Linear Algebra and its Applications 615 (2021) 42–76

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

A characterization of trace-zero sets realizable by compensation in the SNIEP

Carlos Marijuán [a,1], Julio Moro [b,*,2]

[a] *Departamento de Matemática Aplicada, Universidad de Valladolid/IMUVA, Valladolid, Spain*
[b] *Departamento de Matemáticas, Universidad Carlos III de Madrid, Leganés, Spain*

ARTICLE INFO

Article history:
Received 16 May 2017
Accepted 18 December 2020
Available online 5 January 2021
Submitted by R. Brualdi

MSC:
15A18
15A42
65F15
65F18

Keywords:
Symmetric nonnegative matrix
Symmetric nonnegative inverse eigenvalue problem
C-realizability

ABSTRACT

The symmetric nonnegative inverse eigenvalue problem (SNIEP) is the problem of characterizing all possible spectra of entry-wise nonnegative symmetric matrices of given dimension. A list of real numbers is said to be symmetrically realizable if it is the spectrum of some nonnegative symmetric matrix. One of the most general sufficient conditions for realizability is so-called C-realizability, which amounts to some kind of compensation between positive and negative entries of the list. In this paper we present a combinatorial characterization of C-realizable lists with zero sum, together with explicit formulas for C-realizable lists having at most four positive entries. One of the consequences of this characterization is that the set of zero-sum C-realizable lists is shown to be a union of polyhedral cones whose faces are described by equations involving only linear combinations with coefficients 1 and −1 of the entries in the list.

* Corresponding author.
E-mail addresses: marijuan@mat.uva.es (C. Marijuán), jmoro@math.uc3m.es (J. Moro).

[1] The research of this author was partially supported by the Spanish Ministerio de Economía y Competitividad under grants PGC2018-096446-B-C21 and MTM2017-90682-REDT, and by the Consejería de Educación de la Junta de Castilla y León (Spain) under grant VA166G18.
[2] The research of this author was partially supported by the Spanish Ministerio de Economía y Competitividad under grants MTM2017-84098-P and MTM2017-90682-REDT.

https://doi.org/10.1016/j.laa.2020.12.021
0024-3795/

1. Introduction

A self-conjugate list of complex numbers is *realizable* if it is the spectrum of some entrywise nonnegative matrix. The *nonnegative inverse eigenvalue problem* is the problem of characterizing all possible realizable lists. If the list is real we have the *real nonnegative inverse eigenvalue problem* (hereafter RNIEP). A complete solution to these problems is known only for spectra of size $n \leq 4$ (see [11], [19], [9]) and $n = 5$ with trace 0 (see [10], [19]).

If, in the RNIEP, we require that the nonnegative matrix be symmetric, we have the *symmetric nonnegative inverse eigenvalue problem* (hereafter SNIEP). Both problems, RNIEP and SNIEP, are equivalent for $n \leq 4$ and are different and remain open for $n \geq 5$ (see [8], [4], [9]). One of the most general sufficient conditions for the SNIEP follows from the *Soules approach* by means of the so-called Soules matrices, first introduced in [17], and later characterized by Elsner, Nabben and Neumann in [6]. The SNIEP for size $n = 5$ with trace 0 has been solved by Spector [18].

Many different points of view have been adopted to find sufficient conditions for both the RNIEP and the SNIEP. In [12–14], the authors construct maps of several sufficient conditions for the RNIEP and SNIEP, respectively, in which they prove inclusion or independence relations between them. Two of the strongest sufficient conditions are (i) the so called *C- realizability* (see Definition 2.1 below), due to Borobia, Moro and Soto [2], which might be roughly summarized as *realizability by compensation*, and (ii) the sequence of *Soto p* conditions, due to Soto [16]. According to these maps, C-realizability and the union of all Soto p conditions contain, as a particular case, most of the sufficient conditions in the literature for the RNIEP.

Recently, Ellard and Šmigoc [5] have extended the Soules approach to what they call *piecewise* Soules, and have proposed a new recursive method to construct symmetrically realizable lists. The main contribution in [5] is proving the equivalence among the four sufficient conditions mentioned above: C-realizability, the union of all Soto p, piecewise Soules and the Ellard-Šmigoc method. As a consequence, C-realizability is shown to be a criterion of *symmetric* realizability. This turn of events is hardly uncommon in the area: several sufficient conditions which were first obtained for the RNIEP have later turned out to be sufficient conditions for the SNIEP as well.

For all of the above, it seems clear that the set of C-realizable lists is a reasonably large part of the set of all symmetrically realizable lists, and that understanding its structure and properties could shed some light on the problem of characterizing the full set of real realizable lists. The fact that this set can be approached from different points of view is an additional advantage when studying it. The main goal of this paper is to characterize, using a mostly combinatorial approach, the subset of C-realizable lists *with zero sum*. In particular, this set will be shown to be a union of polyhedral cones, with each of these cones given by an inequality which only involves linear combinations with coefficients 1 and -1 of the entries in the list under study. Hopefully, this will be a first step towards characterizing the whole set of C-realizable lists.

The paper is organized as follows: in Section 2 we define C-realizability and briefly present some of its basic properties. Section 3 presents the basic ideas behind the combinatorial procedure we will use to characterize C-realizability. In particular, it is shown that one can associate both a directed rooted tree and a so-called nested bracket structure to each C-realizing procedure. These will be the main concepts in the characterization. The general combinatorial procedure leading to the characterization is completely described in Section 4, while Section 5 contains the statement and proof of our main result, Theorem 5.1. Finally, Section 6 presents by way of example the explicit formulas derived from Theorem 5.1 when there are four positive entries or less in the original candidate list. Appendix A contains the proof of a somewhat fundamental result (Lemma 3.2) which removes a basic ambiguity and justifies the overall approach to C-realization taken in this paper.

2. C-realizability: definition and basic properties

In what follows, a *list* is a collection $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ of real numbers with possible repetitions. C-realizability is a kind of *realizability by compensation* based on the following three known results:

- *Rule 1*: Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be a realizable list with $\lambda_1 \geq |\lambda|$ for $\lambda \in \Lambda$ and let $\epsilon > 0$. Then $\{\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n\}$ is also realizable.
- *Rule 2*: Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be a realizable list with $\lambda_1 \geq |\lambda|$ for $\lambda \in \Lambda$ and let $\epsilon > 0$. Then $\{\lambda_1 + \epsilon, \lambda_2 - \epsilon, \lambda_3, \ldots, \lambda_n\}$ is also realizable (see [7]).
- *Rule 3*: Let Λ_1 and Λ_2 be realizable lists. Then the list $\Lambda_1 \cup \Lambda_2$ is realizable.

This suggests considering three types of 'moves', transforming realizable lists into other realizable lists. Suppose, as above, that $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \subset \mathbb{R}$ is a realizable list with $\lambda_1 \geq |\lambda|$ for $\lambda \in \Lambda$, and let $\epsilon > 0$. We consider:

Move of type 1: $\Lambda \mapsto \{\lambda_1 + \epsilon, \lambda_2, \ldots, \lambda_n\}$;
Move of type 2: $\Lambda \mapsto \{\lambda_1 + \epsilon, \lambda_2 - \epsilon, \ldots, \lambda_n\}$;

And, if Λ_1 and Λ_2 are realizable lists, the third type of move is just the union:

Move of type 3: $(\Lambda_1, \Lambda_2) \mapsto \Lambda_1 \cup \Lambda_2$.

Definition 2.1. (see [2]) Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be a list of real numbers. We say that Λ is **C-realizable** if it can be reached starting from the n realizable lists

$$\{0\}\ \{0\}\ \ldots\ \{0\} \tag{1}$$

and successively applying, in any order and any number of times, any of the moves of types 1, 2 or 3.

Example. The list $\{10, 7, 3, -5, -6, -8\}$ is C-realizable, since it can be reached starting from $\{0\}$ $\{0\}$ $\{0\}$ $\{0\}$ $\{0\}$ $\{0\}$ and performing the moves

$$\begin{aligned}
&\{0,0\}\ \{0,0\}\ \{0,0\}\\
&\{8,-8\}\ \{6,-6\}\ \{3,-3\}\\
&\{8,-8\}\ \ \{6,-6,3,-3\}\\
&\{8,-8\}\ \ \{7,-6,3,-4\}\\
&\{8,-8,7,-6,3,-4\}\\
&\{9,-8,7,-6,3,-5\}\\
&\{10,-8,7,-6,3,-5\}
\end{aligned}$$

It is clear that any C-realizable list is, in particular, realizable, since the three types of move preserve realizability. However, not all realizable lists are C-realizable (see below). Nevertheless, C-realizability turns out to be one of the strongest sufficient conditions for the RNIEP, in the sense that it includes any other known sufficient condition except the ones given by Perfect in [15] (see [2,12,14]). Recall that, as mentioned in the Introduction, any C-realizable list is, in particular, realizable by a symmetric matrix.

Several necessary conditions are known for C-realizability, some of which are connected to the concept of majorization:

Definition 2.2. Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ be two vectors in $\mathbb{R}^n$ with their entries ordered decreasingly. We say that x **weakly majorizes** y if

$$\sum_{j=1}^{k} x_j \geq \sum_{j=1}^{k} y_j, \qquad k = 1, \ldots, n$$

We say that x **majorizes** y if, additionally,

$$\sum_{j=1}^{n} x_j = \sum_{j=1}^{n} y_j.$$

Negative subdivisions will also play a relevant role in our discussion:

Definition 2.3. The set $\{\rho_1, \ldots, \rho_{i-1}, \gamma, \delta, \rho_{i+1}, \ldots, \rho_n\}$ is a **negative subdivision** of $\{\rho_1, \ldots, \rho_{i-1}, \rho_i, \rho_{i+1}, \ldots, \rho_n\}$ if $\gamma + \delta = \rho_i$ with $\gamma, \delta, \rho_i < 0$.

Some necessary conditions for C-realizability are as follows (see [2]):

Lemma 2.4. *If Λ is a C-realizable list, then so is any list obtained by successively applying negative subdivisions on Λ.*

Lemma 2.5. *Let $\Lambda = \{\lambda_1, \dots, \lambda_n, -\mu_m, \dots, -\mu_1\}$ be a C-realizable list such that $\lambda_1 \geq \cdots \geq \lambda_n > 0$ and $\mu_1 \geq \cdots \geq \mu_m \geq 0$. Set $\alpha = (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n$ and $\beta = (\mu_1, \dots, \mu_m) \in \mathbb{R}^m$ and let $\widetilde{\alpha}, \widetilde{\beta} \in \mathbb{R}^p, p = max\{n, m\}$, be the vectors obtained respectively from α, β by adjoining an appropriate number of zeros to one of them. Then $\widetilde{\alpha}$ weakly majorizes $\widetilde{\beta}$.*

Example. The list $\{4, 1, 1, -3, -3\}$ is known to be symmetrically realizable [18], but does not satisfy weak majorization. Hence, it is not C-realizable. This shows that *the set of C-realizable lists is a strict subset of the set of symmetrically realizable ones.*

Lemma 2.6. *Let Λ be any list of real numbers, and let $\widetilde{\Lambda}$ be the same list with any number of zeros appended to it. Then $\widetilde{\Lambda}$ is* C*-realizable if and only if Λ is.*

Proof. If Λ is C-realizable, then a number of moves of type 3 leads from Λ to $\widetilde{\Lambda}$, so $\widetilde{\Lambda}$ is C-realizable. Conversely, suppose that $\widetilde{\Lambda}$ is C-realizable via a sequence of moves of types 1, 2 or 3. The same moves, only removing the appended zeros from the intermediate lists, show the C-realizability of Λ. □

It should be noted that this property of not being affected by the addition (or removal) of zero entries in the list is true for C-realizability, but not for realizability: the existence of non-realizable lists which are the nonzero part of the spectra of some nonnegative matrices of large enough dimension is well known (see Boyle & Handelman [3]). Thus, adding or removing zeros from a list can make a difference with regard to realizability, but not for C-realizability. Therefore, in what follows we will only consider lists with no zero entries.

