J Comb Optim (2006) 11: 321–339
DOI 10.1007/s10878-006-7911-5

One-dimensional optimal bounded-shape partitions for Schur convex sum objective functions

F. H. Chang* · H. B. Chen · J. Y. Guo* · F. K. Hwang* · Uriel G. Rothblum

Received: 8 April 2005 / Accepted: 10 January 2006

Abstract Consider the problem of partitioning n nonnegative numbers into p parts, where part i can be assigned n_i numbers with n_i lying in a given range. The goal is to maximize a Schur convex function F whose ith argument is the sum of numbers assigned to part i.

The shape of a partition is the vector consisting of the sizes of its parts, further, a shape (without referring to a particular partition) is a vector of nonnegative integers $(n_1, \ldots, n_p)$ which sum to n. A partition is called size-consecutive if there is a ranking of the parts which is consistent with their sizes, and all elements in a higher-ranked part exceed all elements in the lower-ranked part. We demonstrate that one can restrict attention to size-consecutive partitions with shapes that are nonmajorized, we study these shapes, bound their numbers and develop algorithms to enumerate them. Our study extends the analysis of a previous paper by Hwang and Rothblum which discussed the above problem assuming the existence of a majorizing shape.

Keywords Optimal partition · Bounded-shape partition · Sum partition · Schur convex function

1. Introduction

Throughout, let n and p be positive integers and let $\theta_1 \ldots, \theta_n$ be real numbers satisfying $\theta_1 \le \theta_2 \le \cdots \le \theta_n$. Consider a partition π of the indices $1, \ldots, n$ into p nonempty parts $\pi_1, \ldots, \pi_p$. Such a partition is called *consecutive* if each part consists of consecutive integers. For example, with $n = 6$ and $p = 3$, $\pi_1 = \{4, 5\}$, $\pi_2 = \{6\}$, $\pi_3 = \{1, 2, 3\}$. Given a partition

*This research is partially supported by ROC National Science grant NSC 92-2115-M-009-014.

F. H. Chang · H. B. Chen · J. Y. Guo (✉)· F. K. Hwang
Department of Applied Mathematics, National Chiaotung University, Hsinchu, Taiwan, 300
e-mail: davidguo.am90g@nctu.edu.tw

U. G. Rothblum
Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel

π, the vector $(|\pi_1|, \ldots, |\pi_p|)$ is called the *shape* of π, and for each i, $|\pi_i|$ is called the *size* of π_i. For convenience, denote $n_i = |\pi_i|$. Of course, a vector of nonnegative integers which sum to n is a potential shape of a partition and we refer to such a vector as a *shape*. A consecutive partition is called *size-consecutive* (*reverse-size-consecutive*) if $n_i > n_j$ implies that every member in π_i is larger (smaller) than every member in π_j. Of course, given any integer vector $(n_1, \ldots, n_p)$ which satisfies $\sum_{i=1}^{p} n_i = n$, there exist a size-consecutive and a reverse-size-consecutive partition with shape $(n_1, \ldots, n_p)$; in fact, they are unique whenever the n_i's and the θ_i's are distinct.

For a vector a in $\mathbb{R}^p$ and $i = 1, \ldots, p$, let $a_{[i]}$ be the ith largest member of $\{a_1, \ldots, a_p\}$. Given vectors a and b in $\mathbb{R}^p$, we say that *a majorizes b* if

$$\sum_{i=1}^{k} a_{[i]} \geq \sum_{i=1}^{k} b_{[i]} \text{ for } k = 1, \ldots, p-1 \tag{1.1}$$

and

$$\sum_{i=1}^{p} a_i = \sum_{i=1}^{p} b_i \,. \tag{1.2}$$

We say that *a strictly majorizes b* if a majorizes b and one of the inequalities in (1.1) holds strictly. A real-valued function f on $\mathbb{R}^p$ is *Schur convex* if $f(a) \geq f(b)$ whenever a majorizes b. A Schur convex function is known to be symmetric. These two properties are the only ones we need in this paper (see Marshall and Olkin (1979) for further details about majorization and Schur convexity).

Schur convexity has been proved to be a powerful tool in maximizing set functions. There is a large body of literature on the so-called "sum partition problem" (see Hwang and Rothblum (to appear) for references), i.e., to maximize the objective function

$$F(\pi) = f\left(\sum_{j\in\pi_1} \theta_j,\ \sum_{j\in\pi_2} \theta_j, \ldots, \sum_{j\in\pi_p} \theta_j\right), \tag{1.3}$$

over partitions π having shape in a prescribed set. One application of this problem is to the optimal assembly of a system where each component is a series subsystem (see Hwang and Rothblum (1994) for a summary of literature). Hwang and Rothblum (to appear) studied the sum partition problem with objective function being Schur convex. In particular, they proved ([Theorem 3.2]) that given a set of shapes Γ and a majorizing shape $(n_1, \ldots, n_p)$ in that set (one that majorizes all other shapes in that set), one has the following:

(a) if $\theta_i \geq 0$ for each i, then every size-consecutive partition with shape $(n_1, \ldots, n_p)$ is optimal; and
(b) if $\theta_i \leq 0$ for each i, then every reverse-size-consecutive partition with shape $(n_1, \ldots, n_p)$ is optimal

In particular, once $n_1, \ldots, n_p$ are ordered so that $n_1 \leq \cdots \leq n_p$, the following explicit partitions are optimal under (a) and (b), respectively

$$\pi_i = \left(\sum_{j=1}^{i-1} n_j + 1, \ldots, \sum_{j=1}^{i} n_j\right) \text{ for } i = 1, \ldots, p \tag{1.4}$$

and

$$\pi_i = \left(n - \sum_{j=1}^{i} n_j + 1, \ldots, n - \sum_{j=1}^{i-1} n_j\right) \text{ for } i = 1, \ldots, p. \tag{1.5}$$

Of course, this is the case when Γ contains a single shape.

Barnes et al. (1992) first considered the problem where the size of each part must lie in a range, i.e., nonnegative integer vectors $L = (L_1, \ldots, L_p)$ and $U = (U_1, \ldots, U_p)$ are given where

$$\sum_{i=1}^{p} L_i \leq n \leq \sum_{i=1}^{p} U_i, \tag{1.6}$$

and the shape $(n_1, \ldots, n_p)$ of a feasible partition must satisfy

$$L_i \leq n_i \leq U_i, \quad i = 1, \ldots, p. \tag{1.7}$$

They proposed the partition polytope approach. Hwang et al. (1998) further explored the issues of representations and characterization of vertices. For nonnegative integer p-vectors $L = (L_1, \ldots, L_p)$ and $U = (U_1, \ldots, U_p)$ that satisfy (1.6), define $\Gamma(L, U)$ as the set of all partitions whose shape $(n_1, \ldots, n_p)$ satisfies (1.7). Hwang and Rothblum (to appear) gave the example where $n = 9$, $p = 3$, $L = (1, 2, 2)$ and $U = (5, 4, 4)$ to show that a majorizing shape may not exist (neither of the two shapes $(5, 2, 2)$ and $(1, 4, 4)$ majorizes the other). Still, they provided the following sufficient condition for the existence of the majorizing shape. Without loss of generality, assume

$$L_1 \leq L_2 \leq \cdots \leq L_p. \tag{1.8}$$

The condition that was determined to suffice for the existence of a majorizing shape is that

$$U_1 \leq U_2 \leq \cdots \leq U_p; \tag{1.9}$$

further, when this condition holds, an explicit simple expression for a majorizing shape was provided. The result stated above then shows how to obtain, when the θ_i's are one-sided, a corresponding size-consecutive/reverse-size-consecutive optimal partition. We note that it takes $O(p \ln p)$-time to order $L_1, \ldots, L_p$ and $O(p)$-time to check condition (1.9); if met, another $O(p)$-time is required to identify the majorizing shape.

