{p}_c}{\|\nabla f(\mathbf{x}_c)\| \|\mathbf{p}_c\|} \geqslant (1 - \frac{1}{4\tau}) \frac{\|\nabla f(\mathbf{x}_c)\|}{\|\mathbf{p}_c\|} \geqslant \frac{1}{M_0}(1 - \frac{1}{4\tau}) \equiv C_0. \tag{4.44}$$

The above inequalities mean that

$$\cos \varphi_k \geqslant C_0 > 0.$$

Hence from (4.39) the result:

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

is obtained. □

### 4.3.2 Further results when $p$ is even

In this subsection, we study the convergence property in the case $p$ is even. The graph of the function $f(\mathbf{x})$ involved in (4.3) is

$$\mathrm{Graph}\, f = \{(\mathbf{x}, \lambda) \in \mathbb{R}^n \times \mathbb{R} : [(r-1)!\mathcal{A}\mathbf{x}^r]^p = \lambda^p (\sum_i |x_i|^p)^r\}.$$

Since Graph $f$ is a semialgebraic set, $f(\mathbf{x})$ is a semialgebraic function and satisfies the Łojasiewicz inequality in (3.26) (Absil et al., 2005; Bolte et al., 2007). The next theorem shows the sequence $\{\mathbf{x}_c\}$ converges to a first-order stationary point when $p$ is even.

**Theorem 4.3.** *Let $\{\mathbf{x}_c\}$ be a sequence produced by the CSRH method. Then we have*

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

*with $\mathbf{x}_*$ being a first-order stationary point.*

*Proof.* Based on (4.43), (4.25) and (4.26) we get

$$\begin{aligned} \alpha_c &\geqslant \frac{1-c_2}{2L}(1-\frac{1}{4\tau})\frac{\|\nabla f(\mathbf{x}_c)\|^2}{\|\mathbf{p}_c\|^2} \\ &\geqslant \frac{1-c_2}{2LM_0^2}(1-\frac{1}{4\tau}) \\ &\equiv \alpha_{\min} > 0. \end{aligned}$$

By combining (4.31) and (4.25) we have

$$\begin{aligned} f(\mathbf{x}_{c+1}) - f(\mathbf{x}_c) &\geqslant c_1 \alpha_c \nabla f(\mathbf{x}_c)^\top \mathbf{p}_c \\ &\geqslant c_1 \alpha_{\min} (1 - \frac{1}{4\tau}) \|\nabla f(\mathbf{x}_c)\|^2 \\ &> 0, \end{aligned} \tag{4.45}$$

which indicates that

$$[f(\mathbf{x}_{c+1}) = f(\mathbf{x}_c)] \Rightarrow [\mathbf{x}_{c+1} = \mathbf{x}_c]. \tag{4.46}$$

Therefore from (4.26), (4.31) and (4.41) we have

$$\begin{aligned} f(\mathbf{x}_{c+1}) - f(\mathbf{x}_c) &\geqslant c_1\alpha_c(1 - \frac{1}{4\tau})\frac{\|\nabla f(\mathbf{x}_c)\|\|\mathbf{p}_c\|}{M_0} \\ &\geqslant (1 - \frac{1}{4\tau})\frac{c_1}{2M_0}\|\nabla f(\mathbf{x}_c)\|\|\mathbf{x}_{c+1} - \mathbf{x}_c\|. \end{aligned} \tag{4.47}$$

From (4.46) and (4.47), as well as the Łojasiewicz inequality (3.26), the results are obtained based on Theorem 3.2 in Absil et al. (2005). □

In order to study the probability of the event that the CSRH approach get an exact $p$-spectral radius, we introduce the next Lemma.

**Lemma 4.5.** *Assume $\mathbf{x}_*$ is a stationary point of $f(\mathbf{x})$ and $\mathscr{B}(\mathbf{x}_*, \rho) = \{\mathbf{x} \in \mathbb{S}^{n-1} : \|\mathbf{x}-\mathbf{x}_*\| \leqslant \rho\} \subseteq \mathscr{U}$, in which $\mathscr{U}$ is a neighborhood of $\mathbf{x}_*$ defined in (3.26). Let*

$$C \equiv \frac{2M_0C_K}{c_1(1-\theta)(1-\frac{1}{4\tau})},$$

*and $\mathbf{x}_0$ be the initial point such that*

$$\rho > \rho(\mathbf{x}_0) \equiv C|f(\mathbf{x}_0) - f(\mathbf{x}_*)|^{1-\theta} + \|\mathbf{x}_0 - \mathbf{x}_*\|. \tag{4.48}$$

*Then we have*

$$\mathbf{x}_c \in \mathscr{B}(\mathbf{x}_*, \rho), \quad c = 0, 1, 2, \ldots, \tag{4.49}$$

*and*

$$\sum_{c=0}^{\infty}\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \leqslant C|f(\mathbf{x}_0) - f(\mathbf{x}_*)|^{1-\theta}. \tag{4.50}$$

*Proof.* Let $\phi(t) \equiv \frac{C_K}{1-\theta}|f(\mathbf{x}_*) - t|^{1-\theta}$. Then it is easy to verify that $\phi(t)$ is a concave function for $f(\mathbf{x}_*) > t$. For $i = 0, 1, \ldots, c$, we have

$$\begin{aligned}
\phi(f(\mathbf{x}_i)) - \phi(f(\mathbf{x}_{i+1})) &\geqslant \phi'(f(\mathbf{x}_i))(f(\mathbf{x}_i) - f(\mathbf{x}_{i+1})) \\
&= C_K|f(\mathbf{x}_*) - f(\mathbf{x}_i)|^{-\theta}(f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i)) \\
\text{[The Łojasiewicz inequality]} &\geqslant \|\nabla f(\mathbf{x}_i)\|^{-1}(f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i)) \\
\text{[For (4.31) and (4.25)]} &\geqslant c_1\alpha_i(1 - \frac{1}{4\tau})\|\nabla f(\mathbf{x}_i)\| \\
\text{For [(4.26) ]} &\geqslant c_1\alpha_i(1 - \frac{1}{4\tau})\frac{1}{M_0}\|\mathbf{p}_i\| \\
\text{For [(4.41) ]} &\geqslant \frac{1}{2}c_1(1 - \frac{1}{4\tau})\frac{1}{M_0}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|.
\end{aligned}$$

Then we obtain

$$\begin{aligned}
\|\mathbf{x}_{c+1} - \mathbf{x}_*\| &\leqslant \sum_{i=0}^{c}\|\mathbf{x}_{i+1} - \mathbf{x}_i\| + \|\mathbf{x}_0 - \mathbf{x}_*\| \\
&\leqslant \frac{2M_0}{c_1(1 - \frac{1}{4\tau})}\sum_{i=0}^{c}[\phi(f(\mathbf{x}_i)) - \phi(f(\mathbf{x}_{i+1}))] + \|\mathbf{x}_0 - \mathbf{x}_*\| \\
&\leqslant \frac{2M_0}{c_1(1 - \frac{1}{4\tau})}\phi(f(\mathbf{x}_0)) + \|\mathbf{x}_0 - \mathbf{x}_*\| \\
&< \rho.
\end{aligned}$$

Hence, we get $\mathbf{x}_{c+1} \in \mathscr{B}(\mathbf{x}_*, \rho)$ and (4.49) holds. Moreover,

$$\sum_{c=0}^{\infty}\|\mathbf{x}_{c+1} - \mathbf{x}_c\| \leqslant \frac{2M_0}{c_1(1 - \frac{1}{4\tau})}\sum_{k=0}^{\infty}[\phi(f(\mathbf{x}_c)) - \phi(f(\mathbf{x}_{c+1}))] \leqslant \frac{2M_0}{c_1(1 - \frac{1}{4\tau})}\phi(f(\mathbf{x}_0)).$$

The inequality (4.50) is valid. □

**Theorem 4.4.** *If we start the CSRH algorithm from $N$ uniformly distributed initial points on $\mathbb{S}^{n-1}$, and choose the largest one among the results of these trails as the* *p-spectral radius of its corresponding problem, then the probability of getting a true p-spectral radius is*

$$1-(1-\varsigma)^N,$$

*in which the parameter $\varsigma$ satisfies $\varsigma \in (0,1]$. When $N$ is large enough, the probability is close to* 1*.*

