DE GRUYTER Spec. Matrices 2019; 7:263–271

Research Article Open Access

Special Issue Dedicated to Charles R. Johnson

Brian D. Sutton*

Numerical construction of structured matrices with given eigenvalues

https://doi.org/10.1515/spma-2019-0020
Received June 24, 2019; accepted November 30, 2019

Abstract: We consider a structured inverse eigenvalue problem in which the eigenvalues of a real symmetric matrix are specified and selected entries may be constrained to take specific numerical values or to be nonzero. This includes the problem of specifying the graph of the matrix, which is determined by the locations of zero and nonzero entries. In this article, we develop a numerical method for constructing a solution to the structured inverse eigenvalue problem. The problem is recast as a constrained optimization problem over the orthogonal manifold, and a numerical optimization routine seeks its solution.

Keywords: inverse eigenvalue problem, eigenvalue, multiplicity, graph, numerical, optimization

MSC: 15A29, 65F18, 15A18, 15B99, 90C30

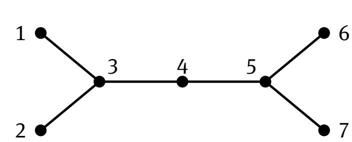
This article is about a structured inverse eigenvalue problem (IEP) and a numerical algorithm for its solution. The following example shows where we're headed.

Example 1. Does there exist a real symmetric matrix with eigenvalues $-2, -1, -1, 0, 1, 1, 2$ and the following zero-nonzero pattern?

$$\begin{bmatrix} ? & 0 & * & 0 & 0 & 0 & 0 \\ 0 & ? & * & 0 & 0 & 0 & 0 \\ * & * & ? & * & 0 & 0 & 0 \\ 0 & 0 & * & ? & * & 0 & 0 \\ 0 & 0 & 0 & * & ? & * & * \\ 0 & 0 & 0 & 0 & * & ? & 0 \\ 0 & 0 & 0 & 0 & * & 0 & ? \end{bmatrix} \quad (1)$$

In the pattern, the 0's indicate entries that must be zero; the $*$'s indicate entries that must be nonzero; and the diagonal entries are marked by ?'s to indicate that they are not constrained either way. If there is such a matrix, we wish to exhibit one.

There is a substantial literature involving patterns like the one above, in which every off-diagonal entry is constrained to be zero or nonzero and every diagonal entry is left unconstrained. Such a pattern is identified with a graph on n vertices in which distinct vertices i and j are adjacent if and only if the (i, j)-entry of the pattern is a $*$. Pattern (1) in particular is identified with the following graph:



***Corresponding Author: Brian D. Sutton:** Randolph-Macon College, Ashland, Virginia

Using methods developed later in this article, we have found numerical evidence supporting the existence of such a matrix. Let $a = 0.823577710903906$, $b = 0.281967046458124$, and $c = 0.776972844158924$, and construct the matrix

$$\begin{bmatrix} 1 & 0 & a & 0 & 0 & 0 & 0 \\ 0 & 1 & a & 0 & 0 & 0 & 0 \\ a & a & b & c & 0 & 0 & 0 \\ 0 & 0 & c & 0 & c & 0 & 0 \\ 0 & 0 & 0 & c & -b & a & a \\ 0 & 0 & 0 & 0 & a & -1 & 0 \\ 0 & 0 & 0 & 0 & a & 0 & -1 \end{bmatrix}.$$

This matrix is symmetric; it has the correct zero-nonzero pattern; and its eigenvalues have been computed numerically to be the following:

$$\begin{array}{r} -2.000000000000000 \\ -1.000000000000001 \\ -1.000000000000000 \\ 0.000000000000000 \\ 1.000000000000000 \\ 1.000000000000001 \\ 2.000000000000001 \end{array}$$

These are very nearly equal to the desired eigenvalues $-2, -1, -1, 0, 1, 1, 2$—as close as possible given the standard precision of modern desktop computers. □

A construction problem similar to the one above was solved previously using the implicit function theorem [8]; the method used in this article is very different. In addition, the final section of the paper considers a larger example whose solution may be new.