In the current paper, we consider the general bounded-shape case without imposing the consistency condition of (1.8)–(1.9). Given nonnegative integer p-vectors L and U, a *nonmajorized shape for* (L, U) is a shape in $\Gamma(L, U)$ which is not strictly majorized by any other shape in $\Gamma(L, U)$. We will show that when f is Schur convex and the θ_i's are one-sided, one can restrict attention to (reverse) size-conscutive partitions having a nonmajorized shape. As a (reverse) size-consecutive partition with a given shape is easy to determine (see (1.4) and (1.5)), the problem of finding an optimal partition is reduced to the task of identifying a set of shapes that contains all nonmajorized ones.

Schur convex functions are symmetric. Thus, they do not differentiate between partitions that are obtained by part-permutations as long as the corresponding coordinate-permutations of the shapes are feasible. Thus, we may, in effect, restrict attention to representatives of *shape-types* which are the equivalence classes of (feasible) shapes with respect to coordinate-permutations (rather than to shapes).

In Section 2, we explore properties of nonmajorized shapes and shape-types. In Section 3, we obtain a 2^{p-1} bound on the number of nonmajorized shape-types. In Section 4, we provide an $O(2^p + p2^{p-5} \log p)$-time algorithm and $O(p^3 2^{2p})$-time an algorithm for enumerating, respectively, all nonmajorized shape-types and shapes. Finally, in Section 5, we give an $O(p^2)$-time algorithm that determines the existence of a majorizing shape and identifies one when the answer is positive.

2. Nonmajorized shapes

In this section, we explore the relation between shape-majorization and the optimization problem over partitions introduced in the Introduction. In particular, we explore the role of nonmajorized shapes, in particular, with respect to sets of the form $\Gamma(L, U)$.

Motivated by (1.3), for a partition $\pi = (\pi_1, \ldots, \pi_p)$ let

$$\theta^\pi = \left(\sum_{j \in \pi_1} \theta_j, \ldots, \sum_{j \in \pi_p} \theta_j \right) \tag{2.1}$$

Proposition 2.1. *Suppose f is Schur convex, Γ is a set of positive integer p-vectors that sum to n and π is a partition with shape in Γ which is majorized by a shape $(n_1, \ldots, n_p) \in \Gamma$.*

(a) If $\theta_i \geq 0$ for $i = 1, \ldots, n$, then every size-consecutive partition π' with shape $(n_1, \ldots, n_p)$ has $f(\theta^{\pi'}) \geq f(\theta^\pi)$.
(b) If $\theta_i \leq 0$ for $i = 1, \ldots, n$, then every reverse-size-consecutive partition π' with shape $(n_1, \ldots, n_p)$ has $f(\theta^{\pi'}) \geq f(\theta^\pi)$.

Proof: The proof is identical to that of Theorem 3.2 of Hwang and Rothblum (to appear). □

Corollary 2.2. *Suppose f and Γ are as in Proposition 2.1.*

(a) If $\theta_i \geq 0$ for $i = 1, \ldots, n$, then there is a nonmajorized shape in Γ such that any corresponding size-consecutive partition is optimal.
(b) If $\theta_i \leq 0$ for $i = 1, \ldots, n$, then there is a nonmajorized shape in Γ such that any corresponding reverse-size-consecutive partition is optimal.

Proof: We consider only the case where the θ_i's are nonnegative. Let π' be an optimal partition. As majorization is transitive and Γ is finite, Γ contains a shape $(n_1, \ldots, n_p)$ which majorizes the shape of π' and is not majorized by any other shape in Γ. By Proposition 2.1, any size-consecutive partition with shape $(n_1, \ldots, n_p)$ has $f(\theta^\pi) \geq f(\theta^{\pi'})$, so the optimality of π' assures that π is also optimal. □

Corollary 2.2 implies that when f is Schur convex and the θ_i's are one-sided, it suffices to restrict attention to (reverse) size-consecutive partitions whose shape is nonmajorized. Of course, the symmetry of Schur convex functions implies that all size-consecutive partitions with the same shape have the same objective value F (as determined by (1.3)). We conclude that the underlying optimization problem over partitions can be solved by obtaining a list that contains all nonmajorized shapes, determining corresponding size-consecutive partitions, and evaluating the right-hand side of (1.3) for each one of them. Further, it suffices to consider only representatives of all nonmajorized shape-types. The remainder of our paper will focus on studying and identifying nonmajorized shapes and shape-types with respect to sets of the form $\Gamma(L, U)$.

The next (standard) lemma is useful in exploring properties of nonmajorized vectors. For a vector $a \in \mathbb{R}^p$ and $J \subseteq \{1, \ldots, p\}$, let a_J denote the subvector of a consisting of the coordinates indexed by J.

Lemma 2.3. *Consider vectors a and b in $\mathbb{R}^p$ with $\sum_{i=1}^p a_i = \sum_{i=1}^p b_i$ and a set $J \subseteq \{1, \ldots, p\}$ for which $a_i = b_i$ for each $i \in \{1, \ldots, p\} \setminus J$. Then*

$$[a_J \text{ majorizes } b_J] \Leftrightarrow [a \text{ majorizes } b]\,; \tag{2.2}$$

further (2.2) holds with "majorizes" replaced by "strictly majorizes".

Proof: Suppose a_J majorizes b_J. Let $k \in \{1, \ldots, p-1\}$ be given and let K be a subset of $\{1, \ldots, p\}$ with $\sum_{i=1}^k b_{[i]} = \sum_{i \in K} b_i$. Set $m \equiv |K \cap J|$. As a_J majorizes b_J we have that $\sum_{i=1}^m (a_J)_{[i]} \geq \sum_{i=1}^m (b_J)_{[i]} \geq \sum_{i \in K \cap J} b_i$, hence, the assertion $a_i = b_i$ for each $i \in \{1, \ldots, p\} \setminus J$ implies that

$$\begin{aligned}\sum_{i=1}^k a_{[i]} &\geq \sum_{i=1}^m (a_J)_{[i]} + \sum_{i \in K \cap J^c} a_i \\ &\geq \sum_{i \in K \cap J} b_i + \sum_{i = K \cap J^c} b_i = \sum_{i \in K} b_i = \sum_{i=1}^k b_{[i]}\,.\end{aligned}$$

As $k \in \{1, \ldots, p-1\}$ was selected arbitrarily and (by assumption) $\sum_{i=1}^p a_i = \sum_{i=1}^p b_i$, we conclude that a majorizes b.

Next, assume that a majorizes b. As $a_i = b_i$ for each $i \in \{1, \ldots, p\} \setminus J$ and $\sum_{i=1}^p a_i = \sum_{i=1}^p b_i$, we have that $\sum_{i \in J} a_i = \sum_{i \in J} b_i$. Next, let $k \in \{1, \ldots, |J|-1\}$ be given and let K be a subset of J with $\sum_{i \in K} a_i = \sum_{i=1}^k (a_J)_{[i]}$. Consider the set W consisting of all indices $i \in \{1, \ldots, p\} \setminus J$ for which $a_i \geq \min\{a_i : i \in K\}$, and let $m \equiv |W|$ ($W = \emptyset$ and $m = 0$ is possible). For $k' = k + m$, we have that $\sum_{i=1}^{k'} a_{[i]} = \sum_{i \in K} a_i + \sum_{i \in W} a_i$. Consider any set $H \subseteq J$ with $|H| = k$. As a majorizes b,

$$\sum_{i \in K} a_i + \sum_{i \in W} a_i = \sum_{i=1}^{k'} a_{[i]} \geq \sum_{i=1}^{k'} b_{[i]} \geq \sum_{i \in H} b_i + \sum_{i \in W} b_i\,.$$

As $a_i = b_i$ for each $i \in \{1, \ldots, p\} \setminus J \supseteq W$, we conclude that

$$\sum_{i=1}^k (a_J)_{[i]} = \sum_{i \in K} a_i \geq \sum_{i \in H} b_i\,.$$

The freedom in selecting H and k allows us to conclude that a_J majorizes b_J.