*Proof.* With the aid of Lemma 4.5, we can prove the result similarly according to Theorem 3.9. □

## 4.4 Numerical results

In this section, we represent the numerical performances of the CSRH method for computing the $p$-spectral radius when $p > 1$ and estimating the Lagrangian ($p = 1$) of a hypergraph. For eigenvalue problems of adjacency tensors ($p = 2$ or r), we compare the CSRH method with several other algorithms. Further, we assign other different values to $p$, and calculate the $p$-spectral radii of $\beta$-stars. In Subsection 4.4.2, the CSRH method is applied to approximating the Lagrangian of a hypaergraph. We carry out the tests by MATLAB. Experiments in Subsection 4.4.1 are terminated when

$$\|\nabla f(\mathbf{x})\|_2 \leqslant 10^{-8} \quad \text{or} \quad \|\lambda^{(p)} - \lambda_*^{(p)}(G)\| \leqslant 10^{-12},$$

in which $\lambda^{(p)}$ is the numerical result given by the CSRH method and $\lambda_*^{(p)}(G)$ is the analytical solution from relevant theorems or conclusions. For the tests in Subsection 4.4.2 and Section 4.5, the termination criterion is

$$\|\nabla f(\mathbf{x})\|_2 \leqslant 10^{-6}.$$

In addition, all experiments, except those in Tensor Toolbox, are terminated when the maximum number of iterations reaches 1000. As the experiments in Section 3.5, for each test we run the CSRH method one hundred times from one hundred initial points which are uniformly distributed on the unit sphere $\mathbb{S}^{n-1}$, and get one hundred approximated solutions $\lambda_1^{(p)}, \ldots, \lambda_{100}^{(p)}$. The largest one is taken to be the numerical result of the $p$-spectral radius. Based on this data, we denote the accuracy rate of a method as

$$\text{Accu.} \equiv \left| \left\{ i : \frac{|\lambda_i^{(p)} - \lambda_*^{(p)}(G)|}{|\lambda_*^{(p)}(G)|} \leqslant 10^{-8} \right\} \right| \times 1\%. \tag{4.51}$$

We report the number of iterations (Iter.), the total computational time (Time) for all one hundred runs, the accuracy rate (Accu.), and the relative error (Err.) between a computational value and a real solution.

### 4.4.1 Calculation of $p$-spectral radii of hypergraphs

When $p$ equals two, the 2-spectral radius of an $r$-graph is $(r-1)!$ times the Z-eigenvalue of its adjacency tensor. When $p$ is even and equivalent to $r$, the $r$-spectral radius of the $r$-graph is $(r-1)!$ times the H-eigenvalue of its adjacency tensor. For the Z-eigenvalue problems of adjacency tensors, we compare the CSRH method with the adaptive shifted power method (Power M.).

**Eigenvalues of adjacency tensors.** Given a 3-graph $G_1$ with its vertex set $V$ being $V = \{1, 2, 3, 4\}$ and edge set $E$ being $E = \{123, 234\}$, the exact largest Z-eigenvalue of its adjacency tensor is provided in Xie and Chang (2013b). For the two loose paths $G_2$ with $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{123, 345, 567\}$, $G_3$ with $V = \{1, 2, 3, 4, 5\}$ and $E = \{123, 345\}$, as well as the 4-vertex 3-uniform completed hypergraph in Figure 2.4, Pearson and Zhang (2014) gave the the exact largest Z-eigenvalues of their adjacency tensors. Here we renamed the hypergraph in Figure 2.4 as $G_4$. The results of CSRH method and Power M. method for finding the largest Z-eigenvalues of adjacency tensors of $G_1$, $G_2$, $G_3$, and $G_4$ are shown in Table 4.1. The relative errors in the Err. column are given based on the the exact largest Z-eigenvalue of the corresponding adjacency tensors in Xie and Chang (2013b); Pearson and Zhang (2014). In terms of the computing time, the CSRH method seems more stable and efficient than the Power M. method.

Table 4.1: Computing $\lambda^{Z}_{\max}(\mathcal{A})$ of small scale hypergraphs.

| Hypergraph | CSRH | | | | Power M. | | | |
|---|---|---|---|---|---|---|---|---|
| | Iter. | Time(s) | Accu. | Err. | Iter. | Time(s) | Accu. | Err. |
| $G_1$ | 13593 | 3.35 | 1.00 | $5.44 \times 10^{-16}$ | 2668 | 4.89 | 1.00 | $5.44 \times 10^{-16}$ |
| $G_2$ | 1257 | 0.78 | 1.00 | $3.85 \times 10^{-16}$ | 18610 | 32.58 | 0.94 | $3.85 \times 10^{-16}$ |
| $G_3$ | 674 | 0.42 | 1.00 | $3.85 \times 10^{-16}$ | 731 | 1.61 | 1.00 | $7.69 \times 10^{-16}$ |
| $G_4$ | 8901 | 2.23 | 0.18 | $1.48 \times 10^{-16}$ | 2317 | 4.38 | 0.22 | $2.96 \times 10^{-16}$ |

The main goal of the next experiment is to verify that the probability of the event that the CSRH approach find an exact $p$-spectral radius increases along with the trail number. By randomly choosing the initial points from uniformly distributed points, we apply the CSRH method to computing $\lambda^{Z}_{\max}(\mathcal{A})$ of the hypergraph $G_4$. When the relative error of our computational largest Z-eigenvalue is equivalent to or less than $10^{-8}$, we terminate the trail and write down the number of run times. The test is run one thousand times repeatly. Denote $\sigma(i)$ as the occurrence of tests whose trail time is the integer $i$, and $\nu_i$ as the frequency of touching the exact Z-eigenvalue when running $i$ times as follows:

$$\nu_i = \frac{\sum_{j \leqslant i} \sigma(j)}{1000}. \tag{4.52}$$

The frequencies of different running times are displayed in Figure 4.2. It illustrates the connection between trail times and success probability. We can see that the frequencies increase with the trail times $i$. This phenomenon is consistent with the result in Proposition 4.4.

**Large scale examples with various** $p$. Since the $p$-spectral radii of $\beta$-stars can be attained from Theorem 2.5, we report the numerical performance of the CSRH method for calculating the $p$-spectral radii of $\beta$-stars.

We compute the $p$-spectral radii of $\beta$-stars with different number of orders and edges. Table 4.2 describes the numerical results of 3-spectral radius of 3-uniform

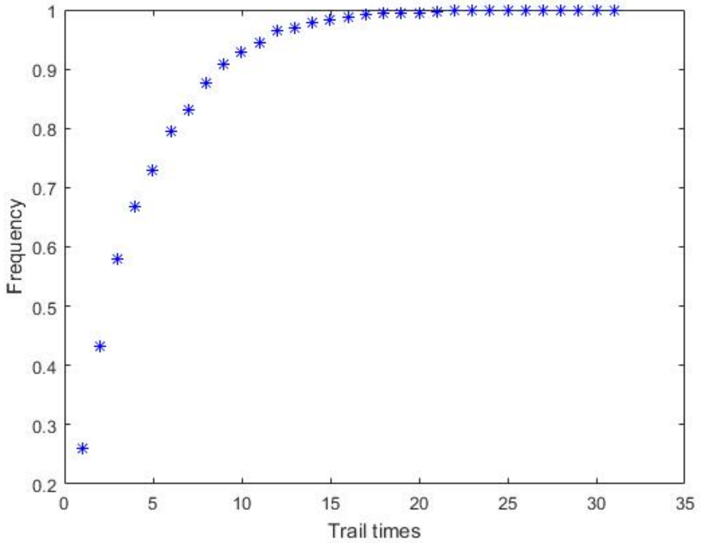


Figure 4.2: Probability of finding $\lambda^Z_{\max}(\mathcal{A}(G_4))$.