1 Introduction

Note the presence of repeated eigenvalues in Example 1. A "generic" matrix has no repeated eigenvalues, and it is now well known that the graph of a matrix can severely restrict its eigenvalue multiplicities. It is impossible, for example, to construct a real symmetric matrix with pattern (1) and three eigenvalues of multiplicity 2, or with eigenvalues of multiplicity 2 and 3, or with an eigenvalue of multiplicity 4. Classifying the eigenvalue multiplicities that are achievable with a given graph is a big problem that has received much attention and is now the subject of a book [7].

The problem considered in this article is the following: Given a pattern such as (1) and a list of real numbers $\lambda_1, \dots, \lambda_n$, does there exist a real symmetric matrix with the given pattern and eigenvalues? We call this problem "the IEP." It is an example of a *structured inverse eigenvalue problem* [5]. Our problem is motivated by the literature on graphs and eigenvalue multiplicities, but it differs in three major ways:

1. The pattern may contain any numerical entries, not just zeros, indicating that the desired matrix must have the specified values in the specified locations;
2. Entries may be constrained or unconstrained regardless of whether they are on or off the main diagonal; and
3. Eigenvalues are given specific numerical values rather than just multiplicities. (A multiple eigenvalue can be indicated by including a numerical value multiple times in the eigenvalue list.)

We develop a numerical routine for constructing a matrix with a given pattern and given eigenvalues, when one exists. The routine is also helpful in determining when no such solution exists.

Our approach is to reformulate the inverse eigenvalue problem as an optimization problem. Below, the optimization problem is developed, an algorithm for its solution is described, and an additional example is investigated.

2 Optimization problem

Consider a symmetric pattern consisting of $*$'s, numerical values, and ?'s, e.g.,

$$\begin{bmatrix} \pi & ? & * & 0 \\ ? & ? & * & ? \\ * & * & ? & 0 \\ 0 & ? & 0 & * \end{bmatrix} \tag{2}$$

This example specifies a real symmetric matrix with nonzero entries in locations $(1,3)$, $(2,3)$, $(3,1)$, $(3,2)$, and $(4,4)$; the numerical value π in the $(1,1)$-entry; and 0's in entries $(1,4)$, $(3,4)$, $(4,1)$, and $(4,3)$. The remaining entries $(1,2)$, $(2,1)$, $(2,2)$, $(2,4)$, $(3,3)$, and $(4,2)$ are unconstrained and can be any real numbers, zero or nonzero.

Let I_* be the set of index pairs (i,j) that locate $*$'s. Let I_c be the set of index pairs (i,j) that locate numerical values, and let c_{ij} denote the numerical value in the pattern at location (i,j). For pattern (2), we have $I_* = \{(1,3),(2,3),(3,1),(3,2),(4,4)\}$, $I_c = \{(1,1),(1,4),(3,4),(4,1),(4,3)\}$, and $c_{11} = \pi$, $c_{14} = c_{34} = c_{41} = c_{43} = 0$.

Let $\lambda_1,\dots,\lambda_n$ be real numbers, the desired eigenvalues with repetitions indicating multiplicities.

Our problem is to find a real symmetric matrix $X = [x_{ij}]$ in which

$$x_{ij} \neq 0 \text{ for all } (i,j) \in I_*,$$
$$x_{ij} = c_{ij} \text{ for all } (i,j) \in I_c,$$

and whose eigenvalues are $\lambda_1,\dots,\lambda_n$.

Often, if there is one such matrix, then there are infinitely many. (For example, in the heavily-studied case of a pattern whose graph is a tree, there are $2n-1$ unknowns but just n eigenvalue constraints.) This provides some freedom in a numerical search, and we choose to seek a solution whose nonzero entries are far from zero in order to provide more convincing numerical evidence. Let f be a function that levies an infinite penalty at zero, e.g.,

$$f(x) = -\log|x|.$$

We seek a solution $X = [x_{ij}]$ to the IEP that further solves the following optimization problem:

$$\begin{aligned} &\text{minimize} && \sum_{(i,j)\in I_*} f(x_{ij}) \\ &\text{subject to} && x_{ij} = c_{ij} \text{ for all } (i,j) \in I_c, \\ &&& \lambda(X) = (\lambda_1,\dots,\lambda_n). \end{aligned} \tag{3}$$

There is a solution to the IEP if and only if there exists a matrix in the feasible region where the objective function is finite. Furthermore, as we argue below, the objective function is bounded below. Therefore, there is a solution to the IEP if and only if the objective function has a global minimum over the feasible region.