The strict version of (2.2) follows directly from the weak version and the observation that a vector u strictly majorizes another vector v if and only if u majorizes v and v does not majorize u. □

Lemma 2.3 will be particularly used with sets J consisting of two elements.

Throughout the remainder of this section, let L and U be nonnegative integer p-vectors that satisfy (1.6)–(1.7). In particular, we refer to a nonmajorized shape under $\Gamma(L, U)$ as a *nonmajorized shape*. We next explore the properties of such shapes.

Lemma 2.4. *Consider the following properties of a shape $s = (n_1, \ldots, n_p)$:*
(a) s is nonmajorized;
(b) there exist no distinct i and j such that

$$L_j < n_i < U_i \ \text{ and } \ L_j < n_j < U_i\,; \tag{2.3}$$

(c) if for distinct i and j, $L_j < n_j$ and $n_i < U_i$, then $n_i < n_j$; and
(d) there exists at most one index i with $L_i < n_i < U_i$.

Then (a) ⇒ (b) ⇒ (c) ⇒ (d).

Proof: (a) ⇒ (b). Suppose n_i and n_j satisfy (2.3) where $i \neq j$. Without loss of generality, assume that $n_i \geq n_j$. Then s is majorized by the shape obtained from s by increasing n_i to $\max\{n_i, n_j\} + 1$, and decreasing n_j to $\min\{n_i, n_j\} - 1$ (see Lemma 2.3).

(b) ⇒ (c). Suppose condition (b) holds, and i and j are indices satisfying $L_j < n_j, n_i < U_i$ and $i \neq j$. By condition (b), either $L_j \geq n_i$ or $n_j \geq U_i$. In the former case, $n_i \leq L_j < n_j$ and in the latter case $n_j \geq U_i > n_i$.

(c) ⇒ (d). Suppose condition (c) holds, and i and j are indices satisfying $L_i < n_i < U_i$, $L_j < n_j < U_j$ and $i \neq j$. We will establish a contradiction. Indeed, if $n_i \geq n_j$ we get a direct violation of (c) and if $n_i < n_j$ we get a violation of (c) with the roles of i and j reversed. □

The following examples shows that condition (b) of Lemma 2.4 does not imply condition (a).

Example 2.1. Let $U = (5, 5, 5, 2)$, $L = (1, 4, 3, 1)$, $s = (5, 4, 3, 1)$ and $s' = (2, 5, 5, 1)$. It is easy to verify that s is majorized by s'. To see that there exist no i and j satisfying (2.3) for s, observe that the only coordinate of s that is strictly larger than the lower bound is the first one, so if (2.3) is satisfied, necessarily $j = 1$. But, n_1 is not strictly below any upper bound.

For a given shape s, call part i an *upper part*, a *middle part* or a *lower part* if, respectively, $n_i = U_i$, $L_i < n_i < U_i$, $n_i = L_i$. If part i has $L_i = U_i$, each shape $(n_1, \ldots, n_p) \in \Gamma(L, U)$ has $n_i = L_i = U_i$. Thus, in search of nondominated shapes under (L, U), one can ignore such parts. Of course, when $L \ll U$ (i.e., $L_i < U_i$ for each i), the parts are classified uniquely. Lemma 2.4 shows that a nonmajorized shape can have at most one middle part.

Suppose $L \ll U$. Given a shape $s = (n_1, \ldots, n_p)$, let $B(s)$ stand for the p-vector whose elements are the symbols L, M and U constructed in the following way: For a permutation $i_1, \ldots, i_p$ of the coordinates for which $n_{i_1} \geq n_{i_2} \geq \cdots \geq n_{i_p}$, let $B(s)_t$ for $t = 1, \ldots, p$ be L, M, U according to i_t being an upper, middle or lower part. The next result shows that no ambiguity can arise in the definition of $B(s)$, i.e., it is uniquely defined, and that $B(s)$ has a simple structure.

Lemma 2.5. *Suppose $L \ll U$ and $s = (n_1, \ldots, n_p)$ is a nonmajorized shape. Let $(i_1, \ldots, i_p)$ be a permutation of $(1, \ldots, p)$ such that $n_{i_1} \geq n_{i_2} \geq \cdots \geq n_{i_p}$. Then:*

(a) $n_{i_r} = n_{i_t}$ for $r, t \in \{1, \ldots, p\}$ implies i_r and i_t are either both upper parts or both lower parts.

(b) $B(s)$ has the form $(U, \ldots, U, M, L, \ldots, L)$ or $(U, \ldots, U, L, \ldots, L)$.

Proof: (a) If $n_{i_r} = n_{i_t}$, i_r is a lower-part and i_t is not, then $L_{i_t} < n_{i_t} = n_{i_r} = L_{i_r} < U_{i_r}$, in contradiction to implication (a) $\Rightarrow$ (b) in Lemma 2.4. A similar argument applies to prove that if i_r is an upper-part, so is i_t.

(b) The implication (a) $\Rightarrow$ (c) in Lemma 2.4 assures that if $n_j = U_j > L_j$ and $n_i < U_i$, then $n_i < n_j$, and that if $n_i = L_i < U_i$ and $n_j > L_j$, then $n_i < n_j$. It follows that for every permutation $i_1, \ldots, i_p$ of $1, \ldots, p$ with $n_{i_1} \geq \cdots \geq n_{i_p}$ and $r, t \in \{1, \ldots, p\}$

$$\left[n_{i_r} = U_{i_r} \text{ and } n_{i_t} < U_{i_t}\right] \Rightarrow [r < t]$$

and

$$\left[n_{i_t} = L_{i_t} \text{ and } n_{i_r} > L_{i_r}\right] \Rightarrow [r < t].$$

These implications establish the asserted structure of $B(s)$. $\square$

We conclude this section with an observation about a necessary difference between two nonmajorized shapes.

Lemma 2.6. *Two distinct nonmajorized shapes $s = (n_1, \ldots, n_p)$ and $s' = (n'_1, \ldots, n'_p)$ must differ in at least two coordinates; further, if such s and s' differ in exactly two coordinates, say coordinate i and coordinate j, where $n_i > n'_i$, then s' is obtained from s by permuting these coordinates,*

$$n_i = U_i \quad or \quad n_j = L_j \tag{2.4}$$

and

$$n'_i = L_i \quad or \quad n'_j = U_j\,. \tag{2.5}$$

Proof: Suppose shapes s and s' differ in only one part, then $\sum_i n_i \neq \sum_i n'_i$, contradicting the fact that both are shapes and their coordinate sum is n.

Next, assume that $s = (n_1, \ldots, n_p)$ and $s' = (n'_1, \ldots, n'_p)$ are nonmajorized shapes that differ only in coordinates i and j. As neither strictly dominates the other (they are nonmajorized), we have that s' is obtained from s by permuting two coordinates, say coordinates i and j. Now, suppose $n_i < n'_i = n_j$. As $L_j \leq n_j = n'_i < n_i \leq U_i$, the implication (a) $\Rightarrow$ (b) in Lemma 2.4 assures that either $n_i = U_i$ or $n_j = L_j$, and (applying the result on s' with the roles of i and j reversed), either $n'_j = U_j$ or $n'_i = L_i$. $\square$

We say that two shapes are *equivalent* if one is obtained from the other by coordinate-permutation. Of course, not all coordinate-permutations of a shape in $\Gamma(L, U)$ are necessarily in $\Gamma(L, U)$.