In this paper, we focus our attention on the C-realizability of lists *with zero trace* (i.e., *with zero sum*) as a first step towards analyzing the general case of arbitrary nonnegative trace. One special feature of C-realizability in the case of zero trace is the following:

Corollary 2.7. *Let the list $\Lambda = \{\lambda_1, \dots, \lambda_n, -\mu_m, \dots, -\mu_1\}$ be such that $\lambda_1 \geq \cdots \geq \lambda_n > 0$ and $\lambda_1 \geq \mu_1 \geq \cdots \geq \mu_m > 0$. Suppose that Λ has zero trace, i.e.*

$$\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{m} \mu_i, \tag{2}$$

and that $n > m$. Then, Λ is not C*-realizable.*

Proof. If Λ were C-realizable and $n > m$, the majorization guaranteed by Lemma 2.5 would imply

$$\sum_{i=1}^{m} \lambda_i \geq \sum_{i=1}^{m} \mu_i,$$

which contradicts (2), since $\lambda_{m+1} > 0$. □

This means that we may assume that there are at least as many negative entries in Λ as there are positive ones. This, together with the exclusion of zero entries, suggests the following definition:

Definition 2.8. We say that a list

$$\Lambda = \{\lambda_1, \ldots, \lambda_n, -\mu_m, \ldots, -\mu_1\},$$

of real numbers is **$\mathbf{T}_0$-admissible** if $n \leq m$, $\lambda_1 \geq \ldots \geq \lambda_n > 0$, $\lambda_1 \geq \mu_1 \geq \ldots \geq \mu_m > 0$, and

$$\sum_{i=1}^{n} \lambda_i = \sum_{j=1}^{n} \mu_j. \tag{3}$$

3. Preliminaries

The main idea underlying C-realizability is that of *compensation*, i.e., the fact that certain lists of real numbers having a positivity surplus may transfer part of that excess to other lists having a positivity deficit, thereby decreasing, or eventually overcoming, such deficit (see [1]). Systematizing this transfer procedure will lead us to a characterization of the set of C-realizable lists of real numbers with zero sum: given a T_0-admissible list

$$\Lambda = \{\lambda_1, \ldots, \lambda_n, -\mu_m, \ldots, -\mu_1\}, \tag{4}$$

as defined above, we want to determine whether it is C-realizable or not.

We begin with a toy example to illustrate our ideas. Consider the following three lists:

$$\Lambda_1 = \{12, -3, -8\}, \qquad \Lambda_2 = \{6, -5\}, \qquad \Lambda_3 = \{9, -5, -5\}.$$

None of them has zero sum, but both $\Lambda_1 \cup \Lambda_3$ and $\Lambda_2 \cup \Lambda_3$ have. The sum of the entries in either Λ_1 or Λ_2 is $+1$, so we call Λ_1 and Λ_2 *positive* lists, while the sum of entries in Λ_3 is -1, so we say the list Λ_3 is *negative*. Since the sum of entries in both $\Lambda_1 \cup \Lambda_3$ and $\Lambda_2 \cup \Lambda_3$ is zero, we say they are *neutral*.

We shall now show that $\Lambda_1 \cup \Lambda_3$ is C-realizable by exhibiting a C-realization, i.e., a sequence of moves of types 2 and 3, as described in Section 2, leading from the zero spectrum to $\Lambda_1 \cup \Lambda_3$. On the other hand, we shall see that $\Lambda_2 \cup \Lambda_3$ is not C-realizable, and why it is not.

In order to C-realize $\Lambda_1 \cup \Lambda_3$ we may start, for instance, with Λ_3 by using a move of type 2 to construct the set $\{5, -5\}$, and then merge it with an initial $\{0\}$ set to obtain $\{5, 0, -5\}$. Then, another move of type 2 leads to $\{9, -4, -5\}$. Notice that the target, 9, for the dominant entry has been reached, but one of the negative entries, -4, has not yet

reached its final value of -5. Also, the difference between -4 and -5 is precisely the sum -1 of all entries in Λ_3. Next, we try to C-realize Λ_1 on its own by using moves of type 2 on $\{0, 0, 0\}$ to obtain $\{11, -3, -8\}$. Now both negative entries -3, -8 have reached their final value, while the dominant one, 11, has not yet reached the desired value of 12. Again, the difference between 11 and 12 is equal to the sum $+1$ of the entries in Λ_1. The idea now is that, since Λ_1 has a surplus $+1$, this surplus should be transferred to Λ_3 in order to compensate its deficit -1. We can do this by merging the two lists into $\{11, -3, -8, 9, -4, -5\}$ with a move of type 3, and then making a last move of type 2 to complete $\{12, -3, -8, 9, -5, -5\} = \Lambda_1 \cup \Lambda_3$. This proves that $\Lambda_1 \cup \Lambda_3$ is C-realizable.

On the other hand, if we take Λ_2 and Λ_3, we may do the same to Λ_3 as above, leading to $\{9, -4, -5\}$, and proceed with Λ_2 in a similar way to that done with Λ_1, leading to $\{5, -5\}$. However, once we merge both lists into $\{9, -4, -5, 5, -5\}$, we notice that the 5 entry cannot be raised up to the desired level of 6, because it is no longer dominant in the merged list, and the only moves allowing us to increase a number in the list are moves of type 2, which only increase the largest entry in the list. In fact, one can easily show that $\Lambda_2 \cup \Lambda_3$ is not C-realizable by using Theorem 6.1 below (see also [18], Theorem 5), which characterizes C-realizable spectra with two positive entries.

These examples illustrate the basic positivity transfer mechanism which underlies any C-realizing procedure, as well as the need to impose certain conditions on the entries in the list in order to guarantee that the compensation procedure can be successfully performed. Thus, we characterize all C-realizable T_0-admissible lists by systematizing the description of all possible C-realizing procedures as follows:

1. first, we prove in §3.1 that every possible C-realization procedure can be represented by a rooted tree;
2. then we show in §4 that, given any rooted tree representing one such C-realizing procedure, one can write down a set of explicit conditions (see (21) and (22) below) on the entries of a generic T_0-admissible list Λ in such a way that any list Λ satisfying that set of conditions will be C-realizable via the procedure represented by that rooted tree.

This allows us to describe the set of all C-realizable T_0-admissible lists by exhaustion.

In order to keep track of the set of conditions mentioned in step 2 above, and to be able to write them down explicitly, we present a systematic procedure in Section 4 that is based on the concept of 'target list' (see §3.2 for a formal definition), and which describes all possible C-realizing procedures compatible with the given set of target lists. Each of these different C-realizing procedures will give rise to different sets of conditions.

The procedure presented in §4 is organized in levels: at each level new target lists are constructed by merging one positive target list with one or more negative target lists, all from the previous level. As we have seen in the example above, these merging steps may or may not lead to compensation, depending on the actual values of the entries in the original list Λ. In order to achieve compensation, certain conditions have to be met

by the actual entries of the sublists merging at each level. Some of them will be *sign conditions*, requiring that each new target list constructed via moves of type 3 has a suitable sign. Others will be *merging conditions*, which ensure that appropriate moves of type 2 can be performed after merging. Theorem 5.1 will show that sign conditions and merging conditions are both necessary and sufficient for C-realizability.

3.1. C-realization and trees

The following result shows that a rooted tree can be associated to each C-realizing procedure:

Theorem 3.1. *Any C-realizing procedure for a T_0-admissible list Λ can be uniquely represented through a rooted tree.*

Proof. Any C-realization of Λ starts with a list

$$\{0\}\ \{0\}\ \ldots\ \{0\}$$

of $\mathrm{Card}(\Lambda)$ zero lists and, after a sequence of moves of types 2 and 3, concludes in the list Λ. We may associate a digraph $T = (V, A)$ to this procedure, where the vertices in V are all sublists appearing throughout the C-realization procedure. Each move of type 2, applied to a list u and producing a list v, is described by a directed arc (u, v). Likewise, each move of type 3 applied on the lists $u_1, \cdots, u_p$ to obtain the list $u = \bigcup_{j=1}^{p} u_j$ is described by the p directed arcs $(u_1, u), \cdots, (u_p, u)$.

Notice that moves of type 2 do not modify the cardinality of the lists involved, while moves of type 3 strictly increase it. On the other hand, moves of type 2 produce lists with dominant entry strictly larger than that of the list it comes from. As a consequence of this, the digraph T is acyclic.

Furthermore, since all vertices of V have outdegree 1, there is in T a unique directed path connecting each vertex u with the final vertex Λ. This proves that T is connected and, consequently, a directed tree. Finally, T is rooted because it has a unique maximal vertex Λ. □

Let us illustrate this theorem with a specific example: consider the list

$$\Lambda = \{16,\ 13,\ 10,\ 10,\ 8,\ 4,\ -2,\ -3,\ -5,\ -6,\ -6,\ -6,\ -9,\ -12,\ -12\} \tag{5}$$

and the following C-realizing procedure for Λ:

$$
\begin{array}{c}
\{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\} \\
\{0,\,0,\,0\} \quad \{0,\,0\} \quad \{0,\,0\} \quad \{0,\,0,\,0\} \quad \{0,\,0,\,0\} \quad \{0,\,0\} \\
\{11,\,-2,\,-9\}\ \{12,\,-12\}\ \{10,\,-10\}\ \{8,\,-3,\,-5\}\ \{8,\,-2,\,-6\}\ \{4,-4\} \\
\{11,-2,\,-9,\,10,-10,\,4,\,\,-4\}\ \{12,\,-12\}\{8,\,-3,\,-5,\,8,\,-2,\,-6\} \\
\{15,\,-2,\,-9,\,10,\,-12,\,4,\,-6\}\ \{12,\,-12\}\{10,\,-3,\,-5,\,8,\,-4,\,-6\} \\
\{15,\,-2,\,-9,\,10,\,-12,\,4,\,-6\}\ \{12,\,-12,\,10,\,-3,\,-5,\,8,\,-4,\,-6\} \\
\{15,\,-2,\,-9,\,10,\,-12,\,\,4,\,-6\}\ \{13,\,-12,\,10,\,-3,\,-5,\,8,\,-5,\,-6\} \\
\{15,\,-2,\,-9,\,10,\,-12,\,4,\,-6,\,13,\,-12,\,10,\,-3,\,-5,\,8,\,-5,\,-6\} \\
\Lambda = \{16,\,-2,\,-9,\,10,\,-12,\,4,\,-6,\,13,\,-12,\,10,\,-3,\,-5,\,8,\,-6,\,-6\}
\end{array}
\tag{6}
$$

We now construct a rooted tree associated with the above C-realizing procedure: its vertices are the sublists appearing in the sequence above (ignoring the first row of zeros), and a vertex u is adjacent to a vertex v whenever v is obtained from u via a move of type either 2 or 3 (recall that, since our lists have zero sum, no move of type 1 can be performed). The tree is organized by levels, each level corresponding to one row in the process (6) above, and is represented in Fig. 1. Numbers in red indicate that the desired final value has not yet been reached, while numbers in black indicate that it has.

Remark 1. It should be noted that, for each C-realizing procedure, there is a variety of trivially equivalent C-realizing procedures which should ideally be represented by the same tree. For instance,

- if several moves of type 2 are applied to the same list before performing a move of type 3, all those moves of type 2 may be merged into a single one (two consecutive moves of type 2 with $\epsilon = \epsilon_1$ and $\epsilon = \epsilon_2$, for instance, are equivalent to one single move of type 2 with $\epsilon = \epsilon_1 + \epsilon_2$);
- if several moves of type 3 are applied to a collection $u_1, \ldots, u_p$ of lists, leading eventually to a list u without applying any move of type 2 in between, all those moves of type 3 are equivalent to a single move of type 3 on the lists $u_1, \ldots, u_p$ to obtain u.
- each row in (6) records moves of one single type, either 2 or 3, performed on lists from the previous, say $(i-1)$-st, row. If any list in the $(i-1)$-st row is not involved in any of those moves, then that list is preserved in the i-th row below, and an arc is included connecting them.