Table 4.2: Computing three-spectral radii of three-uniform $\beta$-stars.

| $n$ | $p=3$, $r=3$ $(p>r-1)$ | | | |
|---|---|---|---|---|
| | Iter. | Time(s) | Accu. | Err. |
| 21 | 1835 | 0.34 | 1.00 | $5.38\times10^{-16}$ |
| 201 | 2609 | 0.60 | 1.00 | $3.55\times10^{-15}$ |
| 2,001 | 3539 | 1.87 | 1.00 | $4.33\times10^{-14}$ |
| 20,001 | 4475 | 12.93 | 1.00 | $6.39\times10^{-14}$ |
| 200,001 | 6038 | 263.39 | 0.98 | $1.93\times10^{-11}$ |
| 2,000,001 | 20018 | 15437.99 | 1.00 | $1.22\times10^{-10}$ |

$\beta$-stars, while Table 4.2 displays the consequences of 4-spectral radius of 6-uniform $\beta$-stars. The relative errors listed in the Err. column is provided via the exact solution of the corresponding problem obtained from Theorem 2.5. We show that the CSRH method is able to compute the 3- and 4-spectral radii of $\beta$-stars with millions of vertices with high probability and efficiency.

### 4.4.2 Approximation of the Lagrangian of a hypergraph

The Lagrangian of a hypergraph $G$ denoted in (2.5) is actually its 1-spectral radius. As $f(\mathbf{x})$ is nonsmooth at the point $\mathbf{x}$ which has zero elements, we prefer to

Table 4.3: Computing four-spectral radii of four-uniform $\beta$-stars.

| $n$ | $p = 4$, $r = 6$ $(p < r - 1)$ | | | |
|---|---|---|---|---|
| | Iter. | Time(s) | Accu. | Err. |
| 51 | 14747 | 4.79 | 0.99 | $1.59 \times 10^{-11}$ |
| 501 | 26019 | 14.52 | 0.98 | $9.56 \times 10^{-12}$ |
| 5,001 | 30108 | 57.82 | 0.99 | $2.01 \times 10^{-11}$ |
| 50,001 | 32387 | 426.60 | 0.95 | $1.08 \times 10^{-11}$ |
| 500,001 | 30070 | 6309.58 | 0.99 | $4.49 \times 10^{-11}$ |
| 5,000,001 | 51609 | 125869.02 | 0.97 | $2.40 \times 10^{-10}$ |

approximate $\lambda^{(1)}(G)$ by $\lambda^{(p_\vartheta)}(G)$, where $p_\vartheta$ is defined as

$$p_\vartheta = 1 + \frac{1}{2\vartheta + 1}, \text{ for } \vartheta = 1, 2, \dots. \tag{4.53}$$

Since $\lim_{\vartheta\to\infty} p_\vartheta = 1$, then we get

$$\lim_{\vartheta\to\infty} \lambda^{(p_\vartheta)}(G) = \lambda^{(1)}(G),$$

by the result in Proposition 4.1. Therefore, it is rational that we estimate the Lagrangian of $G$ by the $p_\vartheta$-spectral radius of $G$. Moreover, due to continuous and differentiable probability of the function $f_{p_\vartheta}(\mathbf{x})$, the CSRH method is suitable for the $p_\vartheta$-spectral radius problem. If we denote a vector $\mathbf{w}$ as

$$w_i = x_i^{\frac{1}{2\vartheta+1}}, \quad \text{for } i = 1, \dots, n,$$

and substitute the variable $\mathbf{x}$ by $\mathbf{w}^{[2\vartheta+1]}$, the function $f_{p_\vartheta}(\mathbf{x}) = f_{p_\vartheta}(\mathbf{w}^{[2\vartheta+1]})$ is a semialgebraic function and satisfies the Łojasiewicz inequality in (3.26). Hence, the convergence results in Section 4.3 also apply to the $p_\vartheta$-spectral radius problem in this subsection.

We illustrate the feasibility of the CSRH method estimating Lagrangian of a hypergraph by two steps. At first, examples of the CSRH method computing the $p_\vartheta$-spectral radius of a uniform hypergraph are given. Second, we demonstrate the

Table 4.4: Computing $p_\vartheta$-spectral radii.

| $p_\vartheta$ | Iter. | Time(s) | Accu. | Err. |
|---|---|---|---|---|
| $p_\vartheta = \frac{12}{7}$ | 3037 | 0.99 | 1.00 | 0.00 |
| $p_\vartheta = \frac{14}{9}$ | 13271 | 17.88 | 1.00 | $3.08 \times 10^{-16}$ |
| $p_\vartheta = \frac{10}{7}$ | 51018 | 110.53 | 1.00 | $1.85 \times 10^{-16}$ |
| $p_\vartheta = \frac{4}{3}$ | 84848 | 88.85 | 1.00 | $3.07 \times 10^{-14}$ |

numerical performance of the CSRH method for approximating the Lagrangians of complete hypergraphs via $p_\vartheta$-spectral radius.

The results of the CSRH method calculating the $p_\vartheta$-spectral radius of a 3-uniform $\beta$-star with 10 edges are reported in Table 4.4. The accurate values of $p_\vartheta$-spectral radii is attained from Theorem 2.5. We can see that all tests get the exact $p_\vartheta$-spectral radius with accuracy rate being 100%, and the relative errors of all computational results are not greater than $3.07 \times 10^{-14}$.

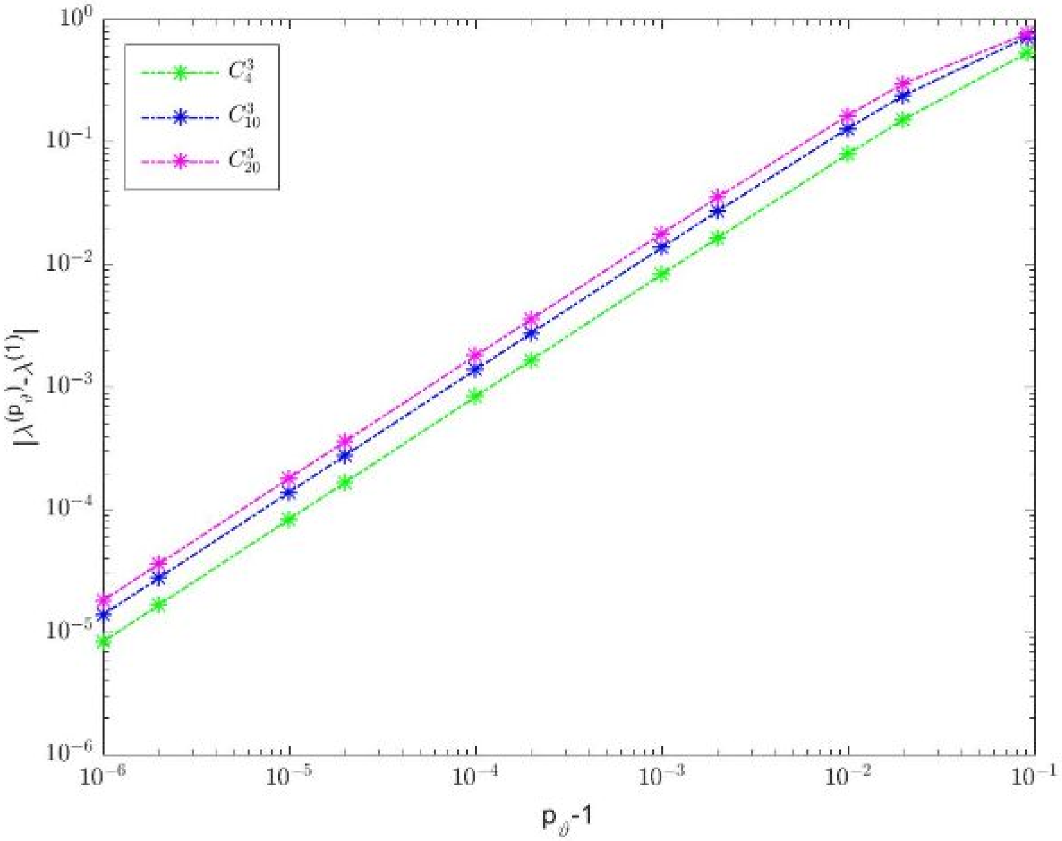


Figure 4.3: Approximating the 1-spectral radii of complete hypergraphs.