The following is an immediate corollary of Lemma 2.6.

Corollary 2.7. *If s and s' are nonmajorized shapes which are not equivalent, then they differ in at least 3 coordinates.*

3. The number of nonmajorized shape-types

In the current section, we continue to assume that L and U are integer p-vectors satisfying (1.6) and $L \ll U$. We recall that two shapes are equivalent if one is obtained as a coordinate permutation of the other. We refer to the resulting equivalence classes as *shape-types*. As strict-majorization is (clearly) invariant of the corresponding shape-types, we can and will refer to *nonmajorized* shape-types. The purpose of the current section is to derive a bound on the number of nonmajorized shape-types.

We note that a single nonmajorized shape-type may correspond to many shapes as the following example suggests.

Example 3.1. Let $L = (1, \ldots, 1)$, $U = (2, \ldots, 2)$ and $p < n < 2p$. Then all nonmajorized shapes are equivalent and each such shape, say $(n_1, \ldots, n_p)$ is determined by a set J of $\{1, \ldots, p\}$ consisting of $n - p$ elements, where $n_i = 2$ if $i \in J$ and $n_i = 1$ otherwise. So, there is a single nonmajorized shape-type that corresponds to $\binom{p}{n-p}$ nonmajorized shapes.

A (nonmajorized) shape-type can be identified with the multiset $\{n_1, \ldots, n_p\}$ where $(n_1, \ldots, n_p)$ is any corresponding shape. It is noted that not every ordering of $n_1, \ldots, n_p$ necessarily yields a feasible shape, that is, one that satisfies the lower and upper bounds.

For a nonmajorized shape $s = (n_1, \ldots, n_p)$, let $U(s)$, $M(s)$ and $L(s)$ be set of corresponding upper-, middle- and lower-parts of s, that is, $U(s) = \{j = 1, \ldots, p : n_j = U_j\}$, $M(s) = \{j = 1, \ldots, p : L_j < n_j < U_j\}$ and $L(s) = \{j = 1, \ldots, p : n_j = L_j\}$.

Lemma 3.1. *Suppose $s = (n_1, \ldots, n_p)$ and $s' = (n'_1, \ldots, n'_p)$ are nonmajorized shapes that are not equivalent. Then:*
(a) $U(s) \neq U(s')$, and
(b) if $U(s')$ is included in $U(s)$, then $M(s')$ contains a single element j' that satisfies

$$U_{i'} > n'_{j'} \text{ for every } i' \text{ in } U(s') \tag{3.1}$$

and

$$U_i \le n'_{j'} \text{ for every } i \text{ in } U(s) \backslash U(s') . \tag{3.2}$$

Proof: (a) Lemma 2.4 assures that $|M(s)| \le 1$ and $|M(s')| \le 1$. Thus, if $U(s) = U(s')$, then s and s' can differ in at most 2 coordinates; it then follows from Corollary 2.7 that s and s' are equivalent, in contradiction to the assertion that they are not.

(b) Suppose $U(s) \supseteq U(s')$. As $s' \neq s$, there is a coordinate j' with $n'_{j'} > n_{j'}$. We will show that such a j' must be in $M(s')$. Indeed, such j' cannot be in $U(s')$ for the assertion $U(s) \supseteq U(s')$ would imply $j' \in U(s)$ and $n'_{j'} > n_{j'} = U_{j'}$; such j' can neither be in $L(s')$ because $n'_{j'} > n_{j'} \geq L_{j'}$. So, j' must be in $M(s')$. By Lemma 2.4, there can be at most a single part in $M(s')$. Thus, $M(s') = \{j'\}$ and j' is the single coordinate for which s' exceeds s.

Now, for i' in $U(s')$, $n_{i'}' = U_{i'} > L_{i'}$. As $n_{j'}' < U_{j'}$, the (a) $\Rightarrow$ (c) part of Lemma 2.4 implies that $n'_{j'} < n'_{i'} = U_{i'}$, proving (3.1).

Next, assume that i is in $U(s) \setminus U(s')$. As s and s' differ by at least 3 coordinates (Corollary 2.7), as j' is the single coordinate for which s' exceeds s and as $n_i = U_i > n'_i$, we have that $i \neq j'$ and

$$n'_{j'} - n_{j'} > n_i - n'_i = U_i - n_i \,. \tag{3.3}$$

Assume that $U_i > n'_{j'}$ and we will establish a contradiction. By summing $U_i > n'_{j'}$ and (3.3), we get that $n_i > n_{j'}$. As i is not in $U(s')$, $n'_i < U_i$. Consider the shape obtained from s' by increasing n'_i to U_i and decreasing $n'_{j'}$ to $n'_{j'} - [U_i - n'_i]$. As $U_i > n'_{j'}$, this shape majorizes s' (recall Lemma 2.3). Further, (3.3) implies that $n'_{j'} - [U_i - n'_i] > n_{j'} \geq L_{j'}$, assuring that the new shape is in $\Gamma(L, U)$. As s' is assumed to be nonmajorized, we have derived a contradiction which established (3.2). □

Corollary 3.2. *Suppose s, s' and s'' are nonmajorized shapes where no pair consists of two equivalent shapes, and suppose $U(s')$ and $U(s'')$ are both included in $U(s)$. Then $U(s')$ and $U(s'')$ are ordered by set-inclusion.*

Proof: Let $s' = (n'_1, \ldots, n'_p)$ and $s'' = (n''_1, \ldots, n''_p)$. Part (b) of Lemma 3.1 assures that $M(s')$ and $M(s'')$ are nonempty. Let $M(s') = \{j'\}$ and $M(s''') = \{j''\}$. Without loss of generality, assume that $n'_{j'} \leq n''_{j''}$. By Lemma 3.1(a), $U(s') \neq U(s'')$. Suppose $U(s') \not\supseteq U(s)$. Then there exists $k \in U(s'') \cap (U(s) \setminus U(s'))$. By Lemma 3.1(b), $n'_{j'} \geq U_k > n''_{j''}$, contradicting our assumption $n'_{j'} \leq n''_{j''}$. □

We next explore the combinatorial restriction imposed by the conclusion of Corollary 3.2. For that purpose, for each integer $p \geq 1$, let $f(p)$ be the maximal size of a class C of subsets of $\{1, \ldots, p\}$ which satisfies the conclusion of Corollary 3.2, that is, every pair of subsets in C that are included in a third subset of C must be comparable by set-inclusion. The next table lists values of $f(p)$ for $p = 0, 1, 2, 3, 4, 5, 6$.

We next obtain an upper bound on $f(p)$.

Theorem 3.3. $f(p) \leq 2^{p-1}$.