Given any C-realization procedure, if we apply these simplifications, we obtain a unique, simpler procedure whose associated rooted tree is also unique and has the property that all maximal paths (i.e., those connecting the leaves of the tree – the zero lists in the C-realizing procedure on top of Fig. 1 – with the root vertex) are alternating sequences of moves of types 2 and 3 with the same length. Note that at some step we may skip the move of type 2 just by taking $\epsilon = 0$. Similarly, some sublists may be left out of the union corresponding to that move of type 3. Without loss of generality, we

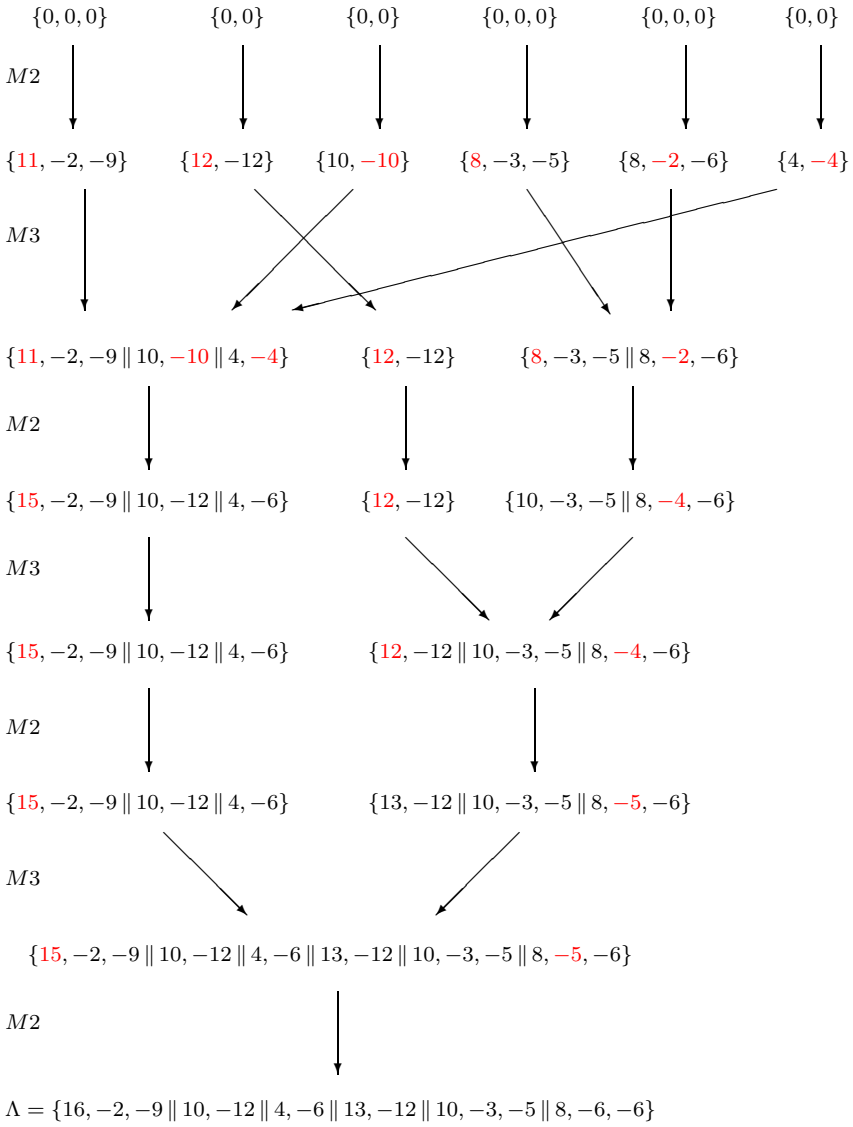

Fig. 1. Rooted tree associated with the C-realization procedure (6). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

may assume in what follows that any C-realization has already been simplified in this sense. The tree in Fig. 1, for instance, is already in this simplified form.

3.2. Target lists and QCRs

Given any C-realization procedure for a T_0-admissible list Λ, and given its corresponding associated tree, we now focus on each leaf of the tree (i.e., each set of zeroes at the top) and traverse the tree downwards in order to identify which entries of Λ are produced

by those specific zeros (in order to make it easier to trace the origin of each entry in Λ we separate with a double vertical bar those entries coming from the same subset in the third uppermost line of the tree). For instance, in the example above, one can easily see that the leftmost triplet of zeros results in the list $\{16, -2, -9\}$, the pair of zeros contiguous to it produces $\{13, -12\}$, and so on. These sets we will call *target lists*, since the whole C-realizing procedure can be interpreted as an attempt to gradually approximate those sets as final goals, to be reached at the end of the process. In view of the last line in Fig. 1, the target lists for the C-realization procedure above are

$$\{16, -2, -9\},\ \{10, -12\},\ \{4, -6\},\ \{13, -12\},\ \{10, -3, -5\},\ \{8, -6, -6\}.$$

Notice that the union of all lists is just the T_0-admissible list Λ in (5). Furthermore, a sign is assigned to each target list: a target list is said to be *positive* (resp., *negative*) if the sum of its entries is positive (resp., negative): the first, fourth and fifth lists above, for example, are positive, while the remaining three are negative. Target lists with zero sum are said to be *neutral*. At this point it is worth mentioning that whenever we have a neutral target list, the C-realizing problem can be simplified by putting aside the neutral target list and dealing with the rest of the entries: since the neutral list can be trivially C-realized via moves of type 2, if the rest of the list can be C-realized in some way, then the full list can also be C-realized. In fact, we shall see in Appendix A that the set Λ in (5) can be C-realized in a completely different way, with one of the target lists being $\{8, -3, -5\}$, which happens to be neutral (and is therefore disregarded in the analysis of Appendix A).

The approximation to Λ is made gradually as we go down the nodes in the tree, all of which are lists with zero sum. Some of these lists, namely the ones obtained after a move of type 2, we call *quasi C-realizations* (henceforth *QCRs*) since they are intermediate steps in the process of reaching the target lists. The QCR $\{15, -2, -9, 10, -12, 4, -6\}$, for instance, is an approximation to the union

$$\{16, -2, -9\} \cup \{10, -12\} \cup \{4, -6\}$$

of target lists. Thus, we may think of the C-realization procedure as being organized in subsequent levels: at each level, the process creates new QCRs by first merging old ones from the previous level via a move of type 3, and then performing a move of type 2. This will provide an improved approximation of the union of the target lists indicated by the move of type 3. Therefore, it makes sense to consider that, at each level, we first create a new target list, which is the union of the target lists in the previous level, merged by the move of type 3. We then improve the approximation to that new target list by creating a new QCR via a move of type 2.

3.3. C-realization and bracket structures

What we now show is that *any rooted tree in simplified form can be represented by what we call its corresponding* **nested bracket structure**. In order to understand this, let us go back to the procedure (6) above, and start using the crucial concept of target list, already introduced in §3.2 above: we claim that the whole C-realizing procedure (6) above corresponds to simply choosing the initial target lists

$$\begin{aligned} &\Lambda_1^{(0)} = \{16, -2, -9\}, \quad \Lambda_2^{(0)} = \{13, -12\}, \quad \Lambda_3^{(0)} = \{10, -12\}, \\ &\Lambda_4^{(0)} = \{10, -3, -5\}, \quad \Lambda_5^{(0)} = \{8, -6, -6\}, \quad \Lambda_6^{(0)} = \{4, -6\} \end{aligned} \tag{7}$$

(which can be read off from the bottom line in the tree) and then repeatedly generating successive QCRs until the union of these target lists is reached. The first nonzero row in (6), for instance, is just the sequence of QCRs for the initial target lists $\Lambda_j^{(0)}$, $i = 1, \cdots, 6$.

The second nonzero row in (6) is obtained by merging some QCRs, e.g., the one of $\Lambda_1^{(0)}$ with those of $\Lambda_3^{(0)}$ and $\Lambda_6^{(0)}$. This is done with the prospect of later being able to transfer positivity from $\Lambda_1^{(0)}$ into $\Lambda_3^{(0)}$ and $\Lambda_6^{(0)}$, which are both negative, via a move of type 2. The same motivation lies behind merging the QCRs of (positive) $\Lambda_4^{(0)}$ with that of (negative) $\Lambda_5^{(0)}$. This first round of mergings can be represented through the brackets

$$[1, 3, 6], [2], [4, 5], \tag{8}$$

where the one in the middle indicates that the QCR of $\Lambda_2^{(0)}$ has not been subject to any merging operation at this stage. The leftmost bracket corresponds to creating a new, aggregated, target list

$$\Lambda_1^{(1)} = \Lambda_1^{(0)} \cup \Lambda_3^{(0)} \cup \Lambda_6^{(0)} = \{16, -2, -9, 10, -12, 4, -6\}, \tag{9}$$

which is again positive (the total sum of its entries is +1). The target list $\Lambda_2^{(0)} = \Lambda_2^{(1)}$ remains unchanged (thus positive), while the third bracket [4, 5] corresponds to an updated target list

$$\Lambda_3^{(1)} = \Lambda_4^{(0)} \cup \Lambda_5^{(0)} = \{10, -3, -5, 8, -6, -6\},$$

which is negative (its total sum is −2). Again, the sign of each of the updated target lists $\Lambda_j^{(1)}$, $j = 1, 2, 3$ leads to a sign condition on the corresponding entries of Λ, which has to be satisfied in order to be able to continue applying the C-realizing procedure (6).

The next row in (6) (i.e., its third nonzero row) corresponds to another round of QCRs, this time for the three updated target lists $\Lambda_j^{(1)}$, $j = 1, 2, 3$.

The fourth nonzero row in (6) corresponds to merging positive and negative target lists, namely,

$$\Lambda_2^{(2)} = \Lambda_2^{(1)} \cup \Lambda_3^{(1)} = \{13,\, -12,\, 10,\, -3,\, -5,\, 8,\, -6,\, -6\,\},$$

which turns out to be a new negative target list with sum -1. The target list $\Lambda_1^{(2)} = \Lambda_1^{(1)}$ remains unchanged at this stage. This second level of merging can be represented through the brackets

$$[\,[1,\, 3,\, 6]\,]\;,\;[\,[2]\,,\,[4,\, 5]\,]\,. \tag{10}$$

The transition from the fourth to the fifth nonzero row in (6) corresponds to generating a QCR for $\Lambda_2^{(2)}$, while the one from the fifth to the sixth nonzero row is another move of type 3 merging $\Lambda_2^{(2)}$ with $\Lambda_1^{(2)}$, which makes $\Lambda_1^{(3)} = \Lambda_1^{(2)} \cup \Lambda_2^{(2)} = \Lambda$ the final target list. This can be denoted with the bracket structure

$$[\;[\,[1,\, 3,\, 6]\,]\;,\;[\,[2]\,,\,[4,\, 5]\,]\;]\,, \tag{11}$$

which represents the whole procedure in a more compact way than the rooted tree. The last target list $\Lambda = \Lambda_1^{(3)}$ is finally reached via a move of type 2, which concludes the C-realizing procedure.

3.4. Sign and merging conditions

We have seen in §3.3 above how the rooted tree in Fig. 1 has an associated bracket structure (11). However, that structure by itself is not enough to guarantee that the C-realizing procedure (6) succeeds, since it only reflects which moves of type 3 are performed throughout the procedure. The moves of type 2 that can actually be performed in order to complete the C-realizing procedure are codified via some further conditions, namely those we shall call *sign* and *merging* conditions.

The question here is: given the C-realizing procedure (6), which T_0-admissible lists, other than the specific list Λ given in (5), can be C-realized using that exact same procedure? Which conditions have to be satisfied by the specific entries of those lists to guarantee that the moves of type 2 in procedure (6) can be performed in their entirety, proving those lists to be C-realizable?

First, since the way each QCR is generated in (6) depends on the sign of the successive target lists, it is clear that the sign of each target list at each level must be imposed if (6) is to be run to completion. These we call *sign conditions*, and have to be imposed starting from the initial level 0.

On the other hand, once a move of type 3 is performed, a move of type 2 typically follows, but this can only be done if the dominant entry of the QCR corresponding to the only positive target list being merged is not smaller than the largest dominant entry among the other negative target lists participating in the merger. Take, for instance the move of type 3 given by (9), and the corresponding list $\{11, -2, -9, 10, -10, 4, -4\}$. The C-realizing procedure can move on to the next QCR, $\{15, -2, -9, 10, -12, 4, -6\}$,

only because the dominant entry 11 of the QCR $\{11, -2, -9\}$ of the positive target list $\Lambda_1^{(0)}$ is not smaller than 10, the largest dominant entry among the QCRs of the two negative target lists $\Lambda_3^{(0)}$ and $\Lambda_6^{(0)}$. These conditions we call *merging conditions*, and have to be imposed at every level of the C-realizing procedure.

3.5. Moves of type 2 decrease negative entries

When performing moves of type 2

$$\{\lambda_1, \lambda_2, \ldots, \lambda_n\} \longrightarrow \{\lambda_1 + \epsilon, \lambda_2 - \epsilon, \ldots, \lambda_n\},$$

the entry λ_2 which is decreased can, in principle, have any sign. However, we will show that without loss of generality we may always assume that λ_2 is negative.