Since the exact Lagrangian of a completed hypergraph could be attained via Proposition 2.1, then we use $\lambda^{p_\vartheta}$ to estimate the Lagrangian of 3 completed hypergraphs $C_4^3$, $C_{10}^3$ and $C_{20}^3$, and show the absolute errors of our approximated solutions in Figure 4.3. The horizontal line means the difference between $p_\vartheta$ and 1, while the vertical line represents the difference between the numerical result $\lambda^{p_\vartheta}$ and the actual Lagrangian of the corresponding completed hypergraph. The tests indicate that the $p_\vartheta$-spectral radii can estimate the Lagrangians of hypergraphs well when $p_\vartheta$ is close to 1.

## 4.5 Application in network analysis

Given a hypergraph, both the optimal value of $f(\mathbf{x})$ in (4.3), i.e., the $p$-spectral radius, and the optimal point $\mathbf{x}$, reflect its structure. Recall that an optimal point is also called a $p$-optimal weighting in (2.4). The ranking of elements of the $p$-optimal weighting implies the significance of the relative vertices in the hypergraph. Hence, we regard the $j$th entry of the $p$-optimal weighting as the impact factor of the $j$th vertex. The ranking results vary with the value of $p$. When $p$ is comparatively large, the $p$-optimal weighting show the importance of vertices individually. When $p$ is comparatively small, the ranking result tends to give the importance of groups in the vertex set. In this section, we compile the CSRH method 10 times for each problem, and the $p$-optimal weighting is determined by the vector associated with the largest computational result.

### 4.5.1 A toy example

The $p$-spectral radius model is first applied to a toy problem to demonstrate the influence of $p$ on the ranking results. In Figure 4.4, a 6-graph is created with the weight of its each edge being 1, except the last edge which is assigned a weight of $\frac{3}{2}$. It can be seen that the vertices numbered 1, 31, and 26 are different from other

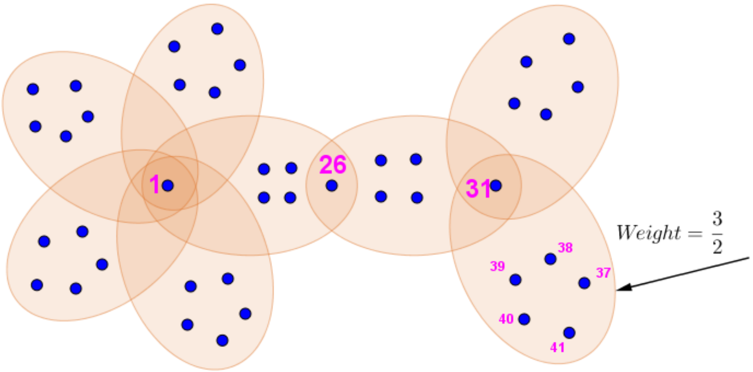


Figure 4.4: A 6-uniform hypergraph.

Table 4.5: Top ten vertices in Figure 4.4.

| Ranking | $p = \frac{4}{3}$ | | $p = 5$ | | $p = 16$ | |
|---|---|---|---|---|---|---|
| | Num. | Val. | Num. | Val. | Num. | Val. |
| 1 | 39 | 0.4082483175 | 41 | 0.4081204985 | 1 | 0.1709715830 |
| 2 | 38 | 0.4082482858 | 39 | 0.4081204985 | 31 | 0.1678396311 |
| 3 | 31 | 0.4082482855 | 31 | 0.4081204983 | 26 | 0.1618288319 |
| 4 | 41 | 0.4082482854 | 38 | 0.4081204982 | 39 | 0.1600192388 |
| 5 | 40 | 0.4082482849 | 40 | 0.4081204973 | 38 | 0.1600192387 |
| 6 | 37 | 0.4082482834 | 37 | 0.4081204958 | 41 | 0.1600192387 |
| 7 | 24 | 0.0000000000 | 28 | 0.0073198868 | 40 | 0.1600192386 |
| 8 | 34 | 0.0000000000 | 30 | 0.0073192175 | 37 | 0.1600192385 |
| 9 | 23 | 0.0000000000 | 26 | 0.0073061265 | 23 | 0.1550865094 |
| 10 | 3 | 0.0000000000 | 29 | 0.0071906282 | 22 | 0.1550865094 |

vertices in terms of degree, and the edge $\{31, 37, 38, 39, 40, 41\}$ is distinct because of its weight. With the aid of different $p$-optimal weightings, the ranking results of vertices are given in Table 4.5. The numbers of the vertices are listed in the Num. columns, while the values of impact factors of the corresponding vertices are displayed in the Val. columns.

When $p = \frac{4}{3}$, the top 6 vertices come from the edge with largest weight, and the impact factor of the top 6 vertices are far more greater than the impact factor of others. All the elements of the $\frac{4}{3}$-optimal weighting, except those associated with the top 6 vertices , are less than $5 \times 10^{-10}$. Due to the predominance of the impact factors related to the dominant vertices from the largest weighted edge, the influence of all the other vertices can be ignored. In other words, in this case the ranking results provide the most influential group among the vertices of this weighted hypergraph. When $p = 5$, although the top 6 vertices remain top 6, the gaps among the impact factors dramatically reduce and the 26th vertex enters top 10. When $p = 16$, results are clearly changed with the top 3 vertices being 1, 31 and 26. The impact factor of a vertex are more relevant to its own degree rather than to the influence of its group. Thus, we argue that the 16-spectral radius reveals the individual importance of all vertices.

### 4.5.2 Author ranking

In this subsection, we generate the rank of a number of authors in accordance with their collaborations in the publication information which can be downloaded from DBLP[2]. By using the same data set, Ng et al. have given different ranking results according citations of authors, category concepts, collaborations, and papers separately. We will compare our sorting results with theirs.

We store the cooperation information in a weighted 3-uniform multi-hypergraph $G_A$, whose vertex set is composed of 10305 authors. If any three authors have cooperations under a topic, then they are in the same edge. The weight of an edge is determined by the total collaboration times among the three authors in this edge. In this way, $G_A$ becomes a 3-graph with $1,243,443$ edges, and its adjacency tensor has 1.17% nonzero elements.

It can be seen from the example in Subsection 4.5.1 that by calculating different $p$-optimal weighting, we get different ranking results of the vertices in a hypergraph. First, we demonstrate the consequence of author group ranking via 2-optimal weighting in Figure 4.5. The stars show the 2-optimal impact factors of the corresponding authors. The majority components of the 2-optimal weighting approach zero, while

[2] http://www.informatik.uni-trier.de/ ley/db/

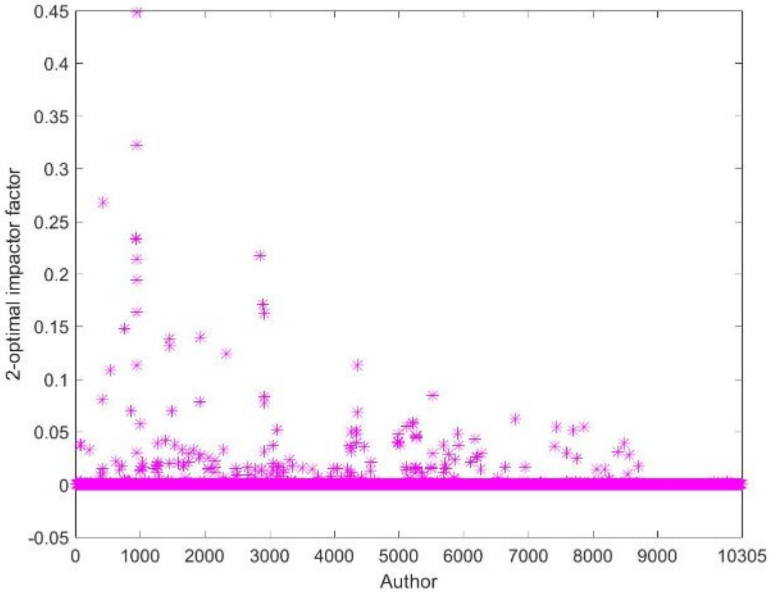


Figure 4.5: 2-optimal points.

only dozens of stars are above the horizontal line $y = 0.1$. Considering the value of the 2-optimal impactor factors , 97.2% of them are less than $10^{-3}$, while only 18% are greater than 0.1. As shown in the picture, several uppermost stars are at the top and the value of the largest impact factor is 0.4481. It implies that a tiny fraction of the elements of the 2-optimal weighting plays a leading role, and we treat the corresponding dominant vertices in the hypergraph as a group. We also display the top ten authors in the light of the 2-optimal impact factors in the second column of Table 4.6. Among these top ten authors, each two authors collaborate 8.533 times on average, while among the whole 10305 authors each two authors collaborate only $9.76 \times 10^{-4}$ times. Due to the intimate cooperation of the top ten authors, we regard the authors listed in the second column as not only a group but also the most powerful group.