Our method for solving the optimization problem is developed in three parts:

1. To enforce the eigenvalue constraints, we represent X implicitly by its eigenvalue decomposition $X = Q\Lambda Q^T$ with $\Lambda = \operatorname{diag}(\lambda_1,\dots,\lambda_n)$ and search over Q in the special orthogonal group.
2. The constraints $x_{ij} = c_{ij}$ are enforced by the *augmented Lagrangian method,* which is related to the classical method of Lagrange multipliers.
3. An off-the-shelf package for optimizing over manifolds, e.g., Manopt [4], is used in solving the reformulated optimization problem.

3 Geometry of the orthogonal group

The simplest optimization algorithm is gradient descent. Imagine an ant sitting at a point in the search space. The ant computes the gradient of the objective function at its current location and then steps in the opposite direction. Step after step, the objective function becomes smaller and smaller, until the ant reaches a local minimum.

When the search space is a non-Euclidean manifold, the ant needs to know some geometry [1, 6].

In the current setting, our manifold is the special orthogonal group $\mathrm{SO}(n)$, whose elements are n-by-n orthogonal matrices with determinant equal to 1. A point Q on the manifold is a potential eigenvector matrix, and the real symmetric matrix X is defined implicitly as $X = Q\Lambda Q^T$ for a fixed eigenvalue matrix Λ.

A geodesic path emanating from Q is a function $Q(t) = e^{tA}Q$ for some antisymmetric matrix A. Its derivative is

$$Q'(t) = Ae^{tA}Q = AQ(t),$$

and at time $t = 0$, this takes the value

$$Q'(0) = AQ(0) = AQ.$$

The possible velocities from the point Q are thus identified with the antisymmetric matrices A. The inner product on this tangent space is $\langle A, B\rangle = \mathrm{tr}(A^TB)$. To simplify terminology below, assume that $\|A\| = \sqrt{\langle A, A\rangle} = 1$.

As Q travels along its geodesic, X travels along the path

$$X(t) = Q(t)\Lambda Q(t)^T,$$

and its velocity is

$$X'(t) = Q'(t)\Lambda Q(t)^T + Q(t)\Lambda Q'(t)^T = AQ(t)\Lambda Q(t)^T + Q(t)\Lambda Q(t)^T A^T = [A, X(t)],$$

in which $[A, B]$ denotes the commutator $AB - BA$ and we've used the fact that $A^T = -A$. At time $t = 0$, the directional derivative is

$$X'(0) = [A, X(0)] = [A, X].$$

We can express the derivative of the (i, j)-entry $x_{ij}(t)$ of $X(t)$ at time $t = 0$ as

$$x'_{ij}(0) = e_i^T[A, X]e_j,$$

in which e_j is the jth standard basis vector.

The objective function in (3) is a sum of $f(x_{ij}) = -\log|x_{ij}|$. Later in this article, a more complicated objective function will appear, but it will still be constructed from functions acting on individual entries of X: $\sum_{i,j} g_{ij}(x_{ij})$. Remember that X is defined to be the product $X = Q\Lambda Q^T$, and our search will be conducted over orthogonal matrices Q. We understand how $g_{ij}(x_{ij})$ varies by taking a directional derivative. As above, let $Q(t) = e^{tA}Q$ be a geodesic and define $X(t) = [x_{ij}(t)] = Q(t)\Lambda Q(t)^T$. Then

$$\frac{\mathrm{d}}{\mathrm{d}t} g_{ij}(x_{ij}(t)) = g'_{ij}(x_{ij}(t))x'_{ij}(t),$$

and so

$$\left.\frac{\mathrm{d}}{\mathrm{d}t} g_{ij}(x_{ij}(t))\right|_{t=0} = g'_{ij}(x_{ij})e_i^T[A, X]e_j. \tag{4}$$

Optimization algorithms rely heavily on gradients and Hessians. The gradient is an element of the tangent space whose dual computes directional derivatives.

Theorem 1. *Let g_{ij} be a function of a real variable and define $X = [x_{ij}] = Q\Lambda Q^T$ as above. Considering $g_{ij}(x_{ij})$ as a function on the space of real orthogonal eigenvector matrices Q, its gradient is*

$$\mathrm{grad}\, g_{ij}(x_{ij}) = g'(x_{ij})[E_{ij}, X], \tag{5}$$

in which $E_{ij} = (1/2)(e_ie_j^T + e_je_i^T)$.