Lemma 3.2. *Let Λ be a C-realizable T_0-admissible list such that there exists a C-realizing procedure containing moves of type 2 with $\lambda_2 > 0$. Then Λ can be C-realized through another procedure where all moves of type 2 are performed on negative λ_2s.*

Since this result is of independent interest, and its proof requires a somewhat different notation than the one employed up to this point, we defer its proof to Appendix A. In any case, from now on, whenever a move of type 2 is considered, it will be assumed that the entry λ_2 which is decreased is negative.

4. The general procedure

We are now in a position to describe in detail a general combinatorial procedure by which necessary and sufficient conditions for C-realizability are generated for T_0-admissible lists.

4.1. Step zero

Let

$$\Lambda = \{\lambda_1, \ldots, \lambda_n, -\mu_m, \ldots, -\mu_1\} \tag{12}$$

be such a T_0-admissible list (recall that $n \leq m$). First, we partition Λ into n **target lists** $\Lambda_j^{(0)}$, $j = 1, \ldots, n$ **for level zero** in such a way that each $\Lambda_j^{(0)}$ contains exactly one positive entry. Thus, we have the initial partition Π_0:

$$\Lambda = \Lambda_1^{(0)} \cup \ldots \cup \Lambda_n^{(0)}. \tag{13}$$

For each j, we say the list $\Lambda_j^{(0)}$ is **positive** or **negative** according to the sign of the sum of its entries (as already mentioned, sublists with zero sum may be set aside to be C-realized separately).

Next, we identify each $\Lambda_j^{(0)}$ with a corresponding index $j^{(0)}$, which, by extension, is also said to be positive or negative, according to the sign of the list $\Lambda_j^{(0)}$. The list of indices

$$1^{(0)}, \dots, n^{(0)} \tag{14}$$

together with the n-vector $(+, \cdots, -)$ of their signs, constitutes the so-called **configuration C_0 at level** 0.

We further associate with each index $j^{(0)}$ the quantity

$$\lambda_j^{(0)} = \lambda_j,$$

i.e., the dominant entry in the target list $\Lambda_j^{(0)}$.

To each such index $j^{(0)}$ we associate its so-called **local quasi C-realization (QCR.j)**, which is a list defined as follows: suppose $\Lambda_j^{(0)} = \{\lambda_j, -\mu_{j_1}, -\mu_{j_2}, \dots, -\mu_{j_{k_j}}\}$; then a partial realization of $j^{(0)}$ is either

- [**QCR.**j^+]: the list $\left\{\sum_{s=1}^{k_j} \mu_{j_s}, -\mu_{j_1}, -\mu_{j_2}, \dots, -\mu_{j_{k_j}}\right\}$ whenever $j^{(0)}$ is positive, or
- [**QCR.**j^-]: a list of the form $\{\lambda_j, -\widetilde{\mu}_1, -\widetilde{\mu}_2, \dots, -\widetilde{\mu}_{k_j}\}$, where

$$0 \leq \widetilde{\mu}_s \leq \mu_{j_s},\ s = 1, \dots, k_j \quad \text{and} \quad \sum_{i=1}^{k_j} \widetilde{\mu}_i = \lambda_j,$$

  whenever the index $j^{(0)}$ is negative.

Such local quasi-C-realizations can be trivially achieved via moves of types 3 and 2 on $n + m$ zero singletons (1), starting with n moves of type 3 creating n lists of zeros with the respective cardinalities of $\Lambda_1^{(0)}, \dots, \Lambda_n^{(0)}$. The reunion of the local QCRs for all $j^{(0)},\ j = 1, \dots, n$ is the QCR at level zero, denoted $\mathbf{QCR}_0$.

To conclude the zero-th step, if the index $j^{(0)}$ is positive, we additionally associate with $j^{(0)}$ a second quantity

$$S_j^{(0)} = \sum_{-\mu_k \in \Lambda_j^{(0)}} \mu_k, \tag{15}$$

namely, the dominant entry in the local QCR.j^+ of $j^{(0)}$ (we make the convention that $S_j^{(0)} = 0$ if $\Lambda_j^{(0)} = \{\lambda_j\}$). Thus, each negative index has one quantity attached to it, while each positive index has two.

4.2. Step one: partition rules

Next, we group the signed indices in the configuration C_0 into an **ordered partition** Π_1, which is the result of distributing the indices in C_0 into a certain number, say n_1, of packages, separated by brackets, according to the following three rules:

R1: each bracket contains at most one positive index,
R2: that positive index (if there is one) is the smallest among the indices within that bracket, and
R3: the rightmost positive index in the configuration is grouped with at least one negative index.

The reason for imposing rule R1 is that there cannot be two or more positive indices in the same bracket, because then some of the positivity would be wasted, since only the dominant entry can be increased via moves of type 2 (recall that, due to the zero-trace condition, no positivity can possibly be wasted in the compensation process if we want to C-realize). It may happen, however, that some package contains only one positive single index (see, for instance, the bracket $[\,2\,]$ in (8) above). That is admissible, *unless that package is the rightmost one*: the rightmost positive index cannot stand on its own because, again, in that case its positivity would be wasted, making C-realizability impossible. This is why rule R3 needs to be imposed. It may even be that *no positive index is contained in some of the brackets*: consider, for instance, the list $\Lambda = \{7, 5, 2, -4, -4, -6\}$ partitioned as the union of

$$\Lambda_1^{(0)} = \{7, -4\}, \quad \Lambda_2^{(0)} = \{5, -6\}, \quad \Lambda_3^{(0)} = \{2, -4\},$$

which is C-realizable via the procedure

$$\begin{array}{ccc} \{0,\,0\} & \{0,\,0\} & \{0,\,0\} \\ \{4,\,-4\} & \{5,\,-5\} & \{2,\,-2\} \\ \{4,\,-4\,\|\,2,\,-2\} & \{5,\,-5\} & \\ \{6,\,-4\,\|\,2,\,-4\} & \{5,\,-5\} & \\ \{6,\,-4\,\|\,2,\,-4\,\|\,5,\,-5\} & & \\ \{7,\,-4\,\|\,2,\,-4\,\|\,5,\,-6\} & & \end{array} \tag{16}$$

which corresponds to the bracket structure $[\,[\,1,\,3\,],\,[\,2\,]\,]$, where the index 2 is negative and stands alone, with no positive index in its bracket. Notice that the bracket structure $[\,[\,1,\,2,\,3\,]\,]$ gives no valid C-realizing procedure, since the first move of type 3 would bring together

$$\{4,\,-4\,\|\,5,-5\,\|\,2,-2\}$$

and neither of the 4 or 2 entries in that list can possibly be increased to 5, since they are no longer dominant in the merged list. Situations like this one motivate rule R2 above, which tries to make sure that each positive entry in the corresponding sublist is dominant within that list.

Finally, although they are not essential, it will be convenient to make the two following conventions: within each bracket, indices are written in increasing order, and brackets are written by increasing order of its only first positive index (in case there is one).

Our next step is to define new, updated **target lists for level** 1**:** if the j-th bracket in Π_1 is $[j_1^{(0)}, j_2^{(0)}, \dots, j_{s_j}^{(0)}]$, we define

$$\Lambda_j^{(1)} = \bigcup_{t=1}^{s_j} \Lambda_{jt}^{(0)}.$$

Furthermore, we identify the j-th bracket in Π_1 with a new index $j^{(1)}$. This gives rise to a new **configuration** C_1 **at level** 1 with indices

$$1^{(1)}, 2^{(1)}, \dots, n_1^{(1)},$$

together with the n_1-vector of their signs, where each $j^{(1)}$ has the same sign, positive or negative, of the corresponding target list $\Lambda_j^{(1)}$. Thus, we have transformed the initial configuration C_0 into a new configuration, C_1.

Next, we update the dominant λs: for each index $j^{(1)} = [j_1^{(0)}, j_2^{(0)}, \dots, j_{s_j}^{(0)}]$ in Π_1, we set $\lambda_j^{(1)}$ as the dominant entry in $\Lambda_{j_1}^{(0)} \cup \Lambda_{j_2}^{(0)} \cup \ldots \cup \Lambda_{j_{s_j}}^{(0)}$. In the most common case when the first index $j_1^{(0)}$ is positive, one can easily check that $\lambda_j^{(1)}$ is just the dominant entry in $\Lambda_{j_1}^{(0)}$.

The next step in our procedure would be to obtain a new, local QCR for each index $j^{(1)}$ in Π_1. All these local QCRs of the indices $j^{(1)}$ for $j = 1, \dots, n_1$ will be the $\mathbf{QCR}_1$ associated with the configuration C_1. For the sake of conciseness, we will not go into details at this stage (see step 4 in §4.3 below for more details).

Finally, step one is completed by updating the sums $S_j^{(0)}$ as defined in (15), only for those $j^{(1)}$ in Π_1 which happen to be positive: suppose $j^{(1)} := [j_1^{(0)}, j_2^{(0)}, \dots, j_{s_j}^{(0)}]$ is one such positive index. Then

$$S_j^{(1)} = S_{j_1}^{(0)} - \sum_{\gamma \in \Lambda_{j_2}^{(0)} \cup \cdots \cup \Lambda_{j_{s_j}}^{(0)}} \gamma, \tag{17}$$

i.e., at level 1, the corresponding $S_j^{(1)}$ is the result of adding the negativity of all negative indices in the bracket at level zero, generating $j^{(1)}$ to the sum $S_{j_1}^{(0)}$ associated with the only dominant positive index in that bracket at level 1.

Notice that the entries in Λ have to satisfy certain conditions if we want the C-realizing procedure to continue: for instance, the entries have to produce the positivities and

negativities given by the indices $j^{(0)}$ at level zero: thus, each bracket $[j_1^{(0)}, j_2^{(0)}, \ldots, j_{s_j}^{(0)}]$ in the ordered partition Π_1 gives rise to $s_j - 1$ *sign conditions*

$$\sum_{\gamma \in \Lambda_{j_t}^{(0)}} \gamma < 0, \qquad t = 2, \ldots, s_j, \tag{18}$$

plus one additional condition $\sum_{\gamma \in \Lambda_{j_1}^{(1)}} \gamma > 0$ if the first index $j_1^{(0)}$ is positive (respectively, $\sum_{\gamma \in \Lambda_{j_1}^{(1)}} \gamma < 0$ if the first index $j_1^{(0)}$ is negative too). Also, we need the *merging condition*

$$S_{j_1}^{(1)} \geq \lambda_{j_2}^{(1)} \tag{19}$$

whenever the index $j_1^{(1)}$ is positive, since otherwise we cannot use a move of type 2 to construct the local QCRs at level 1.

4.3. Step $k+1$

The overall C-realizing procedure we propose consists in successively generating new configurations C_{k+1} from already existing ones, C_k, for each $k = 1, 2, \ldots$, while updating in the process all quantities associated with the configuration. In the process we shall identify which conditions ensure that this C-realizing procedure is not interrupted when passing from C_k to C_{k+1}. The procedure finishes once we reach a configuration with one single index, whose QCR is the original list Λ in (12).