The 12-optimal impact factors of vertices, which represent the 10305 authors, are shown in Figure 4.6. It has an entirely different distribution from the 2-optimal impact factors in Figure 4.5. Stars in Figure 4.6 spread evenly over the internal

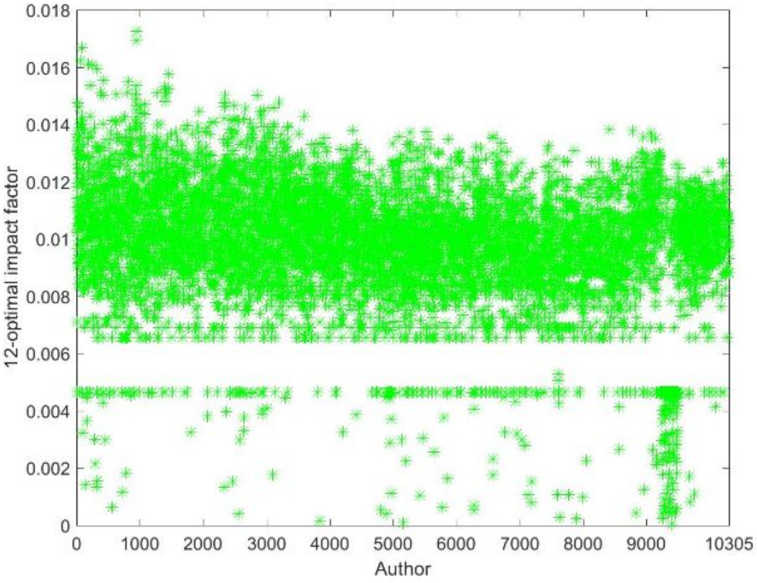


Figure 4.6: 12-optimal points.

between 0.006 and 0.014. In reality, the times of collaboration between different cooperators are mainly one or two. Thus the balance and concentration of the impact factors is consistent with the cooperation information. The top ten authors produced according to the 12-optimal weighting are shown in the third column of Table 4.6. In accordance with the collaboration and category information, Ng et al. also provide the top ten authors based on the same data set as ours, which are presented in the MultiRank column. 6 authors among 10 in MultiRank column appear in the top ten list of our 12-optimal rank.

## 4.6 Conclusion

For the $p$-spectral radius optimization problem, we transform its constraint from $p$-norm into a spherical one, and introduce an iterative method, called CSRH, to solve it. Also, we prove that in the curvilinear search, a suitable step length satisfying the analogue Wolfe conditions exists. In terms of the convergence property, it is shown that the sequence of the gradients converges to zero. Further convergence results

Table 4.6: Top 10 authors.

| Ranking | Author Name | | |
|---|---|---|---|
| | $p = 2$ | $p = 12$ | MultiRank |
| 1 | Zheng Chen | Wei-Ying Ma | C. Lee Giles |
| 2 | Wei-Ying Ma | Zheng Chen | Philip S. Yu |
| 3 | Qiang Yang | Jiawei Han | Wei-Ying Ma |
| 4 | Jun Yan | Philip S. Yu | Zheng Chen |
| 5 | Benyu Zhang | C. Lee Giles | Jiawei Han |
| 6 | Hua-Jun Zeng | Jian Pei | Christos Faloutsos |
| 7 | Weiguo Fan | Christos Faloutsos | Bing Liu |
| 8 | Wensi Xi | Yong Yu | Johannes Gehrke |
| 9 | Dou Shen | Qiang Yang | Gerhard Weikum |
| 10 | Shuicheng Yan | Ravi Kumar | Elke A. Rundensteiner |

are given under the condition that $p$ is even. The CSRH method performs well in solving the $p$-spectral radius problems in the numerical experiments. Finally, the CSRH method successfully provide the rank of 10305 authors from real world data set by calculating the $p$-optimal weighting of a weighted hypergraph with millions of edges.

# Chapter 5

# Conclusions and future work

For the computational problems related to tensors arising from hypergraphs, we investigated the eigenvalues of such tensors and the $p$-spectral radius of hypergraphs via adjacency tensors. The CEST method proposed for computing eigenvalues of adjacency tensor, Laplacian tensor and signless Laplacian tensor of a uniform hypergraph is well designed. The sequence of the function value and iterative points generated by the CEST method converge to an eigenvalue and its associated eigenvector respectively. Numerical experiments show that the CEST method is efficient. The $p$-spectral radius of a hypergraph is expressed as an adjacency tensor related maximization problem. The CSRH method is proved to be convergent and it can calculate $p$-spectral radius for different numbers assigned to $p$, which are greater than one, effectively. Further, the CSRH method is able to approximate the Lagrangian of a hypergraph accurately. In our numerical experiment, both the CSRH and the CEST method can deal with hypergraphs with millions of vertices.

Just as we have mentioned in the introduction part, tensor is a useful tool in the study of hypergraph field. Not only the eigenvalue problems and $p$-spectral radius problem, but also many other problems in hypergraph theory and application can be formulated and tackled by tensors. Such kind of unsolved problems, as well as the development in tensor theory, motivate us to further solve computing problems in hypergprah area with the aid of tensors. For example, the $k$-way hypergraph partitioning problems (Papa and Markov, 2007), which are widely studied in machine learning, computer vision, and very large scale integration (VLSI) design. We restate the problem as follows.

**Definition 5.1** ($k$-way hypergraph partitioning problem)**.** *Given a hypergraph, group all vertices of this hypergraph into $k$ disjoint sets, such that a certain cost function is minimized.*

In Chen et al. (2017), the hypergraph bi-partitioning problem is studied by eigenvalues of Laplacian tensor of the corresponding hypergraph. The general $k$-way hypergraph partitioning problem can be divided into two subproblems. First is to construct the minimization model, specifically the cost function, while the second is to solve this model effectively. Based on spectral clustering knowledge, a popular method for $k$-way hypergraph partitioning is given in Zhou et al. (2007). In this method, matrix plays a central role in storing the hypergraph information and generating the cost function. Since usually a tensor can represent a hypergraph better than a matrix do, it is natural that we construct a tensor related model to solve the $k$-way hypergraph partitioning problem. Actually, this is our ongoing work.

# Bibliography

Absil, P.-A., Mahony, R. and Andrews, B. (2005) Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, **16**, 531–547. URL https://doi.org/10.1137/040605266.

Absil, P.-A., Mahony, R. and Sepulchre, R. (2008) *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ. URL https://doi.org/10.1515/9781400830244. With a foreword by Paul Van Dooren.

Agnarsson, G. and Greenlaw, R. (2007) *Graph theory: Modeling, applications, and algorithms*. Pearson/Prentice Hall.

Aldous, J. M. and Wilson, R. J. (2003) *Graphs and applications: an introductory approach*, vol. 1. Springer Science & Business Media.

Attouch, H. and Bolte, J. (2009) On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming. A Publication of the Mathematical Programming Society*, **116**, 5–16. URL https://doi.org/10.1007/s10107-007-0133-5.

Attouch, H., Bolte, J., Redont, P. and Soubeyran, A. (2010) Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, **35**, 438–457. URL https://doi.org/10.1287/moor.1100.0449.

Barzilai, J. and Borwein, J. M. (1988) Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, **8**, 141–148. URL https://doi.org/10.1093/imanum/8.1.141.

Bolte, J., Daniilidis, A. and Lewis, A. (2007) The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, **17**, 1205–1223. URL https://doi.org/10.1137/050644641.