Proof. Let B be the right-hand side of (5).

B belongs to the tangent space because it is antisymmetric:

$$[E_{ij}, X]^T = -[E_{ij}^T, X^T] = -[E_{ij}, X].$$

Let A be any antisymmetric matrix with unit norm. We have

$$\begin{aligned}
\operatorname{tr}(B^T A) &= \operatorname{tr}((g'(x_{ij})[E_{ij}, X])^T A) \\
&= g'(x_{ij}) \operatorname{tr}((E_{ij}X - XE_{ij})^T A) \\
&= g'(x_{ij})(\operatorname{tr}(XE_{ij}A) - \operatorname{tr}(E_{ij}XA)) \\
&= g'(x_{ij})(\operatorname{tr}(AXE_{ij}) - \operatorname{tr}(XAE_{ij})) \\
&= g'(x_{ij}) \operatorname{tr}([A, X]E_{ij}) \\
&= g'(x_{ij})(1/2) \operatorname{tr}([A, X](e_i e_j^T + e_j e_i^T)) \\
&= g'(x_{ij})(1/2)(e_j^T [A, X]e_i + e_i^T [A, X]e_j).
\end{aligned}$$

Because $[A, X]$ is symmetric ($[A, X]^T = -[A^T, X^T] = [A, X]$), we have $e_j^T[A, X]e_i = e_i^T[A, X]e_j$ and therefore $\operatorname{tr}(B^T A) = g'(x_{ij})e_i^T[A, X]e_j$. According to (4), this equals the directional derivative of $g_{ij}(x_{ij})$ when Q moves in the direction A, just as required. □

The Hessian is the symmetric operator that computes the directional derivative of the gradient. It is the key component in the second-order term of a Taylor series.

Theorem 2. *Define g_{ij} and X as in the previous theorem. The Hessian of $g_{ij}(x_{ij})$ acts as follows on an antisymmetric matrix A:*

$$\left(\operatorname{hess} g_{ij}(x_{ij})\right)[A] = g'_{ij}(x_{ij})([E_{ij}, [A, X]] + [[E_{ij}, A], X]) + g''_{ij}(x_{ij})(e_i^T[A, X]e_j)[E_{ij}, X]. \tag{6}$$

The proof is omitted.

Armed with values of the objective function and its gradient and Hessian at a point, a numerical solver can construct a second-order model of the objective function in a local neighborhood. This enables a search for a local minimum.

One more note about geometry: We promised to prove that the objective function is bounded below. This is about compactness. The entries of $X = Q\Lambda Q^T$ are continuous functions of Q, and the domain is the compact space $\mathrm{SO}(n)$. The possible entries of X thus form a compact space—most importantly a bounded subset of Euclidean space—and $f(x) = -\log|x|$ is bounded below when $|x|$ is bounded.

4 Augmented Lagrangian method

Two basic ideas for enforcing constraints $x_{ij} = c_{ij}$, $(i, j) \in I_c$ are the following:

1. An *external penalty function* such as $(\rho/2)(x_{ij} - c_{ij})^2$ could be added to the objective function for each constraint:
$$\sum_{(i,j)\in I_*} f(x_{ij}) + \frac{\rho}{2} \sum_{(i,j)\in I_c} (x_{ij} - c_{ij})^2.$$
As $\rho \to +\infty$, the penalty becomes more and more severe, eliminating candidate matrices that do not satisfy the constraints.
2. A *Lagrange multiplier* term $y_{ij}(x_{ij} - c_{ij})$ could be added to the objective function for each constraint:
$$\sum_{(i,j)\in I_*} f(x_{ij}) + \sum_{(i,j)\in I_c} y_{ij}(x_{ij} - c_{ij}).$$
At a stationary point, the partial derivative condition $(\partial/\partial y_{ij})[y_{ij}(x_{ij} - c_{ij})] = 0$ forces $x_{ij} = c_{ij}$.

Neither approach alone leads to a satisfactory numerical method. In the external penalty approach, the objective function becomes less and less smooth as $\rho \to \infty$, causing problems for a numerical solver based on gradients and Hessians. In the Lagrange multiplier approach, the desired stationary point sits at a saddle point rather than a local minimum, and saddle points can be hard to find.