Proof: Consider any $p \in \{1, 2, \ldots\}$ and let $F(p)$ realize $f(p)$. Also, let $F_0(p) = \{U \in F(p) : p \notin U\}$ and $F_1(p) = \{U \in F(p) : p \in U\}$. As $F_0(p)$ and $\{U\backslash\{p\} : U \in F_1(p)\}$ are classes of subsets of $\{1, \ldots, p-1\}$ with the property that every pair of sets in class that are included in a third set in the class must be comparable by set-inclusion, we have that $|F_0(p)| \leq f(p-1)$ and $|F_1(p)| \leq f(p-1)$, implying that $f(p) = |F(p)| = |F_0(p)| + |F_1(p)| \leq 2f(p-1)$. As $f(4) = 8 = 2^3$, we conclude that $f(p) \leq 2^{p-1}$ for each $p \geq 4$. □

Table 1 $f(p)$ for $0 \leq p \leq 6$

p	$f(p)$	A realizing class for $f(p)$
0	1	$\emptyset$
1	2	$\{1\},\ \emptyset$
2	3	$\{1,2\},\ \{1\},\ \emptyset$
3	5	$\{1,2\},\ \{1\ 3\},\ \{2,3\},\ \{1\},\ \emptyset$
4	8	$\{1,2\},\ \{1\ 3\},\ \{1,4\},\ \{2,3\},\ \{2,4\},\ \{3,4\},\ \{1\},\ \emptyset$
5	14	$\{1,2,5\},\ \{1,3,5\},\ \{1,4,5\},\ \{2,3,5\},\ \{2,4,5\},\ \{3,4,5\}$
		$\{1,2\},\ \{1,3\},\ \{1,4\},\ \{2,3\},\ \{2,4\},\ \{3,4\},\ \{1\},\ \emptyset$
6	23	All subsets of $\{1,\ldots,6\}$ of size 3, $\{1,2\},\ \{1\},\ \emptyset$

Table 2 The set of nonmajorized shapes

n_1	n_2	n_3	n_4	n_5
20	19	3	4	5
20	2	18	4	7
20	2	3	17	9
1	19	18	4	9
1	19	3	17	11
1	2	18	17	13

Corollary 3.4. *For $p \geq 4$, there are at most 2^{p-1} nonmajorized shape-types.*

Proof: Corollary 3.2 and Lemma 3.1 show that $f(p)$ bounds the number of nonmajorized shape-types and Theorem 3.3 shows that $f(p) \leq 2^{p-1}$. $\square$

The proof of Corollary 3.4 relies on the facts that 2^{p-1} is an upper bound on $f(p)$ (for $p \geq 4$) and that $f(p)$ is an upper bound on the number of unmajorized shape-types. Table 1 demonstrates that 2^{p-1} is not a tight bound on $f(p)$ and we believe that neither is the second bound. In fact, we conjecture that the number of nonmajorized shape-types can be bounded by $\binom{p-1}{\lfloor (p-1)/2 \rfloor}$, a smaller expression than 2^{p-1}. (By the Sperner's lemma (Sperner, 1928)), $\binom{p-1}{\lfloor (p-1)/2 \rfloor}$ is the maximum number of independent subsets in the lattice of subsets of $\{1, \ldots, p-1\}$ with set-inclusion as the partial order.) The following example achieves this (conjectured) bound.

Example 3.2. Let $U = (20, 19, 18, 17, 16), L = (1, 2, 3, 4, 5), n = 51$. Using the algorithms of Section 4 (see Example 4.1), one can show that the set of all nonmajorized shapes contains 6 shapes that are listed below in Table 2.

4. Identifying all nonmajorized shapes and shape-types

In the current section, we provide two algorithms for enumerating all nonmajorized shapes. The first algorithm relies on condition (d) of Lemma 2.4. We consider only bound vectors L and U with $L \ll U$, as indices i with $L_i = U_i$ can be eliminated by setting $n_i = L_i = U_i$ and updating n correspondingly.

Algorithm 1 (For enumerating all nonmajorized shapes in $\boldsymbol{\Gamma(L,U)}$).*:*
The input for the algorithm consists of integer p-vectors L and U that satisfy (1.6) and $L \ll U$.

(a) For $u = 1, \ldots, p$ and $A \subseteq \{1, \ldots, p\}\backslash\{u\}$ do:

(i) Set $B = \{1, \ldots, p\}\backslash A\backslash\{u\}$, $U_A = \sum_{i\in A} U_i$, $L_B = \sum_{i\in B} L_i$ and $M_u = n - U_A - L_B$.
(ii) If $L_u \le M_u \le U_u$, set

$$n_j = \begin{cases} U_j & \text{for } j \in A, \\ L_j & \text{for } j \in B, \\ M_u & \text{for } j = u, \end{cases}$$

and include $(n_1, \ldots, n_p)$ in a temporary list that we denote TEMP.

(b) Test each shape in TEMP for being nonmajorized by testing if it is majorized by any shape in TEMP.

The next lemma analyzes Algorithm 1. For the complexity analysis, computational effort counts arithmetic operations and comparisons.

Lemma 4.1.
(a) At the end of step (a), TEMP contains all nonmajorized shapes.
(b) The output of Algorithm 1 consists of all nonmajorized shapes in $\Gamma(L,U)$.
(c) The computational time in executing step (a) of Algorithm 1 is bounded by $O(p2^{p-1})$, *and the computational time in executing the complete algorithm is bounded by* $O(p^3 2^{2p})$.

Proof:

(a) Lemma 2.4 (part (b)) assures that at the completion of step (a), TEMP contains all nonmajorized shapes.
(b) As transitivity of the majorization relation assures that a majorized shape is majorized by some nonmajorized shape, a test for a shape to be nonmajorized is to compare it with all the shapes in TEMP.
(c) The number of iterations within step (a) is $p2^{p-1}$. The initial calculation of the quantity U_A, L_B and M_u requires $p-1$ addition/subtraction and the updates within each iteration requires $O(1)$ computational time. Hence, the total time to execute step (a) is $O(p2^{p-1})$ and the output may contain up to $p2^{p-1}$ shapes.

In step (b), each output shape of step (a) is tested against all others. The test requires determining the order statistics of the shapes, creating their partial sums, and executing p comparisons for each pair of shapes. The total time is then bounded by $O[(p + p\lg p)p2^{p-1} + (p2^{p-1})^2 p] = O[p^3 2^{2p}]$. □

Remark 1. One can thin TEMP (and save on computational time in executing Step (b) of Algorithm 1) by using sufficient conditions for nonmajorized shapes and avoiding repetitions. For example, consider a shape $s = (n_1, \ldots, n_p)$ generated in Step (a) corresponding to $u \in \{1, \ldots . p\}$ and $A \subseteq \{1, \ldots, p\}\backslash\{u\}$. With $B = \{1, \ldots, p\}\backslash A\backslash\{u\}$, it is possible to exclude s from TEMP if either $n_u \le \min_{i\in A} U_i$ or $n_u \ge \max_{i\in B} L_i$. Indeed, if either condition holds and $L_u < n_u < U_u$, then Lemma 2.5 assures that s is majorized. Alternatively, if $U_u = n_u >$

$\min_{i \in A} U_i = U_v$ or $L_u = n_u < \max_{i \in B} L_i = L_v$, then s will be reproduced and kept when executing Step (a) corresponding to v and $(A \backslash \{v\}) \cup \{u\}$.

Given integer p-vectors L and U satisfying (1.6)–(1.7), the set of *floating indices of* (L, U) is defined as $\{i = 1, \ldots, p : L_i < U_i\}$. Also, if G is the set of indices of (L, U) which are not floating, we refer to $n - \sum_{i \in G} L_i (= n - \sum_{i \in G} U_i)$ as the *availability* under (L, U). We say that the *upper bound of index i* is *effective* for (L, U) if

$$U_i + \sum_{j \neq i} L_j \leq n; \tag{4.1}$$

when the upper bound of index i is not effective, we refer to the replacement of U_i by $n - \sum_{j \neq i} L_j \geq L_i$ as *the adjustment of the upper bound of* i. Similarly, we say that the *lower bound of index i* is *effective* for (L, U) if

$$L_i + \sum_{j \neq i} U_j \geq n, \tag{4.2}$$

and if the lower bound of index i is not effective, we refer to the replacement of L_i by $n - \sum_{j \neq i} U_j \leq U_i$ as *the adjustment of the upper bound of* i. Evidently, (1.6) and (1.7) stay in effect when an upper bound or a lower bound is adjusted.