Bondy, J. A. and Murty, U. S. R. (1976) *Graph theory with applications*, vol. 290. Citeseer.

Bretto, A. and Gillibert, L. (2005) Hypergraph-based image representation. In *International Workshop on Graph-Based Representations in Pattern Recognition*, 1–11. Springer.

Brown, W. G. and Simonovits, M. (1984) Digraph extremal problems, hypergraph extremal problems, and the densities of graph structures. *Discrete Mathematics*, **48**, 147–162. URL https://doi.org/10.1016/0012-365X(84)90178-X.

Bu, C., Fan, Y. and Zhou, J. (2016) Laplacian and signless laplacian z-eigenvalues of uniform hypergraphs. *Frontiers of Mathematics in China*, **11**, 511–520.

Bulo, S. R. and Pelillo, M. (2009) New bounds on the clique number of graphs based on spectral hypergraph theory. In *International Conference on Learning and Intelligent Optimization*, 45–58. Springer.

Bunke H., Dickinson P., K. M. N. M. S. M. (2008) Matching of hypergraphsalgorithms, applications, and experiments. In *Applied Pattern Recognition*, 131–154. Springer.

Caraceni, A. (2011) Lagrangians of hypergraphs. URL http://alessandracaraceni.altervista.org/MyWordpress/wp-content/uploads/2014/05/Hypergraph_Lagrangians.pdf. [Online; accessed 19-February-2018].

Chang, J., Chen, Y. and Qi, L. (2016) Computing eigenvalues of large scale sparse tensors arising from a hypergraph. *SIAM Journal on Scientific Computing*, **38**, A3618–A3643.

Chang, J., Ding, W., Qi, L. and Yan, H. (2018) Computing the p-spectral radii of uniform hypergraphs with applications. *Journal of Scientific Computing*, **75**, 1–25.

Chang, K. C., Pearson, K. and Zhang, T. (2008) Perron-Frobenius theorem for nonnegative tensors. *Communications in Mathematical Sciences*, **6**, 507–520. URL http://projecteuclid.org/euclid.cms/1214949934.

Chang, K.-C., Pearson, K. and Zhang, T. (2009) On eigenvalue problems of real symmetric tensors. *Journal of Mathematical Analysis and Applications*, **350**, 416–422.

Chang, K.-C., Pearson, K. J. and Zhang, T. (2011) Primitivity, the convergence of the NQZ method, and the largest eigenvalue for nonnegative tensors. *SIAM Journal on Matrix Analysis and Applications*, **32**, 806–819. URL https://doi.org/10.1137/100807120.

Chen, L., Han, L. and Zhou, L. (2016a) Computing tensor eigenvalues via homotopy methods. *SIAM Journal on Matrix Analysis and Applications*, **37**, 290–319. URL https://doi.org/10.1137/15M1010725.

Chen, Y., Dai, Y.-H. and Han, D. (2016b) Fiber orientation distribution estimation using a peaceman–rachford splitting method. *SIAM Journal on Imaging Sciences*, **9**, 573–604.

Chen, Y., Qi, L. and Wang, Q. (2016c) Computing extreme eigenvalues of large scale Hankel tensors. *Journal of Scientific Computing*, **68**, 716–738. URL https://doi.org/10.1007/s10915-015-0155-8.

Chen, Y., Qi, L. and Zhang, X. (2017) The fiedler vector of a laplacian tensor for hypergraph partitioning. *SIAM Journal on Scientific Computing*, **39**, A2508–A2537.

Cooper, J. and Dutle, A. (2012) Spectra of uniform hypergraphs. *Linear Algebra and its Applications*, **436**, 3268–3292. URL https://doi.org/10.1016/j.laa.2011.11.018.

Cui, C., Li, Q., Qi, L. and Yan, H. (2018) A quadratic penalty method for hypergraph matching. *Journal of Global Optimization*, **70**, 237–259.

Cui, C., Luo, Z., Qi, L. and Yan, H. (2016) Computing the analytic connectivity of a uniform hypergraph. *arXiv preprint arXiv:1611.01372*.

Cui, C.-F., Dai, Y.-H. and Nie, J. (2014) All real eigenvalues of symmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, **35**, 1582–1601. URL https://doi.org/10.1137/140962292.

Dai, Y.-H. (2014) A positive bb-like stepsize and an extension for symmetric linear systems. In *Workshop on Optimization for Modern Computation, Beijing, China*, 160.

Ding, C., He, X., Husbands, P., Zha, H. and Simon, H. (2003) Pagerank, hits and a unified framework for link analysis. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, 249–253. SIAM.

Duchenne, O., Bach, F., Kweon, I.-S. and Ponce, J. (2011) A tensor-based algorithm for high-order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, **33**, 2383–2395.

Ducournau, A., Bretto, A., Rital, S. and Laget, B. (2012) A reductive approach to hypergraph clustering: An application to image segmentation. *Pattern Recognition*, **45**, 2788–2803.

Easley, D. and Kleinberg, J. (2010) *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.

Erdös, P. and Stone, A. H. (1946) On the structure of linear graphs. *Bulletin of the American Mathematical Society*, **52**, 1087–1091. URL https://doi.org/10.1090/S0002-9904-1946-08715-7.

Fan, Y.-Z., Tan, Y.-Y., Peng, X.-X. and Liu, A.-H. (2016a) Maximizing spectral radii of uniform hypergraphs with few edges. *Discussiones Mathematicae Graph Theory*, **36**, 845–856.

Fan, Y.-Z., Tan, Y.-Y. et al. (2016b) The h-spectra of a class of generalized power hypergraphs. *Discrete Mathematics*, **339**, 1682–1689.

Fan, Y.-Z., Tan, Y.-Y. et al. (2016c) The largest h-eigenvalue and spectral radius of laplacian tensor of non-odd-bipartite generalized power hypergraphs. *Linear Algebra and its Applications*, **504**, 487–502.

Fan, Y.-Z. et al. (2015) On the spectral radius of a class of non-odd-bipartite even uniform hypergraphs. *Linear Algebra and Its Applications*, **480**, 93–106.

Fischer, E., Matsliah, A. and Shapira, A. (2010) Approximate hypergraph partitioning and applications. *SIAM Journal on Computing*, **39**, 3155–3185.

Frankl, P., Peng, Y., Rödl, V. and Talbot, J. (2007) A note on the jumping constant conjecture of erdős. *Journal of Combinatorial Theory. Series B*, **97**, 204–216. URL https://doi.org/10.1016/j.jctb.2006.05.004.

Frankl, P. and Rödl, V. (1984) Hypergraphs do not jump. *Combinatorica*, **4**, 149–159.

Friedland, S., Gaubert, S. and Han, L. (2013) Perron-Frobenius theorem for nonnegative multilinear forms and extensions. *Linear Algebra and its Applications*, **438**, 738–749. URL https://doi.org/10.1016/j.laa.2011.02.042.

Friedland, S., Nocedal, J. and Overton, M. L. (1987) The formulation and analysis of numerical methods for inverse eigenvalue problems. *SIAM Journal on Numerical Analysis*, **24**, 634–667. URL https://doi.org/10.1137/0724043.

Friedman, J. and Wigderson, A. (1995) On the second eigenvalue of hypergraphs. *Combinatorica*, **15**, 43–65.

Gao, D. Y. (2016) On unified modeling, canonical duality-triality theory, challenges and breakthrough in optimization. *arXiv preprint arXiv:1605.05534*.

Gao, Y., Wang, M., Tao, D., Ji, R. and Dai, Q. (2012) 3-d object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, **21**, 4290–4303.

Ghoshdastidar, D. and Dukkipati, A. (2017) Uniform hypergraph partitioning: Provable tensor methods and sampling techniques. *The Journal of Machine Learning Research*, **18**, 1638–1678.

Golub, G. H. and Van Loan, C. F. (2013) *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edn.

Govindu, V. M. (2005) A tensor decomposition for geometric grouping and segmentation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 1150–1157. IEEE.

Gross, J. L. and Yellen, J. (2004) *Handbook of graph theory*. CRC press.