The *augmented Lagrangian method* combines the two ideas, avoiding their pitfalls [2, 3, 9]. To apply this method, we form the *augmented Lagrangian*

$$L_\rho(Q,y) = \sum_{(i,j)\in I_*} f(x_{ij}) + \frac{\rho}{2}\sum_{(i,j)\in I_c}\left[(x_{ij}-c_{ij}) + \frac{y_{ij}}{\rho}\right]^2.$$

As before, $X = [x_{ij}] = Q\Lambda Q^T$.

A very rough outline of the augmented Lagrangian method is the following:

1. Initialize ρ, Q, and $y = [y_{ij}]$.
2. Until convergence,
   (a) Search for the optimal Q given the current values of ρ and y.
   (b) Update y using a simple formula.
   (c) Update ρ.

The process is designed for Q to converge to a local minimum of the original optimization problem (3) and for y to converge to the multipliers in the classical Lagrange multiplier method. At each step, a rule determines whether ρ is increased or left alone.

Two nice things happen: First, under certain conditions, the process converges for bounded ρ. Second, because the Lagrange multipliers are estimated separately from Q, they are not part of the search process and do not introduce a saddle point.

The search for Q can be executed using a numerical solver for unconstrained problems. Such a solver typically requires an objective function and its gradient, and some solvers can be accelerated by providing the Hessian as well. In this context, the objective function is the augmented Lagrangian $L_\rho(Q,y)$, which can also be expressed as follows:

$$L_\rho(Q,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}(x_{ij})$$

with

$$g_{ij}(x) = \begin{cases} f(x), & (i,j)\in I_*, \\ (\rho/2)((x-c_{ij}) + (y_{ij}/\rho))^2, & (i,j)\in I_c, \\ 0 & \text{otherwise.}\end{cases} \tag{7}$$

Note that

$$g'_{ij}(x) = \begin{cases} f'(x), & (i,j)\in I_*, \\ \rho((x-c_{ij}) + (y_{ij}/\rho)), & (i,j)\in I_c, \\ 0 & \text{otherwise.}\end{cases} \tag{8}$$

and

$$g''_{ij}(x) = \begin{cases} f''(x), & (i,j)\in I_*, \\ \rho, & (i,j)\in I_c, \\ 0 & \text{otherwise.}\end{cases} \tag{9}$$

5 Implementation

Finally, we tie it all together.

We start with the optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in I_*} f(x_{ij}) \\ \text{subject to} \quad & x_{ij} = c_{ij} \text{ for all } (i,j)\in I_c, \\ & \lambda(X) = (\lambda_1,\dots,\lambda_n), \end{aligned}$$

in which the domain X consists of real symmetric matrices. We enforce the eigenvalue constraints by the change of variables $X = Q\Lambda Q^T$:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in I_*} f(x_{ij}) \\ \text{subject to} \quad & x_{ij} = c_{ij} \text{ for all } (i,j) \in I_c, \end{aligned}$$

in which x_{ij} is defined to be the (i,j)-entry of $Q\Lambda Q^T$ and the search domain is SO(n). We move the remaining constraints to the objective function, forming an augmented Lagrangian

$$L_\rho(Q,y) = \sum_{(i,j)\in I_*} f(x_{ij}) + \frac{\rho}{2} \sum_{(i,j)\in I_c} \left[(x_{ij} - c_{ij}) + \frac{y_{ij}}{\rho}\right]^2$$

and then abbreviate this as follows:

$$L_\rho(Q,y) = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}(x_{ij}).$$

To aid in the minimization, we find the gradient and Hessian of the objective function using linearity and (5)–(6) and (7)–(9).

Our computer implementation follows Algorithm 4.1 of [3], adhering to Assumptions 5.2, 7.6, and 7.9. Most of the work is involved in successive searches for Q. These are conducted over SO(n) using the freely available Manopt package [4].

6 Nonexistence and failure

What happens when the IEP has no solution? Before the method grinds to a halt, we commonly see the objective function increase, some $*$'ed entries approach zero, and the Lagrange multipliers grow without bound. The growth of the Lagrange multipliers is the most striking phenomenon, and their behavior makes sense. Pinning the entries in place requires more and more force as the algorithm proceeds toward certain failure.