Lemma 4.2. *Consecutive adjustment of bounds results in a pair of vectors for which all bounds are effective, and this outcome is independent of the order in which bounds are adjusted.*

Proof: Trivially, consecutive adjustment of bounds must terminate with a pair of vectors for which all bounds are effective.

Evidently, (1.6) and (1.7) stay in effect when a bound is adjusted. Further, if the upper bound of i needs adjustment, all the lower bounds of indices $j \neq i$ are effective throughout any sequence of adjustments; this is the case because a decrease in an upper bound does not invalidate the effectiveness of a lower bound and an increase in a lower bound does not invalidate effectiveness of an upper bound. We conclude that if an upper/lower bound of i is adjusted, no lower/upper bound of another $j \neq i$ will require adjustment. Further, the order of consecutive adjustment of upper bounds or of lower bounds has no effect on the outcome. The only remaining case is the adjustment of the upper bound and the lower bound of a particular i—it is easy to verify that here, too, the order of executing these adjustments does not influence the outcome. □

We refer to the operation that is described in Lemma 4.2 as an *adjustment of the bounds*. We observe that (1.6) assures that the bounds of indices that are not floating, are always effective and will therefore not be affected by an adjustment of the bounds. But, bound-adjusting can reduce the set of floating indices.

Algorithm 2 (For enumerating all nonmajorized shape-types in $\boldsymbol{\Gamma(L, U)}$)**:**
The input for the algorithm consists of integer p-vectors L and U that satisfy (1.6). Set $r = 1$.

Iteration r:

(a) Adjust the bounds (L, U). Let F and v be the set of floating indices and the availability with respect to the adjusted bounds and set $n_i = L_i = U_i$ for each $i \in \{1, \ldots, p\} \backslash F$.
If $F = \emptyset$, set $r = p$ and go to step (c). Otherwise, set $\alpha \equiv \max_{k \in F} U_k$ and $\beta \equiv \min_{k \in F} L_k$.
(b) Execute, in parallel and record separately the outcome of the following three steps:

(i) Select i as any index that maximizes the lower bound among those whose upper bound is α. Set $n_i \leftarrow U_i$ and $L_i \leftarrow U_i$.
(ii) Select i as any index that minimizes the upper bound among those whose lower bound is β. Set $n_i \leftarrow L_i$ and $U_i \leftarrow L_i$.
(iii) This option is executed only if one identifies an index i that satisfies $U_i = \alpha > U_j$ for each $j \neq i$, $L_i = \beta < L_j$ for each $j \neq i$ and $F \backslash \{i\}$ can be partitioned into two sets A and B such that

$$|A| \geq 2\,,\ |B| \geq 2\,, \tag{4.3}$$

$$\max_{k \in B} U_k \leq n - \sum_{j \in A} U_j - \sum_{k \in B} L_k \leq \min_{j \in A} L_j \tag{4.4}$$

and

$$L_i < n - \sum_{j \in A} U_j - \sum_{k \in B} L_k < U_i\,. \tag{4.5}$$

When the above holds with (4.4) in strict inequalities, do for each such pair A, B the following: Set $n_t \leftarrow U_t$ and $L_t \leftarrow U_t$ for $t \in A$, $n_s \leftarrow L_s$ and $U_s \leftarrow L_s$ for $s \in B$, and $n_i \leftarrow \mu \equiv n - \sum_{j \in A} U_j - \sum_{k \in B} L_k$, $U_i \leftarrow \mu$ and $L_i \leftarrow \mu$.

Let n_i denote the middle part of (4.4). Suppose $n_i = \max_{k \in B} U_k \equiv U_x$. Check the existence of a part y in $B \backslash \{x\}$ such that $|(L_x, U_x) \cap (L_y, U_y)| \geq 2$. If no such y exists, then output this shape-type as in the (4.4) in strict inequalities case.

Similarly, suppose $n_i = \min_{j \in A} L_j = L_z$. Check the existence of a part w in $A \setminus \{z\}$ such that $|(L_z, U_z) \cap (L_w, U_w)| \geq 2$. If no such w exists, then output this shape-type.

(c) If $r = p$, output the shape-types of all generated shapes in step (b)(i) and (b)(ii). Otherwise, replace r with $r + 1$ and go to step (a) with each outcome of step (b)(i) and of step (b)(ii).

Remarks.

(1) Step (b) of Algorithm 2 allows a selection between 3 options. Option (iii) can be executed only if one identifies an index i with $U_i > U_j$ and $L_i < L_j$ for each $j \neq i$. When such an index i is identified, options (i) and (ii) will be executed with this particular selection of i. Option (iii) will then be followed for each partition of $F \backslash \{i\}$ into sets A and B that satisfy (4.3)–(4.5). It is possible to have no such pair A, B, or alternatively, to have multiple pairs.

(2) Ambiguity can occur in Algorithm 2 only in steps (b)(i) and (b)(ii) when there is more than one index i with $U_i = \alpha$ and $L_i = \max\{L_k : U_k = \alpha\}$ or, respectively, with $L_i = B$ and $U_i = \min\{U_k : L_k = \beta\}$. In these cases, the corresponding outputs of the algorithm will obviously generate the same shape-types.

(3) Whenever option (b)(iii) is completed with a particular selection of A, B, there will be no free variables in the next iteration and the algorithm will stop.

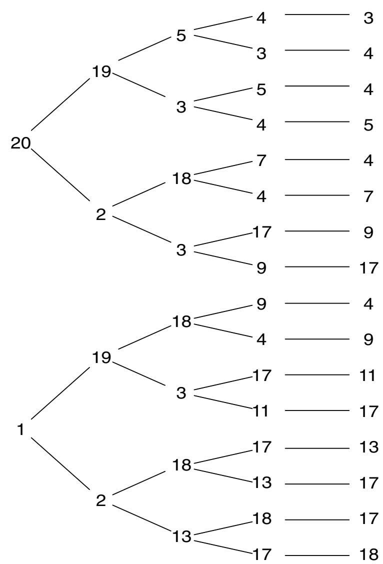
(4) If in a given iteration, option (b)(i)/(b)(ii) selects index i whose upper/lower bound was adjusted in that iteration, then the next iteration will have $F = \emptyset$ and the algorithm will stop.

(5) If at the beginning of an iteration there is only one index i with $L_i < U_i$, then the adjustment of the bounds will result in $F = \emptyset$ and the algorithm will stop. In particular, as each iteration eliminates at least one free index, one will never enter step (b) in iteration p.

We refer to option (i), (ii) and (iii) in Algorithm 2 as, respectively, a U-*step*, an L-*step* and an E-*step*. We refer to an E-shape as one that is determined when an E-step is executed.

The next example shows now Algorithm 2 is executed without the need for an E-step.

*Example 4.1 (*Continuing Example 3.2). Applying Algorithm 2 to Example 3.2 is summarized in the following figure.



The corresponding nonmajorized shapes are listed in Table 1. □

The following examples demonstrate that there may be more than one option in executing step (b)(iii) of Algorithm 2 and that step (b)(i) (or (b)(ii)) may be followed even when step (b)(iii) is possible.

Example 4.2. $U = (13, 12, 12, 8, 8, 4, 4)$, $L = (1, 10, 10, 6, 6, 2, 2)$ and $n = 49$. Then the nonmajorized shapes (13, 10, 10, 6, 6, 2, 2) and (1, 12, 12, 8, 8, 4, 4) are determined by following a U-step and an L-step, respectively, in the first iteration. We also find two shapes (5, 12, 12, 8, 8, 2, 2) and (9, 12, 12, 6, 6, 2, 2), by initial use of E-steps, corresponding respectively to the partitions $A = \{2, 3, 4, 5\}$, $B = \{6, 7\}$ and $A' = \{2, 3\}$, $B' = \{4, 5, 6, 7\}$.