Suppose C_k is one such intermediate configuration

$$1^{(k)}, 2^{(k)}, \ldots, n_k^{(k)},$$

with the corresponding n_k-vector of their signs, where each $j^{(k)}$ is associated with a (positive or negative) target sublist

$$\Lambda_j^{(k)} = \bigcup_r \Lambda_r^{(k-1)},$$

which is a union of target sublists from the previous level $k-1$. For each $\Lambda_j^{(k)}$, there is a corresponding $\mathbf{QCR}_k$, which is obtained in exactly the same way as previously described in [QCR.j^+] and [QCR.j^-] above. Furthermore, each $j^{(k)}$ has a dominant $\lambda_j^{(k)}$ attached to it, whereas positive indices have an additional, second number $S_j^{(k)}$ attached to them. Let us describe step by step the transition from C_k to the new configuration C_{k+1}:

step 1 (Partition Π_{k+1}): Choose one of the possible **ordered partitions** Π_{k+1} by grouping the indices of the configuration C_k into packages, separated by brackets, according to rules R1, R2 and R3, and update the **target lists for level** $k+1$: if the j-th bracket in Π_{k+1} is $[j_1^{(k)}, j_2^{(k)}, \ldots, j_{s_j}^{(k)}]$, then we update the target sublists

$$\Lambda_j^{(k+1)} = \bigcup_{t=1}^{s_j} \Lambda_{j_t}^{(k)}.$$

step 2 (Configuration C_{k+1}): Identify each bracket $[j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$ of Π_{k+1} with one single index $j^{(k+1)}$, giving rise to a list

$$1^{(k+1)}, 2^{(k+1)}, \dots, n_{k+1}^{(k+1)}$$

of indices for level $k+1$, where each index $j^{(k+1)} := [j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$ is negative if

$$\sum_{\gamma \in \Lambda_j^{(k+1)}} \gamma < 0$$

or positive otherwise. The list of indices above, together with the n_{k+1}-vector containing the corresponding signs, constitutes the **configuration C_{k+1} at level $k+1$.**
step 3 (Update the dominant λs): For each index $j^{(k+1)} := [j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$, update

$$\lambda_j^{(k+1)} = \lambda_{j_1}^{(k)},$$

i.e., at level $k+1$, the corresponding $\lambda_j^{(k+1)}$ is just the $\lambda_{j_1}^{(k)}$ associated with the only dominant positive index $j_1^{(k)}$ in the corresponding bracket at level k.
step 4 (Update QCRs for level $k+1$): For each $j^{(k+1)}$, $j = 1, \dots, n_{k+1}$, construct its associated local QCR as follows:

(1) use moves of type 3 to merge all local QCRs corresponding to indices of level k in the bracket $j^{(k+1)} := [j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$, and
(2) make moves of type 2 in the following way:
   (2.a) if the index $j^{(k+1)}$ is positive, increase the dominant entry up to the point where all negative entries in the list are fully reached. This leads to a list where all entries are realized, except for the first, dominant one.
   (2.b) if $j^{(k+1)}$ is negative, increase the dominant entry until it is fully reached, and decrease the negative entries collectively by that same amount, i.e., in such a way that the sum of the decreases of the negative entries equals the increase in the dominant one.

   The collection of all **local quasi C-realizations** of the indices $j^{(k+1)}$ for $j = 1, \dots, n_{k+1}$ is the $\mathbf{QCR}_{k+1}$ associated with the configuration C_{k+1}.

step 5 (Update sums S): For each positive index $j^{(k+1)} := [j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$, update

$$S_j^{(k+1)} = S_{j_1}^{(k)} - \sum_{\gamma \in \Lambda_{j_2}^{(k)} \cup \dots \cup \Lambda_{j_{s_j}}^{(k)}} \gamma, \tag{20}$$

i.e., at level $k+1$, the corresponding $S_j^{(k+1)}$ is the result of adding the negativity of all negative indices in the k-th level bracket generating $j^{(k+1)}$ to the sum $S_{j_1}^{(k)}$ associated with the only dominant positive index in that k-th level bracket.

As announced in §3.4, the creation of each level requires additional conditions which have to be met by the entries in Λ if we want the C-realizing procedure to continue: each bracket $[j_1^{(k)}, j_2^{(k)}, \dots, j_{s_j}^{(k)}]$ in the ordered partition Π_{k+1} gives rise to s_j **sign conditions**. The $s_j - 1$ last indices in Π_1 dictate

$$\sum_{\gamma \in \Lambda_{j_t}^{(k)}} \gamma < 0, \qquad t = 2, \dots, s_j, \tag{21}$$

while the s_j-th sign condition depends on whether the first index $j_1^{(k)}$ is positive or negative: it amounts to imposing the sum $\sum_{\gamma \in \Lambda_{j_1}^{(k)}} \gamma$ to be positive or negative, accordingly. Furthermore, the **merging condition**

$$S_{j_1}^{(k)} \geq \lambda_{j_2}^{(k)}, \tag{22}$$

which only applies if the index $j_1^{(k)}$ is positive and $s_j \geq 2$, ensures that the corresponding local QCR for $j_1^{(k)}$ can be constructed.

Notice that, since each of the quantities above is just a combination of the original entries $\lambda_j, -\mu_j$ of Λ, every one of these conditions can be written ultimately in terms of those original entries. Furthermore, each of these conditions is a linear inequality, involving only linear combinations of those entries with coefficients 1 or -1.

The whole process stops once we arrive at a level L whose configuration C_L consists of one single index $1^{(L)}$ whose $\mathbf{QCR}_L$ is Λ. The outcome of the process can be taken to be either a rooted tree, as explained in §3.1 above or, equivalently, a nested bracket structure, analogous to (11), obtained after successively blowing up all ordered partitions $\Pi_L, \Pi_{L-1}, \dots, \Pi_0$, together with the conditions (21) and (22) associated with each of those levels.

5. The main result

It is clear from the discussion in §4 above that, at each level, there are many possible choices for the ordered partition, each of them giving rise to different conditions (21) and (22). This produces many different sets of sufficient conditions guaranteeing C-realizability. What we shall see now is that, not only is each of these conditions sufficient for C-realizability, but collectively they are also necessary, in the sense of the following result:

Theorem 5.1. *Let $\Lambda = \{\lambda_1, \dots, \lambda_n, -\mu_m, \dots, -\mu_1\}$ be a T_0-admissible list. Then Λ is C-realizable if and only if there exists a partition (13) and a nested bracket structure*

for the indices $1, \cdots, n$ *such that the entries in* Λ *satisfy all conditions* (21) *and* (22) *associated with that bracket structure for* $k = 0, 1, 2, \ldots, L$*, where* L *is the highest level in the nested bracket structure.*

Proof. First, recall that, due to Lemma 3.2, whenever we speak of a C-realizing procedure we may assume that all moves of type 2 are done with λ_2 negative, as in the procedure described in §4.

Suppose the T_0-admissible list Λ is such that there exists a partition (13) and a nested bracket structure for the indices $1, \cdots, n$ such that the entries in Λ satisfy all conditions (21) and (22) associated with that bracket structure for $k = 0, 1, 2, \ldots, L$. Then, as shown in Section 4, the list Λ is trivially C-realizable, since the conditions (21) and (22) are precisely the guarantee that the general procedure described in §4 can progress step by step until it reconstructs the full list Λ via moves of types 2 and 3.

Conversely, suppose Λ is C-realizable and consider any C-realizing procedure for Λ. Then, by Theorem 3.1, there exists a rooted tree associated with that procedure, which we may assume to be simplified in the sense described in Remark 1. Furthermore, the tree has a corresponding nested bracket structure, as explained in §3.3. We have already seen that, for each level $k = 0, 1, 2, \ldots$, each bracket $[j_1^{(k)}, j_2^{(k)}, \ldots, j_{s_j}^{(k)}]$ in the ordered partition Π_{k+1} gives rise to s_j sign conditions (21) plus one merging condition (22). Finally, the initial partition (13), i.e., the initial target lists $\Lambda_j^{(0)}$, can be easily read from the bottom line in the tree. Obviously, the indices $1, \cdots, n$ involved in the nested bracket structure correspond to the n positive entries in Λ and the maximum level L of the nested bracket structure is the number of moves of type 2 in the associated rooted tree minus one (since the first moves of type 2 produce the QCR_0). □

Remarks to Theorem 5.1:

- Since, for every k, both the sign conditions (21) and the merging condition (22) are sums and differences of the original entries λ_j and $-\mu_j$ of Λ, it is clear that every one of these conditions can be easily rewritten in terms of the original entries. Moreover, since each of (21) and (22) is a linear inequality between sums of λs and sums of μs, all conditions together define a union of polyhedral cones. Therefore, *Theorem 5.1 identifies the set of zero-sum C-realizable lists as a union of polyhedral cones.*
  In fact, these cones are rather special, since every equation defining one of its faces is just a linear combination of λs and μs with (respective) coefficients 1 and -1.
- It should be noted that dependence relations may exist among the conditions (21) and (22). We have chosen to include all conditions in the statement of Theorem 5.1, even if some of them may be mutually dependent, or even trivially satisfied, in order to be able to systematically enumerate the conditions. However, once a full set of conditions is obtained, it could be purged *a posteriori* until only nontrivial, mutually independent conditions remain (see, for instance, Corollary 6.4 below).

- Since C-realizability is preserved by negative subdivision (see Definition 2.3 above), we may take advantage of this by writing the following result, which trivially follows from Theorem 5.1 and Lemma 2.4:

Corollary 5.2. *Let* $\Lambda = \{\lambda_1, \dots, \lambda_n, -\mu_p, \dots, -\mu_1\}$ *be a list of real numbers with* $n \leq p$, $\lambda_1 \geq \cdots \geq \lambda_n > 0$ *and* $\mu_j > 0$ *for* $j = 1, \dots, p$. *Suppose there is a partition*

$$I = \bigcup_{k=1}^{m} I_k$$

of the index set $I = \{1, \dots, p\}$ *such that the partial sums*

$$S_k = \sum_{j \in I_k} \mu_j, \qquad k = 1, \dots, m, \qquad m \geq n$$

lead to a T_0*-admissible list* $\{\lambda_1, \dots, \lambda_n, -S_m, \dots, -S_1\}$ *satisfying the conditions of Theorem 5.1. Then* Λ *is* C*-realizable.*

6. The cases $n = 2, 3, 4$

The systematic application of Theorem 5.1 allows all conditions corresponding to all possible nested bracket structures to be written explicitly in cases of low dimension. In this section, we illustrate this by analyzing the simplest cases in which the number n of positive entries in Λ is $2, 3$ or 4. We include two tables displaying all possible initial configurations C_0, all possible ordered partitions Π_1 compatible with C_0, and all C-realizing nested bracket structures which synthesize the results described in the corollaries solving the cases $n = 3$ and $n = 4$.

The case $n = 2$ was already solved in [2] (see Theorem 2.2), even for spectra with arbitrary trace. A version adapted to our context (and notation) is as follows:

Theorem 6.1. *Let* $\Lambda = \{\lambda_1, \lambda_2, -\mu_m, \dots, -\mu_2, -\mu_1\}$ *be a* T_0*-admissible list. Then* Λ *is C-realizable if and only if there is a partition* $\Lambda = \Lambda_1^{(0)} \cup \Lambda_2^{(0)}$ *as in* (13) *such that*

$$\lambda_1 \geq \max \left\{ \sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k, \sum_{-\mu_k \in \Lambda_2^{(0)}} \mu_k \right\}.$$

Note that $\lambda_1 \geq \displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k$ is a sign condition (including the neutral case); the other negative sign condition is deduced from zero trace. Furthermore, $\lambda_1 \geq \displaystyle\sum_{-\mu_k \in \Lambda_2^{(0)}} \mu_k$ is equivalent to the merging condition $\displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_2$.

The case $n = 3$ is slightly more convoluted, but can be easily described via the following Corollary, which is a straightforward consequence of Theorem 5.1:

Corollary 6.2. *Let $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, -\mu_m, \ldots, -\mu_2, -\mu_1\}$ be a T_0-admissible list. Then Λ is C-realizable if and only if there is a partition $\Lambda = \Lambda_1^{(0)} \cup \Lambda_2^{(0)} \cup \Lambda_3^{(0)}$ as in (13) satisfying either of the following two sets of conditions:*

(a) $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \le 0,\ j = 2, 3$, *and one of the following:*

(a.1) $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \ge \lambda_2$;

(a.2) $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \ge \lambda_3$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_3^{(0)}} \gamma \ge \lambda_2$;

(b) $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \ge 0,\ j = 1, 2$, $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \ge \lambda_3$, $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \ge \lambda_2$.

Proof. There are two possible initial configurations $1^{(0)}$, $2^{(0)}$, $3^{(0)}$ with sign vectors $(+,-,-)$ and $(+,+,-)$.

a) The configuration with sign vector $(+,-,-)$ admits three ordered partitions Π_1:

$[1^{(0)}, 2^{(0)}, 3^{(0)}]$; $[1^{(0)}, 2^{(0)}]$, $[3^{(0)}]$ and $[1^{(0)}, 3^{(0)}]$, $[2^{(0)}]$ according to rules R1, R2 and R3.

All the sign conditions of the corresponding bracket structures [1, 2, 3]; [[1, 2], [3]] and [[1, 3], [2]] can be reduced to the two conditions in part (a) (see (21)). The remaining sign conditions are trivially satisfied due to the zero trace.

The condition in (a.1) is just the merging condition of the bracket structure [1, 2, 3]: $S_1^{(0)} \ge \lambda_2^{(0)}$, where $S_1^{(0)}$ is as defined in (15).