Gunopulos, D., Mannila, H., Khardon, R. and Toivonen, H. (1997) Data mining, hypergraph transversals, and machine learning. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 209–216. ACM.

Hager, W. W. and Zhang, H. (2005) A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, **16**, 170–192. URL https://doi.org/10.1137/030601880.

Hager, W. W. and Zhang, H. (2006) *Pacific Journal of Optimization. An International Journal*, **2**, 35–58.

Han, L. (2013) An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors. *Numerical Algebra, Control and Optimization*, **3**, 583–599. URL https://doi.org/10.3934/naco.2013.3.583.

Hao, C. L., Cui, C. F. and Dai, Y. H. (2015) A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors. *Numerical Linear Algebra with Applications*, **22**, 283–298. URL https://doi.org/10.1002/nla.1949.

Hillar, C. J. and Lim, L.-H. (2013) Most tensor problems are NP-hard. *Journal of the ACM*, **60**, Art. 45, 39. URL https://doi.org/10.1145/2512329.

Hu, S. and Qi, L. (2012) Algebraic connectivity of an even uniform hypergraph. *Journal of Combinatorial Optimization*, **24**, 564–579.

Hu, S. and Qi, L. (2014) The eigenvectors associated with the zero eigenvalues of the Laplacian and signless Laplacian tensors of a uniform hypergraph. *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, **169**, 140–151. URL https://doi.org/10.1016/j.dam.2013.12.024.

Hu, S. and Qi, L. (2015) The laplacian of a uniform hypergraph. *Journal of Combinatorial Optimization*, **29**, 331–366.

Hu, S., Qi, L. and Shao, J.-Y. (2013) Cored hypergraphs, power hypergraphs and their laplacian h-eigenvalues. *Linear Algebra and Its Applications*, **439**, 2980–2998.

Hu, S., Qi, L. and Xie, J. (2015) The largest laplacian and signless laplacian h-eigenvalues of a uniform hypergraph. *Linear Algebra and its Applications*, **469**, 1–27.

Huang, Y., Liu, Q., Zhang, S. and Metaxas, D. N. (2010) Image retrieval via probabilistic hypergraph ranking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3376–3383. IEEE.

Jiang, B. and Dai, Y.-H. (2015) A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Mathematical Programming*, **153**, 535–575. URL https://doi.org/10.1007/s10107-014-0816-7.

Kang, L., Nikiforov, V. and Yuan, X. (2015) The $p$-spectral radius of $k$-partite and $k$-chromatic uniform hypergraphs. *Linear Algebra and its Applications*, **478**, 81–107. URL https://doi.org/10.1016/j.laa.2015.03.016.

Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. (1999) Multilevel hypergraph partitioning: applications in vlsi domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **7**, 69–79.

Kayaaslan, E., Pinar, A., Çatalyürek, Ü. and Aykanat, C. (2012) Partitioning hypergraphs in scientific computing applications through vertex separators on graphs. *SIAM Journal on Scientific Computing*, **34**, A970–A992.

Keevash, P. (2011) Hypergraph Turán problems. In *Surveys in combinatorics 2011*, vol. 392 of *London Math. Soc. Lecture Note Ser.*, 83–139. Cambridge Univ. Press, Cambridge.

Keevash, P., Lenz, J. and Mubayi, D. (2014) Spectral extremal problems for hypergraphs. *SIAM Journal on Discrete Mathematics*, **28**, 1838–1854. URL https://doi.org/10.1137/130929370.

Khan, M.-u.-I. and Fan, Y.-Z. (2015) On the spectral radius of a class of non-odd-bipartite even uniform hypergraphs. *Linear Algebra and its Applications*, **480**, 93–106. URL https://doi.org/10.1016/j.laa.2015.04.005.

Khan, M.-u.-I., Fan, Y.-Z. and Tan, Y.-Y. (2016) The $H$-spectra of a class of generalized power hypergraphs. *Discrete Mathematics*, **339**, 1682–1689. URL https://doi.org/10.1016/j.disc.2016.01.016.

Kim, C., Bandeira, A. S. and Goemans, M. X. (2017) Community detection in hypergraphs, spiked tensor models, and sum-of-squares. In *Sampling Theory and Applications (SampTA), 2017 International Conference on*, 124–128. IEEE.

Klamt, S., Haus, U.-U. and Theis, F. (2009) Hypergraphs and cellular networks. *PLoS computational biology*, **5**, e1000385.

Kolda, T. G., Bader, B. W. and Kenny, J. P. (2005) Higher-order web link analysis using multilinear algebra. In *Data Mining, Fifth IEEE International Conference on*, 8–pp. IEEE.

Kolda, T. G. and Mayo, J. R. (2011) Shifted power method for computing tensor eigenpairs. *SIAM J. Matrix Anal. Appl.*, **32**, 1095–1124. URL https://doi.org/10.1137/100801482.

Kolda, T. G. and Mayo, J. R. (2014) An adaptive shifted power method for computing generalized tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, **35**, 1563–1581. URL https://doi.org/10.1137/140951758.

Konstantinova, E. and Skorobogatov, V. (1998) Molecular structures of organoelement compounds and their representation as labeled molecular hypergraphs. *Journal of structural chemistry*, **39**, 268–276.

Konstantinova, E. V. and Skorobogatov, V. A. (1995) Molecular hypergraphs: The new representation of nonclassical molecular structures with polycentric delocalized bonds. *Journal of chemical information and computer sciences*, **35**, 472–478.

Konstantinova, E. V. and Skorobogatov, V. A. (2001) Application of hypergraph theory in chemistry. *Discrete Mathematics*, **235**, 365–383.

Krohn-Grimberghe, A., Drumond, L., Freudenthaler, C. and Schmidt-Thieme, L. (2012) Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 173–182. ACM.

Lee, J., Cho, M. and Lee, K. M. (2011) Hyper-graph matching via reweighted random walks. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1633–1640. IEEE.

Li, G., Qi, L. and Yu, G. (2013) The z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory. *Numerical Linear Algebra with Applications*, **20**, 1001–1029.

Li, H., Shao, J.-Y. and Qi, L. (2016) The extremal spectral radii of $k$-uniform supertrees. *Journal of Combinatorial Optimization*, **32**, 741–764. URL https://doi.org/10.1007/s10878-015-9896-4.

Li, W., Cooper, J. and Chang, A. (2017) Analytic connectivity of $k$-uniform hypergraphs. *Linear and Multilinear Algebra*, **65**, 1247–1259. URL https://doi.org/10.1080/03081087.2016.1234575.

Lim, L.-H. (2005) Singular values and eigenvalues of tensors: a variational approach. In *Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE International Workshop on*, 129–132. IEEE.

Lim, L.-H. (2008) Eigenvalues of tensors and some very basic spectral hypergraph theory. URL https://www.stat.uchicago.edu/~lekheng/work/mcsc2.pdf.

Lin, H., Mo, B., Zhou, B. and Weng, W. (2016a) Sharp bounds for ordinary and signless laplacian spectral radii of uniform hypergraphs. *Applied Mathematics and Computation*, **285**, 217–227.

Lin, H., Zhou, B. and Mo, B. (2016b) Upper bounds for h-and z-spectral radii of uniform hypergraphs. *Linear Algebra and Its Applications*, **510**, 205–221.

Liu, D. C. and Nocedal, J. (1989) On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, **45**, 503–528. URL https://doi.org/10.1007/BF01589116.

Liu, Y., Zhou, G. and Ibrahim, N. F. (2010) An always convergent algorithm for the largest eigenvalue of an irreducible nonnegative tensor. *Journal of Computational and Applied Mathematics*, **235**, 286–292. URL https://doi.org/10.1016/j.cam.2010.06.002.

Lojasiewicz, S. (1963) Une propriété topologique des sous-ensembles analytiques réels. *Les équations aux dérivées partielles*, **117**, 87–89.

Lu, L. and Man, S. (2014) Hypergraphs with spectral radius at most $(r-1)!\sqrt[r]{2+\sqrt{5}}$. *arXiv preprint arXiv:1412.1270.*

Lu, L. and Man, S. (2016a) Connected hypergraphs with small spectral radius. *Linear Algebra Appl.*, **509**, 206–227. URL https://doi.org/10.1016/j.laa.2016.07.013.