There are two groups {2, 3}, {4, 5}, {6, 7} of parts in Example 4.2 having, respectively, the same bounds. In general, g groups would yield up to $g - 1$ partitions.

Example 4.3. $U = (11, 10, 10, 10, 7, 7, 7, 5, 3, 3, 3)$, $L = (1, 9, 9, 9, 6, 6, 6, 4, 2, 2, 2)$ and $n = 66$. Then the nonmajorized shapes are: $s_1 = (11, 9, 9, 9, 6, 6, 6, 4, 2, 2, 2)$, $s_2 = (8, 10, 10, 10, 6, 6, 6, 4, 2, 2, 2)$, $s_3 = (5, 10, 10, 10, 7, 7, 7, 4, 2, 2, 2)$, $s_4 = (4, 10, 10, 10, 7, 7, 7, 5, 2, 2, 2)$ and $s_5 = (1, 10, 10, 10, 7, 7, 7, 5, 3, 3, 3)$. Then s_2 is an example of an E-shape with strict inequalities in (4.4), and s_3 and s_4 are examples of an E-shape with nonstrict inequalities in (4.4).

Example 4.4. $U = (11, 9, 8, 10, 4, 4, 4, 4, 4)$, $L = (3, 6, 6, 0, 2, 2, 2, 2, 2)$ and $n = 43$. If one starts with a U-step, an output can be determined in the next iteration by an E-step, or a U-step resulting, respectively, in the output $(11, 9, 8, 5, 2, 2, 2, 2, 2)$ and $(11, 6, 6, 10, 2, 2, 2, 2, 2)$. Alternatively, one may start with an L-step, which will eliminate the option of an E-step with $i = 4$; then L_1 will be adjusted to 6, and the output $(11, 9, 8, 0, 4, 4, 4, 3, 2)$ can be generated.

The next lemma refers to sensitivity of being nonmajorized.

Lemma 4.3. *Let $\{(L_j, U_j) \mid j = 1, \ldots, p\}$ and $\{(L'_j, U'_j) \mid j = 1, \ldots, p\}$ be two sets of bounds which differ only in one bound corresponding to part j where either $L_j = L'_j$ and $U_j < U'_j$, or $L_j > L'_j$ and $U_j = U'_j$. Then, for a given n, every shape in $\Gamma(L, U)$ is majorized by a nonmajorized shape in $\Gamma(L', U')$.*

Proof: Let s be a nonmajorized shape in $\Gamma(L, U)$. Then s is also a shape in $\Gamma(L', U')$. Thus, it is either a nonmajorized shape, or is majorized by a nonmajorized shape in $\Gamma(L', U')$. □

By Lemma 4.3, we order the upper bounds such that $U_i \succ U_j$ either if $U_i > U_j$ or $U_i = U_j$ but $L_i > L_j$. Similarly, $L_i \prec L_j$ either if $L_i < L_j$ or $L_i = L_j$ but $U_i < U_j$. Obviously, if $U_i = U_j$ and $L_i = L_j$, then the order between i and j does not matter. Under $\prec$, we have a linear order for the upper(lower) bounds.

Lemma 4.4. *Let s be a shape output by Algorithm 2. Suppose N_k, consisting of j upper bounds and $k - j$ lower bounds, is the set of values obtained before an E-step in s (if no E-step occurs, then $k = p$). Let s' be any other shape. If s' majorizes s, then the j^{th} largest n'_i and the $(k - j)^{\text{th}}$ smallest n'_i must be equivalent to N_k.*

Proof: We prove Lemma 4.4 by induction on k. The case $k = 1$ is trivial. Consider a general k. Without loss of generality, assume the first step of s is taking the largest upper bound $U_{[1]}$. If the largest $n'_i < U_{[1]}$, then s' cannot majorize s. If they are equal, then by Lemma 4.3 we may assume s' takes the same part as s. Delete this part from the problem and k is reduced to $k - 1$. Use induction. □

Corollary 4.5. *A regular shape output by Algorithm 2 is nonmajorized.*

Theorem 4.6.

(a) Every shape that is constructed by Algorithm 2 is nonmajorized.
(b) For every nonmajorized shape, there is an equivalent shape that is constructed by Algorithm 2.
(c) The number of outputs of the algorithm is bounded by 2^{p+1} (duplications are possible).

(d) The computational time of all executions of Algorithm 2 is bounded by $O(2^p + p2^{p-5} \log p)$.

Proof: (a) By Corollary 4.5, we only need to consider an E-shape s. Suppose to the contrary that s' majorizes s. By Lemma 4.4, s' majorizes s in the remaining $p - k$ parts. But this is impossible by our construction of an E-shape whose largest k-sum, $1 \leq k \leq |A|$, is $\geq$ the largest k-sum of s', and whose smallest k-sum, $1 \leq k \leq |B|$, $\leq$ the smallest $|B|$-sum of s'. This proves that for the remaining parts, s either majorize s' or they are equivalent.

(b) Now, suppose at a given iteration, there exists a nonmajorized shape s which contains neither the maximum upper bound U_i nor the minimum lower bound L_j. Suppose $i \neq j$. Let s choose $n_i < U_i$ and $n_j > L_j$. Since $U_i > n_j$ and $n_i > L_j$, we can choose $n_i' = \max\{n_i, n_j\} + 1$ and $n_j{}' = \min\{n_i, n_j\} - 1$ to obtain a shape majorizing s, contradicting the assumption that s is nonmajorized.

Assume $i = j$ but s takes n_i such that $L_i < n_i < U_i$. By the comment after Lemma 4.3, $U_i \succ U_j$ and $L_i \prec L_j$ for any remaining part j.

Suppose there exists a part j such that $L_j < n_i < U_j$. Without loss of generality, assume $n_j = U_j$. Then s is majorized by s' with $n_i' = U_j + 1$ and $n_j' = n_i - 1$.

Next suppose $L_j = n_i$, which implies $n_j = U_j$, i.e., $j \in A$. Suppose that there exists another part x in A such that $(L_j, U_j) \cap (L_x, U_x) \neq \emptyset$. Then s is majorized by s' with $n'_i = \max\{U_j, U_x\} + 1$, $n'_j = L_j$, $n'_x = U_x - (n'_i - U_j)$. Note that if $n'_i = U_x + 1$, then $n'_x = U_j - 1 \geq L_x$ implies the part-j range and the part-x range must overlap by at least 2. We have shown that s can be a nonmajorized shape only if condition (4.4) is satisfied.

Finally, we justify (4.3). Suppose that there exists an E-shape s with $|A| = 1$. Without loss of generality, assume $U_1 = \max\{U_i\}$, $L_1 = \min\{L_i\}$, $A = \{2\}$, $B = \{3, 4, \ldots, p\}$, $L_2 > n_1 > U_i$ for all $i \in B$, and $n = n_1 + U_2 + (L_3 + L_4 + \cdots + L_p)$. Then U_1 is adjusted to U'_1 such that $U'_1 < U_2$ because $U_1 + (L_2 + L_3 + \cdots + L_p) > n$. Then s, as an non-E-shape, will be generated by selecting the largest upper bound U_2. Therefore we can restrict our construction of E-shape under the conditions $|A| \geq 2$ and $|B| \geq 2$.

(c) The underlying graph of the part of Algorithm 2 yielding regular shapes is a complete binary tree with depth $p - 1$ (n_i of the last part is determined by the previous $p - 1$ choices). Hence there are at most 2^{p-1} terminal points yielding 2^{p-1} regular shapes. At every path and every stage i, $1 \leq i \leq p - 4$, an E-step may occur. The reason for the upper bound of i is due to (4.3), which specifies that at lest 5 parts remain for an E-shape to exist. The maximum number of E-shapes at stage i is $1 + (n - i - 4)$, since the first A-set and the last B-set must contain at lest two parts, while the other $A(B)$-set can increase by 1. Summing over i, we obtain $2^{p-1} + \sum_{i=1}^{p-4} 2^i(n - i - 3) = (\frac{3}{2}) \times 2^{p-1} + 1$.