The partition $[1^{(0)}, 2^{(0)}]$, $[3^{(0)}]$ has the configuration $C_1 : \ 1^{(1)} = [1^{(0)}, 2^{(0)}]$, $2^{(1)} = [3^{(0)}]$ with sign vector $(+,-)$ and the final configuration $C_2 : \ 1^{(2)} = [1^{(1)}, 2^{(1)}] = [[1^{(0)}, 2^{(0)}], [3^{(0)}]]$.

At level one we have:

- dominant λs: $\lambda_1^{(1)} = \lambda_1^{(0)} = \lambda_1$; $\lambda_2^{(1)} = \lambda_3^{(0)} = \lambda_3$;
- update sums: $S_1^{(1)} = S_1^{(0)} - \sum_{\gamma\in\Lambda_2^{(0)}} \gamma = \sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_2^{(0)}} \gamma$ (see (20)) and
- merging condition: $S_1^{(0)} \ge \lambda_2^{(0)}$, i.e., $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \ge \lambda_2$.

At level two we have a neutral list with merging condition: $S_1^{(1)} \ge \lambda_2^{(1)}$, i.e., $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_2^{(0)}} \gamma \ge \lambda_3$. However, this merging condition is weaker than the previous one, since

$$\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k - \sum_{\gamma \in \Lambda_2^{(0)}} \gamma \geq \sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_2 \geq \lambda_3.$$

Then, the bracket structure [[1, 2], [3]] is C-realizable under the same sign and merging conditions as the bracket structure [1, 2, 3], so they can be considered equivalent (see Definition 6.3 below).

In a similar way, the partition $[1^{(0)}, 3^{(0)}]$, $[2^{(0)}]$ is C-realizable by the bracket structure [[1, 3], [2]] under the merging conditions given in (a.2): the first one is the merging condition $S_1^{(0)} \geq \lambda_3^{(0)} = \lambda_3$ of the bracket $[1, 3]$, and the second one is the merging condition $S_1^{(1)} = S_1^{(0)} - \sum_{\gamma \in \Lambda_3^{(0)}} \gamma \geq \lambda_2^{(1)} = \lambda_2^{(0)} = \lambda_2$ of the bracket $[[1, 3], [2]]$, which is independent of the previous one.

b) The configuration with sign vector $(+, +, -)$ only admits the ordered partition $\Pi_1 = [1^{(0)}]$, $[2^{(0)}, 3^{(0)}]$ which is C-realizable by the bracket structure [[1], [2, 3]]. The sign conditions can be reduced to the first condition given in part (b), and the two latter conditions in (b) are the merging conditions of the two positive indices: $S_2^{(0)} \geq \lambda_3^{(0)} = \lambda_3$ and $S_1^{(1)} = S_1^{(0)} \geq \lambda_2^{(1)} = \lambda_2^{(0)} = \lambda_2$. □

We have already seen, in the proof of Corollary 6.2, that certain nested bracket structures are equivalent, in the following sense:

Definition 6.3. Two nested bracket structures are said to be *equivalent*, and we denote this with the symbol $\sim$, if they induce the same sign and merging conditions.

In view of this, most of these equivalent structures are omitted in the tables below, including only one representative for each equivalence class.

From this point on, the notation for the configurations will be simplified by omitting the $^{(k)}$ superindices and by replacing sign vectors with colors (for interpretation of the colors see the web version of this article): more precisely, each index written in blue (respectively, in red) is understood to be positive (respectively, negative): for instance, the configuration $1^{(0)}$, $2^{(0)}$, $3^{(0)}$, $4^{(0)}$ at level 0 with sign vector $(+, +, -, -)$ will be written $1, 2, 3, 4$. Moreover, once an ordered partition is chosen, the corresponding brackets will also be colored, according to the sign of the new target list created by each bracket: the ordered partition $[\,1^{(0)}, 3^{(0)}\,]\,[\,2^{(0)}, 4^{(0)}\,]$, giving rise to the configuration $1^{(1)}$, $2^{(1)}$ at level 1 with sign vector $(+, -)$ will be described by $[1, 3]\,[2, 4]$.

Using this sign/color criterion in Table 1 we synthesize the possible non-equivalent C-realizations for the case $n = 3$ studied in the previous Corollary, where the leftmost column displays the two possible C_0 configurations $(+, -, -)$ and $(+, +.-)$, i.e., $1, 2, 3$ and $1, 2, 3$, the middle column shows the ordered partitions Π_1 corresponding to each configuration C_0 and the rightmost column shows the non-equivalent nested bracket structures corresponding to each ordered partition Π_1.

Table 1
Possible non-equivalent ordered partitions and nested bracket structures for $n = 3$.

Initial configurations C_0	Ordered partitions Π_1	Nested bracket structures
$1, 2, 3$	$[1, 2, 3]$ $[1, 3]\ [2]$	$[1, 2, 3]$ $[[1, 3], [2]]$
$1, 2, 3$	$[1]\ [2, 3]$	$[\,[1], [2, 3]\,]$

Table 2
Non-equivalent ordered partitions and bracket structures for $n = 4$.

Initial configurations C_0	Ordered partitions Π_1	Nested bracket structures
$1, 2, 3, 4$	$[1, 2, 3, 4]$ $[1, i, 4]\ [j],\ \{i, j\} = \{2, 3\}$ $[1, i]\ [2, j],\ \{i, j\} = \{3, 4\}$ $[1, i]\ [2]\ [j],\ \{i, j\} = \{3, 4\}$	$[1, 2, 3, 4]$ $[[1, i, 4], [j]],\ \{i, j\} = \{2, 3\}$ $[[1, i], [2, j]],\ \{i, j\} = \{3, 4\}$ $[[[1, i], [j]], [2]],\ \{i, j\} = \{3, 4\}$
$1, 2, 3, 4$	$[1, 2]\ [3, 4]$ $[1]\ [2]\ [3, 4]$	$[\,[1, 2], [3, 4]\,]$ $[\,[[1], [3, 4]], [2]\,]$ $[\,[1], [[3, 4], [2]\,]$
$1, 2, 3, 4$	$[1]\ [2, 3, 4]$ $[1, j]\ [2, i],\ \{i, j\} = \{3, 4\}$ $[1]\ [2, i]\ [j],\ \{i, j\} = \{3, 4\}$	$[\,[1], [2, 3, 4]\,]$ $[\,[1, j], [2, i]\,],\ \{i, j\} = \{3, 4\}$ $[\,[1], [2, i], [j]\,],\ \{i, j\} = \{3, 4\}$ $[\,[1], [[2, i], [j]]\,],\ \{i, j\} = \{3, 4\}$ $[\,[[1], [j]], [2, i]\,],\ \{i, j\} = \{3, 4\}$
$1, 2, 3, 4$	$[1]\ [2]\ [3, 4]$	$[\,[1], [[2], [3, 4]]\,]$

Table 2 synthesizes the possible non-equivalent C-realizations studied in Corollary 6.4 below, which summarizes the list of conditions which Theorem 5.1 provides for the case $n = 4$.

Corollary 6.4. *Let $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, -\mu_m, \ldots, -\mu_2, -\mu_1\}$ be a T_0-admissible list. Then Λ is C-realizable if and only if there is a partition $\Lambda = \Lambda_1^{(0)} \cup \Lambda_2^{(0)} \cup \Lambda_3^{(0)} \cup \Lambda_4^{(0)}$ satisfying any of the following four sets of conditions:*

(a) $\displaystyle\sum_{\gamma \in \Lambda_j^{(0)}} \gamma \leq 0,\ j = 2, 3, 4$; *and one of the following:*

(a.1) $\displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_2$;

(a.2) $\displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_i;\quad \sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k - \sum_{\gamma \in \Lambda_i^{(0)} \cup \Lambda_4^{(0)}} \gamma \geq \lambda_j,$ *with* $\{i, j\} = \{2, 3\}$;

(a.3) $\displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_i;\quad \sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k - \sum_{\gamma \in \Lambda_i^{(0)}} \gamma \geq \lambda_2,$ *with* $i = 3, 4$;

(a.4) $\displaystyle\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_i;\quad \sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k - \sum_{\gamma \in \Lambda_3^{(0)} \cup \Lambda_4^{(0)}} \gamma \geq \lambda_2,$ *with* $i = 3, 4$;

(b) $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0$, $j=1,3$; $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \leq 0$, $j=2,4$; *and one of the following:*

(b.1) $\sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2$; $\sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4$;

(b.2) $\sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0$; $\sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_3^{(0)}\cup\Lambda_4^{(0)}} \gamma \geq \lambda_2$;

(b.3) $\sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \geq 0$; $\sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4$; $\sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \geq \lambda_2$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_3$.

(c) $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0$, $j=1,2$; $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \leq 0$, $j=3,4$; *and one of the following:*

(c.1) $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_3$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2$;

(c.2) $\sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_i^{(0)}} \gamma \leq 0$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_j$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_i$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq \lambda_2$;

(c.3) $\sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_i^{(0)}} \gamma \leq 0$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_i$;

(c.4) $\sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_i^{(0)}} \gamma \geq 0$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_i$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_i^{(0)}} \gamma \geq \lambda_j$.

(c.5) $\sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_i^{(0)}} \gamma \leq 0$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_j$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_i$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq \lambda_2$;

(d) $\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0$, $j=1,2,3$; $\sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4$; $\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_3$; $\sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2$.

Proof. There are four possible initial configurations: the conditions in part (a) correspond to the initial configuration $1,2,3,4$, the ones in part (b) to $1,2,3,4$, those in part (c) to $1,2,3,4$, and those in part (d) to $1,2,3,4$.

a) The configuration $1,2,3,4$ admits seven non-equivalent ordered partitions Π_1 according to rules R1, R2 and R3: the full one $[1,2,3,4]$; with two parts: $[1,i,4]\,[j]$, with $\{i,j\}=\{2,3\}$, and $[1,i]\,[2,j]$ with $\{i,j\}=\{3,4\}$; and with three parts: $[1,i]\,[2]\,[j]$, with $\{i,j\}=\{3,4\}$.

The bracket structure $[1,2,3,4]$ represents the four bracket structures, $[[1,2],[3],[4]]$, $[[1,2],[3,4]]$, $[[[1,2],[3]],[4]]$ and $[[1,2,3],[4]]$, which are equivalent according to Definition 6.3. Similarly, $[[1,i],[2,j]] \sim [[1,i],[2],[j]] \sim [[[1,i],[2]],[j]]$, for each $\{i,j\} = \{3,4\}$.

All the sign conditions of the corresponding nested bracket structure (see Table 2) can be reduced to the three conditions in part (a). The remaining sign conditions are satisfied due to the zero-trace condition.

The conditions from (a.1) to (a.4) are the merging conditions corresponding to the four types of nested bracket structures.

b) In the configuration $1,2,3,4$, by rule R3, the part $[3,4]$ is necessary, then we only have two independent ordered partitions Π_1: $[1,2]\,[3,4]$ and $[1]\,[2]\,[3,4]$.

In the corresponding bracket structures, when the part $[3,4]$ is in the middle it can be positive or negative, so we have three possible non-equivalent C-realizations: $[\,[1,2],[3,4]\,]$, $[\,[[1],[3,4]],[2]\,]$ and $[\,[1],[[3,4],[2]]\,]$. Note that $[\,[1],[2],[3,4]\,] \sim [\,[[1],[2]],[3,4]\,] \sim [\,[1,2],[3,4]\,]$.

The sign conditions of the initial configuration $1,2,3,4$ are given in (b). The conditions from (b.1) to (b.3) include the sign conditions of the part $[3,4]$ and the merging conditions corresponding to the other three nested bracket structures, respectively.

c) The configuration $1,2,3,4$ admits five independent ordered partitions Π_1 according to rules R1, R2 and R3: with two parts: $[1]\,[2,3,4]$ and $[1,i]\,[2,j]$, with $\{i,j\} = \{3,4\}$; and with three parts: $[1]\,[2,i]\,[j]$, with $\{i,j\} = \{3,4\}$.

The three ordered partitions with two parts clearly determine the corresponding three nested bracket structures. However, each one of the partitions with three parts produces different nested bracket structures, depending on the sign of the part $[2,i]$. Note that $[\,[[1],[2,i]],[j]\,] \sim [\,[1],[2,i],[j]\,]$.

The sign conditions of the initial configuration $1,2,3,4$ are given in (c). In (c.1) we give the merging conditions of the first nested bracket structure. Whereas the conditions from (c.2) to (c.5) include the sign condition of the part $[2,i]$ and the merging conditions corresponding to the other four nested bracket structures, respectively.

d) The necessary sign conditions of the initial configuration $1,2,3,4$ and the merging conditions of the corresponding bracket structure $[\,[1],[[2],[3,4]]\,]$ are included in (d).