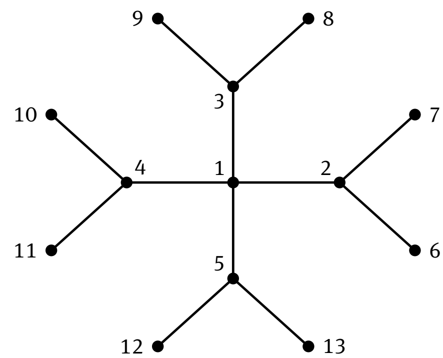
Conversely, how should the user interpret failure of the method? This question requires more research. This article was motivated by the desire to better understand the eigenvalue multiplicity lists permitted by a graph. When the numerical method fails to construct a matrix with a sequence of eigenvalues matching a desired multiplicity list, each of the following explanations is conceivable: (1) no matrix with the multiplicity list exists, (2) the numerical values for the eigenvalues are incompatible with the graph but the multiplicity list is possible with other numerical values, or (3) a matrix exists and the algorithm failed to find it.

By the way, we have tried including the eigenvalues' numerical values in the search process so that only their multiplicities must be specified in the problem statement. However, we have not had much success with this approach. Additional constraints must accompany the new variables to keep the eigenvalues bounded and possibly separated and/or ordered. These additional constraints tend to add wrinkles to the objective function and discourage convergence.

7 Additional example

We close with a second example.

Example 2. Does there exist a real symmetric matrix with eigenvalues of multiplicities $3, 3, 2, 2, 1, 1, 1$ and the following graph?



The answer appears to be yes. Let

$$
\begin{aligned}
a &= 4.243503532482240,\\
b &= 4.295559595710024,\\
c &= 3.564056939327627,\\
d &= 2.578927778314404,\\
e &= 4.317952154165707,\\
f &= 1.148138550259417,\\
g &= 0.994692505341474,\\
h &= 1.601375695035941,\\
i &= 1.515110423602855,\\
j &= 0.843140135542088,\\
k &= 0.882675562174211.
\end{aligned}
$$

The matrix

$$
\left[\begin{array}{c|c|c|c|c|cc|cc|cc|cc}
a & f & f & g & h & & & & & & & & \\ \hline
f & 4 & & & & 1 & 1 & & & & & & \\ \hline
f & & 4 & & & & & 1 & 1 & & & & \\ \hline
g & & & b & & & & & & i & i & & \\ \hline
h & & & & c & & & & & & & j & k \\ \hline
 & 1 & & & & 3 & & & & & & & \\
 & 1 & & & & & 3 & & & & & & \\ \hline
 & & 1 & & & & & 3 & & & & & \\
 & & 1 & & & & & & 3 & & & & \\ \hline
 & & & i & & & & & & 4 & & & \\
 & & & i & & & & & & & 4 & & \\ \hline
 & & & & j & & & & & & & d & \\
 & & & & k & & & & & & & & e
\end{array}\right]
$$

is symmetric and has the requested graph, and its eigenvalues have been computed to be

$$\begin{array}{c} 1.000000000000001 \\ 2.000000000000000 \\ 2.000000000000000 \\ 2.000000000000002 \\ 2.999999999999999 \\ 3.000000000000000 \\ 3.000000000000000 \\ 4.000000000000000 \\ 4.000000000000000 \\ 5.000000000000000 \\ 5.000000000000002 \\ 6.000000000000001 \\ 6.999999999999999 \end{array}$$

There are four clusters of eigenvalues: three eigenvalues near $\lambda = 2$, three eigenvalues near $\lambda = 3$, two eigenvalues near $\lambda = 4$, and two eigenvalues near $\lambda = 5$. All clusters are narrower than 3×10^{-15}. □

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] Dimitri P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1982.

[3] E. G. Birgin and J. M. Martínez. *Practical augmented Lagrangian methods for constrained optimization*, volume 10 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[4] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

[5] Moody T. Chu. Inverse eigenvalue problems. *SIAM Rev.*, 40(1):1–39, 1998.

[6] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1999.

[7] Charles R. Johnson and Carlos M. Saiago. *Eigenvalues, multiplicities and graphs*, volume 211 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2018.

[8] Charles R. Johnson, Brian D. Sutton, and Andrew J. Witt. Implicit construction of multiple eigenvalues for trees. *Linear Multilinear Algebra*, 57(4):409–420, 2009.

[9] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.