Lu, L. and Man, S. (2016b) Connected hypergraphs with small spectral radius. *Linear Algebra and its Applications*, **509**, 206–227. URL https://doi.org/10.1016/j.laa.2016.07.013.

Motzkin, T. S. and Straus, E. G. (1965) Maxima for graphs and a new proof of a theorem of Turán. *Canadian Journal of Mathematics. Journal Canadien de Mathématiques*, **17**, 533–540. URL https://doi.org/10.4153/CJM-1965-053-6.

Mubayi, D. (2006) A hypergraph extension of Turán's theorem. *Journal of Combinatorial Theory. Series B*, **96**, 122–134. URL https://doi.org/10.1016/j.jctb.2005.06.013.

Newman, M. (2010) *Networks: an introduction.* Oxford university press.

Ng, M., Qi, L. and Zhou, G. (2009) Finding the largest eigenvalue of a nonnegative tensor. *SIAM Journal on Matrix Analysis and Applications*, **31**, 1090–1099. URL https://doi.org/10.1137/09074838X.

Ng, M. K.-P., Li, X. and Ye, Y. (2011) Multirank: co-ranking for objects and relations in multi-relational data. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1217–1225. ACM.

Nguyen, Q., Gautier, A. and Hein, M. (2015) A flexible tensor block coordinate ascent scheme for hypergraph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5270–5278.

Ni, Q. and Qi, L. (2015) A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map. *Journal of Global Optimization*, **61**, 627–641. URL `https://doi.org/10.1007/s10898-014-0209-8`.

Nikiforov, V. (2007) Bounds on graph eigenvalues. II. *Linear Algebra and its Applications*, **427**, 183–189. URL `https://doi.org/10.1016/j.laa.2007.07.010`.

Nikiforov, V. (2013) An analytic theory of extremal hypergraph problems. *arXiv preprint arXiv:1305.1073*.

Nikiforov, V. (2014) Analytic methods for uniform hypergraphs. *Linear Algebra and Its Applications*, **457**, 455–535.

Nocedal, J. (1980) Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, **35**, 773–782. URL `https://doi.org/10.2307/2006193`.

Nocedal, J. and Wright, S. J. (2006) *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edn.

Page, L. (1999) The page rank citation ranking: Bring order to the web [ol]. *Tech. rep.*, Stanford Digital Libraries Working Paper, http://www-diglib. stanford. edu.

Papa, D. A. and Markov, I. L. (2007) Hypergraph partitioning and clustering. In *Handbook of approximation algorithms and metaheuristics*, 61(1)–61(19). Chapman & Hall/CRC, Boca Raton, FL.

Pearson, K. J. and Zhang, T. (2014) On spectral hypergraph theory of the adjacency tensor. *Graphs and Combinatorics*, **30**, 1233–1248. URL `https://doi.org/10.1007/s00373-013-1340-x`.

Peng, Y. (2008) Using Lagrangians of hypergraphs to find non-jumping numbers. I. *Annals of Combinatorics*, **12**, 307–324. URL `https://doi.org/10.1007/s00026-008-0353-2`.

Peng, Y. and Zhao, C. (2008) Generating non-jumping numbers recursively. *Discrete Applied Mathematics. The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, **156**, 1856–1864. URL `https://doi.org/10.1016/j.dam.2007.09.003`.

Pliakos, K. and Kotropoulos, C. (2015) Weight estimation in hypergraph learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 1161–1165. IEEE.

Qi, L. (2005a) Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, **40**, 1302–1324.

Qi, L. (2005b) Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, **40**, 1302–1324. URL https://doi.org/10.1016/j.jsc.2005.05.007.

Qi, L. (2014) $H^+$-eigenvalues of Laplacian and signless Laplacian tensors. *Communications in Mathematical Sciences*, **12**, 1045–1064. URL https://doi.org/10.4310/CMS.2014.v12.n6.a3.

Qi, L. and Luo, Z. (2017) *Tensor analysis: Spectral theory and special tensors*, vol. 151. SIAM.

Qi, L., Shao, J.-Y. and Wang, Q. (2014) Regular uniform hypergraphs, s-cycles, s-paths and their largest laplacian h-eigenvalues. *Linear Algebra and Its Applications*, **443**, 215–227.

Qi, L., Wang, F. and Wang, Y. (2009) $Z$-eigenvalue methods for a global polynomial optimization problem. *Mathematical Programming*, **118**, 301–316. URL https://doi.org/10.1007/s10107-007-0193-6.

Shao, J.-Y., Shan, H.-Y. and Wu, B.-f. (2015) Some spectral properties and characterizations of connected odd-bipartite uniform hypergraphs. *Linear and Multilinear Algebra*, **63**, 2359–2372. URL https://doi.org/10.1080/03081087.2015.1009061.

Shashua, A., Zass, R. and Hazan, T. (2006) Multi-way clustering using super-symmetric non-negative tensor factorization. In *European conference on computer vision*, 595–608. Springer.

Sidorenko, A. F. (1987) On the maximal number of edges in a homogeneous hypergraph that does not contain prohibited subgraphs. *Akademiya Nauk SSSR. Matematicheskie Zametki*, **41**, 433–455, 459.

Sun, W. and Yuan, Y.-X. (2006) *Optimization theory and methods: nonlinear programming*, vol. 1. Springer Science & Business Media.

Turán, P. (1941) Eine extremalaufgabe aus der graphentheorie. *Mat. Fiz. Lapok*, **48**, 61.

Turán, P. (1961) Research problems. *Közl MTA Mat. Kutató Int*, **6**, 417–423.

Wen, Z. and Yin, W. (2013) A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, **142**, 397–434. URL https://doi.org/10.1007/s10107-012-0584-1.

Xie, J. and Chang, A. (2013a) H-eigenvalues of signless Laplacian tensor for an even uniform hypergraph. *Frontiers of Mathematics in China*, **8**, 107–127. URL https://doi.org/10.1007/s11464-012-0266-6.

Xie, J. and Chang, A. (2013b) On the Z-eigenvalues of the adjacency tensors for uniform hypergraphs. *Linear Algebra and its Applications*, **439**, 2195–2204. URL https://doi.org/10.1016/j.laa.2013.07.016.

Xie, J. and Chang, A. (2013c) On the Z-eigenvalues of the signless Laplacian tensor for an even uniform hypergraph. *Numerical Linear Algebra with Applications*, **20**, 1030–1045. URL https://doi.org/10.1002/nla.1910.

Xie, J. and Qi, L. (2015) The clique and coclique numbers bounds based on the h-eigenvalues of uniform hypergraphs. *International Journal of Numerical Analysis & Modeling*, **12**, 318–327.

Xu, Y. and Yin, W. (2013) A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, **6**, 1758–1789. URL https://doi.org/10.1137/120887795.

Yuan, X., Zhang, M. and Lu, M. (2015) Some upper bounds on the eigenvalues of uniform hypergraphs. *Linear Algebra and Its Applications*, **484**, 540–549.

Yue, J., Zhang, L. and Lu, M. (2016) Largest adjacency, signless Laplacian, and Laplacian H-eigenvalues of loose paths. *Frontiers of Mathematics in China*, **11**, 623–645. URL https://doi.org/10.1007/s11464-015-0452-4.

Zhang, L., Qi, L. and Zhou, G. (2014) $M$-tensors and some applications. *SIAM Journal on Matrix Analysis and Applications*, **35**, 437–452. URL https://doi.org/10.1137/130915339.

Zhou, D., Huang, J. and Schölkopf, B. (2007) Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, 1601–1608.

Zhou, G., Qi, L. and Wu, S.-Y. (2013a) Efficient algorithms for computing the largest eigenvalue of a nonnegative tensor. *Frontiers of Mathematics in China*, **8**, 155–168. URL https://doi.org/10.1007/s11464-012-0268-4.

Zhou, G., Qi, L. and Wu, S.-Y. (2013b) On the largest eigenvalue of a symmetric nonnegative tensor. *Numerical Linear Algebra with Applications*, **20**, 913–928. URL https://doi.org/10.1002/nla.1885.