Redundant conditions, such as, for instance, $\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k - \sum_{\gamma \in \Lambda_2^{(0)}} \gamma \geq \lambda_3$ in part (b.1), which follows from $\sum_{-\mu_k \in \Lambda_1^{(0)}} \mu_k \geq \lambda_2$, have been removed from the statement. □

Note that, in each ordered partition Π_1, the first bracket is always positive and the last one always negative. Up to this point, for $n \leq 4$, not many ordered partitions had more than two brackets. For $n \geq 5$, however, many ordered partitions (at different levels in the C-realization procedure) have three or more brackets, and the brackets in the middle may be either positive or negative, which usually opens several possibilities for consideration. For the sake of conciseness, we choose not to write the analogous table and corollary corresponding to the case $n = 5$, which contains 132 non-equivalent nested bracket struc-

tures. In any case, note that it is easy to obtain the sign and merging conditions associated to each particular nested bracket structure. For the initial configuration $1, 2, 3, 4, 5$ and the ordered partition $[1][2, 5][3, 4]$, for example, there are three different corresponding bracket structures, namely $[\,[1], [2, 5], [3, 4]\,]$, $[\,[1], [[2, 5], [3, 4]]\,]$ and $[\,[[1], [3, 4]], [2, 5]\,]$ (notice that $[\,[[1], [2, 5]], [3, 4]\,] \sim [\,[1], [2, 5], [3, 4]\,]$).

The conditions for the first one are

$$\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0,\ j = 1, 2, 3;\quad \sum_{\gamma\in\Lambda_j^{(0)}} \gamma \leq 0,\ j = 4, 5;\quad \sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_5^{(0)}} \gamma \leq 0;\quad \sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0;$$

$$\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_5;\quad \sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4;\quad \sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2,$$

the conditions for the second one are

$$\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0,\ j = 1, 2, 3;\quad \sum_{\gamma\in\Lambda_j^{(0)}} \gamma \leq 0,\ j = 4, 5;\quad \sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_5^{(0)}} \gamma \geq 0;$$

$$\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_5;\quad \sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4;\quad \sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_5^{(0)}} \gamma \geq \lambda_3;\quad \sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_2,$$

while the conditions for the third one are

$$\sum_{\gamma\in\Lambda_j^{(0)}} \gamma \geq 0,\ j = 1, 2, 3;\quad \sum_{\gamma\in\Lambda_j^{(0)}} \gamma \leq 0,\ j = 4, 5;\quad \sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_5^{(0)}} \gamma \leq 0;\quad \sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0;$$

$$\sum_{-\mu_k\in\Lambda_2^{(0)}} \mu_k \geq \lambda_5;\quad \sum_{-\mu_k\in\Lambda_3^{(0)}} \mu_k \geq \lambda_4;\quad \sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k \geq \lambda_3;\quad \sum_{-\mu_k\in\Lambda_1^{(0)}} \mu_k - \sum_{\gamma\in\Lambda_3^{(0)}\cup\Lambda_4^{(0)}} \gamma \geq \lambda_2.$$

In the second case, we do not include the sign conditions $\sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0$ and $\sum_{\gamma\in\Lambda_2^{(0)}} \gamma + \sum_{\gamma\in\Lambda_5^{(0)}} \gamma + \sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \leq 0$ because they can be deduced from zero trace. Analogously, in the third case, we do not include the sign condition $\sum_{\gamma\in\Lambda_1^{(0)}} \gamma + \sum_{\gamma\in\Lambda_3^{(0)}} \gamma + \sum_{\gamma\in\Lambda_4^{(0)}} \gamma \geq 0$.

Declaration of competing interest

There is no competing interest of a financial nature, or of any other nature for that matter.

Appendix A. Moves of type 2 decrease negative entries

As already mentioned in §3.5, when we perform a move of type 2

$$\{\lambda_1, \lambda_2, \ldots, \lambda_n\} \longrightarrow \{\lambda_1 + \epsilon, \lambda_2 - \epsilon, \ldots, \lambda_n\},$$

the entry λ_2 which is decreased can be either positive or negative.[3] Our goal in this appendix is to show that, without loss of generality, we may assume that λ_2 is always negative. We observe, however, that choosing one sign or the other is not entirely irrelevant, in fact one could say it responds to different approaches to C-realization: suppose we have a target list $\{9, -5, -6\}$, for instance, and we start with a triplet $\{0, 0, 0\}$ of zeros. One possible course of action is to *fully realize the negative entries*, starting with $\{11, -5, -6\}$, and then bring down the 11 entry down to 9, applying one or more moves of type 2 *to a positive entry*. Alternatively, one can choose to *fully realize the positive entry*, transforming the triplet into $\{9, -4, -5\}$ and then bring the two negative entries down to -5 and -6 using one or more moves of type 2 *on negative entries*. This latter approach is actually the one which drives our overall strategy throughout this paper. We have chosen it because it leads to fewer conflicts with the dominant entries of the lists generated throughout the C-realizing procedures, and gives rise to simpler merging conditions. Below we prove Lemma 3.2, which ensures that any realizing procedure performing moves of type 2 on positive λ_2s can be replaced by an alternative realizing procedure, where all moves of type 2 are performed on negative λ_2s.

The proof of Lemma 3.2 below is actually constructive, i.e., we shall see how to modify the original C-realizing procedure in order to achieve this. The main idea of the proof is to *backtrack*, tracing the moves of type 2 on positive λ_2s back in time, and replacing every one of them by another move of type 2 involving an appropriate *negative* λ_2.

Before we present the proof, let us first illustrate the situation with a specific example: take the T_0-admissible list

$$\widetilde{\Lambda} = \{16,\ 13,\ 10,\ 10,\ 4,\ -2,\ -6,\ -6,\ -6,\ -9,\ -12,\ -12\},$$

which is, in spirit, the same list already introduced in (5), only removing the subset $\{8, -3, -5\}$, which is a *neutral* target list and, therefore, can be put aside and be C-realized independently at the very end of the process. We consider the following C-

[3] We disregard the case $\lambda_2 = 0$ because any zero entry throughout the C-realizing procedure can be transformed into a negative one by 'backtracking' (see below for a detailed description of the 'backtracking' procedure).

realizing procedure, which corresponds to the approach of always trying to fully realize the negative entries:

$$
\begin{array}{c}
\{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\} \\
\{0,\,0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0,\,0\} \\
\{11,\,-2,\,-9\}\ \{12,\,-12\}\ \{6,\,-6\}\ \{12,\,-12\}\ \{12,\,-6,\,-6\} \\
\{\textcircled{11},-2,\,-9\,\|\,\boxed{12},-12\,\|\,\textcircled{6},\ -6\} \qquad \{\boxed{12},\,-12\,\|\,\textcircled{12},\,-6,\,-6\} \\
\{\textcircled{10},-2,\,-9\,\|\,\boxed{15},-12\,\|\,\textcircled{4},\ -6\} \qquad \{\boxed{13},\,-12\,\|\,\textcircled{11},\,-6,\,-6\} \\
\{10,-2,\,-9\,\|\,\boxed{15},-12\,\|\,4,\,-6\,\|\,13,\,-12\,\|\,\textcircled{11},\,-6,\,-6\} \\
\Lambda = \{10,-2,\,-9\,\|\,\boxed{16},-12\,\|\,4,\,-6\,\|\,13,\,-12\,\|\,\textcircled{10},\,-6,\,-6\}.
\end{array}
\tag{23}
$$

As before, entries in black have already been realized, while colored entries still need to be modified in order to achieve their final value in $\widetilde{\Lambda}$. In this case, such entries are colored in blue. Also, double vertical bars separate entries which come from the same initial subset in the third line of (23) (in other words, the QCRs at level zero as defined in §4). Moves of type 2 are indicated by surrounding the entries affected: the dominant entry λ_1 is boxed, while the corresponding λ_2 is circled. Finally, note that the five target lists associated with the procedure (23) above are

$$\Lambda_1^{(0)} = \{10,\,-2,\,-9\}, \quad \Lambda_2^{(0)} = \{16,\,-12\}, \quad \Lambda_3^{(0)} = \{4,\,-6\},$$

$$\Lambda_4^{(0)} = \{13,\,-12\}, \quad \Lambda_5^{(0)} = \{10,\,-6,\,-6\}.$$

One can easily see that the procedure above involves three moves of type 2 which decrease *positive* entries. Our goal is to replace, one by one, each of those three moves by another move of type 2 which decreases an appropriately chosen *negative* entry, and still leads to the same final list $\widetilde{\Lambda}$. We do this by *backtracking* from the bottom up: the first move of type 2 we find in the procedure from the bottom up is the one transforming the dominant entry 15 in line 6 of the procedure into 16, and the positive entry 11 into 10. We choose to replace that move by another move of type 2: the change in the dominant entry remains unchanged but, instead of taking the entry 10 in line 7 as the result of decreasing the entry 11 in line 6, we assume there was already a 10 in line 6, and decrease instead a new entry -5 in line 6 into the entry -6 in line 7. We can do this *for every negative entry within the same double vertical bars as* 11. We depict in red those negative entries (such as the newly created -5) which have had to be modified in order to achieve their final target value. For the sake of consistency, we also depict in red positive entries, such as 15, in need of modification on line 6, or below. This allows us to visualize the gradual transition from the 'completely blue' realizing procedure (where all moves of type 2 involve positive λ_2) to a hopefully 'completely red' procedure (where all moves of type 2 involve negative λ_2) in the end.

Of course, it is not enough just to change those few particular entries in lines 6 and 7: we must also propagate all the 'repercussions on the past' of these changes, which amount to (i) subtracting one unit from the positive entry within those same vertical bars for every single line above it, and (ii) adding one unit to the first negative entry on every line within those same vertical bars. Notice that, since we only modify entries within the same vertical bars, and the amounts by which we increase and decrease are the same, we still get a valid C-realizing procedure. Observe, however, that the new realizing procedure contains only two (instead of three) moves of type 2 involving positive λ_2:

$$\begin{array}{c}
\{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\\
\{0,\,0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0,\,0\}\\
\{11,\,-2,\,-9\}\ \{12,\,-12\}\ \{6,\,-6\}\ \{12,\,-12\}\ \{11,\,-5,\,-6\}\\
\{(11),\,-2,\,-9\,\|\,\boxed{12},\,-12\,\|\,(6),\,-6\}\qquad\{\boxed{12},\,-12\,\|\,(11),\,-5,\,-6\}\\
\{(10),\,-2,\,-9\,\|\,\boxed{15},\,-12\,\|\,(4),\,-6\}\qquad\{\boxed{13},\,-12\,\|\,(10),\,-5,\,-6\}\\
\{10,\,-2,\,-9\,\|\,\boxed{15},\,-12\,\|\,4,\,-6\,\|\,13,\,-12\,\|\,10,\,(-5),\,-6\}\\
\Lambda=\{10,\,-2,\,-9\,\|\,\boxed{16},\,-12\,\|\,4,\,-6\,\|\,13,\,-12\,\|\,10,\,(-6),\,-6\}.
\end{array}$$