(d) For easier analysis of time complexity, we write the subroutine which separates the remaining parts into A and B in pseudo code. Suppose the inputs are $U = (U_1, \ldots, U_p)$, $L = (L_1, \ldots, L_p)$, and n. The outputs are all possible combinations of A and B.

Obtain $U_1 \geq U_2 \geq \cdots \geq U_p$ by sorting U.
$sep := L_1$
Determine the order statistic, say r, of sep in U.
for $i = 2$ to p **do**
  if i=r **then**
    $sep := L_r$
  else if $i = r - 1$ **then**
    Output $A = \{1, 2, \ldots, i\}$, $B = \{i + 1, i + 2, \ldots, p\}$
  else if $L_i < sep$ **then**

10: $sep := L_i$
11: Determine the order statistic, say r, of sep in U.
12: **end if**
13: **end for**

The running time in Line 1 needs $O(p \log p)$ to sort. Line 3 needs $O(\log p)$ by using binary search. The loop from Line 5 to 13 runs $p - 1$ times. Inside loop body, every line runs constant time except Line 12 which needs $O(\log p)$ by using binary search. The total time is $p \log p + \log p + (p-1) \log p = O(p \log p)$.

Furthermore, back to Algorithm 2, for every output of A and B from above, we need to check whether (4.4) and (4.5) hold. We count $\sum L_i$ before the algorithm starts. Then count $\sum_{j \in A} U_j$ and $\sum_{j \in A} L_j$ in every loop. Once Line 8 is executed, count $\sum_{j \in B} L_j = \sum L_i - \sum_{j \in A} L_j$. Thus we save the checking time to constant time.

Therefore, an E-step takes $O(p \log p)$ time. There are $O(2^p)$ steps in Algorithm 2 with at most $O(2^{p-5})$ of them containing an E-step. The generation of regular shapes takes constant time at every step. Therefore the total time is

$$O(2^p) + O(2^{p-5}) O(p \log p) = O(2^p + p 2^{p-5} \log p).$$

□

5. Determine the existence of a majorizing shape

In some problems, the goal is to find a majorizing shape, or to determine if one exists. If Algorithm 2 given in Section 3 yields a single shape, then it is the majorizing shape. However, there is a much faster way of finding out whether a majorizing shape exists, and identifying it if it exists. Even if our goal is to find all nonmajorized shapes, we can still use the faster algorithm as preprocessing. In case it finds a majorizing shape, then there is no need to go through Algorithm 2.

This procedure constructs two nonmajorized shapes in $\Gamma(L, U)$, i.e., the one which goes the upper bound route as much as possible in Algorithm 2 and the one which goes the lower bound route as much as possible. We will refer to them as the *top shape* and the *bottom shape*. Note that in constructing the top shape s_T, we need only to adjust upper bounds; and in constructing the bottom shape s_B, only to adjust lower bounds.

Theorem 5.1. *If s_T and s_B are equivalent, then either of them is a majorizing shape; if not, then no majorizing shape exists.*

Proof: (i) $s_T = s_B$. Suppose $U_i = \max_{1 \le j \le p} U_j$. Consider the reduced problem where part i is deleted and n changes to $n - U_i$. Let s'_T, s'_B be the two shapes identified by our procedure in the reduced problem. Clearly, $s'_T = s_T \backslash \{U_i\}$. We prove $s'_B = s_B \backslash \{U_i\}$ (here we refer to shape-types as *multisets*).

A lower bound L_v will be adjusted in the reduced problem only if

$$L_v + \sum_{j \neq i, v} U_j < n - U_i$$

or equivalently,

$$L_v + \sum_{j \neq v} U_j < n,$$

which is the criterion of adjusting L_v in the original problem. Therefore, the adjustment of lower bounds in choosing s'_B is the same as s_B, which implies $s'_B = s_B \backslash \{U_i\}$.

Next we prove by induction on p that all regular shapes generated by Algorithm 2 are equivalent to s_T. It is trivially true for $p = 1$. Assume that it holds for general $p - 1 \geq 1$, we prove it for p.

Suppose to the contrary, that $s' \neq s_T$ is also a nonmajorized regular shape. Then s' chooses U_i or L_k. Without loss of generality, assume s' chooses U_i. By induction, $s \backslash \{U_i\}$ majorizes $s' \backslash \{U_i\}$. Hence, s majorizes s'.

Finally, we prove that no E-shape can exist. Let the common regular shape contains r upper bounds and t lower bounds where $r + t = p - 1$ or p. Suppose to the contrary that an E-step occurs at stage $j + k$ after j upper bounds and k lower bounds are selected. Among the remaining parts, the largest (in the $\prec$ ordering) effective upper bound is $U_{[j+1]}$ and the smallest effective lower bounds is $L_{[k+1]}$. Necessarily, $j < r + 1$ and $k < t + 1$, or $s(s')$ would not agree with the common regular shape. If $U_{[j+1]}$ and $L_{[k+1]}$ are from the same part, then selecting one means not selecting the other in a shape. In particular, $L_{[k+1]}$ would not be in s and $U_{[j+1]}$ not in s', contradicting the common regular shape.

(ii) If $s_T \neq s_B$, then Theorem 5.1 assures that both s_T and s_B are nonmajorized shapes; in particular no majorizing shape exists. □

If we calculate $\sum L_i$ at the beginning, then $U_i{}' = \min\{U_i, n - (\sum L_i - L_i)$ can be computed with one subtraction. Therefore, adjusting each U_i takes a constant time. It takes $O(p)$ time to adjust all U_i in each calling of the algorithm and $O(p)$ time to select maximum of $\{U_i{}'\}$. The algorithm is called p times to obtain s_T, so the total time is $O(p(p + p)) = O(p^2)$. The time complexity of constructing s_B is the same. Finally, checking $s_T = s_B$ takes $O(p)$ time.

An improvement of this algorithm is to sort $\{U_i\}$, and to sort $\{L_j\}$ among those parts with the same upper bound at the beginning, so that we don't have to do it at every stage. But the running time is still $O(p^2)$.

Example 5.1. In Example 4.1, $s_T = (20, 19, 3, 4, 5)$ and $s_B = (1, 2, 18, 17, 13)$. Hence no majorizing shape exists.

Example 5.2. $U = (100, 90, 60, 50, 17)$, $L = (10, 70, 10, 48, 10)$. If $n = 228$, we obtain $s_T = s_B = \{90, 70, 10, 48, 10\}$ which is a majorizing shape. But, if $219 \leq n \leq 226$, then there is no majorizing shape.

References

Barnes ER, Hoffman AJ, Rothblum UG (1992) Optimal partitions having disjoint convex and conic hulls. Math Program 54:69–86

Hwang FK, Onn S, Rothblum UG (1998) Representations and characterizations of the vertices of bounded-shape partition polytopes. Lin Alg Appl 278:263–284

Hwang FK, Rothblum UG (to appear) Partition-optimization with Schur-convex sum objective functions. SIAM J Disc Math

Hwang FK, Rothblum UG (1994) Optimality of monotone assemblies for coberant systems composed of series modules. Oper Res 42:709–720

Marshall AW, Olkin I (1979) Inequalties. Theory of majorization and its applications. Academic Press, New York

Sperner E (1928) Ein Satz über Untermengen einer endlichen Menge. Math Z 27:544–548