In a second stage, we move up in the procedure and replace the move of type 2 which increased the dominant entry 12 in line 4 into the entry 15 in line 5, and decreased both 11 into 10, and 6 into 4. As before, we keep the same change in the dominant entries and, instead of decreasing 11, we assume that the entry 10 had already been reached, and decrease instead *by the exact same amount* a negative entry within the same double-bar-block. For instance, we may take -9 and assume it is the result of decreasing an entry -8 above it. Also, instead of decreasing the entry 6 two units into 4, we assume that 4 had already been achieved, and -6 is the result of decreasing an entry -4 above it. For both changes, we also update their repercussions on the lines above them. As a result, we obtain a new procedure

$$\begin{array}{c}
\{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\\
\{0,\,0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0\}\qquad\{0,\,0,\,0\}\\
\{10,\,-2,\,-8\}\ \{12,\,-12\}\ \{4,\,-4\}\ \{12,\,-12\}\ \{11,\,-5,\,-6\}\\
\{10,\,-2,\,(-8)\,\|\,\boxed{12},\,-12\,\|\,4,\,(-4)\}\qquad\{\boxed{12},\,-12\,\|\,(11),\,-5,\,-6\}\\
\{10,\,-2,\,(-9)\,\|\,\boxed{15},\,-12\,\|\,4,\,(-6)\}\qquad\{\boxed{13},\,-12\,\|\,(10),\,-5,\,-6\}\\
\{10,\,-2,\,-9,\,\|\,\boxed{15},\,-12,\,\|\,4,\,-6,\,\|\,13,\,-12,\,\|\,10,\,(-5),\,-6\}\\
\Lambda=\{10,\,-2,\,-9,\,\|\,\boxed{16},\,-12,\,\|\,4,\,-6,\,\|\,13,\,-12,\,\|\,10,\,(-6),\,-6\},
\end{array}$$

which is again a completely valid C-realizing procedure for the same list $\widetilde{\Lambda}$, but containing one single move of type 2 decreasing a positive entry: the one increasing the entry 12 in

line 4 to 13 in line 5, and decreasing 11 to 10. As before, we replace this last move by another one which decreases instead one of the negative entries within the same double-bar-block as 11. Notice that we may either take the -5 as a result of decreasing a -4 in the line above, or the -6 as a result of decreasing a former -5. Either is acceptable. If we choose the latter, and backtrack its repercussions, we arrive, as announced, at the C-realizing procedure

$$
\begin{array}{c}
\{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\}\ \{0\} \\
\{0,\,0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0\} \qquad \{0,\,0,\,0\} \\
\{10,\,-2,\,-8\}\ \{12,\,-12\}\ \{4,\,-4\}\ \{12,\,-12\}\ \{10,\,-5,\,-5\} \\
\{10,-2,\,(-8)\,\|\,\boxed{12},-12\,\|\,4,\,(-4)\} \qquad \{\boxed{12},\,-12\,\|\,10,\,-5,\,(-5)\} \\
\{10,-2,\,(-9)\,\|\,\boxed{15},-12\,\|\,4,\,(-6)\} \qquad \{\boxed{13},\,-12\,\|\,10,\,-5,\,(-6)\} \\
\{10,-2,\,-9,\,\|\,\boxed{15},-12,\,\|\,4,\,-6,\,\|\,13,\,-12,\,\|\,10,\,(-5),\,-6\} \\
\Lambda = \{10,-2,\,-9,\,\|\,\boxed{16},-12,\,\|\,4,\,-6,\,\|\,13,\,-12,\,\|\,10,\,(-6),\,-6\}.
\end{array}
$$

for $\widetilde{\Lambda}$ such that all its moves of type 2 decrease only negative entries.

The same ideas employed in this example lead to a proof for Lemma 3.2:

Proof of Lemma 3.2 in §3.5. Consider a T_0-admissible realizable list Λ such that there exists a C-realizing procedure for Λ which contains moves of type 2 performed on some positive λ_2, and consider the tree representation of that C-realizing procedure as described in Theorem 3.1. In fact, by appropriately rearranging the QCRs in the second row from the top of the tree, we may avoid any 'crossing arrows' in the tree, leading to an even simpler representation of the C-realizing procedure, where moves of type 3 always merge sets which are contiguous. Thus, we may condense the procedure into a representation, like (23) above, where the entries within the same pair of double vertical bars (i.e., those originating in the same QCR at level 0) are stacked vertically on top of each other. Thus, as we go down within two pairs of vertical bars we may track the evolution of any single specific entry as the procedure advances.

Our goal is to find another C-realizing procedure for Λ, all of whose moves of type 2 are performed on negative entries. We do this by replacing every single move of type 2 on a positive λ_2 with another move of type 2 on an appropriate negative λ_2 leading to the same outcome. The keys for doing this are: (i) doing it from the bottom up, (ii) choosing the negative entry to be decreased within the same pair of double bars as the positive one, and (iii) updating past entries in vertical to compensate for the change. The combination of these three actions is what we call *backtracking*. Moves of type 3 will not be changed at all, so without loss of generality, we may even replace the set delimiters by double vertical bars, which will make tracking even easier.

For notational purposes, let us denote by $Q_1^{(0)}, Q_2^{(0)}, \ldots, Q_p^{(0)}$ the QCRs at level zero in the third row from the top in a representation of the form (23). Each $Q_j^{(0)}$ is bounded

by double vertical bars. Going down the procedure, we denote by $Q_k^{(l)}$ the k-th QCR from the left in line l of the procedure. Notice that in this setting all the QCRs $Q_k^{(l)}$ for the same k are stacked on top of each other, as in

$$\begin{array}{c} \| \, Q_1^{(0)} \, \| \, Q_2^{(0)} \, \| \, Q_3^{(0)} \, \cdots \| \, Q_p^{(0)} \, \| \\ \| \, Q_1^{(1)} \, \| \, Q_2^{(1)} \, \| \, Q_3^{(1)} \, \cdots \| \, Q_p^{(1)} \, \| \\ \cdots \\ \| \, Q_1^{(t)} \, \| \, Q_2^{(t)} \, \| \, Q_3^{(t)} \, \cdots \| \, Q_p^{(t)} \, \| \end{array} .$$

We now carefully describe the action of backtracking: suppose the lowest lines containing a move of type 2 which decreases a positive entry are the s-th and the $(s+1)$-th line. Suppose further that the corresponding move decreases a positive entry $\lambda_{kr}^{(s)} \in Q_k^{(s)}$ into another entry $\lambda_{kr}^{(s)} - \delta \in Q_k^{(s+1)}$ with $\delta > 0$. Then, we first replace the quantity $\lambda_{kr}^{(s)}$ in line s with $\lambda_{kr}^{(s)} - \delta$, and update accordingly all entries $\lambda_{kr}^{(l)}$, $l = 0, 1, 2, \ldots, s-1$ stacked above it by subtracting δ from every single one of them. Also, we choose any negative entry $-\mu_{kt}^{(s+1)} \in Q_k^{(s+1)}$ in line $s+1$ which has been already realized, and replace the corresponding negative entry $-\mu_{kt}^{(s)} \in Q_k^{(s)}$ in the line above it by the negative[4] entry $-\mu_{kt}^{(s+1)} + \delta$. Once this change is made, we update accordingly all entries $-\mu_{kt}^{(l)}$, $l = 0, 1, 2, \ldots, s-1$ stacked above it by adding δ to every one of them. Schematically, we replace the procedure

$$\begin{array}{ccccc} \vdots & & \vdots & & \vdots \\ Q_{k-1}^{(s-2)} & \| & \lambda_{kr}^{(s-2)}, \ldots, -\mu_{kt}^{(s-2)}, \ldots & \| & Q_{k+1}^{(s-2)} \, \| \\ Q_{k-1}^{(s-1)} & \| & \lambda_{kr}^{(s-1)}, \ldots, -\mu_{kt}^{(s-1)}, \ldots & \| & Q_{k+1}^{(s-1)} \, \| \\ Q_{k-1}^{(s)} & \| & \lambda_{kr}^{(s)}, \ldots, -\mu_{kt}^{(s)}, \ldots & \| & Q_{k+1}^{(s)} \, \| \\ Q_{k-1}^{(s+1)} & \| & \lambda_{kr}^{(s)} - \delta, \ldots, -\mu_{kt}^{(s+1)}, \ldots & \| & Q_{k+1}^{(s+1)} \, \| \end{array}$$

by the procedure

[4] It may be necessary to modify more than one such negative entry if δ is larger than the absolute value of every negative entry in $Q_k^{(s+1)}$, but this is acceptable as long as all these negative entries belong to the same $Q_k^{(s+1)}$.

$$
\begin{array}{cccc}
\vdots & \vdots & \vdots & \vdots \\
Q_{k-1}^{(s-2)} & \| \lambda_{kr}^{(s-2)} - \delta, \ldots, -\mu_{kt}^{(s-2)} + \delta, \ldots \| & & Q_{k+1}^{(s-2)} \| \\
Q_{k-1}^{(s-1)} & \| \lambda_{kr}^{(s-1)} - \delta, \ldots, -\mu_{kt}^{(s-1)} + \delta, \ldots \| & & Q_{k+1}^{(s-1)} \| \\
Q_{k-1}^{(s)} & \| \lambda_{kr}^{(s)} - \delta, \ldots, -\mu_{kt}^{(s+1)} + \delta, \ldots \| & & Q_{k+1}^{(s)} \| \\
Q_{k-1}^{(s+1)} & \| \lambda_{kr}^{(s)} - \delta, \ldots, -\mu_{kt}^{(s+1)}, \ldots \| & & Q_{k+1}^{(s+1)} \|
\end{array}
\tag{24}
$$

Notice that the $(s+1)$-th line in both procedures coincides, so if from that point on we define the new realizing procedure as the same as the one we started with, then the final outcome will be $\widetilde{\Lambda}$ again. Furthermore, the fact of having taken the positive entries $\lambda_{kr}^{(*)}$ and the negative entries $-\mu_{kt}^{(*)}$ from within the same set $Q_k^{(*)}$ ensures that the procedure represented by (24) is a perfectly valid C-realizing procedure, only with one less move of type 2 decreasing positive entries than the original procedure.

Repetition of the backtracking step, as described above, as many times as the number of moves of type 2 which decrease positive entries leads to the final result. □

References

[1] A. Borobia, J. Moro, R.L. Soto, Negativity compensation in the nonnegative inverse eigenvalue problem, Linear Algebra Appl. 393 (2004) 73–89.
[2] A. Borobia, J. Moro, R.L. Soto, A unified view on compensation criteria in the real nonnegative inverse eigenvalue problem, Linear Algebra Appl. 428 (2008) 2574–2584.
[3] M. Boyle, D. Handelman, The spectra of nonnegative matrices via symbolic dynamics, Ann. Math. 133 (1991) 249–316.
[4] P. Egleston, T. Lenker, S. Narayan, The nonnegative inverse eigenvalue problem, Linear Algebra Appl. 379 (2004) 475–490.
[5] R. Ellard, E. Šmigoc, Connecting sufficient conditions for the symmetric nonnegative inverse eigenvalue problem, Linear Algebra Appl. 498 (2016) 521–552.
[6] L. Elsner, R. Nabben, M. Newmann, Orthogonal bases that lead to symmetric nonnegative matrices, Linear Algebra Appl. 271 (1–3) (1998) 323–343.
[7] W. Guo, Eigenvalues of nonnegative matrices, Linear Algebra Appl. 266 (1997) 261–270.
[8] C.R. Johnson, T.J. Laffey, R. Loewy, The real and the symmetric nonnegative inverse eigenvalue problems are different, Proc. Am. Math. Soc. 124 (1996) 3647–3651.
[9] C.R. Johnson, C. Marijuán, P. Paparella, M. Pisonero, The NIEP, Oper. Theory, Adv. Appl. 426 (2018) 199–220.
[10] T.J. Laffey, E. Meehan, A characterization of trace zero nonnegative 5×5 matrices, Linear Algebra Appl. 302/303 (1999) 295–302.
[11] R. Loewy, D. London, A note on the inverse problem for nonnegative matrices, Linear Multilinear Algebra 6 (1978) 83–90.
[12] C. Marijuán, M. Pisonero, R.L. Soto, A map of sufficient conditions for the real nonnegative inverse eigenvalue problem, Linear Algebra Appl. 426 (2007) 690–705.
[13] C. Marijuán, M. Pisonero, R.L. Soto, A map of sufficient conditions for the symmetric nonnegative inverse eigenvalue problem, Linear Algebra Appl. 530 (2017) 344–365.
[14] C. Marijuán, M. Pisonero, R.L. Soto, Updating a map of sufficient conditions for the real nonnegative inverse eigenvalue problem, Spec. Matrices 7 (2019) 246–256.

[15] H. Perfect, Methods of constructing certain stochastic matrices II, Duke Math. J. 22 (1955) 305–311.
[16] R.L. Soto, A family of realizability criteria for the real and symmetric nonnegative inverse eigenvalue problem, Numer. Linear Algebra Appl. 20 (2013) 336–348.
[17] G. Soules, Constructing symmetric nonnegative matrices, Linear Multilinear Algebra 13 (1983) 241–251.
[18] O. Spector, A characterization of trace zero symmetric nonnegative 5×5 matrices, Linear Algebra Appl. 434 (2011) 1000–1017.
[19] J. Torre-Mayo, M.R. Abril-Raymundo, E. Alarcia-Estévez, C. Marijuán, M. Pisonero, The nonnegative inverse eigenvalue problem from the coefficients of the characteristic polynomial, EBL digraphs, Linear Algebra Appl. 426 (2007) 729–773.